Deep stochastic sentence generation
Resources and strategies

Simon Mille

TESI DOCTORAL UPF / 2014

Director de la tesi

Prof. Leo Wanner
Department of Information and Communication Technologies

UNIVERSITAT POMPEU FABRA

By My Self and licensed under
Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported

You are free to Share – to copy, distribute and transmit the work Under the following conditions:

- **Attribution** – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial** – You may not use this work for commercial purposes.
- **No Derivative Works** – You may not alter, transform, or build upon this work.

With the understanding that:

Waiver – Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain – Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights – In no way are any of the following rights affected by the license:

- Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
- The author's moral rights;
- Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice – For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

The follwowing PhD Committee was appointed by the Rector of the Universitat Pompeu Fabra on ...., 2014.

President: **Eva Hajíčova**

Member: **Owen Rambow**

Secretary: **Kim Gerdes**

Reserve members

Igor Mel'čuk

Horacio Saggion

The thesis defense was held on ...., 2014, at the Universitat Pompeu Fabra

Acknowledgements

I am truly and deeply indebted to many people for their active participation to the work described in this thesis. It would have been impossible for me to complete this work on my own, a state of affairs which the use of "we" throughout the thesis hardly manages to reflect.

First of all, I would like to thank my supervisor Leo Wanner for all his unfailing support through the years. Working by his side has been tremendously enriching, including at levels which go beyond the mere academic life. Leo started giving me responsibilities in European projects at a time when I hardly knew how to implement a single graph-transducer rule, and kept pushing me to go forward with my research while making sure both could coexist without major brain incidents on my side. I feel like I come out of this period with a new confidence as a person, and with enough skills as a researcher to keep doing what I like. So for all this, merci.

I am grateful to the members of the defense committee Eva Hajíčova, Kim Gerdes, Owen Rambow, Igor Mel'čuk and Horacio Saggion. Thank you so much for your interest, and thank you for taking the time to come to Barcelona.

I also feel very lucky to have been surrounded by exceptional people in my small academic world (I know it sounds exaggerated, but it is really not). These few lines are obviously not enough to thank everyone as much as they should be thanked, but I'll give it a try anyway.

Bernd Bohnet was the one who gave me the initial motivation for this work, back in 2007: he was confident that it was a good idea, and he helped me make up my mind. He has been omnipresent, helping me with his graph transducer MATE, which I used a lot, and above all with the experiments on stochastic generation, for which he implemented most systems that I describe in the thesis. Thanks you Bernd for your professionalism, you kindness, your patience, for having me in Stuttgart for a while, for making me ride a bike in the mountains again...

Many thanks to Alicia Burga, with whom we launched the annotation project in 2008. It would not have possible to design an annotation scheme and annotate these 100.000 words without her. The endless discussions we had on “what is what” turned out captivating. We have been talking about syntax any time, anywhere, late at night, while having coffee, in front of a FC Barcelona game, sometimes even at work. Thank you for your talent for finding problems, your will to never give up—including when teaching me proper Peruvian talking—and simply for you daily presence during all this time. I could not have hoped for a better partner!

Many thanks to my other colleagues from the TALN group who got involved at some point, always in an excellent atmosphere: Miguel Ballesteros, indispensable for all parsing-related tasks, and who dared experiencing “the other side”, with the implementation of our most recent deep NLG system (6–2 6–3); Gabriela Ferraro, my “che”, who has always been up for bringing easy solutions to my (numerous) problems, be it with programming or my Spanish level when I got to Barcelona; Roberto Carlini, who was able to turn into a usable interface our annotation tool which wasn't much more than a 3,000 cell table at the time. And of course all the TALN people for the daily stimulation: François Lareau, Katerina Mihailovska, Luz Rello (422 powre—sic), Vanesa Vidal, Gerard Casamayor, Joan Codina, Nadjet Bouayad-Agha, Stefan Bott, Johannes Graën, Horacio Saggion, Sara Rodriguez, Francesco Barbieri, Joan Soler, Luis Espinosa, Mireia Farrus, Monica Dominguez, Francesco Ronzano.

I am grateful to the DTIC crew, who make a great job all year long: Judith, Jana, Bea, Vanesa, Joana, Montse, Magda, Santa, and especially the great Lydia!

Thank you to colleagues from other universities who helped us work on the annotation of other languages, or simply discuss them: Kim Gerdes (French, Spanish), Anton Granvik (Finnish), Xinying Chen and Yue Zhang (Chinese), Marga Alonso (Spanish), Sylvain Kahane (French), Anja Belz and Michael White (English).

I would like to thank warmly our dear Igor Mel'čuk, who put me on the right track by getting me to understand how to annotate properly surface- and

deep- syntactic structures. Thanks for your humor, your constant support, and all the stories you always find a way to tell us until the end!

Finally, thanks to my “families” in France, for the support and the love from the beginning. Thanks to my friends, especially in Barcelona, Bordeaux and the U.S., and to King of Prussia, Big Summer, Voltor Negre, the LotJL, for providing the recurring change of scenery necessary to a good balance in life. This thesis has been written with constant acoustic support from, among others, Behemoth, the Human Abstract, and thevisitor.

And last but not least, very special thanks to Myriam. We’ve been on each other’s side for a long time, and she brings me a strength that nothing else can equal.

Abstract

The present Ph.D. thesis addresses the problem of deep data-driven Natural Language Generation (NLG), and in particular the role of proper corpus annotation schemata for stochastic sentence realization. The lack of multilevel corpus annotation has prevented so far the development of proper statistical NLG systems starting from abstract structures. We first detail a methodology for annotating corpora at different levels of linguistic abstraction (namely, semantic, deep-syntactic, surface-syntactic, topological, and morphological levels), and report on the actual annotation of such corpora, manually for Spanish and automatically for English. Then, using the resulting annotated data for our experiments, we train and evaluate deep stochastic NLG tools which go beyond the current state of the art, in particular thanks to the absence of rules in non-isomorphic transductions. Finally, we show that such data can also serve well other purposes such as statistical surface and deep dependency parsing.

Resumen

La presente tesis aborda el problema de la generación de textos comenzando desde estructuras profundas; se examina especialmente el papel de un esquema de anotación apropiado para la generación estadística de oraciones. La falta de anotación en varios niveles ha impedido hasta ahora el desarrollo de sistemas de generación estadística desde estructuras abstractas. En primer lugar, se detalla la metodología para anotar corpus en varios niveles (representaciones semánticas, sintácticas profundas, sintácticas superficiales, topológicas y morfológicas), y se presenta su proceso de anotación, manual para el Español, y automático para el Inglés. Posteriormente, se usan los datos anotados para entrenar y evaluar varios generadores de textos que van más allá del estado del arte actual, en particular por la ausencia de reglas para transducciones no isomórficas. Por último, se muestra que estos datos se pueden utilizar también para otros objetivos tales como el análisis sintáctico estadístico de estructuras superficiales y profundas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The present Ph.D. thesis addresses the problem of deep Natural Language Generation (NLG), and in particular the role of proper corpus annotation schemes for stochastic sentence realization.

This decade saw a significant increase of interest in corpus-based (i.e. *statistical*, or *stochastic*) Natural Language Processing (NLP). These tendencies have been reflected by the recent organization of (i) the very popular Conferences on Natural Language Learning (CoNLL), so far focused on the analysis of texts (e.g. dependency-based syntactic and semantic parsing, see (Buchholz and Marsi, 2006; Hajič et al., 2009)), and (ii) the first Surface Realization Shared Task (henceforth SRST), for NLG itself (Belz et al., 2011), which challenged research teams to produce some well written English texts from two types of representation, one more superficial (unordered syntactic dependency tree), and one more abstract (approximate predicate-argument structure without some grammatical units).

The first SRST evidenced two crucial points as far as NLG is concerned:

- there is very little research on stochastic NLG: only five teams submitted a system to the challenge, of which only two competed for the deep track; only two systems used mainly statistical methods;
- the training data is so far not adapted to deep NLG: good quality large-scale syntactic dependency annotations are available in many languages, but this is not true for more abstract representations; the organizers had to spend a lot of time to derive the deep input from

existing data, and the resulting annotation was not really satisfying (Belz et al., 2012).

Both problems are obviously related: if there were more ready-to-use resources available, more research could be carried out in the NLG field. This is the reason why the main aims of this thesis are (i) to design an annotation scheme which is adequate for deep generation, (ii) to apply this scheme to the annotation of a mid-size corpus suitable for the training and testing of a stochastic NLG generator; (iii) to validate this scheme and the annotated treebank in stochastic NLG generation experiments. In addition, we also show that such a resource is valuable for other fields of NLP, in particular syntactic parsing.

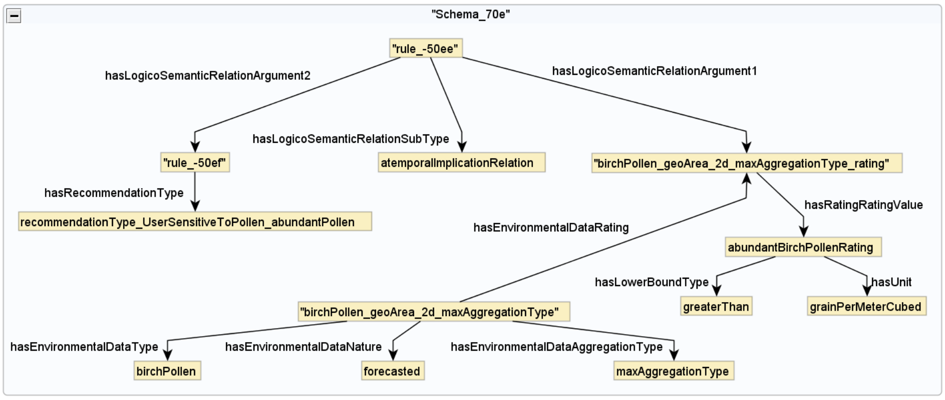
Before going more into details about the content of this thesis, let us start by a brief introduction to what is implied by the notion of "deep generation".

1.1 What is deep Natural Language Generation?

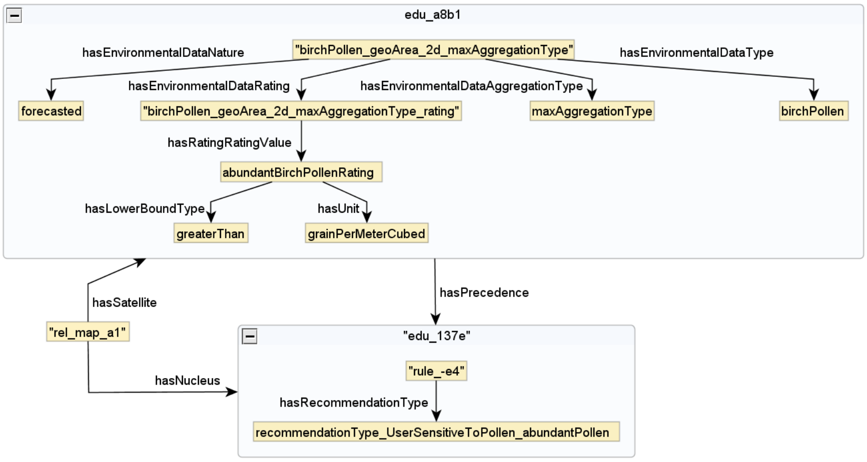
NLG is generally seen as a sequence of subtasks (Reiter, 1994). Deep NLG usually starts from numeric time series, such as sequences of measurements of pollutant concentrations or sequences of turbine pressure values, or from more complex knowledge bases (cf. Figure 1.1a for a representation of a fragment of such knowledge). These deep (abstract) representations are, from a theoretical point of view, independent from language.

Turning a deep input into a well-formed text implies the following tasks:

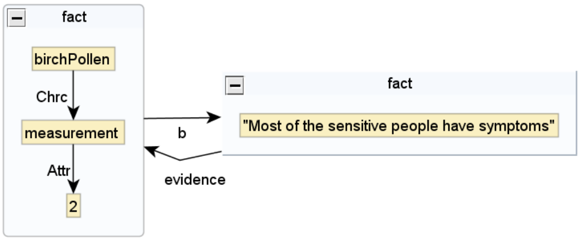
- Selecting the content to be verbalized.
  Since it does not make sense to verbalize an entire knowledge base, the first task of NLG consists in selecting a part of the ontology that will be generated. The **content plan** showed in Figure 1.1a is the result of this process, in the domain of air quality, and more specifically about a forecast on the concentration of birch pollen in the air. Such a representation describes word knowledge, in terms of objects and properties. For instance, the object *birchPollen_geoArea_2d_maxAggregationType_rating* stands for "maximum concentration of birch pollen according to several measuring stations in a certain area". This object has the property *hasEnvironmentalDataType* with the value *birchPollen*, which informs that this object is of the environmental type *birch pollen*. The property *hasEnvironmentalDataRating* indicates



(a) Content Plan



(b) Text Plan



(c) Conceptual Structure

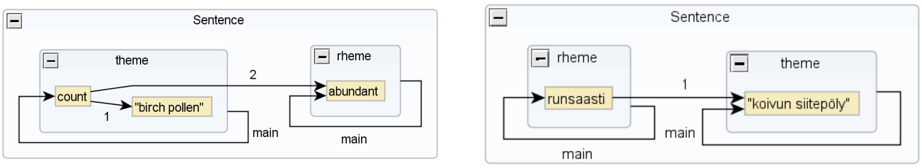
Figure 1.1: An example of deep generation (1): from non-linguistic to linguistic representations

that this maximum concentration has a rating associated to it, which in its turn has the property *hasRatingValue* that points to the node *abundantBirchPollenRating*, which stands for the rating itself, i.e. “abundant”.

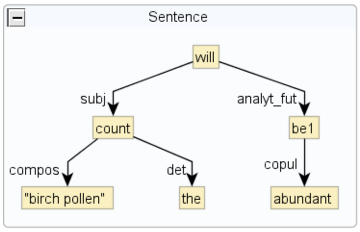
- Organizing the content discursively.
  The content has to be structured discursively, that is, some elementary discursive units (EDUs) have to be determined, and related with one another *via* discursive relations (e.g. *volitionalCause*, *violationOfExpectation*, *evidence*). In the **text plan** in Figure 1.1b, the content plan has been split into two EDUs, one which contains the data about the measurement of the concentration of birch pollen (above in the figure), and a second one about a recommendation for sensitive people due to the high concentration of pollen. The two EDUs are related by a discursive node *rel_map_a1*, which stands for an *implication* between its nucleus and its satellite. In other words, the high concentration is what implies the recommendation delivered to the public.

- Making the representation linguistically motivated.
  Once the content and the discursive organization have been determined, the next step is to obtain a linguistically motivated structure. At this point in the pipeline, this structure should be language-independent in order to allow for multilingual generation. Only the nodes of the text plan that are to be communicated (explicitly or implicitly) in the generated text are kept, with their incoming and outgoing edges; on the contrary, meta-nodes related to a possible user query and to the nature of certain content nodes (as, e.g., *forecasted*) are omitted. This representation is defined in terms of events, processes, states, entities, numerical values, etc.: it is a **conceptual structure** in terms of Sowa (2000). In Figure 1.1c, the first EDU contains only three nodes: the entity *birchPollen*, the relational process *measurement*, and the value *2*. Thematic roles such as *Chrc* and *Attr* relate the nodes with one another: *measurement* is a characteristic (*Chrc*) of *birchPollen*, and has as value *2* through the role *Attr*. Note that from this point, it is possible to use pre-generated text (on the right of the figure, not shown in Figure 1.2 since it does not evolve).

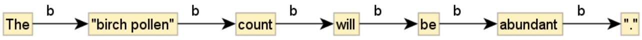
- Making the representation language-specific.
  The next step consists in making the structure language-dependent: every particular language has its own set of meanings and its own rules for their combinations. For example, in English, the meaning



(a) Semantic Structure (ENG) (b) Semantic Structure (FIN)



(c) Syntactic Structure (ENG)



(d) Sentence (ENG)

Figure 1.2: An example of deep generation (2): from abstract language-dependent to surface representations

count can relate *birch pollen* and its value, as shown in the **semantic structure** in Figure 1.2a, while in Finnish there is no possibility of combining the meaning of *concentration* with the meaning of *pollen*; thus, the value of the concentration is directly related to the node of the birch pollen, cf. Figure 1.2b.[1] Figures 1.2a and 1.2b also contain communicative information, which constrains the syntactic structure of the sentence in the next steps: what the sentence talks about (the *theme*) is more likely to be a subject in English, while what is said about the theme, forming the *rheme*, is typically a verb group.

- Determining the syntactic structure of the sentences.
  With this communicative structure at hand, after choosing the lexical unit(s) corresponding to each meaning, it is possible to draw the syntactic structure of the sentence to generate, and consequently to introduce all function words needed to make it grammatical; see the **syntactic structure** in Figure 1.2c, in which a copula had to be introduced, as well as an auxiliary and a determiner. This step is

---

[1] Note that one does not exclude the other: there can be several equivalent semantic structures corresponding to one conceptual structure.

the one which allows for lexical and structural variation, through the choice of one or another lexical unit that has a correspondence with a particular meaning.

- Ordering and inflecting the words.
  Finally, the nodes have to be ordered and inflected (based on the syntactic relations) for the **sentence** to be ready to be delivered (Figure 1.2d).

1.2 How to package the tasks of natural language generation

There are different views on how to divide the task of NLG. For instance, within a **cognitive approach**, Levelt (1989) describes the processes involved in the production of articulated messages: he distinguishes, on the one hand, *macro-* and *microplanning*, which are responsible for selecting and grouping together the information to be delivered, and, on the other hand, *formulating*, which consists in *grammatical encoding* (i.e. word selection, sentence structuring, word inflection) and *phonological encoding* (production of sounds).

The **theoretical linguistic approach** has more direct correlations with what we have described so far. For instance, Rambow and Korelsky (1992) split NLG into three main sequential tasks:

- Text Planning: this module produces a list of language-independent propositions, that is, it is responsible for selecting the content of the message and structuring it at the textual level (e.g. trough the delimitation of sentences).
- Sentence Planning: this is when is determined how the selected content will be expressed in a particular language. It consists in mainly two subtasks: the concepts of the text plan are lexicalized, and the syntactic structure of each sentence is elaborated. This step can involve syntactic aggregation, through coordination or subordination, but also the generation of referring expressions, for instance.
- Linguistic Realization: this last module handles the linearization of the words and the resolution of all morphological interactions between the words of the sentence (agreements, concatenations of words, phonetically motivated graphical modifications, etc.).

Bouayad-Agha et al. (2012a,c) present a very similar architecture, using the classic dichotomy between (i) "what to say" and (ii) "how to say it":

- (i) is called *Text Planning*, and consists of the content selection, the discourse structuring and the "linguisticization", the three steps seen in Figure 1.1; it corresponds to Rambow and Korelsky (1992)'s first subtask.
- (ii) is called *Linguistic Generation*, and covers mainly syntacticization, lexicalization, linearization and morphologization, that is, the steps seen in Figure 1.2; it corresponds to Rambow and Korelsky (1992)'s second and third subtasks, namely Sentence Planning and Linguistic Realization.

In the remainder of this thesis, we use the terminology of Bouayad-Agha et al. (2012a,c).

What we just defined as **Linguistic Generation** is considered **deep NLG** by the state of the art. Thus, from now on, the input to deep generation will be considered to be an abstract structure in which all the content of the text has been determined and distributed in separated groups corresponding to what will be distinct sentences.[2] Note that since the focus of this thesis is deep NLG, pure Machine Translation (MT) or summarization systems, which deal with straightforward text-to-text generation, are out of the scope of this work.[3]

1.3 Methods for deep natural language generation

There are three main ways to generate texts from abstract structures: with templates, rules, and statistical methods. In this section, we justify our decision to focus on the latter.

1.3.1 Handcrafted template-based methods

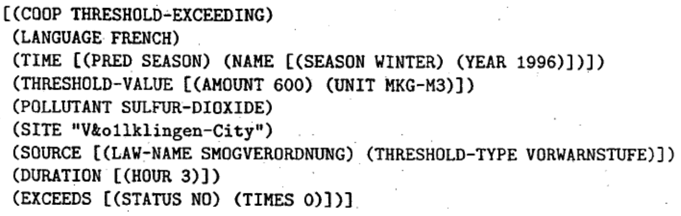
These kinds of methods rely on pre-generated text, using generally little or no linguistic information during the process: sentences are written prior

---

[2] For more discussions and references about the architecture of NLG, see e.g. (Reiter and Dale, 1997), (Oberlander and Brew, 2000) or (Mellish et al., 2006).

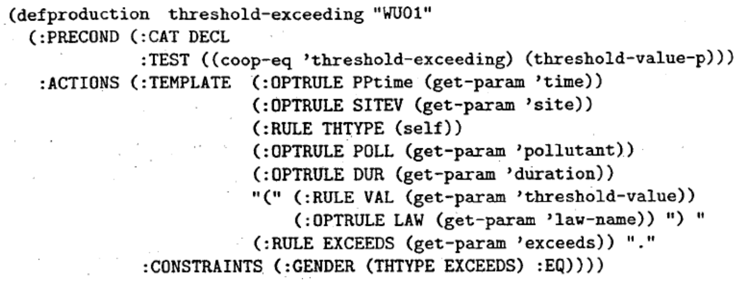
[3] For text-to-text statistical generation with no intermediate structures, see, for instance, the reference paper of Berger et al. (1996).

to generation, with possibly empty slots indexed (with rules) by variables such as, e.g., temperatures, sports scores, flight departure and arrival information, etc. Figure 1.3 illustrates how such a system, TG/2 (Busemann and Horacek, 1998), works. In this case, an English user made a query to compare thresholds for sulfur dioxide with measurements from the 1996/97 winter at Völklingen City, specifying that the applicable legislation should originate from Germany. The system retrieves the requested information, and stores it in an intermediate structure, as shown in Fig 1.3a. The system then tries to match the values of the *coop-eq* slot in this intermediate representation with the value of the COOP slot contained in the rules. The rule shown in Figure 1.3b does provide such match (value *THRESHOLD-EXCEEDING*), and since the intermediate representation also contains information about the *THRESHOLD-EXCEEDING*, the rule can apply and fill the slots of the corresponding English template. The text which is returned to the user is the following: *During the winter season 1996/97, at the measurement station of Völklingen City, the early warning threshold for sulfur dioxide at an exposition of three hours (600 μg/m*^{3} *according to the German decree "Smogverordnung") was not exceeded.*



```
[(COOP THRESHOLD-EXCEEDING)
 (LANGUAGE FRENCH)
 (TIME [(PRED SEASON) (NAME [(SEASON WINTER) (YEAR 1996)])])
 (THRESHOLD-VALUE [(AMOUNT 600) (UNIT MKG-M3)])
 (POLLUTANT SULFUR-DIOXIDE)
 (SITE "V&o1lklingen-City")
 (SOURCE [(LAW-NAME SMOGVERORDNUNG) (THRESHOLD-TYPE VORWARNSTUFE)])
 (DURATION [(HOUR 3)])
 (EXCEEDS [(STATUS NO) (TIMES 0)])]
```

(a) Intermediate representation (Busemann and Horacek, 1998, p.242)



```
(defproduction  threshold-exceeding "WU01"
  (:PRECOND (:CAT DECL
             :TEST ((coop-eq 'threshold-exceeding) (threshold-value-p)))
   :ACTIONS (:TEMPLATE  (:OPTRULE PPtime (get-param 'time))
                        (:OPTRULE SITEV (get-param 'site))
                        (:RULE THTYPE (self))
                        (:OPTRULE POLL (get-param 'pollutant))
                        (:OPTRULE DUR (get-param 'duration))
                        "(" (:RULE VAL (get-param 'threshold-value))
                            (:OPTRULE LAW (get-param 'law-name)) ") "
                        (:RULE EXCEEDS (get-param 'exceeds)) "."
             :CONSTRAINTS (:GENDER (THTYPE EXCEEDS) :EQ))))
```

(b) Rule defining a sentence template (Busemann and Horacek, 1998, p.244)

Figure 1.3: Details of the TG/2 template-based system

This kind of system can easily map the same intermediate structure onto another language, if the templates in said language have been defined, as, e.g., in French: *En hiver 1996/97, la station de mesure de Völklingen City, le seuil d'avertissement pour le dioxide de soufre pour une exposition de trois heures (600 $\mu g/m^3$ selon le décret allemand "Smogverordnung") n'a pas été dépassé.*

The advantage of template-based systems is that the quality of the text is completely controlled, and that generation is just a matter of finding the good template in the database and fill its empty slots, so it is simple and fast. The important drawback is that the set of templates needed for generating texts grows very fast beyond controllable as soon as one wants to cover a variety of domains, linguistic styles, languages, etc. Some systems based on templates are described in (Van Deemter and Odijk, 1997), (White and Caldwell, 1998), (Theune et al., 2001), (Busemann and Horacek, 1998), (McRoy et al., 2003), and (Narayan et al., 2011). Diversity of template-based systems and their opposition to other NLG systems is discussed in more details in (Van Deemter et al., 2005).

1.3.2 Handcrafted rule-based methods

Rule-based systems map (in one or several steps) an abstract structure onto a well-formed sentence, using linguistic resources such as dictionaries, which describe the basic units of each level of representation (e.g. meaning, lexical units), and grammars, which contain the rules that produce a well-formed text from the input structure, according to the knowledge found in dictionaries.[4]

Figure 1.4 shows the general architecture of the MATE generator (Wanner et al., 2010), which is able to generate air quality reports in eight languages: abstract (semantic) structures are mapped step by step onto surface structures (respectively through deep-syntactic, surface-syntactic, topological, morphological structures). Each mapping is performed thanks to a graph transduction grammar, coupled with semantic, lexical and morphological resources for each language. One dictionary (*semanticon*) contains different possible lexicalizations of a particular meaning, which allows for lexical and structural variation in the output texts. Another dictionary (*lexicon*) contains the syntactic description of all lexical units used for a particular language, especially, how they combine with other lexical units (i.e., what

---

[4] Note that rules and dictionaries are not necessarily implemented as different components: encyclopedic knowledge can be represented as rules too.

functional nodes or features have to be introduced in the syntactic tree in order to generate a grammatical sentence).

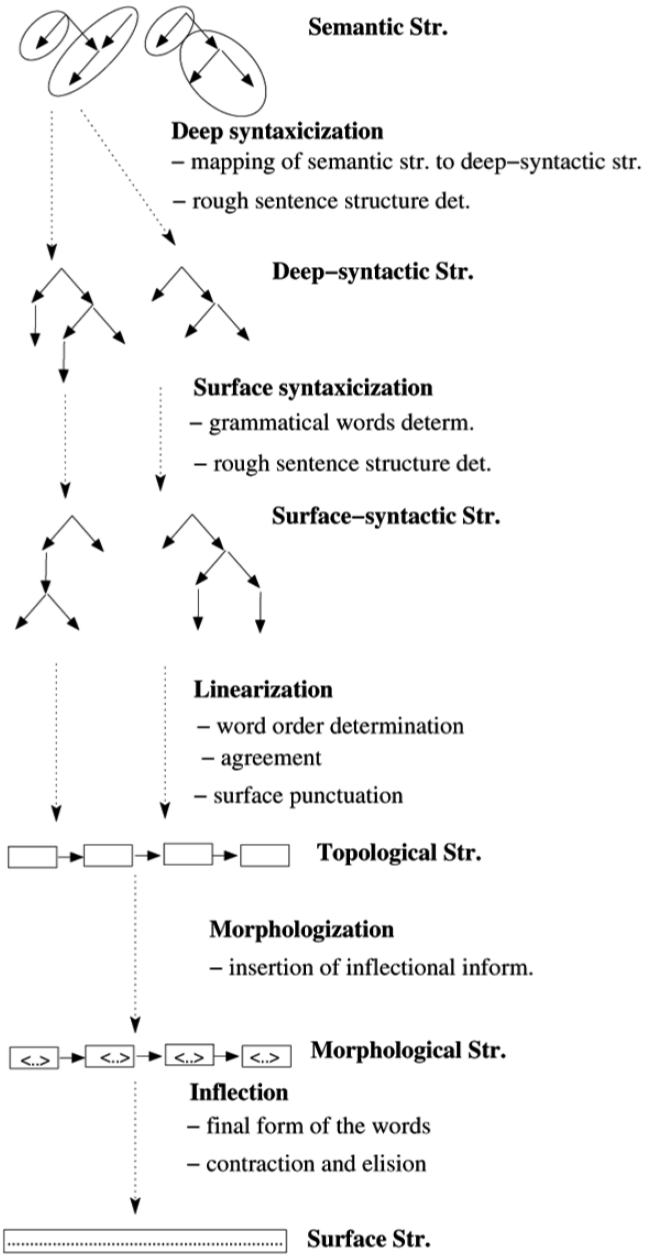


Figure 1.4: The MATE rule-based generator (Wanner et al., 2010, p.938)

In a rule-based system, if the rules are generic enough, a wide variety of outputs can be produced by a rather small rule set (or *grammar*); for instance, in English, the main verb will always agree with the subject, and one single rule can handle this agreement for any configuration found in the input structure. However, building a rule-based system is costly in two respects: first, solid linguistic knowledge is needed in order to build a grammar, e.g. syntactic, morphological, lexical, etc.; second, a complex grammar can be slow to produce an output, since the system has to find the best combination of rules for the corresponding input. Furthermore, while it is easy to control the precision of a rule-based system, its coverage is often an issue: if an input contains a configuration which is underspecified or not foreseen by the system, the generation will most probably fail. For details on handcrafted rule-based generators, see, among others, SURGE (Elhadad and Robin, 1996), Realpro (Lavoie and Rambow, 1997), KPLM (Bateman, 1997), MATE (Wanner et al., 2010) or SurReal (Gervás, 2011).

1.3.3 Corpus-based methods

Machine learning algorithms produce models which are able to predict what an abstract structure will look like at the sentence level. These systems rely on a pre-existing large-scale annotation of reference (*gold standard*) data from which it calculates probabilities. These probabilities, which all together constitute the models, can be calculated over simple word cooccurrences in a sentence, but also over more complex features, such as the Part-of-Speech of a node and/or the surrounding ones, and/or the syntactic relation(s) it is involved in, etc. For example, in order to calculate the order between words, knowledge such as "in 100% of the cases, the definite determiner "the" appears *before* its nominal syntactic governor in an English sentence" is needed. From annotated data, it is also possible to "learn" wide-coverage grammars, which can be used for rule-based generators. These corpus-based systems do not always produce well-formed texts, since the quality of their outputs relies heavily on the selection of complex feature combinations and on the quality of the annotated data itself. But they are faster to build than handcrafted grammars, and their coverage is wider since they are able to produce an output for a completely new input configuration, see, e.g. (Langkilde and Knight, 1998), (Bangalore and Rambow, 2000), (Corston-Oliver et al., 2002), (Nakanishi et al., 2005), (Belz, 2005), (White et al., 2007), (Mairesse et al., 2010), (Bohnet et al., 2010). A more extensive state of the art of corpus-based generators is presented in Chapter 2.

1.3.4 Summary of the pros and cons of the different methods

To summarize what has been outlined above, two big advantages of a generator with non corpus-based methods are that (i) individual rules can be tuned so as to favor high quality outputs, and especially (ii) no previous resource is needed for the system. The main problems are that it is costly to develop, it is difficult to obtain a wide coverage, and rule-based systems tend to be slow and unstable (one small change in a rule can affect other rules). In contrast, a stochastic system requires annotated resources for its training, but it has considerable advantages over a traditional realizer that uses handcrafted rules in that: (i) it is more robust, (ii) it usually has a significantly larger coverage, (iii) it is much faster to build (once again, when an adequate corpus is available), (iv) once built, the system is easier to maintain, and (v) if trained on a representative corpus, it is domain-independent. The grammar-learning approach combines advantages and disadvantages of both systems: it can have a wide coverage if the appropriate corpora are available, it is fast to build, and the rules can be individually improved and the quality of the output better controlled, but it is slower and difficult to maintain, since the rules which are extracted can be tens of thousands.

As rightly pointed out by Belz (2008), traditional wide coverage realizers such as KPML (Bateman et al., 2005), FUF/SURGE (Elhadad and Robin, 1996) and RealPro (Lavoie and Rambow, 1997), which were also intended as off-the-shelf plug-in realizers still tend to require a considerable amount of work for integration and fine-tuning of the grammatical and lexical resources. Deep stochastic sentence realizers have the potential to become real off-the-shelf modules.

We believe that if the training material already exists, choosing the statistical method is advantageous. And since, as stated at the beginning of this introduction, we also aim at building such training material, a corpus-based approach has naturally been chosen for this thesis. In addition, since it is important to us that the NLG system presented here can be used in a wider text generation project such as a paraphrasing system or a summarization tool, the speed of the system is crucial. As a consequence, in this thesis we are primarily interested in a system that avoids to resort to rules. But in order to test an hybrid approach, we also try to combine the derivation of rule sets and of fully probabilistic submodules.

1.4 Linguistically motivated approach

The basic assumption underlying this work is that it is crucial to develop a theoretically-motivated approach for deep natural language generation. At the beginning of the XX^{th} century, Ferdinand de Saussure clearly established syntax as an independent level of description (De Saussure, 1989). A few decades later, Tesnière (1959) and, with a different point of view, Chomsky (1965) refined this idea. Both approaches, namely dependency and constituency syntax, have largely contributed to the development of the Natural Language Processing field. In this thesis, we also assume that it is necessary to separate clearly the different levels of representation of language. For this reason, our work on deep natural language generation is based on the Meaning-Text Theory (MTT) theoretical framework (Mel'čuk, 1988). The MTT is a dependency-based framework which postulates the existence of various level of representation between deep inputs as we defined them in Section 1.2 and a full-fledged sentence. Having several intermediate structures at hand, we do not need to consider all linguistic phenomena at play at once. On the contrary, at each level, each linguistic phenomenon can then be treated separately (e.g. semantics, syntax, morphology, etc.).

Furthermore, Mel'čuk (1988) argues that separating the different levels of abstraction allows for modeling adequately the process of utterance production. Indeed, as explained in this introduction, NLG is usually seen as a sequence of subtasks which aim at transforming an abstract structure into a well-formed text. Figures 1.1 and 1.2 on pages 3 and 5 illustrate to what extent an input and an output differ. The idea is that, for instance, it is less difficult to transform respectively 1.2a into 1.2c and 1.2c into 1.2d than to transform 1.2a into 1.2d in just one step. Using intermediate structures dictated by a linguistic model allows us to facilitate the transition between one and another, which is crucial for deep NLG.

The major shortcoming so far for such an approach to NLG is the lack of resources, in spite of the increasing popularity of dependency treebanks in NLP applications. Dependency annotated corpora are currently available for many languages, as shown in Table 1.1. But most dependency treebanks were meant to be used for syntactic parsing, for which only morpho-syntactic and linear order annotations are necessary. Only very recently there has been an increasing need for dealing with deeper levels of representation, due to experiments on automatic semantic role labeling (Surdeanu et al., 2008). To respond to this need, the initially purely syntactic corpora were enriched with partial semantic annotation, without prior discussion re-

Language	**Name**	**Reference**
Arabic	Quranic Arabic DT	(Dukes et al., 2010)
Arabic	Prague Arabic DT	(Hajic et al., 2004)
Basque	3LB	(Aduriz Agirre et al., 2003)
Bulgarian	BulTreeBank	(Chanev et al., 2006)
Catalan	AnCora	(Taulé et al., 2008)
Chinese	Sinica	(Chen et al., 2003)
Chinese	CDT	(Chang et al., 2009)
Croatian	Croatian DT	(Tadić, 2007)
Czech	Prague DT	(Hajič, 2005), (Hajič et al., 2006)
Danish	Danish DT	(Kromann, 2003)
Dutch	Alpino	(Van der Beek et al., 2002)
English	Penn TreeBank	(Marcus et al., 1999), conversion (Johansson and Nugues, 2007)
Estonian	Arborest	(Bick et al., 2004)
Finnish	Turku DT	(Haverinen et al., 2009)
French	French TreeBank	(Abeillé et al., 2003)
French (Oral)	Rhapsodie Project	(Deulofeu et al., 2010)
German	TIGER	(Brants et al., 2004)
Greek (Modern)	Greek DT	(Prokopidis et al., 2005)
Greek (Ancient)	Ancient Greek DT	(Bamman et al., 2009)
Hebrew	Hebrew DT	(Goldberg and Elhadad, 2009)
Hindi/Urdu	Hindi/Urdu TreeBank	(Bhatt et al., 2009)
Hungarian	Hungarian DT	(Vincze et al., 2010)
Italian	ISST	(Montemagni et al., 2003)
Japanese	Kyoto DT	(Kurohashi and Nagao, 2003)
Latin	Latin DT	(Bamman and Crane, 2006)
Persian	Persian DT	(Rasooli et al., 2013)
Portuguese	Floresta Sintá(c)tica	(Afonso et al., 2002)
Romanian	Romanian DT	(Călăcean, 2008)
Russian	SynTagRus	(Apresjan et al., 2006)
Slovene	Slovene DT	(Džeroski et al., 2006)
Spanish	AnCora	(Taulé et al., 2008)
Swedish	Talbanken05	(Nilsson et al., 2005)
Tamil	Tamil TreeBank	(Ramasamy and Zabokrtskỳ, 2012)
Turkish	Turkish TreeBank	(Oflazer et al., 2003)

Table 1.1: Existing Dependency Treebanks (non-exhaustive)

garding what kind of deep annotation would be appropriate or what exactly should each level of representation deal with. As a result, semantically enhanced annotations such as PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004) on top of the Penn TreeBank prove insufficient, for instance, for NLG (Belz et al., 2011). In addition, word order, syntactic dependencies, morphological features, semantic relations, etc., are phenomena that are rather different in their nature. However, quite often, their annotations are agglomerated in a single structure. Such a structure is deficient from the theoretical (linguistic) point of view, and it reduces the quality of the annotated resources, which in its turn hampers the quality of the applications trained on them.

1.5 Outline of the thesis

The remainder of the thesis is organized as follows.

In Chapter 2, we present the current state of the art of statistical generation and multilayered corpus annotation.

Chapter 3 gives an overview of the Meaning-Text Theory, and describes precisely how we obtained a supervised mid-size corpus of Spanish annotated with predicate-argument, syntactic and morphological information. Special emphasis is put on the surface-syntactic annotation methodology since this layer is the basis for obtaining all other layers. We also show how this kind of annotation can be otained automatically from existing resources.

Chapter 4 reports on our experiments on training different deep statistical generators on the obtained multilayered annotations.

Chapter 5 then shows that an NLG-suitable corpus can easily be used as such for other purposes, in particular for training good quality surface or deep-syntactic statistical parsers.

Finally, in Chapter 6, we summarize the main points of the undertaken research together with its limitations, and outline the perspectives that it opens.

Chapter 2

State of the art

As stated in the introduction, this thesis deals with deep stochastic generation, and in particular with the resources that this task requires. The few existing statistical generators are rather limited, largely due to the lack of adequately annotated resources. Before going more into details about our annotation methodology and its underlying principles, let us have a look at (i) what parts of generation are covered by the state-of-the-art stochastic generators, and (ii) what the available resources look like. This chapter is organized as follows. In Section 2.1 we first give an overview of the evolution of statistical text generation systems. Then, Section 2.2 presents a description of various currently available multilayered corpora which are relevant as points of comparison with our work, be it for the similarity of some principles of annotation, for what kind of information is encoded in the annotation, or for the language annotated. Finally, Section 2.3 points out the problems in the common annotation schemes.

2.1 Stochastic generators

Since the first proposal on stochastic generation (Knight and Hatzivassiloglou, 1995), the state of the art evolved and several techniques have been developed. For the sake of clarity, we classify them in four chronologically motivated groups, even though some systems from different groups may have common characteristics: (i) output ranking, (ii) statistically-driven handcrafted generation, (iii) automatic grammar derivation, and (iv) non grammar-based generation.

2.1.1 First steps: overgeneration and ranking

This section presents systems (i) which use rule-based generators with handcrafted grammars in order to produce several output texts for a given input representation, and (ii) which rank these outputs from best to worst by calculating the similarity with a reference corpus (unannotated texts or syntactic annotations).

The first paper mentioning statistical methods for NLG is (Knight and Hatzivassiloglou, 1995). It describes a statistical ranker which sorts out concurrent outputs of the PENMAN rule-based generator (Penman, 1989), in the framework of Japanese-English Machine Translation (MT). Such ranking is needed when information is missing in the input representation (number, definiteness, etc.), that is, when the input is underspecified and sentences with different meanings could correspond to it (e.g., *the cats sleep* VS *the cat sleeps* if no number for *cat* is specified). The alternative realizations of an input, compactly represented as “word lattices”, are ranked calculating their similarity with the strings of two words (bigrams) found in the reference set of sentences (46 million words corpus from the Wall Street Journal). A limitation of this n-gram approach is that looking only at bigrams is obviously not enough to control more complex, long-distance lexical or syntactic choice. For example, a bigram approach would rule out *a cats* and accept *the cats*, but it would also accept *a splendid cats*, since both *a splendid* and *splendid cats* are perfectly valid bigrams. As a result, the general quality of the output cannot be optimal. However, this approach indirectly allows for constraining lexical cooccurrences, for instance, since it will give more weight to a sentence which contains pairs of words that frequently occur together.

Nitrogen (Langkilde and Knight, 1998) is a system which connects a ranker to a grammar that is able to map Abstract Meaning Representation (AMR) to text via the word lattices already introduced in (Knight and Hatzivassiloglou, 1995). It includes the integration of a recasting mechanism to derive AMRs from other semantically equivalent AMRs so as to allow more flexible generation. The system is still simple and robust, requiring very little linguistic knowledge. The main problem remains the text quality due to the bigram approach, but also the fact that there can be many candidates for each node in the word lattice, multiplying the possible output structures: the longer the sentence, the (exponentially) higher the processing time. A couple of years later, Langkilde (2000) solves this problem showing that in order to rank possible outputs, it makes more sense to look at a tree structure which contains all of them (which she calls *forest*) instead of a graph, avoiding the unnecessary exploration of many paths that takes place in the lattice. The successor of Nitrogen, HALogen (Langkilde-Geary, 2002), provides a broader coverage of English structures thanks to the inclusion of more syntactic features, and gives more importance to the statistical ranking, but the main architecture remains the same.

The FERGUS generator (Bangalore and Rambow, 2000) is focussed on the linearization part of linguistic generation. The authors show that using syntactic trees for learning a model as well as for producing sentences gives better results (Ratnaparkhi (2000) also made this claim roughly at the same time, see next page). They use the XTAG formalism (Doran et al., 1994) and follow Langkilde and Knight (1998) in that they create word lattices and statistically rank them depending on their n-gram similarity with the training corpus. The input to their system is an unlabeled dependency tree which contains all the words, and which they statistically tag with XTAG lexico-syntactic information as a preliminary step. They argue that having access to such syntactic and lexical information helps significantly to improve the output of a realizer since long distance phenomena which are invisible to purely n-gram models are explicitly shown by the dependencies in the syntactic tree, allowing them to handle with more efficiency the agreements between wordforms of the final sentence.

Walker et al. (2002) present SPoT, a trainable sentence planner for dialog systems . The system uses the Meaning-Text Theory's deep-syntactic structures (Mel'čuk, 1988) as intermediate representations, which they consider predicate-argument structures, mapping fragments of text plans onto them by a set of operations in a bottom-up, left-to-right fashion. Several sentence plans are created with a rule-based system, and the best plan is selected and sent to the rule-based RealPro generator (Lavoie and Rambow, 1997) to generate the sentences. Stent et al. (2004) present a similar system. Chen et al. (2002) combine FERGUS and SPoT in order to build a real-time dialog system; in this paper, special attention is also given to the system's integrability and its portability to other domains. Habash (2004) presents a similar approach as in FERGUS, with what he calls structural n-gram models. Finally, the ATT system (Stent, 2011) is a recent realizer also based on FERGUS; it utilizes lexicalized and unlexicalized bag of features, and ranks the outputs with a trigram model. The morphologization is performed thanks to a morphological dictionary obtained automatically from the goldstandard annotation.

2.1.2 Introduction of statistics to the selection of rules

Shortly after the development of the first statistical rankers, intents were made to reduce or eliminate the need for the generation of all possible realizations, so as to avoid to output too many unnecessary candidates and to improve the processing time. In order to do so, some statistical decisions were introduced at some point in the generation pipeline, and used for driving the application of handcrafted generation rules.

NLG3 (Ratnaparkhi, 2000) is a system for the generation of sentences describing flight information in the air travel domain. This generator is based on templates, but some choices are made statistically, namely, a part of word-selection and inter-phrase ordering. Possible linearized outputs are not ranked as do Langkilde and Knight (1998); rather, the input attribute/value pairs are directly mapped onto pre-built sentences using intermediate syntactic information, which means that corpora annotated with attribute/value pairs in the domain of air travel, and also with syntactic structures, are needed. These syntactic structures are unlabeled dependency trees, which express untyped grammatical relations between the words in a sentence. These trees are obtained semi-automatically from an existing corpus; they provide information that allows the system to take better decisions in selecting the appropriate template. The mapping is performed using maximum entropy (ME; more precisely, Iterative Scaling, see (Malouf et al., 2002) for comparison between different ME models). For spoken dialog systems, a similar approach is followed by Oh and Rudnicky (2000) (but using ranking at some point), while Walker (2000) performs the selection among a set of templates through reinforcement learning.

Belz (2005) presents three more or less complex ways to have a rule-based generator produce the best output possible from a single semantic structure, through *p*robabilistic Context-free Representationally Underspecified (*p*CRU) language generation; see also (Belz, 2008). Her starting point are numeric time series in the field of meteorological data. In her experiments, she uses a bigram approach, as in (Langkilde and Knight, 1998), but also statistical data extracted from the application of the rules during the generation of the gold-standard annotation. Indeed, during the generation process, alternative outputs can be created at every step, via the application of alternative rules. Each decision made by the generator (i.e. each rule application) is counted and then converted to a probability, which in its turn is used to give a global weight to a sequence of rule application according to a particular input configuration. This way, *during* the generation process, it is possible to avoid applying some rules that do not usually apply to a particular configuration, according to the data found in the corpus.

This type of data-driven restriction during generation has also been used in order to control the style of the output of a generator, e.g. sentence length and type of referring expressions (Paiva and Evans, 2005), or to predict personality parameters which constrain in different ways the realization of generated text (Mairesse and Walker, 2008). Also worth mentioning is SEGUE (Pan and Shaw, 2004), a "case-based" system in which the rules are only used to produce sentences which have not been generated before. In other cases, the overlap between an input semantic graph and a semantic graph which has already been generated is statistically measured (the training corpus consists in output texts associated with their input graph). This way, it is made possible to use whole previous generations, or adapt them to a similar input.

2.1.3 Automatic derivation of grammars from annotated corpus

In parallel to the increasing availability of statistical applications and annotated corpora, a branch of NLG focused on finding solutions to the time-consuming elaboration of handcrafted generation grammars. Several works describe the automatic derivation of grammars for rule-based systems from annotated corpora:

- generation from domain-specific semantic annotation with dynamic rule selection (Varges and Mellish, 2001);
- content selection and linguistic generation rules for a summarization system (Kan and McKeown, 2002);
- sentence planning and linguistic generation rules for Nitrogen in the context of spoken dialog systems (Oh and Rudnicky, 2002);
- rules for linearization of dependency trees learning from parallel phrase and dependency structures (Bohnet, 2005);
- sentence planning and realization rules from unannotated data (Zhong and Stent, 2005);
- and also some theory-based generators: HPSG (Nakanishi et al., 2005), LFG (Cahill and Van Genabith, 2006) and (Hogan et al., 2007), CCG (White et al., 2007) and (Rajkumar et al., 2011).

All the systems described so far in this chapter imply the presence of rules, be they handcrafted or derived automatically from annotated data. Even if rule-based systems can be optimized through restrictions of rule applications, they not only suffer from speed limitations, but also from maintenance issues due to the increasing number and complexity as soon as decent coverage is desired. This is why the popularity of modules with no linguistic knowledge beyond mere statistical data obtained from annotated corpora has been growing.

2.1.4 Do it without rules: elaboration of fully statistical (sub-)modules

Fully probabilistic modules are black boxes which receive any input similar to what they have been trained on, and return an output almost instantly. The issue with this kind of module is that the quality of the output largely relies on the quality and size of the annotated data it is trained on. And even on a perfectly annotated large corpus, some outputs might be more than questionable for a native speaker. However, the trade-off between quality and coverage/speed is less and less of a problem, and it is an objective of this thesis to illustrate this tendency.

The oldest system, to our knowledge, which includes a fully stochastic unit is Amalgam (Corston-Oliver et al., 2002). It presents a German realizer that maps a logical input onto sentences with intermediate syntactic (phrase-based) representation. The logical input, obtained through deep-parsing of full-fledged sentences, contains communicative, semantic information, but also lexical features such as subcategorization information, which is the part of lexicalization that is handled statistically, through a decision tree classifier. The rest of the sentence planning—defining the structure of the sentence—is rule-based, and the ordering is performed combining rules (for constituent-internal precedence) and decision tree classifiers (for inter-constituent precedence). The authors argue that although they train their model on corpora, they do not need any annotated corpus since they can produce the syntactic trees thanks to a parser and automatically derive from there the logical representation, even though they acknowledge that the quality of this kind of corpus cannot be expected to be optimal. Some further experiments on linearizing constituents in French, German and English are reported in (Ringger et al., 2004).

Marciniak and Strube (2004) present a cascade of classifiers that map so-called *minimal elements* onto a well formed text using Tree Adjoining Grammar (Joshi et al., 1991). They bring down text generation to a set of 8 different operations, associated with 8 different types of features, the value of which is decided during the application of the sequence of 8 classifiers trained on a relatively small corpus (916 clauses). Their minimal elements are pre-ordered constituency trees which encode some combinatorial rules, so they use linguistic knowledge during the generation process. Note that their view of generation could be considered a little simplistic as far as the combination of operations is concerned.

A generator based on an inverted semantic parser is presented in (Wong and Mooney, 2007). Their statistical system, trained on few sentences (880), produces concurrent output sentences from partially ordered meaning representation. To choose the best candidate, they use n-gram models. In that they do not use any intermediate structure, the process is similar to the one used in BAGEL (Mairesse et al., 2010). BAGEL is a statistical language generator which uses dynamic Bayesian networks to assign a semantic part directly to a phrase. The representation is based on stacks that contain the semantic information for a sentence decomposed into phrases, the phrases being already ordered with respect to each other before starting the generation. The Bayesian networks are used to order the phrases and to align semantic parts with them. The model generalizes to some degree since it contains lexicalized backoff features that reduce the needed semantic coverage.

With the recent growth of interest for dependency formalisms and increasing availability of subsequent annotations in many languages (see Table 1.1 above), several studies have been made on the last step of linguistic generation, i.e. surface realization, with unordered syntactic dependency trees as input. For instance, Filippova and Strube (2007) propose a linearization system for German which first identifies the initial word of a sentence and then determines the rest of the ordering. He et al. (2009) and Wan et al. (2009) describe systems which first order the governors and the dependents in a dependency tree, and then order the dependents of the same node with respect to one another, respecting the decisions taken during the first step. The DCU system (Guo et al., 2011a) achieves a wide-coverage linearization and inflection of the words through the use of syntactic structure and morpho-syntactic features. This surface realizer was initially designed to convert the functional representations of the Lexical Functional Grammar framework (Bresnan, 2001), i.e., *f-structures*, into well-formed and linearized sentences (Guo et al., 2011b). F-structures are lexical matrices annotated with syntactic annotation, in which not all function words are considered nodes (for instance, the infinitive marker ‘to’ is just a feature). As a result, the original system is also capable of converting these features to actual words in the sentence; the paper provides an evaluation of this realizer on English and Chinese.

2.1.5 Summary

Table 2.1 shows clearly the evolution of statistical NLG systems, from a simple n-gram-based ranker applied after symbolic generation to a system which uses models in all the steps of sentence planning and linguistic generation. Since one of the objectives of this thesis is to contribute to develop an approach to NLG which favors the absence of rules, we focus here on generators which substitute some parts of the rule system by purely data-driven modules. The generators are described along the following characteristics:

- Which subtasks of sentence planning (syntacticization and lexicalization) and realization (linearization and morphologization) are performed statistically (see Introduction)? Is overgeneration and ranking of outputs used?
- What kind of annotated knowledge is needed in order to train the generator? Note that the annotations of different phenomena can be superimposed.
- What kind of statistical method was used in order to build the models of the corpus-based modules?
- Does a corpus-based module allow for the mapping between non-isomorphic graphs?
- Can the system be used independently of the domain considered in the reference paper?
- What language is concerned by the experiments of the authors?

Even if most of them use intermediate (e.g. syntactic) information, very few generators perform generation in more than one step, that is, generating intermediate structures between the logical form and the text. The integration of other intermediate levels of representation such as syntactic structures has not yet been proven to improve the systems in a way that they can be used on a large scale, but the contrary has not been shown

		Nitrogen 1998	FERGUS 2000	NLG3 2000	Amalgam 2002	Fil & Str 2007	BAGEL 2010
Corpus -based...	syntacticization	-	-/+	-	-	-	+
	lexicalization	-	-	-/+	-/+	-	-/+
	linearization	-	-	-/+	-/+	+	-/+
	morphologization	-	-	-	-	-	-/+
	ranking	+	+	-	-	-	-
Type of annotation	logical	-	-	+	+	-	+
	syntactic	-	+	+	+	+	+
	sentence	+	+	+	+	+	+
Statistical method	n-grams	+	+	-	-	-	-
	decision trees	-	-	-	+	-	-
	dynamic bayes	-	-	-	-	-	+
	maximum entropy	-	-	+	-	+	-
	SVM classifiers	-	-	-	-	-	-
Non-isomorphic mapping		-	-	-	-	-	-
Domain independent		+	+	-	+	+	+
Language		ENG	ENG	ENG	GER	GER	ENG

“-” means “yes”, “+” means “no”, and “-/+” means “partially”

Table 2.1: Overview of features of 6 statistical realizers

either. The preliminary idea defended in our work is that using such intermediate structures can significantly improve the quality of the generated text without sacrificing the robustness of the system or triggering excessive annotation efforts prior to the training stage. Another important remark is that no system so far handles mappings between graph which are not isomorphic; in other words, the statistical production of functional words is not covered by the current state of the art. From this point of view, a system allowing (i) for the use of intermediate levels of representation and thus built on a multi-layer corpus and (ii) for the derivation of non-isomorphic graphs is ideal for experiments. Such a system is presented in Chapter 4.

2.2 Overview of the existing multi-layered annotations

There are many corpora available (cf. Introduction), but the type of annotated resources relevant to this work should have the following basic properties:

- have dependency-based layers: dependencies are a simple and efficient way of representing natural language interactions; a dependency graph is acyclic, directed and labeled, as in $cat - eat \rightarrow mouse$; note that a clear definition of the relation taxonomy is crucial for the expressiveness of the dependency representation (a large part of this thesis is dedicated to this issue); see, e.g., the work of Iordanskaja and Mel'čuk (2009) on French verbal dependencies.
- exhibit (relative) separation of the layers of annotation;
- include at least syntax and some logical representation;

In this section, we detail the few corpora which correspond to this decription and give an example of a different kind of annotations which can be used in order to obtain something similar.

2.2.1 The Prague Dependency Treebank

The Prague Dependency Treebank (Hajič, 2005; Hajič et al., 2006) is the current reference with respect to multi-layered dependency annotation, since it was the first large-scale treebank designed from the start with several layers of linguistic representation in mind. It contains four different levels of annotation of the Czech language: sentence string, morphology, syntax, and a deeper, more abstract stratum. Each stratum is annotated independently of the others and extensively documented. Figure 2.1 shows the general architecture of the annotation.[1]

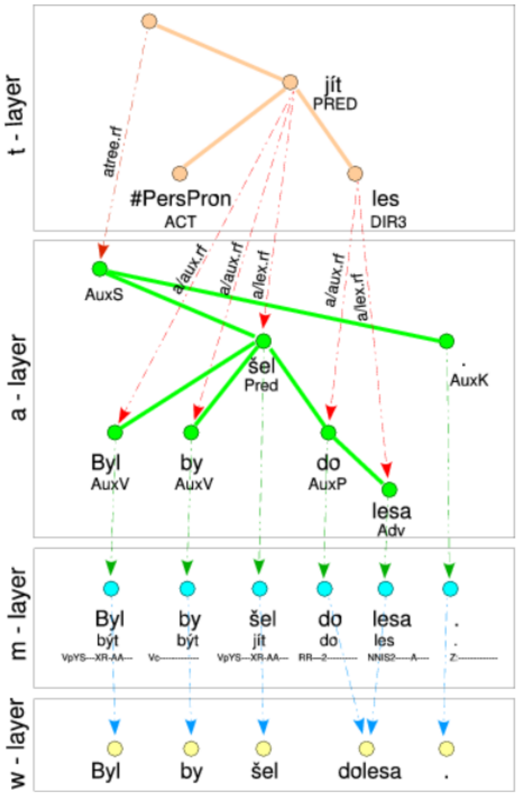


Figure 2.1: The layers of annotation of the PDT

The closest layer to the unannotated sentence (the word layer) is the morphological layer, the m-layer, which contains the following information: the lemma of the wordforms; the Part-of-Speech (PoS) and morphological features such as gender, number, case, person, tense, voice, etc.; the correct form of the token (if there is an error in the source text); some other attributes of minor relevance.

All those features are associated with the words and stored as one single 15-slot chain in which the value (if any) of each individual feature always

---

[1] Source: `http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/html/ch02.html`

appears in the same position (e.g., the PoS always comes first, the gender always comes third, etc). Furthermore, the m-layer nodes contain a correspondence with the word layer and with the analytical layer.

The following level is the analytical layer, the a-layer, “a rooted ordered tree with labeled nodes and edges” which has a strict 1-to-1 correspondence with the m-layer nodes. The nodes are ordered and we can find 28 edge labels (“afun”, for “analytical function attribute”) linking them.

The nodes at the a-layer contain the following 10 attributes:

- the incoming analytical function, only based on syntactic criteria;
- the position in the original sentence;
- the lemma;
- the original word form as found in the sentence and the corrected form if any manual correction was necessary;
- the chain of morphological features as described *supra*;
- 4 markers of coordinations, appositions, and parenthesis interpretations.

The a-layer nodes also contain a correspondence with the m-layer and with the tectogrammatical layer. There is an extensive documentation on the annotation of this layer; see for instance (Hajič et al., 2001).

The deepest level of annotation is the tectogrammatical layer, the t-layer, which “reflects the underlying (deep) structure of the sentence”, in other words, its semantic structure. The representation at this level has the following properties:

- it is a tree, not a graph;
- it only contains *autosemantic words*, that is, meaningful units; in other words, there is no 1-to-1 correspondence with the m-layer nodes, since the functional (or *governed*) prepositions, for instance, are not in the t-layer (while omitted subjects or gapped elements are);
- it is annotated with grammatemes, which represent information about the node that cannot be derived from the structure nor the nodes themselves: for example, grammatical number, tense, etc.;

- a valency frame is assigned to predicates of the corpus (PDT-VALLEX): each entry is linked to its occurrence in the corpus and thus can be used for disambiguation.
- Topic-Focus Articulation (TFA) information is introduced in the tree: "a node can be contextually bound, contrastively contextually bound, or contextually non-bound".

Every non-root node of the t-layer contains: the type of the edge linking the node to its governor, a unique identifier, the tectogrammatical lemma, the Topic-Focus Articulation, the communicative dynamism of nodes (the nodes are ordered), and coreference links with other nodes.

2.2.2 The Penn TreeBank/PropBank/NomBank

In this thesis, we refer as "the Penn TreeBank" (henceforth PTB) to the CoNLL-format dependency version of the original phrase-based Penn TreeBank annotation (Marcus et al., 1993); the automatic mapping from constituency to dependency is described in (Johansson and Nugues, 2007).[2] This dependency PTB has been enriched by disambiguated identifiers and predicate-argument structure for (i) verbal and (ii) nominal predicates (respectively PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004)). Thus, the resulting one-word-per-line file contains several layers of annotation: morphologic, syntactic, and semantic. We describe this corpus with more details because it is used in some of our experiments in Chapters 3 and 4.

The PTB contains a morphological layer, with the lemmas, and 54 tags combining PoS and morpho-syntactic features. As for syntactic representation, each node but the root of the dependency tree receives one of the 37 dependency labels available to the annotators. The semantic annotation comes from PropBank/NomBank (henceforth, PB/NB): each verb and each predicative noun in a distinct usage is assigned a set of its arguments, numbered semantic roles (a *roleset*) starting with 0: Argument 0, Argument 1, Argument 2, …(henceforth *A0*, *A1*, *A2*). The roleset is associated with a set of syntactic frames that specify the variations in the realization of the roles of a given roleset, resulting in a *frameset*. Roughly, each verb/noun

---

[2] See (Hajič et al., 2009) for the description of the 14^{+}-column one-word-per-line CoNLL format.

sense that shows a distinct configuration of roles is distinguished as a frameset. For instance, **decline.02** 'demure, reject' has the roleset (Palmer et al., 2005):

A0:	agent
A1:	rejected thing

The label *A0* is reserved for Agency, such that "agentless" verbs do not possess *A0*; consider, for instance **decline.01** 'go down incrementally':

A1:	entity going down
A2:	amoung gone down by EXT
A3:	start point
A4:	end point

The numbered argument labels can be attributed functional tags *EXT* or *PRD*. *EXT* ("extent") signals that the corresponding argument is numerical; *PRD* ("secondary predication") that the argument is used predicatively; cf.:

Example 2.1. *John received from Mary 10 dollars[A1-EXT].*

Example 2.2. *John considers Mary generous[A2-PRD].*

Syntactic dependents that do not form part of the frame (i.e., are not arguments) of the governor are covered in semantics by "modifier" relations (*AM-. . .*). They represent knowledge related to discourse (cause, purpose, discursive), circumstancials (directional, manner, adverbial, locatives, temporal, extent), predicate structure (modal, reciprocals, secondary predication), or negations.

Example 2.3. *Canada's gross domestic product rose in August as a result of service industry growth:*
rose-AM-TMP→in [August] (temporal modifier)
rose-AM-CAU→as [a result. . .] (causal modifier)

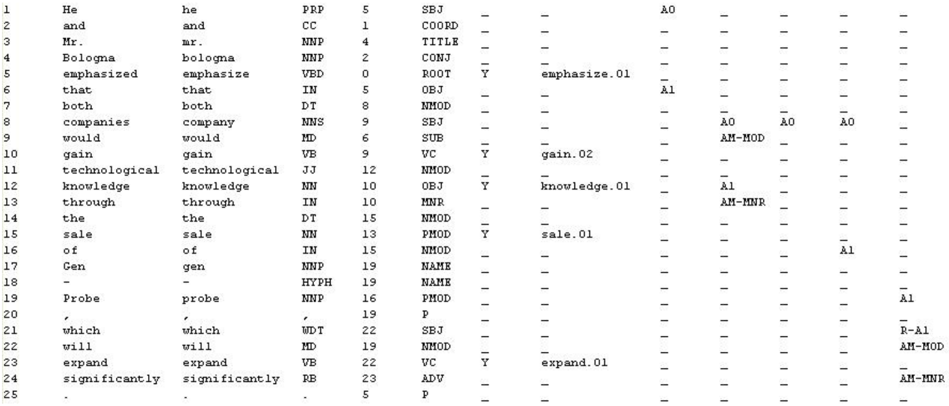
Example 2.4. *Apparently the commission did not really believe in this ideal:*
did-AM-NEG→not (negative modifier)
did-AM-DIS→apparently (sentencial modifier)
did-AM-DIS→really (sentencial modifier)

The dichotomy between the semantic roles and modifiers is also valid for the rest of the relations in the PB/NB annotation: relatives and interrogatives on the one hand (*R-A...* edges), and “continuation” constructions (i.e. split arguments) on the other hand (*C-A...* edges).

Example 2.5. *Civilized discourse and an environment where compromise can begin are lost:*
begin-R-AM-LOC→where

Example 2.6. *He believes in what he plays:*
plays-R-A1→what

Example 2.7. *Labor costs continued to rise more rapidly in service industries than in goods-producing industries:*
continued-A1→costs
continued-C-A1→to [rise]
rise-C-AM-LOC→in [service industries]
rise-C-AM-LOC→than [in ...]



1	He	he	PRP	5	SBJ	_	_	A0	_	_	_	_
2	and	and	CC	1	COORD	_	_	_	_	_	_	_
3	Mr.	mr.	NNP	4	TITLE	_	_	_	_	_	_	_
4	Bologna	bologna	NNP	2	CONJ	_	_	_	_	_	_	_
5	emphasized	emphasize	VBD	0	ROOT	Y	emphasize.01	_	_	_	_	_
6	that	that	IN	5	OBJ	_	_	A1	_	_	_	_
7	both	both	DT	8	NMOD	_	_	_	_	_	_	_
8	companies	company	NNS	9	SBJ	_	_	_	A0	A0	A0	_
9	would	would	MD	6	SUB	_	_	_	AM-MOD	_	_	_
10	gain	gain	VB	9	VC	Y	gain.02	_	_	_	_	_
11	technological	technological	JJ	12	NMOD	_	_	_	_	_	_	_
12	knowledge	knowledge	NN	10	OBJ	Y	knowledge.01	_	A1	_	_	_
13	through	through	IN	10	MNR	_	_	_	AM-MNR	_	_	_
14	the	the	DT	15	NMOD	_	_	_	_	_	_	_
15	sale	sale	NN	13	PMOD	Y	sale.01	_	_	_	_	_
16	of	of	IN	15	NMOD	_	_	_	_	_	A1	_
17	Gen	gen	NNP	19	NAME	_	_	_	_	_	_	_
18	-	-	HYPH	19	NAME	_	_	_	_	_	_	_
19	Probe	probe	NNP	16	PMOD	_	_	_	_	_	_	A1
20	,	,	,	19	P	_	_	_	_	_	_	_
21	which	which	WDT	22	SBJ	_	_	_	_	_	_	R-A1
22	will	will	MD	19	NMOD	_	_	_	_	_	_	AM-MOD
23	expand	expand	VB	22	VC	Y	expand.01	_	_	_	_	_
24	significantly	significantly	RB	23	ADV	_	_	_	_	_	_	AM-MNR
25	.	.	.	5	P	_	_	_	_	_	_	_

Figure 2.2: PTB/PB/NB annotation of the sentence *“He and Mr. Bologna emphasized that both companies would gain technological knowledge through the sale of Gen-Probe, which will expand significantly [...].”*

Figure 2.2 illustrates the PTB/PB/NB annotation in the popular CoNLL format. The first column is the position of the units in the sentence (used as its ID), the second holds the superficial form of each unit, the third its lemmatized form, the fourth indicates its PoS, the fifth the position of its syntactic governor, the sixth the label of the edge from its governor; from the seventh column, we find the semantic annotation, starting with the semantic status of the unit—semantic predicate (“Y”) or not (“_”)—, and then, in the eighth column, its disambiguated meaning. The remaining columns, in this case columns nine to thirteen, stand for respectively each predicate of the sentence (five predicates ⇒ five columns) in the order they appear; for instance, *companies* is Argument 0 of the second (*gain.02*), the third (*knowledge.01*) and the fourth (*sale.01*) predicates.[3]

This corpus has been used for the derivation of the multi-layered annotation provided to the participants of the first Surface Realization Shared Task (Belz et al., 2011). The syntactic annotation is used as such, and the semantic annotation has been adapted as follows: (i) some function words and commas were removed; (ii) when there was no available dependency in PropBank or NomBank, the syntactic dependencies, some labels of which were generalized, were used to connect the deep nodes (for more details, see Section 3.5.1).

2.2.3 The AnCora corpus

As numerous other corpora (Hajič et al., 2009), AnCora follows the same style as the PTB/PB/NB for the Spanish language: all syntactic dependencies between all words are labeled, and a partial predicate-argument annotation is superimposed in the CoNLL'09 format. Note that we describe here in details the 2006 version of AnCora, since this version is the one which has been used as the starting point of our own annotation process: it contains less sentences, only syntactic annotation, not all dependencies are labeled.

In 2006, AnCora contained 3,510 sentences (95,028 tokens) taken from the Lexesp corpus (Sebastián et al., 2000) and from the Spanish news agency EFE (see Figure 2.3 for a sample annotation of a Spanish sentence).

The AnCora corpus is annotated with the following information, split over the ten columns of the CoNLL'06 file (Buchholz and Marsi, 2006):

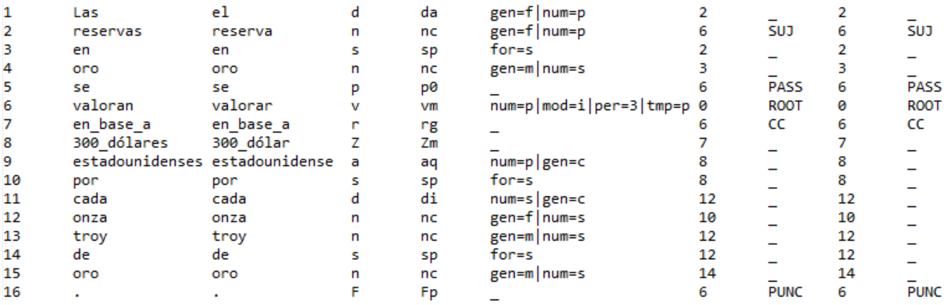
- the first column contains the position of the words in the sentence;
- the second column contains the surface form of the words; some entities are grouped together as one single surface form (e.g. 800_*millones _de_Euros* '800 millions of Euros', *Banco_Central* 'Central Bank', *sin_embargo* 'however') and clitic pronouns are not separated from their anchor (e.g. *verlo* lit. 'see-it');

---

[3] The actual CoNLL format also comprises columns for predicted lemmas, Part-of-Speech, governor, dependency, etc. to be used for the evaluation of statistical tools trained on the corpus. In Figure 2.2, we removed these columns since they don't bring any information with respect to the annotation.

Type	**AnCora coarse-g.**	**AnCora fine-g.**	**Subtype**
adjective	a	aq ao	regular ordinal
conjunction	c	cc cs	coordinating subordinating
determiner	d	da dd de di dn dp dt	definite demonstrative exclamative quantificative numerative possessive interrogative
punctuation	F	Fa/c/e etc.	11 subtypes
interjection	i	i	regular
noun	n	nc np	common proper
pronoun	p	p0 pd pe pi pn pp pr pt px	reflexive demonstrative exclamative quantificative numerative personal relative interrogative possessive
adverb	r	rg rn	regular negative
preposition	s	sp sn	regular null element
verb	v	va vm vs	auxiliary regular copula
date	w	w	regular
unknown	X	X	regular
temperature unit	Y	Y	regular
number	z	z	regular
number and unit	Z	Zp	regular

Table 2.2: The Part-of-Speech tags used in AnCora'06



```
1    Las              el              d  da  gen=f|num=p                 2   _     2   _
2    reservas         reserva         n  nc  gen=f|num=p                 6   SUJ   6   SUJ
3    en               en              s  sp  for=s                       2   _     2   _
4    oro              oro             n  nc  gen=m|num=s                 3   _     3   _
5    se               se              p  p0  _                           6   PASS  6   PASS
6    valoran          valorar         v  vm  num=p|mod=i|per=3|tmp=p     0   ROOT  0   ROOT
7    en_base_a        en_base_a       r  rg  _                           6   CC    6   CC
8    300_dólares      300_dólar       Z  Zm  _                           7   _     7   _
9    estadounidenses  estadounidense  a  aq  num=p|gen=c                 8   _     8   _
10   por              por             s  sp  for=s                       8   _     8   _
11   cada             cada            d  di  num=s|gen=c                 12  _     12  _
12   onza             onza            n  nc  gen=f|num=s                 10  _     10  _
13   troy             troy            n  nc  gen=m|num=s                 12  _     12  _
14   de               de              s  sp  for=s                       12  _     12  _
15   oro              oro             n  nc  gen=m|num=s                 14  _     14  _
16   .                .               F  Fp  _                           6   PUNC  6   PUNC
```

Figure 2.3: AnCora'06 annotation of the sentence *Las reservas de oro se valoran en base a 300 dólares estadounidense por cada onza troy de oro* lit. 'the stocks of gold are valued on the basis of 300 dollars U.S. for each ounce troy of gold ', 'Gold stocks are valued on the basis of U.S.$ 300 per troy ounce'

- the third column contains the lemma of the word in the second column;
- the fourth column contains the coarse-grained Part-of-Speech (PoS) of the word (15 different tags);
- the fifth column contains a fine-grained PoS (46 different tags); Table 2.2 details all coarse- and fine-grained tags and their meanings;
- the sixth column contains a list of morpho-syntactic features; there are 9 different attributes in this column (different features are separated by a vertical bar): case (*case*), gender (*gen*), mood (*mod*), number (*num*), person (*per*), tense (*tmp*), and other features overlapping with others of columns 4, 5 or 6, such as *for*, which marks prepositions, *pari*, which indicates if a word exhibits gender agreement, and *pos*, which is used to mark the possessive elements;
- the seventh and ninth columns contain the identifier (line number) of the governor of the word in the first column;
- the eighth and tenth columns show the dependency label between the word and its governor; there are 17 different labels used in the annotation, plus the *ROOT* label indicating the root of a syntactic tree; 58,131 dependencies (63.52% of the total, excluding the *ROOT* label) are left unlabeled in this early version of the corpus (see Figure 2.3 for details on the labels).

The set of dependency relations is quite classic, with the typical subject, different types of objects and adverbials, etc. However, a number of annota-

AnCora label	**Description**
ATR	complement of copula
CAG	agentive complement
CC	adverb with tight relation to verb
CD	direct object
CD.Q	special direct object
CI	indirect object
CPRED	predicative complement
CPRED.CD	special predicative complement
CREG	prepositional object
ET	textual element
IMPERS	marker of impersonality
MOD	verb modifier (non argumental)
NEG	negative adverbial
PASS	marker of passive
PUNC	puntuation signs
SUJ	subject
VOC	vocative
_	unlabeled
ROOT	root

Table 2.3: The dependency labels used in AnCora'06

tion policies deserve to be pointed out, since they will have a direct impact on the efforts that have to be made in order to produce our annotation. In particular, the following choices have been made:[4]

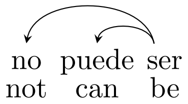
- non-finite verbs in auxiliary, modal and raising/control constructions are the syntactic governors of the whole verb group, such as *ser* 'be' in Figure 2.4a, which governs the modal *puede* 'can'.

- an adjective positioned before a noun is the governor of the adjectival phrase that includes the noun, as, e.g., the adjective *pequeña* 'small' in Figure 2.4b; accordingly, the adjective is considered governor of various dependents of the noun.

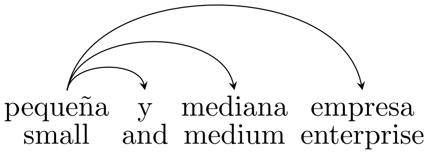
- functional and coordinating conjunctions are considered "transparent" from the perspective of syntax, in that they are not used to connect groups together: they do not have any dependent; for instance,

---

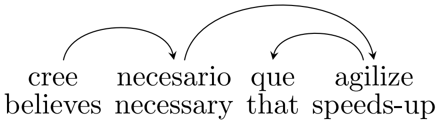
[4] In the illustrations, dependencies are not labeled, since we focus on the direction of the arcs.



(a) Modal verb (‘cannot be’)



(b) Nominal group with anteposed coordinated adjectives (‘small and medium enterprise’)



(c) Functional conjunction (‘she believes necessary that he speeds up’)

Figure 2.4: Sample dependency direction choices in AnCora

in Figure 2.4b, the coodinating conjunction *y* ‘and’ is a dependent of *pequeña* ‘small’, as is the conjunct *mediana* ‘medium’; in Figure 2.4c, the subordinating conjunction *que* ‘that’ is a dependent of the subordinated verb *agilize* ‘speeds up’.

2.2.4 The Stanford Typed Dependencies

The Stanford Typed Dependencies (de Marneffe et al., 2006) originate from other annotation attempts inspired by the Lexical-Functional Grammar (LFG) framework (Bresnan, 2001): GR (Carroll et al., 1998) and PARC 700 (King et al., 2003). As the related annotations, the Stanford approach provides a scheme for syntactic annotation; but from this layer a more abstract representation can be derived, through the use of “collapsed” dependencies.[5] Collapsing the dependencies means that some nodes, which they call “function words”,[6] become dependency relations, so as to bring

---

[5] de Marneffe and Manning (2008) give more details on the differences between the Typed dependencies and PARC and GR.

[6] Actually, the collapsing (i) does not concerns only functional nodes, since these are supposed to have no own independent meaning, but collapsed words such as “because”,

closer non-functional nodes in the representation. What is done to achieve this is to remove all prepositions, conjunctions, and possessive clitics, and replace them by edges labeled with the name of the removed word. The rest of the dependencies remain the same as the original syntactic annotation. Figure 2.5 shows the syntactic annotation (top) and its counterpart collapsed annotation (bottom).

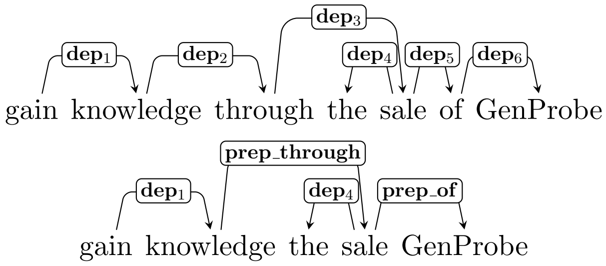


Figure 2.5: Non-collapsed and collapsed representations according to the Stanford scheme

At the syntactic layer, de Marneffe et al. (2006) present a set of 48 dependency labels organized in a hierarchy: the root of the hierarchy is the generic dependency *dep*, which is split into 8 coarse-grained labels, which are in their turn split into 39 fine-grained labels. The grammatical functions are classified according to whether or not the dependency is of the coordinating or subordinating type, and whether it is argumental or not. However, unlike other abstract representations seen in this section, such as PDT, ISST and PropBank, the Stanford scheme is not concerned with specifying the predicate-argument relations at any layer of representation (de Marneffe and Manning, 2008). Instead, the scheme produces, through the collapsing of prepositional nodes, a semantic representation which explicitly encode the type of relation that some words can have with one another, in the same fashion as what can be found in "conceptual" networks such as the one presented in, e.g., (Sowa, 2000).

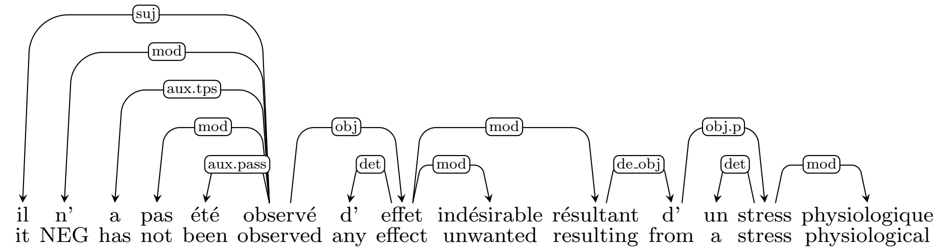
2.2.5 The Sequoia French Treebank

The Sequoia Treebank (Candito and Seddah, 2012) is a constituency and dependency treebank following the same basic guidelines as the French Tree-

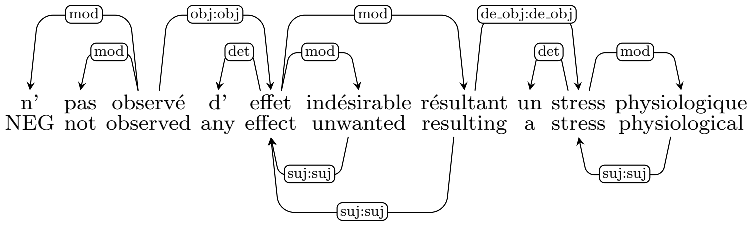
---

"and" or "while" for instance all have a precise meaning, and (ii) does not concern all function words, since functional nodes such as auxiliaries for example, are not collapsed

Bank (FTB, see (Abeillé et al., 2003; Abeillé and Barrier, 2004)) in order to cover more domains than the FTB. Texts from medical, social, political, journalistic and legal domains have been first annotated with morpho-syntactic information and syntactic constituencies, both manually checked. Then, an automatic constituency-to-dependency conversion has been applied (Candito et al., 2010) and the resulting structures have also been subjected to manual revision. There are two levels of morpho-syntactic tags, a coarse-grained one with 14 tags (corresponding to adjectives, adverbs, coordinating and subordinating conjunctions, weak clitic and strong pronouns, determiners, foreign words, interjections, common and proper nouns, prepositions, verbs and punctuations), and a fine-grained one with twice as many tags, which differentiates between various subtypes of verbs, pronouns, adverbs and determiners. The surface-syntactic edge tagset contains 23 different labels called *final Grammatical Functions* (final GFs) which cover syntactic constructions in a quite classical way (subject, object, modifier, relative, coordination, auxiliary construction, etc.); see sample structure in Figure 2.6.



Sequoia surface-syntactic structure



Sequoia deep-syntactic structure

Figure 2.6: Sample Sequoia annotations

As a further step, annotators manually superimposed a semantics-oriented annotation to the final GFs, as explained in (Candito et al., 2014). For this deep annotation, only meaningful nodes are considered; in other words, some functional nodes are ignored: functional prepositions and conjunctions (e.g. *d'* 'from' in Figure 2.6), auxiliaries (e.g. *a* 'has' and *été* 'been'), relative pronouns, empty subjects (e.g. *il* 'it'), etc. Determiner are maintained in the annotation. The annotation scheme differentiates between final GFs and *canonical* GFs, which reflect the underlying argumental structure of deep predicates. For instance, the subject of a passive verb is the final subject but the canonical object of this verb. Sometimes, final GFs are added at the deep layer too: adjectives are final modifiers of their governing noun in both superficial and deep annotations, but in the latter the noun is the final and canonical subject of this adjective. In Figure 2.6, canonical GFs are on the right of the colons; all other GFs are final.[7]

2.2.6 The Italian Syntactic-Semantic Treebank

The Italian Syntactic-Semantic Treebank (Montemagni et al., 2003)—henceforth ISST—is a multilayered corpus of Italian language that contains four manually revised levels of annotation: morpho-syntax, syntactic constituents, syntactic dependencies, and lexical semantics. The nodes at each level are connected through the annotation tool used for the task. Although the most superficial annotation is very similar to that of the PDT, the other layers are quite different from it.

The first layer, the closest to the sentence, contains a morpho-syntactic annotation under the form of tags associated to the components of the sentence. There are 16 basic PoS tags (e.g. *noun*, *verb*) and 31 more precise tags (e.g. *proper noun*, *common noun*), which combine with other morpho-syntactic properties (number, person, gender, etc.), for a total of 236 possible tags. In Italian, some words can combine in a single morphologically complex unit; such units receive particular processing in the annotation, together with some multi-word expressions.

The ISST syntactic annotation is two-fold: it contains both phrase-based and functional representations as completely independent annotations. The phrase-based annotation contains 22 types of constituents; the main difference with classic constituency annotations is the fact that there are no traces in the structure, since they are accounted for in the functional annotation.

---

[7] In the original annotation, *cf.* (Candito et al., 2014, p.4), both superficial and deep structures are superimposed; they are separated here in order to show clearly the differences between them.

The latter is a tree-like dependency structure annotated with labeled oriented edges. The main difference to the PDT annotation, for instance, is that not all words of the morphological layer are used in the annotation: the dependencies only hold between lexical heads, excluding determiners, auxiliaries and some prepositions. Non-lexical items are encoded as attribute/values of the lexical nodes. In other words, the dependencies are recoverable, but they are only partially explicit. The reason for that is that the functional annotation is clearly oriented to deeper predicate-argument structure: the functional labels are divided according to the modifier/argument opposition, as shows the hierarchy of dependency relation used for the task (see Figure 2.7). Figures 2.8 and 2.9 show a sample annotation of the dual syntactic layer in the ISST.

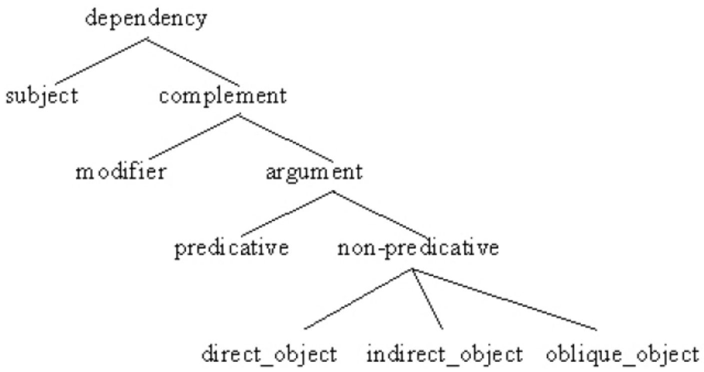
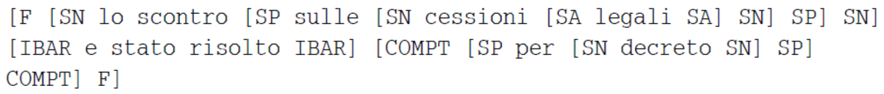


Figure 2.7: The functional tag hierarchy in ISST (Montemagni et al., 2003, p.210)



```
[F [SN lo scontro [SP sulle [SN cessioni [SA legali SA] SN] SP] SN]
[IBAR e stato risolto IBAR] [COMPT [SP per [SN decreto SN] SP]
COMPT] F]
```

Figure 2.8: Sample ISST constituency structure for the sentence *lo scontro sulle cessioni legali è stato risolto per decreto* 'the clash on legal transfers has been resolved by decree' (Montemagni et al., 2003, p.193)

The lexico-semantic annotation is a layer on which the content units are assigned three types of information: (i) sense of the word in its context, linked to the corresponding Italian WordNet entry; (ii) various lexico-semantic tags used for marking figurative usages, the presence of neologisms, etc.; (iii) notes by the annotators aiming at facilitating revisions by other annota-

```
sogg (risolvere.<diatesi=passiva>, scontro)
mod (scontro, cessione.<intro=''su''>)
mod (cessione, legale)
mod (risolvere.<diatesi=passiva>, decreto.<intro=''per''>)
```

Figure 2.9: Sample ISST functional annotation corresponding to Figure 2.8 (Montemagni et al., 2003, p.196)

tors. This information can be associated to single nodes (USS), multi-word expressions (USC), and titles (of newspapers, books, etc., UST).

2.2.7 The DELPH-IN project

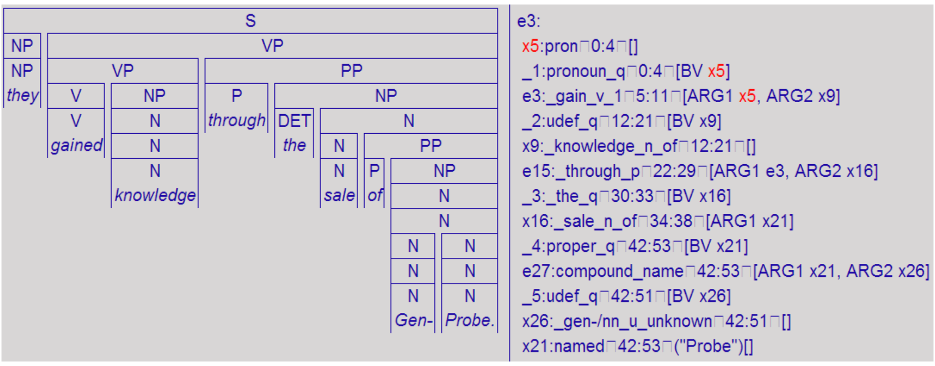
Multilayered corpora also exist in some more complex representations, as it is the case for the Head-driven Phrase Structure Grammar-influenced annotations (HPSG, (Pollard, 1994)). Because the syntactic annotation is provided under the form of constituency trees, it does not seem to fit with the initial goal of this section, i.e., to describe dependency annotations only. However, (Ivanova et al., 2012) show that it is possible to transform the phrase-structures and the logical annotation into respectively labeled trees and labeled graphs containing only bilexical dependencies. The conversions may not be optimal yet, but considering the variety of languages covered by the HPSG Resource Grammars, this project opens interesting perspectives as far as multilayered dependency annotation is concerned (especially taking into account that some work has been done for Spanish already (Marimon, 2010)).

The ambitious DELPH-IN project (Oepen, 2002)[8] aims at creating open-source HPSG grammars for many languages as different as English, Japanese, Spanish, German, Norwegian, Korean, French, Portuguese and Chinese, for instance. These grammars are used to obtain syntactic and semantic (logical) parses from raw text, and combined with a manual selection of the best parser output, they are used for producing gold standard corpus for any language. This has already been done on a large scale with the LinGo Redwoods English corpus (Oepen et al., 2004), thanks to the English Resource Grammar (ERG, (Flickinger, 2000)). For example, the analysis in Figure 2.10 has been obtained through the LinGo ERG of an online demonstrator[9].

---

[8] See http://www.delph-in.net/ for background.

[9] http://erg.delph-in.net/logon



```
e3:
x5:pron¬0:4¬[]
_1:pronoun_q¬0:4¬[BV x5]
e3:_gain_v_1¬5:11¬[ARG1 x5, ARG2 x9]
_2:udef_q¬12:21¬[BV x9]
x9:_knowledge_n_of¬12:21¬[]
e15:_through_p¬22:29¬[ARG1 e3, ARG2 x16]
_3:_the_q¬30:33¬[BV x16]
x16:_sale_n_of¬34:38¬[ARG1 x21]
_4:proper_q¬42:53¬[BV x21]
e27:compound_name¬42:53¬[ARG1 x21, ARG2 x26]
_5:udef_q¬42:51¬[BV x26]
x26:_gen-/nn_u_unknown¬42:51¬[]
x21:named¬42:53¬("Probe")[]
```

Figure 2.10: Sample ERG analysis of the sentence "*They gained knowledge through the sale of Gen-Probe.*"

On the left side of Figure 2.10, a syntactic phrase-based analysis is shown, while the right side is a logical analysis in the format of Minimal Recursion Semantics (MRS, (Copestake et al., 2005))[10]. If it is easy to understand the constituents of the left side, the logical analysis contains some meta-information, which can make it difficult to understand at first sight. Words, parts of words, or groups of words are assigned an internal ID, which appears at the beginning of every line. For instance, the identifier *x5*[11] stands for the string contained between characters 0 and 4 of the sentence (i.e., *they*), and *e3* for the string between the fifth and eleventh characters of the sentence (i.e., *gained*), 5 being the space between *they* and *gained*). In the line of *x5*, there is no further information, but in the line of *e3*, the square brackets are not empty: the ERG identified *x5* as *e3*'s first argument (*ARG1*), and *x9* (*knowledge*) as *e3*'s second argument (*ARG2*). Some meta-nodes such as *udef_q* or *compound_name* are also used in the MRS representation. Note also that functional nodes, such as *of* in *sales of*, are not considered semantic nodes, and consequently do not form part of the logical representation: unlike in NomBank (see Figure 2.2), the first argument of *sale* is *Gen-(Probe)*, and not *of*. One other notable difference with PropBank and NomBank is that not only verbal and nominal predicates receive arguments, but also adjectival and adverbial ones. As a result, the MRS are complete and can form connected graphs with almost only predicate-argument edge

---

[10] We selected what be believed to be the best analysis of the sentence, which was the second suggestion of the online rule-based parser.

[11] *x5* appears in a different color because a variable lights up when the user of the interface points the mouse to it, in order to facilitate the reading of the structure.

labels.

For more discussions between the different semantic analyses of some of the corpora described in this section, see (Ivanova et al., 2012). In this paper they also compare the DELPH-IN analysis with in particular the CoNLL ones and Stanford, based on the multi-annotated PEST corpus released in the framework of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation (Bos et al., 2008).

2.3 Some problems in common annotation schemes

Corpora such as the ones described in the previous subsections are usually annotated in order to be used by NLP tools orientated to language understanding: syntactic and/or semantic parsing, relation extraction, information retrieval, word sense disambiguation, etc.[12] Natural Language Generation is often left aside, so that when it comes to using those resources for NLG, heavy adaptations are required (Belz et al., 2011; Wanner et al., 2012; Belz et al., 2012). Only the PDT authors argue that one should annotate deeper layers in a way that allows for being able to reconstruct the superficial representations, that is, without losing any information, explicitly mentioning NLG. Currently, several corpora (AnCora, Tiger/Salsa, Chinese Treebank / PropBank, etc.) are annotated following the annotation scheme in the PTB corpus, which serves as the reference corpus regarding size and consistency of annotation. Let us discuss what we believe to be the main problems, from the linguistic point of view, of corpora that follow the PTB/PB/NB scheme and of the others cited in the previous subsection. First, we point out the confusions between layers of representation at the level of nodes and edges, and then the incompleteness of some annotations.

2.3.1 Confusion between layers of representation

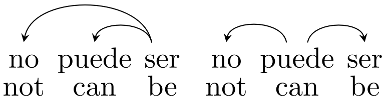
2.3.1.1 Mix of syntactic and semantic edges

First of all, there can be confusions which are due to the directions of the edges. For instance, in Section 2.2.3, we mentioned non-finite verbs in auxiliary constructions, anteposed adjectives, void and coordinating conjunctions in Figure 2.4, all considered syntactic dependents. The annotators

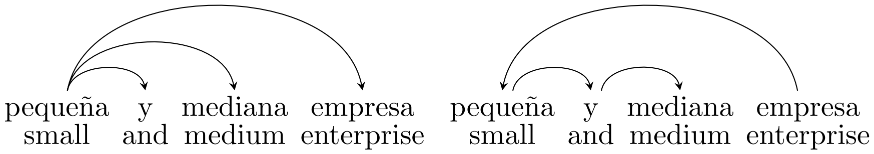
---

[12] The creators of the PropBank, for instance, acknowledge this about their corpus Palmer et al. (2005).

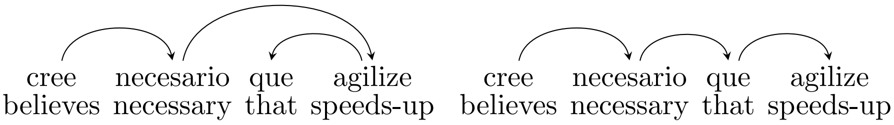
certainly based their choices on semantic criteria: while we believe that an auxiliary is the syntactic governor of the auxiliated verb, the latter is the semantic head, in that it carries the lexical meaning of the verb group. In Sequoia-Deep as well, auxiliaries are considered syntactic dependents of the non-finite verb. Same with functional conjunctions: the way they are represented in AnCora 2006 allows for linking directly verbs and their object(s), which do have a direct semantic relation. At the syntactic level, though, the conjunction (as its name indicates), is the element that connects both groups. And even when the conjunction is not functional, as it is the case with coordinations, it is a conjunction and should be used to link elements together in the syntactic tree. In Figure 2.11, we illustrate what we consider a truly syntactic annotation of the examples shown in Figure 2.4.



(a) Modal verb ('cannot be')



(b) Nominal group with anteposed coordinated adjectives ('small and medium enterprise')



(c) Functional conjunction ('she believes necessary that he speeds up')

Figure 2.11: Left:semantics-oriented / Right:syntax-oriented annotations

Second, the edge labels can mix semantics and syntax (i) at the syntactic level and (ii) at the semantic level, which has consequences for the clarity and transparency of each tagset.

Some syntactic edge labels in PTB/PB/NB encode semantic information. Thus, the preposition *through*, on line 13 of Figure 2.2 on page 31, is annotated as *MNR* of its governor *gain*, i.e. a circumstantial carrying the meaning of manner. Further tags of this kind are, for instance, TeMPoral, LOCation, and PuRPose. All of these circumstantials behave in English syntactically in the same way; hence, their syntactic annotation should be identical. As a consequence, the tags do not reflect the level of idiosyncrasy of the syntactic analysis. Consider, for instance, the case of the *NMOD* relation, which links a noun to any modifier, be it a determiner, an adjective, a numeral, a relative or a PP. For example, a numeral can combine with a determiner, but it is impossible to combine two determiners. Syntactic tags should reflect this kind of difference instead of using different relations to annotate constructions with the same syntactic properties (e.g. circumstantials or appositions), based on their divergent meanings. This problem can actually be quite easily overcome in the case of *MNR*, since it is trivial to generalize the aforementioned tags and use only one syntactic label for all circumstancials, but we believe that by doing so, PTB/PB fails to offer a clear and motivated point of view on English syntax. The same criticism is valid for, e.g., the Stanford annotation scheme.

There is semantics in syntax, but there is also syntax in the semantic annotation, in which some edge labels clearly encode syntactic information. In Sequoia-Deep, for instance, even though the canonical GFs encode predicate-argument relations, the edges receive a label which is identical to the ones which stand for grammatical functions: a noun is the *subject* of an adjective, which means that this noun is its first argument. In addition, for every *subject* edge between an adjective and a noun, there is an edge *modifier* in the opposite direction which maintains the syntactic function of the adjective to the noun, creating cycles in the annotation. In PropBank, a relation such as *AM-MNR* in line 24 of Figure 2.2 implies that the adverb *significantly* is a “modifying argument” of the predicate *expand.01*, ignoring the fact that such an adverb is itself a semantic predicate which takes as argument its syntactic governor. Along the same lines, in line 21, the *R-A1* relation indicates that the semantic argument is a “relative” argument, in the sense that the relative pronoun is co-first argument of *expand.01*, whereas *expand.01* only has one first argument at the semantic level. *AM-...* or *R-...* edges actually reflect the syntactic structure of the sentence, not its semantic structure.

Another confusion induced by the semantic edge label nomenclature is the unjustified distinction between internal and external argument labels, a syntactic notion derived from the Government and Binding framework (Chomsky, 1993). According to the PropBank annotation guidelines, “A0 arguments are the arguments which cause the action denoted by the verb, either agentively or not, as well as those which are traditionally classified as experiencers, i.e. the arguments of stative verbs such as *love*, *hate*, *fear*. *A1* arguments, on the other hand, are those that change due to external causation, as well as other types of patient-like arguments." (Babko-Malaya, 2005). Thus, *Gen-Probe* is *A1* of *expand.01* because it is the entity which "changes due to external causation". As a consequence, *A1* sometimes stands for the first argument of a predicate, but sometimes it is used to annotate a second argument of a predicate (e.g. *knowledge* in line 12 of Figure 2.2). For the sake of consistent and transparent predicate-argument structure, the distinction between *A0* and *A1* should be abandoned.

2.3.1.2 Coexistence of nodes of different levels of abstraction in a same structure

The semantic annotation in PropBank contains not only semantic but also syntactic nodes. For instance, relative pronouns are annotated at the semantic level, in spite of being pure syntactic elements, as are all pronouns (they have no own meaning since their antecedent carries it). Similarly, syntactically *governed* (i.e. required by their governor) prepositions or conjunctions such as *that* and *of*, respectively on lines 6 and 16 in Figure 2.2, receive a semantic arc, whereas the actual semantic arguments are respectively *gain.02* and *Gen-Probe*. Thus, it is not always easy to recover the actual predicate-argument structure (see Section 3.5).

One interesting example from the Spanish corpus AnCora shows a hybrid annotation of morphology and syntax: a verb and a postponed clitic such as *comerlo* lit. 'eat.it', a very productive construction in Spanish, appear as one single node in the syntactic representation, while it should be split into two functional nodes, the verb and the clitic object pronoun.[13]

In the ISST, the functional syntactic representation contains significant information related to predicate-argument structure. Thus, it combines the criticisms we just made on the PTB for edge labels and nodes: (i) by ignoring functional words in the syntactic representation, one cannot account for all syntactic idiosyncracies of the Italian language, and (ii), by annotating predicate-argument structures with functional syntactic labels, one has to make compromises as far as the purity or the representation is concerned. For instance, a subject of an active or a passive verb receives the same label in both cases, and even though the diathesis of the verb is encoded in the annotation.

---

[13] PTB/PB actually split those morphological groupings: *don't=do+n't*.

The Stanford collapsed dependencies aim at making the syntactic tree more "semantic", and for that, the authors preferred structural consistency to linguistic motivations in the choice of what to collapse: all prepositions were converted to edges, for instance, in order to avoid making a difference between prepositional constructions. But as a result, the information in the regular syntactic tree and the collapsed one is exactly the same. In addition, functional words such as auxiliaries are maintained in the collapsed annotation, and from that perspective, only a part of the job of making the representation less close to the surface is done.

2.3.2 Incompleteness of annotations

At the syntactic and semantic levels, the annotations are often incomplete. This is partly due to the confusions mentioned in the previous subsection, but also to some annotation policies. For instance, as it is the case for the PTB/PB/NB, ISST and Stanford, for instance, the semantic annotation does not form a connected structure, because only nominal and verbal predicates are annotated. This is a problem from the perspective of NLG since the algorithms generating from semantic representations must be able to search through an entire structure, which is impossible if some nodes are disconnected. However, this choice is understandable since the other semantic predicates (adjectives, adverbs, numbers, etc.) can be identified in the syntactic structure together with their arguments, which generally are their syntactic governors; that is, if one wants a connected semantic structure, it is obtainable provided an extra mapping step, but the problem is that this is not trivial nor will the final structure be flawless (see evaluation in Section 3.5).

From the perspective of NLG, apart from PDT, the presented annotations also lack two important types of data: communicative and coreferential structures. Communicative structure features—such as theme/rheme, perspective, emphasis, given/new, etc. (Mel'čuk, 2001)- are crucial for NLG since they directly influence the syntactic organization of sentences. They can only be partially derived from the syntactic annotation (see Section 3.5)—which is why they should be explicitly annotated on the semantic layer. Coreferential structure is what controls pronominalization at the syntactic level. In PTB/PB/NB though, it is handled for relative pronouns (cf. *which* in Figure 2.2). For instance, in the sentence *The Japanese government has stated that it wants 10% to 11% of its gross national product to come from biotechnology products*, the two pronouns *it* and *its* are annotated as arguments of *wants* and *product* respectively. In both cases, the argument should be *the Japanese government*, but due to the introduction of syntactic nodes at the semantic level this is not how it is done. A coreference structure, which not only links a pronoun with its antecedent but also nouns that co-refer, would allow the retrieval of this information.

Seen from our point of view, the deeper layer of the PDT could be more abstract: the fact that the representation is already tree-like means that the sentence structure is already in place, in other words, syntactic choices have already been made. We believe that an abstract structure should be freed from such considerations, so that the algorithms which produce syntactic structure from those abstract representations also learn to build the internal structure of the sentence and of its components.

2.3.3 Manual workload

Annotating a corpus on several layers is a very tedious task, which can involve an important number of persons over a large period of time. Even though it is possible to use morpho-syntactic taggers and syntactic or semantic parsers in order to pre-process the structures of each layer, a manual revision cannot be avoided in order to ensure a reliable annotation. For instance, the first version of the PDT was partially automated (Panevová et al., 1999), but it involved the work of up to 17 persons at the same time over a period of five years; the three layers of the 40,000 sentences were annotated separately. For this reason, very few good quality multi-layered corpora are available nowadays. As for the DELPH-IN annotation, it has been realized mainly automatically, but suffers for its format, which does not make it easy to process; the existing conversion from the original format can reduce the quality of the annotation.

Our objective is to reduce as much as possible human intervention while maintaining a very high quality of annotation. In the next chapter, we show that thanks to the theoretical framework that we use and the currently available tools, it is possible to annotate, at least partially, some parts of the corpus automatically and with very good quality.

CHAPTER 3

Multilevel corpus annotation: the AnCora-UPF corpus

In this chapter, we report on the work that has been carried out on the annotation of a corpus which is suitable for our experiments. In Section 3.1, we describe the theoretical framework which underlies our annotation scheme. Section 3.2 details the choices that we make with respect to each of the four layers of the Spanish corpus, and Section 3.3 exposes the criteria used to define the dependency relations of the surface-syntactic layer, which is the most important in our scheme. In Section 3.4, we explain how the multilayer annotation task has been carried out, and finally, in Section 3.5, we show that a it is possible to obtain automatically a similar corpus from existing resources.

3.1 Theoretical framework

Our annotation model is strongly influenced by the Meaning-Text Theory (Mel'čuk, 1988). The MTT model supports fine-grained annotation at the three main levels of the linguistic description of written language: semantics, syntax and morphology, while facilitating a coherent transition between them via intermediate levels of deep-syntax and deep-morphology; such a smooth transition is especially relevant to NLG since we defined deep NLG as a sequence of mappings between an abstract representation and a text. In total, thus five strata are foreseen. At each stratum, a clearly defined type of linguistic phenomena is described in terms of distinct dependency structures.

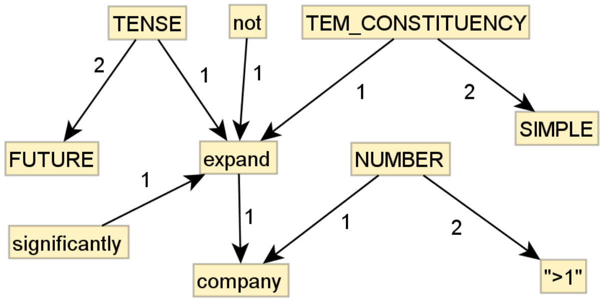
Semantic Structures (SemSs) are predicate-argument structures in which the relations between predicates and their arguments are numbered in accordance with the order of the arguments.[1]
Deep-syntactic structures (DSyntSs) are dependency trees, with the nodes labeled by meaningful ("deep") lexical units (LUs) and the edges 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).
Surface-Syntactic Structures (SSyntSs) are dependency trees in which the nodes are labeled by open or closed class lexemes and the edges by grammatical function relations of the type *subject*, *oblique_object*, *adverbial*, *modifier*, etc.
Deep-Morphological Structures (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 a precedence relation ('b(efore)' in our examples) is defined and which are grouped in terms of constituents.
Surface-Morphological Structures (SMorphSs) are chains of inflected word forms, i.e., sentences as they appear in the corpus, except that orthographic contractions still did not take place. For illustration, consider the representation of the sentence *The companies won't expand significantly* for each MTT-level in Figure 3.1.

The MTT provides a framework for annotation and for transition from a layer to another, but it does not offer particular guidelines, except at the deep-syntactic level, which is the only level in which both nodes and relations are precisely described. At the semantic layer, the set of relations is universal (numbers for argument slots); as for nodes, as long as they their own meaning, they can be part of the structure. At the surface-syntactic layer, if the nodes are clearly defined—all the words or parts of words which have a function in the sentence—, the set of dependency relations is not; we only encode in the dependencies "objective" syntactic properties of the studied language, in our case, Spanish. At the morphological levels, the set of morpho-syntactic attributes associated to the nodes and the morphological interactions between the latter are also designed with respect to the studied language.

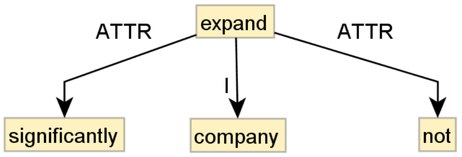
As became clear above, the rich stratification facilitates a clear separation

---

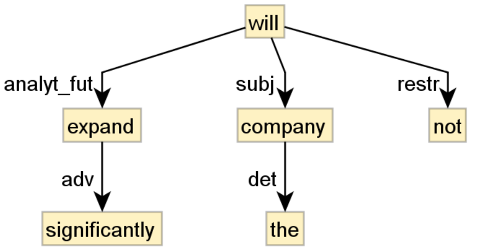
[1] The communicative structure can be superimposed on the semantic structures; see (Bohnet et al., 2013) for automatic annotation of communicative structure on SemSs.



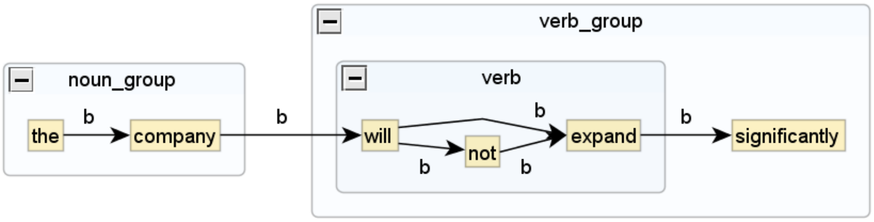
(a) Semantic Structure (SemS)



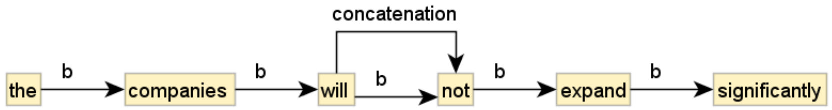
(b) Deep-Syntactic Structure (DSyntS)



(c) Surface-Syntactic Structure (SSyntS)



(d) Deep-Morphological Structure (DMorphS)



(e) Surface-Morphological Structure (SMorphS)

Figure 3.1: The variety of linguistic structures in an MTT-model

of different types of linguistic phenomena and thus a straightforward handling of various NLP-applications. However, this is not to say that our annotation is the only possible one. For instance, the T-layer in Prague Dependency Treebank corresponds, roughly, to MTT's DSyntS & SemS; its A-layer, to MTT's SSyntS & DMorphS; and its M-layer, to MTT's SMorphS. Another possible theory candidate for multilayered annotation would be Lexical Functional Grammar (LFG). LFG's two main structures— f- and c-structure—are complementary but of the same abstraction (namely, syntax), while we view all levels as differing with respect to their abstraction of the linguistic description. This differentiation can be an advantage from the viewpoint of generation. The HPSG framework (Pollard, 1994) does not seem close to ours at first view, since it is phrase-based; still, it has been shown that the output of the various Resource Grammars and the Minimal Recursion Semantics representations can be mapped to typical binary dependency representations (Ivanova et al., 2012), resulting in structures close to our surface-syntax and semantics. The Discourse Representation Theory (DRT, (Kamp and Reyle, 1993)) is also quite similar to the semantic layer of the MTT[2]; in the framework of the Groningen Meaning Bank (Basile et al., 2012), Discourse Representation Structures have been automatically derived from a phrase-based annotation (namely, from Combinatory Categorial Grammar (Steedman, 1996)). Finally, apart from the problems mentioned in Section 2.3, the PTB/PB/NB annotation is somehow comparable to the MTT layers: the PTB corresponds to the SSynt and DMorph layers, while PB and NB form structures which are very close to DSyntSs.

Dependency-based annotation schemes all encode the same information: morpho-syntactic features, word order, functional dependencies, and sometimes argumental dependencies and co-reference resolution. Since our annotation scheme also contains this information, equivalent annotations for other theoretical frameworks can be easily derived from our representations, and our representations can be derived from them without major problems. Phrase-based annotations encode phrase-structures instead of functional dependencies, but, as shown by Gaifman (1965), the latter can be derived from the former. Widely used algorithms such as the one described in (Johansson and Nugues, 2007) confirm that constituencies can be quite safely mapped to dependencies. On the other side, Bosco (2007) and Bos et al. (2009) have already performed the opposite experiment with good results, which shows

---

[2] One notable difference is that some functional words appear in Discourse Representation Structures.

that dependencies can also be mapped to constituencies (respectively Penn TreeBank- and CCG-like).[3]

Hence, MTT in general has one considerable advantage of being somehow equivalent to other types of annotations. But in addition, the MTT model has the property of being transductive (Kahane, 2003), which means that it also provides the instruments for mapping the representation at a given level to the representations at the adjacent levels. This has two crucial consequences as far as corpus annotation is concerned:

- annotating two consecutive strata makes the automatic derivation of a broad-coverage mapping grammar for generation or analysis between those two levels possible; such mapping grammars are an essential component of MTT-based text generation, parsing, paraphrasing, and machine translation.
- starting from a given stratum and a manually created mapping grammar (the coverage does not need to be broad at first), the annotations at the adjacent strata can be easily obtained, which can on their turn be used to derive the annotations at the next strata, and so on. That is, with a corpus of SSyntSs, it is straightforward to derive parallel corpora of DSyntSs and SemSs using for instance a graph transducer.

The second point is particularly relevant, given that corpus annotation is an extremely demanding task; it allows us to reduce the process of annotation to a minimal manual revision of automatically created structures, as shown in Section 3.4.

3.2 The layers of our annotation

Our annotation intends to avoid the problems mentioned in Chapter 2 in a similar way to the PDT, that is, ensuring (i) that a level of representation does not leak onto another one, and (ii) that the annotation is somehow complete in order to allow for easy automatic processing at every layer. We annotated four different layers on top of the sentence level: morphologic, surface-syntactic, deep-syntactic, semantic. In the following, all structures are formally defined following Mel’čuk and Wanner (2006).

---

[3] Note that (i) the dependencies obtained from constituencies cannot be very fine-grained, because many syntactic properties are not encoded in a simple phrase structure; and (ii) that the dependency-to-constituency conversion is so far not made without loss either, as discussed in (Bohnet and Seniv, 2004).

3.2.1 Morphological layer

Features	**Possible values**
dpos	A, Adv, N, V
spos	adjective, adverb, auxiliary, conjunction, copula, determiner, foreign_word, formula, interjection, interrogative_pronoun, noun, number, percentage, preposition, pronoun, proper_noun, punctuation, relative_pronoun, roman_numeral, verb
pos	CC, CD, DT, IN, JJ, N, NN, NP, PP, RB, SYM, UH, VB, VH, VV, WP, formula
id	1 to ∞
surface form	any
lemma	any
gender	C, FEM, MASC
number	PL, SG
mood	IMP, IND, SUBJ
person	1, 2, 3
tense	FUT, PAST, PRES
finiteness	FIN, GER, INF, PART

Table 3.1: Morpho-syntactic features

Definition 3.1 (Morphological Structure, *MorphS*)**.** *Let L_s, G_{sem} and R_{ssynt} be three disjunct alphabets, where L_s is the set of surface lexical units of a language $\mathcal{L}$, F_{morph} is the set of morpho-syntactic features, and P is the precedence relation.*

A MorphS of $\mathcal{L}$, S_{Morph}, is a 4-tuple over $L_s \cup F_{morph} \cup P$ of the following form:

$$S_{Morph} = \langle N, P, \lambda_{l_s \to n}, \gamma_{n \to g} \rangle$$

where

- *the set N of nodes and the directed arcs P form a chain of elements (with a source node n^s and a target node n^t defined for each precedence arc),*

- $\lambda_{l_s \rightarrow n}$ *is a function that assigns to each* $n \in N$ *an* $l_s \in L$,
- $\gamma_{n \rightarrow g}$ *is a function that assigns to the name of each LU associated with a node* $n_i \in N$, $l_i \in \lambda_{n \rightarrow g}(N)$, *a set of corresponding morpho-syntactic features* $F_t \in F_{morph}$.

Surface lexical units are all the items of the vocabulary, in other words, all words as they appear in any monolingual dictionary, and their inflected variants. As for proper nouns, we took the decision not to join them as a single entity. Instead, *Barack Obama* or *Banco de España* 'Bank of Spain' are respectively left as two and three tokens at the Morph layer. In Table 3.1 all possible values of the morpho-syntactic features used in our annotation are detailed.

PoS	**spos**
CC	**conjunction**
CD	cardinal number
DT	determiner
IN	**conjunction** preposition
JJ	adjective
NN	common noun
NP	proper noun
PP	personal pronoun
RB	adverb
SYM	punctuation percentage
UH	interjection
VB	**auxiliary** copula
VH	**auxiliary**
VV	verb
WP	interrogative pronoun relative pronoun
Formula	formula
-	foreign word

Table 3.2: Correspondences between *PoS* and *spos* tagsets

In addition to features such as gender and number, we use three different tagsets for Part-of-Speech: a coarse-grained one, *dpos*, which contains only 4 classes, and two fine-grained ones: *pos* and *spos*. The difference between *pos*, a subset of the well-known Tree-Tagger annotation scheme (Santorini, 1990), and *spos* seems minor, but is not meaningless, since it has an important impact on the results of some parsing experiments reported upon in Chapter 5. Table 3.2 shows these discrepancies: four *PoS* tags have been split into two (*IN*, *SYM*, *VB*, *WP*), while two *spos* tags (namely *conjunction* and *auxiliary*, in bold in the table) correspond to twice as many PoS tags. Table 3.2 allows for visualizing the difference between the two fine-grained PoS tags.

3.2.2 Surface-syntactic layer

Definition 3.2 (Surface-Syntactic Structure, *SSyntS*). *Let L_s, G_{sem} and R_{ssynt} be three disjunct alphabets, where L_s is the set of surface lexical units of a language $\mathcal{L}$, G_{sem} is the set of semantic grammemes, and R_{ssynt} is the set of names of surface-syntactic relations (or grammatical functions).*

An SSyntS of $\mathcal{L}$, S_{SSynt}, is a quintuple over $L_s \cup G_{sem} \cup R_{ssynt}$ of the following form:

$$S_{SSynt} = \langle N, A, \lambda_{l_s \to n}, \rho_{r_s \to a}, \gamma_{n \to g} \rangle$$

where

- *the set N of nodes and the set A of directed arcs (or* branches*) form an unordered dependency tree (with a source node n^s and a target node n^t defined for each arc),*
- *$\lambda_{l_s \to n}$ is a function that assigns to each $n \in N$ an $l_s \in L$,*
- *$\rho_{r_s \to a}$ is a function that assigns to each $a \in A$ an $r_s \in R_{ssynt}$,*
- *$\gamma_{n \to g}$ is a function that assigns to the name of each LU associated with a node $n_i \in N$, $l_i \in \lambda_{n \to g}(N)$, a set of corresponding grammemes $G_t \in G_{sem}$.*

The nodes at this layer have a one-to-one correspondence with the nodes of the morphological level. The 48 surface-syntactic dependency relations (*DepRels*) used for the annotation of this layer[4] are given and briefly ex-

---

[4] So far, we do not have special relations for ellipses; we add a syntactic empty node in order to deal with "impossible" dependencies, so far, only in case of what is commonly known as gapping and right-node-raising.

plained in Tables 3.3 and 3.4.[5] Depending on the application, one can need more or less tags in the annotation; for this reason, we allow for tuning the granularity of the tagset, as shown in Section 3.3.3. The rudimentary semantic grammemes set is a subset of the morpho-syntactic features shown in Table 3.1; it contains number and tense.

For the design of the syntactic tagsets, we use almost exclusively syntactic and morpho-syntactic criteria, which are based on objective properties of Spanish, and are thus theory-independent[6]. Another important point is that unlike the large number of annotation schemes, we do not subdivide a relation into more specific relations based only on the Part-of-Speech of the dependent. Instead of dividing a generic noun modifier relation *modif* into *a-modif*, *n-modif*, *p-modif*, etc. for respectively adjectival, nominal and prepositional modifiers, we split it according to syntactic criteria such as *is there an agreement? can the dependent move on the other side of its governor in the sentence?* etc. The reason for that is that the PoS information is already accessible in the syntactic tree: even if a dependency was unlabeled, one could retrieve the PoS of the dependent simply by looking at its morpho-syntactic features. What is encoded in the dependency relations are syntactic properties which cannot be inferred in a straightforward way from the morpho-syntactic annotation. Obviously, this does not mean that we do not use the PoS; on the contrary, it can be a very important property because it has a direct correlation with other properties (for instance, syntactic agreement between two words only happens with certain types of PoS: noun and adjective or determiner, verb with noun, etc., but not between a noun and a preposition for instance). All the criteria used for obtaining those labels are detailed in Section 3.3.

The annotation of a corpus with SSyntSs also follows a number of basic rules which mainly originate from the notion of dependency and the characteristics of an SSyntS in MTT:

(i) 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 auxiliary *ha* and not of the participle *dejado*, unlike

---

[5] Examples are given in Appendix A, together with the list of properties of each relation.

[6] This statement is equivalent, for instance, to the Minimal Structural Complexity criterion used for the design of the Chinese Sinica Treebank (Huang et al., 2000)

DepRel	Distinctive properties
abbrev	abbreviated apposition
abs_pred	non removable dependent of a noun making the latter act as an adverb
adv	invariant adverbial
adv_mod	adverbial dependent of a verb, which agrees with a sentence-external element
agent	promotable dependent of a participle always introduced by *por*
analyt_fut	preposition *a* governed by future auxiliary
analyt_pass	non finite verb governed by passive auxiliary
analyt_perf	non finite verb governed by perfect auxiliary
analyt_progr	non finite verb governed by progressive auxiliary
appos	non-abbreviated apposed element
attr	right-side modifier dependent of a noun
aux_phras	multi-word marker
aux_refl	reflexive pronoun depending on a verb
bin_junct	for binary constructions
compar	complement of a comparative adjective/adverb, introduced by a governed preposition
compl1	non-removable adjectival object agreeing with subject
compl2	non-removable adjectival object agreeing with direct object
compl_adnom	prepositional dependent of a stranded determiner
conj	any complement of a conjunction which is not of the coordinating type
coord	between a conjunct and the element acting as coordination conjunction
coord_conj	complement of a coordination conjunction
copul	cliticizable dependent of a copula
copul_clitic	cliticized dependent of a copula
det	non-repeatable left-side modifier of a noun, which is the target of an agreement

Table 3.3: 48 dependency relations used at the surface-syntactic layer (1)

the direct object: Gerard←*subj*-ha-*analyt_perf*→dejado-*dobj*→piso-*det*→su. The reason for this is that the syntactic agreement holds between the auxiliary and the subject; the relation between the non-finite verb and the subject is more of a semantic one.

(ii) One lexeme corresponds to one and only one node in the tree: as a consequence, a lexeme with more than one function or multiple lexemes aggregated in a single word should be considered with attention. For instance, in a relative clause, 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*

DepRel	**Distinctive properties**
dobj	verbal dependent that can be promoted, or cliticized with an accusative pronoun
dobj_clitic	accusative clitic pronoun depending on a verb
elect	dependent of a comparative adjective/adverb or number, not introduced by a governed preposition
iobj	verbal dependent that cannot be promoted but can be cliticized with a dative pronoun
iobj_clitic	dative clitic pronoun depending on a verb
juxtapos	links two unrelated groups of the same sentence
modal	non-removable, non-cliticizable infinitive verbal dependent
modif	for an adjective which agrees with its governing noun
num_junct	right-side numerical dependent of another number
obj_copred	adverbial dependent of a verb, which agrees with an object
obl_compl	right-side dependent of a non-verbal element, introduced by a governed preposition
obl_obj	dependent of a verb that cannot be demoted, promoted or cliticized, but is introduced by a governed preposition
prepos	complement of a preposition
prolep	for clause-initial accumulation of elements with no connectors
punc	for non-sentence-initial punctuation signs
punc_init	for sentence-initial punctuation signs
quant	numerical dependent which controls the number of its governing noun
quasi_coord	for coordinated elements with no conjunction or comma
quasi_subj	a “fake” subject next to a grammatical subject
relat	finite verb introduced by a relative pronoun and that modifies a noun
relat_expl	adverbial finite clause introduced by a neutral relative pronoun
sequent	non-removable right-side coordinated adjacent element
subj	dependent that controls grammatical agreement on its governing verb
subj_copred	adverbial dependent of a verb, which agrees with the subject

Table 3.4: 48 dependency relations used at the surface-syntactic layer (2)

duerme 'Igor, who sleeps' is represented as Igor-*relat*-[que]→duerme and duerme-*subj*→ que. Another example: two lexemes which occur within the same word have to be separated, so that each can be assigned its own function. For example, *del* 'of.the' has to be split into *de+el* 'of+the', *haberlo* 'have.it' into *haber+lo* 'have+it', etc. Empty lexemes are not considered at the superficial layer: in case of 0-subject, which is frequent in Spanish, the verb remains without a subject in the surface-syntactic tree.

(iii) Subordinating and coordinating conjunctions, as their names indicate, are syntactic connectors between two groups, and for this reason, depend on the governor of the first group, and govern the one of the second group. This hierarchical approach is considered more syntactic than other approaches that directly link the governors of the two groups, making the conjunction only a dependent of the first one. Indeed, in addition to syntactically linking two groups, a conjunction can impose a grammeme on its dependent: e.g., *cuando llegó* 'when [he/she] arrived', *cuando* 'when' requires that the following main verb be finite, which we believe indicates a strong syntactic link between the two lexemes. The only exception to this is the relative pronouns, as discussed above.[7]

Since we derive our annotation from an existing one which is not necessarily in conformity with these rules (see Section 3.4), special attention must be paid to these phenomena when performing the mapping between one and the other.

3.2.3 Deep-syntactic layer

Definition 3.3 (Deep-Syntactic Structure, *DSyntS*)**.** *Let L_d, G_{dsynt} and R_{dsynt} be three disjunct alphabets, where L_d is the set of deep lexical units (LUs[8]) of a language $\mathcal{L}$, G_{dsynt} is the set of semantic grammemes, and R_{dsynt} is the set of names of deep-syntactic relations.*

---

[7] Interestingly, it has been shown recently that the parsing accuracy is optimal when a statistical dependency parser is trained on material annotated with these principles (Schwartz et al., 2012).

[8] The difference between surface and deep lexical units is that the latter (i) do not include purely functional nodes and (ii) are disambiguated. Note that this is the theoretical view, and that the disambiguation of the LUs is not absolutely necessary for the purposes of our experiments in Chapter 4, which is why we do not make this issue a priority in this thesis.

An DSyntS of $\mathcal{L}$*,* S_{DSynt}*, is a quintuple over* $L_d \cup G_{dsynt} \cup R_{dsynt}$ *of the following form:*

$$S_{DSynt} = \langle N, A, \lambda_{l_s \to n}, \rho_{r_s \to a}, \gamma_{n \to g} \rangle$$

where

- *the set* N *of nodes and the set* A *of directed arcs (or* branches*) form a dependency tree (with a source node* n^s *and a target node* n^t *defined for each arc),*
- $\lambda_{l_s \to n}$ *is a function that assigns to each* $n \in N$ *an* $l_s \in L_d$*,*
- $\rho_{r_s \to a}$ *is a function that assigns to each* $a \in A$ *an* $r_s \in R_{dsynt}$*,*
- $\gamma_{n \to g}$ *is a function that assigns to the name of each LU associated with a node* $n_i \in N$*,* $l_i \in \lambda_{n \to g}(N)$*, a set of corresponding grammemes* $G_t \in G_{dsynt}$*.*

The deep-syntactic dependency relations available are given and shortly explained in Table 3.5.

DepRel	**Short description**
I	first argument
II	second argument
III	third argument
IV	fourth argument
V	fifth argument
VI	sixth argument
APPEND	backgrounded modifier
ATTR	regular modifier
COORD	coordination
coref	coreference relation (optional)

Table 3.5: 9 dependency relations used at the deep-syntactic layer

By its nature, the deep-syntactic layer could be called *shallow semantic.* The deep-syntactic dependency relations are language-independent and thus also more abstract than the surface-syntactic ones. In our corpus, the deep-syntactic layer contains less nodes than the surface-syntactic one since all punctuation signs and functional nodes (governed prepositions and conjunctions, auxiliaries, determiners) have been removed. Removing functional nodes from deeper annotations has two advantages from the perspective of NLG:

- it makes the annotation less syntactic and forces the generators trained on it to introduce non-meaningful nodes;
- it allows the generators to deal with different surface realizations when several are possible (e.g. *give something to Mary* vs *give Mary something*).

The idea is that from the perspective of Natural Language Generation from abstract structure, the system will only have access to non-linguistic data (see, for example, (Bouayad-Agha et al., 2012c,b), in the football and the air quality domains respectively). This implies that a system that generates statistically from those abstract representations MUST be able to learn when to introduce all the functional words, and thus that a corpus claimed to be suitable for training NLG tools takes this into account. Having in parallel two layers, one with all the words, and one without the functional words, is one way to provide the basis for statistical models.

DSynt Feature	**Possible values**
coref_id	1 to ∞
definiteness	DEFINITE — INDEFINITE — N/A
id_ssynt1	1 to ∞
id_ssynt2	1 to ∞
id_ssyntn	1 to ∞
tem_constituency	SIMPLE — PROGRESSIVE — PERFECT — PERFECT PROGRESSIVE
voice	ACTIVE — PASSIVE

Table 3.6: Additional grammemes used in the deep-syntactic annotation

In the following, we discuss more in detail when and how nodes are removed or transformed, and their possible correspondence with the deep-syntactic grammemes.

(a) Governed elements
The presence of a governed preposition is imposed by the subcategorization ("valency") characteristics of its head, as, e.g., the appearance of "TO" in *give TO your friend*), in the sense that the preposition "TO" is the only possible preposition to express the meaning of 'give'.

"TO" in itself is here void of own meaning and should thus not appear in a semantics-oriented structure. This is different in, for instance, *he reads ON the sofa*, where "ON" is not at all required by *read*, but indicates a location. Without "ON", the meaning of the sentence would be perceived more incomplete than *give TO your friend* without "TO". In some cases, a required preposition can also bear its own (or at least a piece of its own) meaning: in *to go INTO/IN FRONT OF/NEXT TO/... your house*, the preposition is meaningful, even though it is governed, and thus, as ON in the previous example, should appear in the deep-syntactic structure. The dependents involved in the following SSynt DepRels are concerned: *agent*, *compar*, *conj*, *dobj*, *iobj*, *obl_compl*, *obl_obj*. We also exclude from the DSyntS all subordinating conjunctions *que* 'that' when they introduce an argument of a predicate.

(b) Auxiliaries
An auxiliary is a syntactic element and should not appear as such in a deep structure. However, in an appropriate syntactic configuration, it expresses semantic grammatical meanings, namely tense (past: *haber* 'have' + past participle; future: *ir* 'go' + preposition *a* 'to' + infinitive), aspect (progressive: *estar* 'be' + present participle) or voice (passive: *ser* 'be' +past participle). These meanings must be reflected in the deep-syntactic structure. For this purpose, corresponding attributes can be introduced to capture tense, aspect and voice: *time* for tense (with as possible values *present*, *future* and *past*); *tem_constituency* for aspect (with as possible values *simple*, *progressive*, *perfect*, *perfect progressive*).[9]; finally, the attribute *voice* with the values *active* or *passive*.[10]

(c) Determiners
Definite *el* 'the', indefinite *un* 'a(n)' and demonstrative *este* 'this', *ese*/*aquel* 'that' determiners should also be excluded from the deep-syntactic annotation: they indicate degrees of givenness and from that respect account for a part of the communicative and coreference structures. The determiners can be replaced by attribute/value pairs

---

[9] See (Comrie, 1976, p.3) for definition of aspect as "different ways of viewing the internal temporal constituency of a situation".

[10] Interestingly, as already mentioned, there are two ways to realize passive voice in Spanish, one with an auxiliary, one with a reflexive pronoun. Hence the mapping between a deep-syntactic verb with *voice=passive* and its superficial counterpart is not straightforward.

on the governing noun in syntax (*givenness=given*, *givenness=new*, etc.). However, there is so far no reliable way to identify automatically the givenness of nouns, since there is no systematic correlation between the presence or the absence of a determiner on a noun and its givenness. A manual annotation of givenness is needed in order for a generator to learn correctly how to deal with their introduction in a superficial structure. For this thesis, we only annotate definiteness on nouns in order to encode the presence, at the surface, of a definite or indefinite determiner. All other determiners—demonstrative, possessives, etc.—are kept in the deep annotation. A possessive can receive any edge in deep-syntax since it can stand for a modifier (*su silla* 'his/her chair') or an argument (first argument: *su traducción* 'his/her translation (of something)' second argument: *su elección* 'his/her election (by someone)', etc.) of the governing noun. All other determiners receive the DSynt DepRel *ATTR*.

(d) Relative Pronouns
Relative pronouns with antecedent should be substituted by their antecedent in the deep-syntactic structure, and a coreference link added between the two.

While some nodes are absent from our deeper annotation, some nodes which do not appear at the superficial layer are shown in the deep-syntactic structure. Indeed, when there is an empty subject, an unlabeled node with the person and number information has to be the first argument of the verb (since the verb takes that information for being inflected); when necessary, this new node may need to be linked to another one with a coreference relation. The coreference relation is described as optional in Table 3.5, since it can be represented as a relation or/and as an attribute (*coref_id* in Table 3.6) with the same value on each of the coreferring nodes.

Finally, the deep-syntactic grammemes comprise the features of the more superficial layers (see Table 3.1), and additional features only used at this level, shown in Table 3.6. The feature(s) *id_ssynt* store the correspondence between the DSynt node and one or more SSynt nodes. The other grammemes, *definiteness*, *tem_constituency* and *voice* are abstract ways of representing the functional nodes at this level.[11]

---

[11] For more technical details on the DSyntS, see (Mel'čuk and Wanner, 2006).

3.2.4 Semantic layer

Definition 3.4 (Semantic Structure, *SemS*)**.** *Let* S *and* R_{sem} *be two disjunct alphabets, where* S *is the set of semantemes of a language* $\mathcal{L}$ *and* R_{sem} *is the set of names of predicate-argument relations.*

An SemS of $\mathcal{L}$*,* S_{Sem}*, is a quintuple over* $S \cup G_{sem} \cup R_{sem}$ *of the following form:*

$$S_{Sem} = \langle N, A, \lambda_{l_s \to n}, \rho_{r_s \to a}, \gamma_{n \to g} \rangle$$

where

- *the set* N *of nodes and the set* A *of directed arcs (or* branches*) form a dependency tree (with a source node* n^s *and a target node* n^t *defined for each arc),*
- $\lambda_{l_s \to n}$ *is a function that assigns to each* $n \in N$ *an* $l_s \in S$*,*
- $\rho_{r_s \to a}$ *is a function that assigns to each* $a \in A$ *an* $r_s \in R_{sem}$*,*
- $\gamma_{n \to g}$ *is a function that assigns to the name of each LU associated with a node* $n_i \in N$*,* $l_i \in \lambda_{n \to g}(N)$*, a set of corresponding identification features* $G_t \in G_{sem}$*.*

In this work, the nodes at the semantic level are the same as the nodes at the deep-syntactic level. In other words, in the framework of this dissertation, we use as semantic node labels words rather than semantemes, i.e., we do not carry out the tasks of generalizing and disambiguating the word labels. Different words which have identical meanings keep different labels in semantics, and isomorphic words with different meanings remain ambiguous. Generalization or disambiguation are very important tasks, and they cannot be avoided on the long term in order to get an acceptable corpus, but they are not crucial for our experiments.

We do not keep any lexical or grammatical information at this level: no PoS[12], no gender, no person, no surface-form, no mood, no finiteness, no agreement information. On the contrary, only what we consider "semantic" information is kept. We add six different meta-nodes in order to encode information stored as feature/values in the previous layers, or to connect non-predicative units to the rest of the structure:

---

[12] Note that since we do not generalize meanings, the node labels at the semantic level most of the time indirectly indicate the PoS...

(1) *ROOT*: it has only one argument, and simply indicates which node of the semantic structure is the communicatively dominant node; it directly relates with the main node of the sentence, that is, the main verb of the main clause.

(2) *TENSE*: the first argument is by convention the event, and the second argument indicates if it was in the past, is in the present, or will be in the future.

(3) *NUMBER*: following the same model as *TENSE*, the first argument is the quantified entity, and the second argument is the value *SINGULAR* or *PLURAL*. Note that this concerns semantic number only, and not grammatical number: nouns keep their number in the SemS, but adjective or determiners for instance do not, since they only get their number (and gender) by an agreement rule imposed by Spanish syntax. [13]

(4) *TEM_CONSTITUENCY*: again, the first argument is by convention the event, and the second argument indicates if it is progressive, perfect, both or none.

(5) *ELABORATION*: this meta-node is used to connect to the semantic graph these non-predicative nodes whose corresponding deep-syntactic nodes receive the relations ATTR or APPEND. The node *ELABORATION* takes the dependent as its second argument, and the governor as its first one. When there is a predicative attribute, such as *este* 'this' in *este chico* 'this boy', the syntactic governor is its first argument and, therefore, no *ELABORATION* node is needed to connect it to the semantic structure. However, in some appositive constructions, for instance, the apposed element cannot take its DSyntS governor as argument: in *Pipo, mi perro* 'Pipo, my dog', we have Pipo-*ATTR*→perro, and *perro* is not a predicate. An extra node is therefore needed to connect it to the structure. The attributive relation in this case stands for the fact that the governor is the name given to the dependent; subsequently, we should have at the semantic level 'Pipo'←*2*-NAME-*1*→'perro'. However, since we did not undertake a manual revision of the semantic layer as yet, we use for now

---

[13] Lexical number should equally not be represented in the SemS: for instance, the number of the word *paro* 'unemployment' in Figure 3.8 is *lexical*; it cannot vary. As a result, it should not be an argument of a node *NUMBER*. However, in this version of the corpus, all nouns receive a number.

the generic label *ELABORATION* in all cases, considering that the second argument somehow elaborates on the first one.

(6) *POSSESS*: when the possessive determiner is not an argument, it usually stands for a possession relation between the governor, which will be the second semantic argument, and the dependent, which will be the first one.

The predicates described in (1–6) are called “meta-” because they encode information that is necessary at the semantic level of representation, but that should not be considered the same as other nodes, since they should not be realized as words in the final sentence. If we would not differentiate one type of node from the other, a generator could end up generating sentences like “The document, the number of which is singular, suggests in a present tense that ...”.

Finally, the semantic features are (i) a unique individual ID, (ii) an ID indicating the correspondence with DSynt nodes, and (iii) an attribute encoding the definiteness of some nouns.

Technically, all this information is still not sufficient in order to reconstruct the sentence as it was on the surface: as mentioned in the Introduction, the communicative structure also constrains the realization of the semantic graph, but this is out of the scope of this thesis, since we see the fact of superimposing a communicative structure on a semantic network as a different task.[14]

DepRel	**Short description**
1	first argument
2	second argument
3	third argument
...	...
n	nth argument

Table 3.7: Predicate-argument relations used at the semantic layer

The nomenclature of predicate-argument relations is given in Table 3.7; an example of each annotation level is shown in Figures 3.6 and 3.8 in Section 3.4. Note that unlike the semantic annotation of PTB/PB/NB, the

---

[14] We performed however some experiments on English in which we use very basic communicative structure, see Section 3.5.

semantic structure in MTT has transparent semantic frames, in the sense that no difference is made between external or internal arguments.

3.3 Our methodology for surface-syntactic annotation

As already mentioned in Section 3.1, since the theoretical model we use is transductive, the annotation of the different layers can be seen as a sequential task. One stratum can thus be the starting point of the whole process. Since nowadays the most common annotation that would be rich enough for our purposes is that of syntactic structures, our general methodology was to annotate first surface-syntactic structures together with morpho-syntactic features, and from that derive the deeper layers of our annotation (deep-syntax and semantics). In other words, the surface-syntactic layer is the most important layer since it will strongly influence the manual workload required for the annotation of the deeper strata. A careful annotation of this layer ensures easy annotation of the other layers.

There are three 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 is extremely costly and not conceivable given the other options.

B Using SSyntS-dependency parsers. Kakkonen (2005), for instance, 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, at the beginning of this project, not a single SSyntS-parser was available.[15] A solution could have been to use another dependency parser, for instance, the JBeaver parser (Herrera et al., 2007a) or an early version of Freeling parser (Atserias et al., 2006), mapping the obtained parse trees onto SSyntSs. However, the error rates of these parsers were quite high. In addition, their output structures are very different from SSyntSs—which implies additional noise during the phase of mapping.

---

[15] It is actually the resources developed in the framework of this thesis which led to the first SSyntS parser of Spanish.

C Starting from an existing treebank, mapping the original annotation (constituency or dependency trees) onto SSyntS dependency trees. For instance, the Spanish constituency corpus Cast3LB (Civit and Martí, 2004) has already been used by Herrera et al. (2007b) for the derivation of dependency annotations. Bohnet (2003) performed a similar task on the German corpus NEGRA (Brants et al., 2003), and more recently, Johansson and Nugues (2007) established the reference conversion for English. The quality of the conversions is usually very high. It is also possible to skip this step and use an existing dependency treebank which has already undergone manual revision, which is what we decided to do with the AnCora-DEP-ES (Taulé et al., 2008).

With a seed corpus at hand, it is only a matter of post-editing the structures it contains. For this, we defined a detailed annotation scheme that allows for relatively easy dependency relation identification, based on easy-to-use criteria. In this section, we first detail the steps prior to the proper annotation, that is, how to identify a dependency and its direction, and then we explain the deep motivation behind our criteria and how to distinguish between different labels.

3.3.1 Establishing the presence and direction of a dependency between two nodes

The central question faced during the establishment of the SSyntS (as in Definition 3.2) for each sentence of the corpus under annotation is related to:

- the elements of A: when is there a dependency between two nodes labeled by the LUs l_i and l_j and what is the direction of this dependency,
- the elements of R_{ssynt}: what are the names of the dependencies, how they are to be assigned to $a \in A$, and how they are to be distinguished,

or, in short, to the determination of SSynt-Dependencies. It is more likely that there is a dependency between two units (i) if the position of one unit in the sentence is established with respect to the other unit (e.g., a determiner has to be positioned *before* the noun it determines, hence a probable dependency between the two), (ii) if the two units have a prosodic link with one another, and (iii) if a unit triggers agreement on the other. The direction of the dependency, i.e., the fact that one unit is the syntactic governor of the other, depends on other parameters, in particular on (i) the passive valency of the group they form together (e.g., a noun and a determiner have the distribution of a noun, so the noun is more likely to be the governor), and (ii) which unit is involved in grammatical agreement with external elements (e.g. mil_{SG} personas_{PL}, lit. ‘one-thousand persons’ as a subject will have a plural agreement on a verb, making the noun prone to be the governor of its quantifier). We address this in terms of Mel’čuk (1988)’s corollaries (pages 129–144).

Corollary 3.5 (Dependency between nodes)**.** *Given any two unordered nodes n_1 and n_2, labeled by the LUs l_1 and l_2 respectively, in the sentence S of the corpus, there is a dependency between n_1 and n_2 if either*

(a) in order to position l_i in S, reference must be made to l_j, with $i, j = 1, 2$ and $i \neq j$ (linear correlation criterion)

and

(b) between l_i and l_j or between syntagms of which l_i and l_j are heads ($i, j = 1, 2$ and $i \neq j$), a prosodic link exists (prosodic correlation criterion)

or

(c) l_i triggers agreement on l_j ($i, j = 1, 2$ and $i \neq j$) (agreement criterion)

Thus, in *Juan ha dormido bien hoy* ‘John has slept well today’, *Juan* has to be positioned before the auxiliary *ha* (or after in a question) and a prosodic link exists between *Juan* and the syntagm headed by *ha*. This means that *Juan* and *ha* are likely to be linked by a dependency relation. *Bien* has to be positioned compared to *dormido* (not compared to *ha*), hence there is a dependency between *dormido* and *bien*.

With respect to agreement, we see that the verb is *ha* and not *han*, as it would be if we had *los chicos* ‘the boys’ instead of *Juan*. This verbal variation in person, which depends on the preverbal element, implies that a dependency links *Juan* and *ha*. This criterion is not sufficient on its own: for instance, in a construction involving a copula, an adjective copular element agrees with the subject, even though it is governed by the verb (criteria (a) and (b)): *Los chicos*$_{MASC-PL}$ *están dormidos*$_{MASC-PL}$ 'The boys are sleepy'.

Once the dependency between two nodes has been established, one must define which node is the governor and which one is the dependent, i.e., the direction of the SSynt arc that links those two nodes. The following corollary handles the determination of the direction of the dependency:

Corollary 3.6 (Direction of a dependency relation)**.** *Given a dependency arc a between the nodes* n_1 *and* n_2 *of the SSyntS of the sentence S in the corpus,* n_1 *is the governor of* n_2*, i.e.,* n_1 *is the source node and* n_2 *is the target node of a if*

(a) the passive valency (i.e., distribution) of the group formed by the LU labels l_1 *and* l_2 *of* n_1/n_2 *and the arc between* n_1 *and* n_2 *is the same as the passive valency of* l_1 *(passive valency criterion)*

or

(b) l_1 *as lexical label of* n_1 *can be involved in a grammatical agreement with an external element, i.e., a label of a node outside the group formed by LU labels* l_1 *and* l_2 *of* n_1/n_2 *and the arc between* n_1 *and* n_2 *(morphological contact point criterion)*

If neither (a) nor (b) apply, the following weak criteria should be taken into account:

(c) if upon the removal of n_1*, the meaning of S is reduced and NOT restructured,* n_1 *is more likely to be the governor than* n_2 *(removal criterion),*

(d) if n_1 *is not omissible in S, it is more likely to be the governor than* n_2 *(omissibility criterion),*

(e) if l_2 *as label of* n_2 *needs ("predicts")* l_1 *as label of* n_1*,* n_2 *is likely to be a dependent of* n_1 *(predictability criterion).*

As illustration of the passive valency criterion,[16] consider the group *the cats*. It has the same distribution as *cats*: both can be used in exactly the

---

[16] For the definition of the notion "passive valency", see (Mel'čuk, 1988).

same paradigm in a sentence. On the other side, *the cats* does not have the distribution of *the.* We conclude that *cats* is the head in the group *the cats.* It is important to note that, for instance, in the case of prepositional groups, the preposition does not have its own passive valency since it always needs an element directly after it. It does not prevent the passive valency criterion from applying since, e.g., the distribution of *from* [*the*] *house* is not the same as the distribution of *house.* It is the presence of the preposition that imposes on the group a particular distribution.

The morphological contact point criterion is used as follows: considering the pair *sólo felinos* in *sólo felinos ronronean* 'only felines$_{PL}$ purr$_{PL}$', *felinos* is the unit which is involved in the agreement with an external element, *ronronean.* As a consequence, *felinos* is more prone to be the governor of *sólo.*

For the other criteria, consider *Juan es el mejor ciclista del mundo* 'John is the best cyclist in the world'. During the first step, we identified that there was a dependency between *mejor* and *del mundo*, since they can form a prosodic group, for instance. Removing *del mundo* only reduces the meaning of the sentence, it does not change it (removal criterion); this makes it more likely to be a dependent. On the other side, removing *mejor* makes the presence of *del mundo* impossible, in other words, this word is not omissible in the sentence, (omissibility criterion), which indicates a strong possibility of directed dependency from *mejor* to *del mundo.* Finally, the determiner *el* "predicts" a noun, in that it most of the time needs a noun to be used in a sentence (predictability criterion). This criterion often gives the same results as the omissibility criterion; it is a little less easy to apply, but it can be used in more contexts. For instance, the group *del mundo*, as any other prepositional group, predicts another element, making it likely to be a dependent when involved in a dependency relation; this goes along the lines of the omissibility criterion, as described above. However, the latter cannot be used in this sentence in order to define the dependency direction between the determiner and its governing noun, since the noun can perfectly be elided in this context: *Juan es el mejor del mundo* 'John is the best in the world'. As for the predictability criterion, it still indicates that the determiner is more likely to be the dependent of the noun. For more details see (Mel'čuk, 1988).

3.3.2 Criteria used for labeling dependencies

When an annotator manages to identify pairs of governor and dependent, an important part remains, which is to label the arc linking them with the correct dependency. We started with the following statement: the granularity of the scheme should be balanced in the sense that it should be fine-grained enough to capture language-specific syntactic idiosyncrasies, but be still manageable by the annotator team.[17] The latter led us target a set of around 50 SSynt DepRels (also abbreviated *SSyntRels*).

In order to be able to identify a particular dependency, the annotator must be provided with some well-defined criteria. In the following, we discuss briefly the parameters which we take into account when it comes to selecting these criteria. First of all, they should be applicable to the largest number of cases possible. For instance, a governor and a dependent always have to be ordered, so a criterion implying order can be applied to every relation whatever it is. One advantage here is to keep a set of criteria of reasonable size, in order to avoid to have to manage a large number of criteria which could only be applied in very specific configurations. The other advantage in favoring generic criteria is that it makes the classification of dependency relations more readable: if a relation is opposed to another using the same set of criteria, the difference between them is clearer.

Second, when applying a criterion, an annotator would rather *see* a modification or the presence of a particular feature. Indeed, we try to use only two types of criteria: the ones that transform a part of the sentence to annotate—promotion, mobility of an element, cliticization, etc.—, and the ones that check the presence or absence of an element in the sentence to annotate (is there an agreement on the dependent? does the governor impose a particular preposition? etc.). In other words, we avoid semantically motivated criteria, and instead favor criteria that are related to the *syntactic* behavior of the nodes.[18] The main consequence of this is the absence of

---

[17] We are thinking here of decision making and inter-annotator agreement rate.

[18] Actually, not only syntactic information constrains the syntactic behaviors of the sentence units, in particular the order between them. Lexical information, for instance, is also of first relevance: some units have individual behaviors which can be different from the rest of the PoS class they belong to (see for instance the case of modifiers in Section 3.3.3). Because, unless we provide extremely large and complex guidelines, it is not possible to give criteria that take into account individual properties of all lexical units, some criteria have to be left unspecified for some relations, and therefore are not really useful when it comes to take a decision on a dependency relation tag. We think however that this does not prevent the annotators to eventually find a correct relation.

opposition complement/attribute as discriminating feature between syntactic relations, unlike what has been done with the available MTT SyntRel sets—see e.g. (Iordanskaja and Mel'čuk, 2009). Note that although we use only syntax-based criteria, we try to account for the predicate-argument relations as much as possible since the goal is to obtain a SemS eventually; most relations actually end up having a direct correlation with complements or attributes in DSynt (*cf* Table 3.14 on page 102).

Finally, once the annotator has applied a criterion, she must be able to make a decision quickly. This is why almost all criteria involve a binary choice.

All of the resulting selected criteria presented below have been used in one sense or the other in the long history of grammar design. However, what we believe has not been tackled up to date is how to conciliate in a simple way fine-grained syntactic description and large-scale application for NLP purposes. In what follows, we present a selection of the most important criteria we use in order to assign a label to a dependency relation; it includes the possibility of cliticization, promotion or demotion, the topological and agreement properties and restrictions of the governor-dependent pair, the omissibility of the dependent, the type of dependency, the required presence of functional elements, the presence of a comma, as well as criteria related to the Part-of-Speech of the governor and the dependent. Then, we show how we use them for the annotation of a Spanish corpus with different levels of detail.[19]

3.3.2.1 Cliticization

Cliticization refers to the possibility for the dependent to be replaced or duplicated by clitic pronouns and refers thus only to elements for which the order between the verbal governor and its dependent is not restricted. For instance, the relation *indirect object* allows cliticization, as opposed to the *oblique object* that does not:

Miente '[He] lies'–*iobj*→ *a* 'to' *Carla* 'Carla'.
Le miente, lit. 'to-her [he] lies.' '[He] lies to her.'
A Carla le miente, lit. 'to Carla to-her [he] lies .' '[He] lies to Carla.'

Invierte '[He] invests'–*obl_obj*→ *en* 'in' *bolsa* 'stock-market'.
**La invierte*, lit. 'in-it [he] invests'
**En bolsa la invierte*, lit. 'in stock-market in-it [he] invests'

---

[19] Values of all criteria for each dependency are shown in Appendix A.

3.3.2.2 Promotion/demotion

Promotion and demotion refer to the possibility of moving an argument up (respectively down) the ordered syntactic actant list (*subject* > *direct object* > *indirect object* > . . .). Thus, the dependent of the relation *direct object* can be promoted to the dependent of the relation *subject* in a passive sentence, and, from the opposite point of view, the subject can be demoted to the dependent of the relation *agent* in a passive sentence:[20]

Juan compuso las canciones 'Juan wrote the songs.'
Las canciones fueron compuestas por Juan 'The songs were written by Juan.'

Cliticization and promotion/demotion is obviously only possible if the governor is a finite verb. From this perspective, these criteria do not seem to be very "generic", that is, widely usable; but since there are many different relations that can hold on a verb, this is not totally true. In addition, they are very efficient from the other perspective, which is that they are easy to apply.

3.3.2.3 Type of linearization

Some relations are characterized by a rigid order between the governor and the dependent (in any direction), whereas some others allow more flexibility with respect to their positioning. Thus, e.g., some relations that connect an auxiliary with the verb imply a fixed linearization: the auxiliary (governor) always appears to the left of the verb (dependent):

He comido mucho, lit'[I] have eaten a-lot.'
**Comido he mucho*, lit '[I] eaten have a-lot.'

On the other hand, even if Spanish is frequently characterized as an SVO language, the relation *subject* does allow flexibility between the governor and the dependent:

Subject on the left: *Juan come manzanas*, lit. 'Juan eats apples'
Subject on the right: *Come Juan manzanas*, lit. 'Eats Juan apples'
Subject on the right: *Come manzanas Juan*, lit. 'Eats apples Juan'

For this criterion, the dependent should be moved with all its own dependents, and the movement is restricted to the phrase its governor is the head

---

[20] In Spanish, only direct objects and agents can be promoted; English, for instance, also allows for the promotion of indirect objects: *John sent a postcard to Paul* vs. *Paul was sent a postcard by John.*

of. Other elements of the sentence may be moved for the movement to be possible; for instance, the copulative element *alto* ‘tall’ can only be moved to the other side of the copula if the subject makes the opposite movement:

Juan es alto, lit. ‘Juan is tall’
**Juan alto es*, lit. ‘Juan tall is’
Alto es Juan, lit. ‘Tall is Juan’

Note that moving the subject, for instance, after its verb is very marked in Spanish, but this kind of consideration is not taken into account: the fact that it is *syntactically* possible to move the subject is sufficient in order to consider the *subj* relation as allowing for a flexible ordering between governor and dependent.

Given that it is possible to apply this criterion to all relations, the linearization criterion is very relevant to our purposes.

3.3.2.4 Canonical order

As just stated, some relations are more flexible than others with respect to the order between governor and dependent. When the order is not restricted, there is usually a canonical order. Thus, although it is possible to have a postverbal subject, the canonical order between the subject and the verb is that the former occurs to the left of the latter. On the other hand, the relations introducing the non-clitic objects have the opposite canonical order, i.e., the object appears to the right of the verb (see *Juan come manzanas* above).

3.3.2.5 Adjacency to the governor

There are some relations that require that the governor and the dependent are adjacent in the sentence, and therefore only accept a very restricted set of elements (namely, other adjacent elements) to be inserted between them. On the other hand, there are some other relations that allow a larger variety of elements to appear between governor and dependent. The fact that a governor has to keep a dependent very close to itself is a distinctive syntactic feature. All the relations involving clitics belong to the first type, while a relation such as *determinative* belongs to the second type:

Cada día, lo miraba, lit. ‘Every day, it [I] watched.’
**Lo cada día miraba*, lit. ‘It each day [I] watched.’
‘I watched it every day’.

Un hombre muy bueno, lit. ‘A man very good’
Un muy buen hombre, lit. ‘A very good man.’
‘A very good man.’

3.3.2.6 Dependent omissibility

This syntactic criterion is defined within an “out-of-the-blue” context, given that otherwise it is very difficult to determine whether a dependent is omissible or not: it is always possible to create pragmatic contexts in which the dependent can be perfectly omitted. There are two cases: on the one hand, relations such as *prepositional* always require the presence of the dependent and, on the other hand, relations as *modifier* do not require the presence of the dependent. Consider:

Juan viene para ‘Juan comes to’*–prepos→* *trabajar* ‘work’.
**Juan viene para*, lit. ‘Juan comes to.’

Tiene ‘[He] has’ *sillas* ‘chairs’*–modif→* *verdes* ‘green’. ‘[He] has green chairs.’
Tiene sillas. ‘He has chairs.’

Note that the meaning of every lexical unit must be maintained for this criterion to be applied: if, after removing a dependent, the meaning of one of the remaining units must be changed for the sentence to remain grammatical, it means that the dependent cannot be removed.

3.3.2.7 Left Dislocation=strong focalization

Left dislocation (with or without comma) is used in order to distinguish in some cases an object from an adverbial. If the dislocated element seems strongly focalized when it is positioned to the left of its governor, the relation is more probably an object. When applying this criterion, the dependency relation should still stand after the dislocation. For instance, it seems possible to dislocate the apposed element in the case of apposition: *el presidente Obama* ‘the president Obama’ gives *Obama, el presidente* ‘Obama, the president’, but in the latter, there would be an inversion of dependency, in that *el presidente* ‘the president’ would now be the apposed element. As a result, the relation *apposition* does not react positively with respect to this criterion.

3.3.2.8 Agreement

Agreement appears when governor and dependent share morphological features such as gender, number, person, etc., which one of the elements passes to the other. Agreement actually depends on two parameters. On the one hand, the target of the agreement must have a PoS which allows agreement. On the other hand, the dependency relation itself must allow it. For example, the *copulative* relation allows agreement, but if the dependent is not an adjective, it is not mandatory; cf.: *Pedro y Carla son relajados* 'Pedro and Carla are relaxed$_{\mathrm{PLU}}$' as opposed to *Pedro y Carla son una pareja* 'Pedro and Carla are a couple$_{\mathrm{SING}}$'. Inversely, the past participle in the perfect analytical construction is intrinsically prone to agreement (as the second example that follows shows), but the relation does not allow it: *Carla está perdida* 'Carla is lost$_{\mathrm{FEM}}$' as opposed to *Carla ha perdido* 'Carla has lost$_{\mathrm{noFEM}}$'. If a relation licenses agreement, this does not mean that any dependent must have agreement, but, rather, that there is agreement if the dependent allows it.

There are different types of agreements allowed by a syntactic relation:

- dependent agrees with governor (i.e., the dependent is the target of the agreement): *sillas* 'chairs'–*modificative*→ *rotas* 'broken$_{\mathrm{FEM.PL}}$',
- governor agrees with dependent (i.e., the dependent controls the agreement): *Juan* 'Juan' ←*subject*–*viene* 'comes',
- dependent agrees with another dependent: *Juan* 'Juan' ←*subject*–*parece* 'seems'–*copulative*→ *enfermo* 'sick$_{\mathrm{MASC.SG}}$'.

Saying that a relation controls or is the target of an agreement means that if one draws the path from the target of the agreement to its controller, one has to go through that DepRel.

When there is an agreement, secondary criteria concerning the type of inflection of the agreeing element can be applied. Thus, in some cases the agreement can vary, in other cases it cannot. Consider for instance the opposition between *subject* and *quotative subject*: the *subject* relation triggers agreement that depends on the morphological (number) and lexical (person, gender, number for proper nouns) properties of the dependent; on the other hand, the *quasi-subject* relation always triggers the same agreement on the verb, 3rd person masculine singular (Variant Inflection criterion).

Further remarks on agreement

- In the case of the *quantificative* relation, there is strictly speaking no agreement, since a singular element such as *mil* 'thousand' triggers a plural form on the noun. We are then forced to consider two types of number/gender/person, one **internal**, for the interactions with the dependents, and one **external**, for the interactions with other nodes, as it happens with coordinate structures (see the example 'Pedro and Carla are relaxed $_{\text{PLU}}$' above). The latter is what is considered when applying this criterion.

- We consider that there is an agreement when a unit takes some feature(s) from another non-coreferring unit (which can be present or absent in the sentence). On the other hand, pronouns for instance can be inflected but it comes from intrinsic properties of the antecedent. In this case, there is no agreement strictly speaking, but only inflection; that's why the _clitic relations are defined as having no agreement involved.

3.3.2.9 Coordinate vs Subordinate

Coordinate structures have always been a problem as far as syntactic annotation is concerned. Numerous studies deal with their syntactic representation: is there a direct dependency between the conjuncts? or should the conjunction stand between the conjuncts in the dependency tree? or are all conjuncts dependent of the conjunction? or is coordination to be annotated on another level, with another dimension in the annotation? All approaches have their pros and cons, and it is not the objective of the thesis to bring a definitive answer to this question. Following the traditional trend in the Meaning-Text Theory, we chose a hierarchical representation of coordinate structures: a coordinating conjunction links two conjuncts together, being governed by the first one and governing the second one. That is, the internal structure of a coordination is the same as the one of subordination. Since it is easy for an annotator to identify if a construction is of the coordinate or the subordinate type, we use this distinction as a criterion. Only five of the dependency relations are of the coordinating type in our scheme.

3.3.2.10 Governed Preposition/ Conjunction/ Grammeme (P/C/G)

There are some relations that require the presence of a preposition, a subordinating conjunction or a grammeme. For instance, the relation *oblique object* implies the presence of a preposition without meaning to introduce the dependent (*invierte* ***en*** *la Bolsa* '[he/she] invests in the stock market'), and the relation *subordinate conjunctive* requires the presence of a feature in the verb indicating that it is finite.

There are simple tests that help decide if an element is syntactically required or not:

- A governed preposition is less prone to be changed, whereas a non-governed preposition usually can be replaced by any other preposition (obviously, not with the same meaning). It is even more drastic for case or finiteness: the case or the finiteness cannot be changed at all.
- If (i) a dependent is part of the definition of its governor, and (ii) a preposition/conjunction/grammeme cannot be avoided in order to introduce said dependent, then this preposition/conjunction/grammeme is "governed". By "being in the definition", we mean "being necessarily mentioned for the meaning of he governor to be complete".

In order to apply this criterion, it is important to ensure that the governor is the one that imposes the governed element. It can happen that an element is required by something else than the governor: (1) by the dependent, for instance for introducing a place name as an adverbial (***en*** *España* 'in Spain'); (2) by the relation, for instance *compl_adnom*, where the preposition is not free but is not required either by the governor *la* ***de*** *la falda azul*, lit. 'the-one of the dress blue', 'the one with the blue dress'. As for the governed grammeme, it is easier to see if a verb has to be finite in its position, for instance, as it is the case after a subordinating conjunction. In order to know the case, replacing the dependent by a personal pronoun indicates the case: NOM=yo,él; ACC=me,lo; DAT=me,le; ABL=mí,él.

3.3.2.11 Presence of a punctuation marks

For this criterion, the annotator should check if there is (or not) a comma, a colon, a semi-colon, a dash, a parenthesis, etc. between the governor and the dependent, considering only those two elements, that is, excluding all types of interpolated groups.

3.3.2.12 Other criteria

As last resort option, we also use some very specific criteria: is the dependency projective? is the dependent the same entity as one of the governor's arguments? can an element be further away from the governor than the dependent? is there a conjunction around? is the dependent an abbreviated form of the governor?

3.3.2.13 Criteria used as filters during the annotation process

Some properties, such as the possible Part-of-Speech of the governor and dependent of a relation, are not purely syntactic but can help ruling out some labels, and others are only useful in order to check if the chosen dependency is the correct one, e.g. the repeatability of a dependency or its prototypical dependent. In this subsection, we give more details about these properties.

Part-of-Speech of the Governor
The actual PoS of the governor is relevant in that there are very few syntactic dependents that behave the same with governors of different syntactic categories once a certain level of detail has been reached in the annotation. In other words, the PoS of the governor has an important impact on the rest of the syntactic properties described in this section; hence, we believe that this should be somehow reflected in the edge label. This criterion helps ruling out some dependencies which are described as not allowing a particular PoS as its governor.

Part-of-Speech to consider (there can be intersections between the PoS classes):

- Adjective: an adjective which is not the governor of a noun group; for some relations (compar, compl_adnom, elect), can be restricted to comparative and/or superlative adjectives.
- Adverb: any adverb.
- Conjunction: finite list of conjunctions: *como* 'as', *que* 'that', *si* 'if', *aunque* 'even though', *porque* 'because', *cuando* 'when'.
- Date: are included day and month.
- Determiner: a determiner abandoned by its original governor, that is, a noun is elided in the group.

- Noun: includes nouns and other elements happening in the paradigm of a noun, such as infinitive verb, adjective, number. Is considered a noun any element that can govern a determiner in the configuration of the sentence; excludes pronoun determiners (see Det above).
- Numeral: only numerals; does not include nouns such as *millón* 'million', *mil* 'thousand', *docena* 'dozen', etc.
- Preposition: finite list of prepositions.
- Finite verb: e.g., only verbs with tense.
- Non finite verb: includes infinitives, gerunds, past participles.

Part-of-Speech of the Dependent

Looking at the PoS of a dependent of a relation also aims at eliminating some labels; for example, a preposition or an adverb will never be at the end of an arc labeled "determinative". Here are the details on the PoS we consider during the annotation process:

- Acronym.
- Adjective: see Gov PoS; excludes "pronominal" adjectives.
- Adverb: see Gov PoS.
- Determiner: includes the following determiners: definite (*la*,*el*, and other inflected forms), indefinite (*una*, *un*, *alguna*, *ninguna*, *demasiada*, *tal* etc.), possessive (*mi*, *tu*, *su*, etc.), demonstratives (*este*, *ese*, *aquel*, etc.); excludes "pronominal" determiners.[21]
- Conjunction: see Gov PoS; in the case of the dependent being limited to coordination conjunction, also consider a comma as possible dependent instead of the conjunction.
- Noun: includes all nouns, including this time in addition to infinitives, pronoun determiners and adjectives, personal pronouns and determiners without their head noun.
- Number: see Gov PoS; excludes "pronominal" numerals.

---

[21] It is impossible to have two determiners depending on the same noun; modifiers and quantifiers can combine with determiners.

- Preposition: see Gov PoS.
- Pronoun (clitic): ACC:lo, los, la, las, os, me, te, nos, DAT:le, les, se, os, me, te, nos.
- Finite verb: unlike for Gov PoS, are included here also finite verbs preceded by a relative pronoun without antecedent, which is why sometimes a finite verb can appear in the same paradigm as a noun.
- Gerund verb
- Infinitive verb: excludes "pronominal" infinitives.
- Past participle: excludes "pronominal" participles.

Prototypical dependent

Following Mel'čuk (1988), we consider that every relation must have a prototypical dependent, that is, it should always be possible to replace the dependent by an element of the prototypical Part-of-Speech of the relation. This criterion is more useful for designing the set of dependency relations than for assigning a tag to a relation since it involves a generalization over a large number of cases which are not accessible during the process of annotation. However, it can be used during annotation as well, especially in order to discard/confirm a relation: if a dependent of a SSyntRel cannot be replaced by the prototypical dependent of this relation, then the relation should be changed.[22]

If the replacement has taken place correctly, it is usually not possible anymore to express the replaced element in the sentence (except in the case of clitics, which can duplicate the corresponding object in Spanish). In order to apply this criterion, (i) the annotator should not allow for a pause between the governor and the prototypical dependent, otherwise it's easy to accept the construction as a quasi-coordination; and (ii) the annotator must ensure that the governor keeps exactly the same meaning (paraphrasing the governor with several more abstract meanings can help), which is why this criterion can be quite difficult to apply. Thus, it is recommended to use it in order to confirm a dependency.

---

[22] We noticed that there seems to be a hierarchy in the PoS nomenclature: A→ Adv→ N→ V. This comes from the observation that it is usually possible to find in the position of a prototypical dependent a prototypical dependent from its right, but not the contrary. For instance, when the prototypical dependent is an adjective, one can usually find an adverb, a noun or a verb in the same syntactic role, but when the prototypical dependent is a verb, it can only be a verb.

For three subordinate relations, there is no prototypical dependent. In the case of *compar*, the PoS of the dependent depends on the governor, whereas in the case of *compar_conj* and *coord_conj*, it depends on the governor of the governor, that is, the governor of the coordinative or comparative construction.

Repeatability of dependency relation
Again following Mel'čuk (1988), we use this criterion as an important feature of a dependency relation: a dependency relation should always be repeatable or never be repeatable. But as it is the case for the prototypical dependent criterion, the annotator can only apply this criterion once she has labeled a dependency. This criterion states that (i) if a dependency relation is defined as unrepeatable, (ii) if the annotator identifies this relation between a governor and a dependent, and (iii) if she can see or introduce another dependent on the same governor with exactly the same properties, i.e., with the same dependency relation, then this dependency relation is not the good one, since it is proven repeatable.

For illustration, consider the following example:

(a) the dependency *determinative* is defined as non-repeatable;

(b) the annotator sees the following group *los tres gatos* 'the three cats';

(c) she identifies the dependency between *gatos* 'cats' and *los* 'the' as being *determinative*;

(d) she wants to annotate the dependency between *gatos* 'cats' and *tres* 'three' as *determinative*.

The conjunction of (c) and (d) means that *determinative* would happen twice under the same governor, which contradicts (a) and makes one of the two *determinative* relations wrong and to be reconsidered.

3.3.3 How to use the criteria: Illustration with selected SSynt DepRel

In this section, we illustrate two different ways of using the criteria we determined above. One is based on a hierarchical layout, in which criteria have to be examined one after the other in a given order; the other approach considers no such hierarchy in order to achieve more flexibility. The two approaches correspond to the two distinct uses we make of the criteria:

respectively (i) describing precisely syntactic properties of a DepRel; (ii) allowing the annotator to label the arcs more easily.

3.3.3.1 The hierarchical approach

Criteria for ¬Fix Lin SSyntRel							**SSyntRel**
Clitic	Prom	Quot					dobj_quot
		¬Quot					dobj
	¬Prom	¬Agree	Gov P				iobj
			¬Gov P				copul_quot
		Agree	Target	Sibling	subj		copul
¬Clitic	¬Remov	Agree	Target	Sibling	subj		compl1
					dobj		compl2
	Remov	ProtD N	¬Agree	Gov P			obl_obj
				¬Gov P			quasi_subj
			Agree	Control	Sibling & Governor	Var	subj
						¬Var	subj_quot
		ProtD A	Agree	Target	Sibling	subj	subj_copred
						dobj	obj_copred
					External Elt		adv_mod
		ProtD Adv	Parenthetical				adjunct
			¬Parenthetical				adv

¬: negation of a criterion;
Agree: Dependent is involved in an agreement;
Clitic: Dependent can be replaced by a clitic pronoun;
Control: (IF AGREE) Dependent controls the agreement on another word;
External Elt: (IF TARGET OF AGREE) Dependent agrees with an element which is in another sentence;
Fix Lin: Governor and dependent always in the same order;
Gov P: the dependent is a governed preposition;
Governor: (IF AGREE) Dependent agrees with its governor;
Parenthetical: the dependent is between brackets or dashes
Prom: Dependent can be promoted;
ProtD A: Prototypical dependent is an adjective;
ProtD Adv: Prototypical dependent is an adverb;
ProtD N: Prototypical dependent is a noun;
Quot: Dependent is quoted;
Remov: Dependent can be removed;
Sibling: (IF AGREE) Dependent agrees with one of its siblings;
Target: (IF AGREE) Dependent is the target of the agreement;
subj - dobj: (IF AGREE WITH SIBLING) Dependent agrees with subject or object;

Table 3.8: A partial hierarchy of syntactic criteria

We organized all criteria into a tree-like hierarchy such that if an annotator identifies a pair governor/dependent but wonders which relation holds between the two, she has merely to follow a path of properties that leads to the relation. The order in which the criteria are applied is only important for expressiveness: it allows for keeping the relations that have the same type close to each other in the graphical representation. In this way, differences between similar relations can be visualized very easily.

In this section, we present only a part of the complete hierarchy, namely, the relations governed by a verb which do not impose a rigid order between governor and dependent. We use here nine criteria: (1) removability of dependent, (2) possible cliticization, (3) agreement type, (4) inflection type, (5) PoS of prototypical dependent, (6) promotion/demotion, (7) presence of governed preposition, (8) presence of quotes, and (9) presence of parentheses or dashes. With this level of detail, we obtain sixteen different relations in which verbs are involved; cf. Table 3.8.

By selecting only a few criteria, it is possible to diminish the number of relations and thus to tune the level of detail of the annotation. For example, keeping only four of the nine criteria presented above (i.e., (1) removability of dependent, (2) possible cliticization, (3) agreement type, (5) PoS of prototypical dependent) we end up with only five relations instead of sixteen; see Table 3.9.[23]

Criteria for ¬Fix Lin SSyntRel					**SSyntRel**
¬Fix Lin	Clitic				Obj1
	¬ Clitic	¬ Remov			Compl
		Remov	ProtD N	Agree (control)	Subj
				¬ Agree	Obj2
			ProtD A/Adv		Mod1

¬: negation of a criterion;
Agree: Dependent is involved in an agreement;
Clitic: Dependent can be replaced by a clitic pronoun;
Control: (IF AGREE) Dependent controls the agreement on another word;
Fix Lin: Governor and dependent always in the same order;
ProtD A: Prototypical dependent is an adjective;
ProtD Adv: Prototypical dependent is an adverb;
ProtD N: Prototypical dependent is a noun
Remov: Dependent can be removed;

Table 3.9: A hierarchy with less criteria

This allows for building quite easily full dependency relation hierarchies, using more or less criteria for defining the relations: Table 3.10 displays

---

[23] In Tables 3.8 and 3.9, each cell corresponds to the application of one criterion; the rightmost column contains the SSyntRels. The path from the root of the tree to one leaf thus indicates a list of properties of this relation. Note that not all properties are listed in these tables: for instance, elements that can be cliticized are usually linearized to the right of their governor, that is, have the property *Canonical Order = RIGHT*.

the correspondence between the dependency relations of more or less fine-grained tagsets (from 48 down to 15 tags).[24] For instance, in the 48-tag column, *obl_obj*,*obl_compl* and *agent* stand for various types of prepositional objects; in the 31-tag column, they are gathered together as prepositional objects governed by verbs or nouns and which do not involve an agreement; in the 15-tag column, they are grouped with all objects which do not pronominalize. This hierarchy is similar to the Stanford hierarchies in English (de Marneffe et al., 2006; de Marneffe and Manning, 2008) and Hebrew (Tsarfaty, 2013) for instance, although they do not justify clearly the *syntactic* differences between the different relations, since they use the argument VS modifier opposition as a criterion.

48 Rels	**31 Rels**	**15 Rels**	**48 Rels**	**31 Rels**	**15 Rels**
abs_pred	abs_ pred	NMOD	iobj	iobj	IOBJ
det	det		iobj_clitic	iobj_clitic	
quant	quant		obl_obj	obl_obj	OOBJ
compl_adnom	compl_adnom		obl_compl		
appos	modif		agent		
abbrev			compar	compar	
attr			compl1	compl	
modif			compl2		
relat			elect	elect	
adv	adv	ADV	subj	subj	SUBJ
relat_expl			quasi_subj	quasi_subj	QSUBJ
prolep			conj	conj	PREPOS
adv_mod			coord_conj		
obj_copred			prepos		
subj_copred			coord	coord	COORD
analyt_fut	analyt_fut	AUX	num_junct		
analyt_pass	analyt_pass		juxtapos	juxtapos	
analyt_perf	analyt_perf		quasi_coord		
analyt_prog	analyt_prog		sequent	sequent	BIN
modal	modal		bin_junct	bin_junct	
dobj_clitic	dobj_clitic	DOBJ	aux_phras	aux_phras	NAME
dobj	dobj		aux_refl	aux_refl	AREFL
copul	copul	COPUL	punc	punc	PUNC
copul_clitic	copul_clitic		punc_init		

Table 3.10: Tag groupings for a hierarchy of syntactic tags (Left=top, right=bottom of table)

Another advantage of the hierarchical approach is that it displays clearly what the differences between dependency relations are. For instance, in

---

[24] The most generic tags (15) are labeled in a way that facilitates a comparison with PTB-like edge nomenclature.

Table 3.8 one can see at one glance that the relations *dobj* and *iobj* allow for the dependent to be moved around and cliticized, but that only *dobj* allows for promotion.

However, it is well known that natural languages cannot be described in their entirety by general rules. Languages evolve independently of the rules formulated by linguists with the goal to capture the observed syntax. In other words, all rules have more or less numerous exceptions. As a result, the criterion hierarchy as it has been presented above has its limitations: not all the instances of a DepRel necessarily exhibit all the properties that appear in the path from the root of the criterion tree. For example, if an annotator finds a dependent that has all the properties of an *obl_obj*, with the exception of one—for instance, that this dependent cannot be removed from the sentence it appears in—she will never arrive at the *obl_obj* relation.

One way to avoid this deadlock would be to add a branch in the criterion hierarchy in order to have another path that arrives at *obl_obj* with the property "not removable dependent". But if we do this for each configuration of properties of each DepRel, the resulting hierarchy will be totally unreadable and lose its main purpose. Therefore, we decided to create a complementary approach that considers bags of properties for each DepRel instead of a hierarchy.

3.3.3.2 The bag-of-properties approach

As its name indicates, this approach simply consists in playing with the set of properties associated to each DepRel, without considering that some properties are more important than others. This time, we do not focus on using the properties that differentiate a DepRel from another one; neither do we impose an order in the use of criteria. Instead, we compile an inventory of all the possible values for each criterion for each DepRel. An SQL-based tool allows the annotator to introduce one value for each criterion of her choice, and returns a classification of dependency relations ordered by (i) similarity based on the selected criteria, and (ii), frequency. The idea behind this inventory of values is that whatever the configuration in the sentence to annotate is, the target DepRel appears at (or close to) the top of the list when the annotator introduces the selected criteria.

As a use case, let us consider one particular DepRel and one sentence. In Table 3.11 the properties of the DepRel *modif*, which holds between a noun and a modifying adjective, are shown.[25]

Criterion	**Possible values**
PoS Gov	N | Date
prototypical Dep	Adj
PoS Dep	V_{Part} | Adj
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	Gov
variant inflection	YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

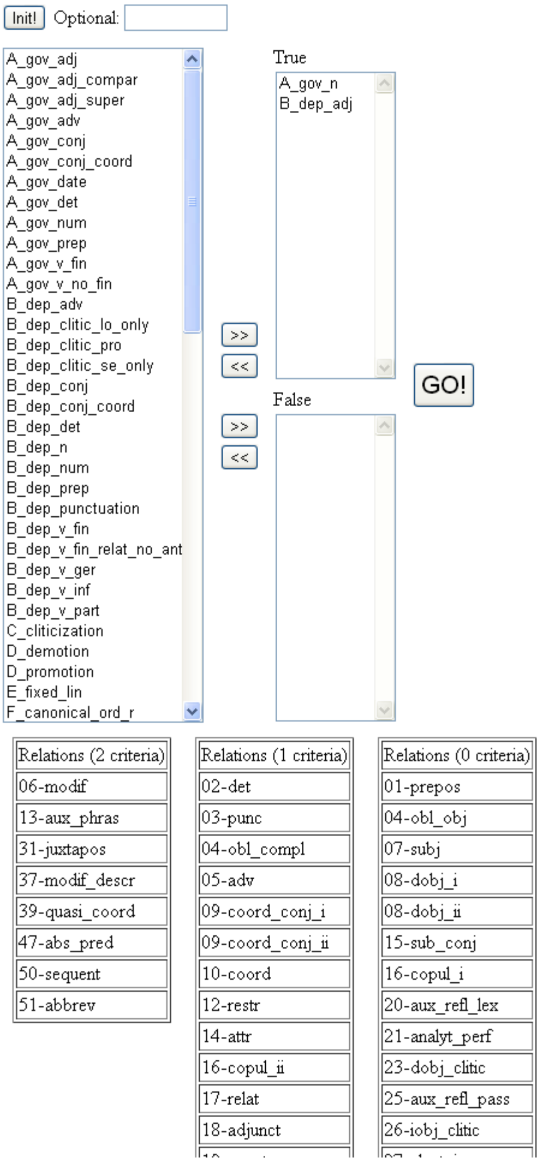
Table 3.11: Distinctive properties of the *modif* SSynt DepRel

In the case of this particular DepRel, the dependent usually appears to the right of its governor, and cannot be moved to the left of it. However, some adjectives (as, e.g., *pequeño* 'small') can appear both to the right and to the left of the governing noun: *niño pequeño* vs. *pequeño niño*, and some can only be found to the left of the noun (cf. quantificative adjectives such as *poco* 'little', which do not behave as numbers: *poco aire* 'little air' vs. **aire poco*, lit. 'air little'). In other words, some lexical properties can overrule syntactic properties. To handle this phenomenon, some criteria can be left unspecified (e.g., in terms of the value N/A for *type of linearization* and *canonical order* in Table 3.11), such that both YES and NO give a match for the DepRel in question. In contrast, if we would use the hierarchical scheme to describe the most probable or prototypical properties of a DepRel, an unconventional construction would erroneously rule out a DepRel (see Section 3.3.3.1).

As for the sentence, let us use *Tiene ojos verdes* '[He] has green eyes'. The

---

[25] The properties of all 48 surface-syntactic relations are shown in Appendix A, together with annotation examples for each of them.



A_gov_adj
A_gov_adj_compar
A_gov_adj_super
A_gov_adv
A_gov_conj
A_gov_conj_coord
A_gov_date
A_gov_det
A_gov_num
A_gov_prep
A_gov_v_fin
A_gov_v_no_fin
B_dep_adv
B_dep_clitic_lo_only
B_dep_clitic_pro
B_dep_clitic_se_only
B_dep_conj
B_dep_conj_coord
B_dep_det
B_dep_n
B_dep_num
B_dep_prep
B_dep_punctuation
B_dep_v_fin
B_dep_v_fin_relat_no_ant
B_dep_v_ger
B_dep_v_inf
B_dep_v_part
C_cliticization
D_demotion
D_promotion
E_fixed_lin
F_canonical_ord_r

True

A_gov_n
B_dep_adj

False

Relations (2 criteria)
06-modif
13-aux_phras
31-juxtapos
37-modif_descr
39-quasi_coord
47-abs_pred
50-sequent
51-abbrev

Relations (1 criteria)
02-det
03-punc
04-obl_compl
05-adv
09-coord_conj_i
09-coord_conj_ii
10-coord
12-restr
14-attr
16-copul_ii
17-relat
18-adjunct

Relations (0 criteria)
01-prepos
04-obl_obj
07-subj
08-dobj_i
08-dobj_ii
15-sub_conj
16-copul_i
20-aux_refl_lex
21-analyt_perf
23-dobj_clitic
25-aux_refl_pass
26-iobj_clitic

Figure 3.2: Sample query in the DepRel identifier tool with two criteria

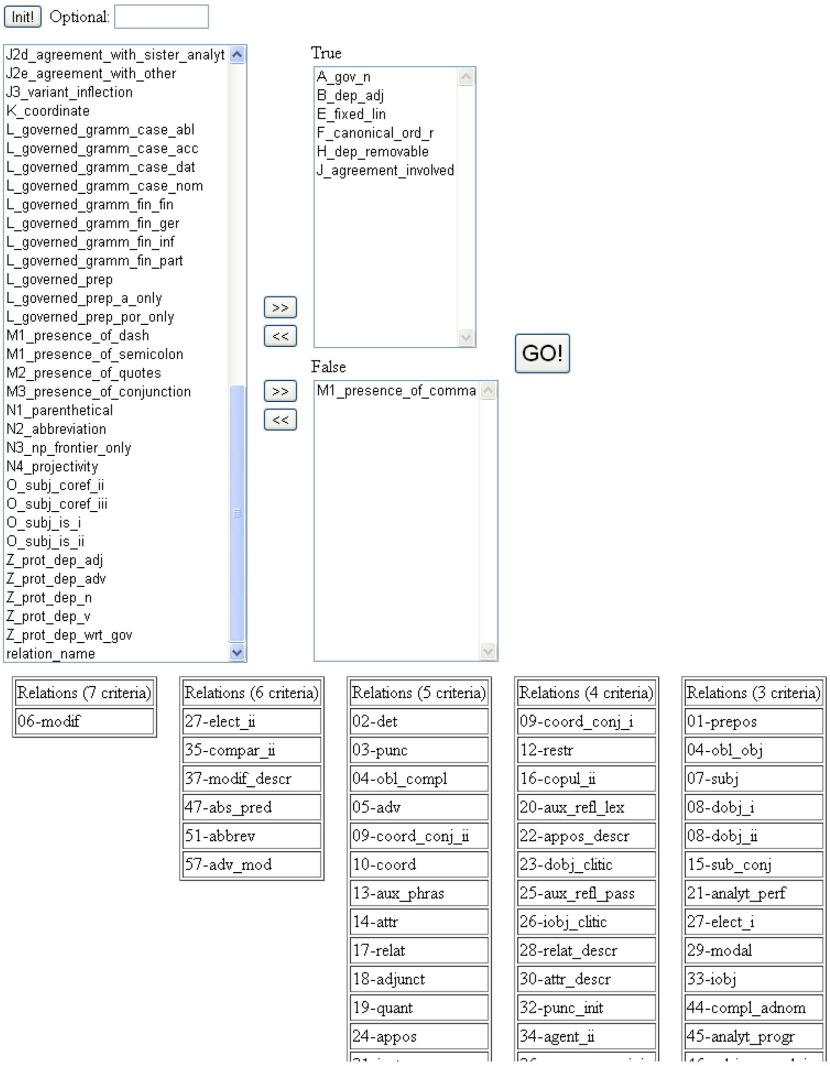
adjective *verdes* 'green$_{PL}$' is positioned with respect to the noun *ojos* 'eyes', more precisely after it (if the noun goes in front of the verb, so does the adjective). *Verdes* forms a prosodic group with *ojos*, and it agrees with it, which indicates a dependency between these two words. The group behaves as a noun, and *ojos* triggers the agreement on *verdes*, which indicates that the latter is the dependent of the relation. In order to label it, an annotator has to perform simple syntactic tests, starting with the indication of the PoS of the governor and the dependent, which are respectively *noun* and *adjective*.[26]

Figure 3.2 shows a screenshot of the result of this query made by the annotator. The tool returns three lists, one with the n DepRels that match the two criteria, namely that noun is the governor and adjective the dependent, one with DepRels that match only one of the two criteria, and one with DepRels that do not match any of the two criteria. Within each frame, the relations are ordered from the most (top) to the least (bottom) frequent. That is, in our example, the most likely label for the query in question is *modif*, while the least probable would be the one at the bottom of the 0-criteria list. The annotator can discard candidates from the most to the least probable, based on the knowledge she has about the labels. She can also refine the query by adding criteria. Figure 3.3 shows a screenshot of the result of such a refined query. In this case, we can see that the annotator considered that it is not possible to move the dependent with respect to the governor (cf. the criterion *fixed_lin*), that the dependent is found to the right of the governor (cf. the criterion *canonical_ord_r*), that it can be removed without causing meaning restructuring nor agrammaticality (cf. the criterion *dep_removable*), that it is involved in an agreement of some kind (cf. the criterion *agreement_involved*), and that there is no comma between the noun and the adjective (cf. the criterion *presence_of_comma* in the "False" column).

At the bottom, we can see that only one relation matches the seven criteria, and that six relations match six criteria. In our example, the correct label is indeed *modif*, but it can happen that the most probable label has to be discarded by the annotator, for instance because no answer would have been given to a criteria which is important for the identification of the DepRel

---

[26] We use letters as prefixes for criteria so as to order them in a way that helps the annotator: the most discriminative and easy-to-use criteria appear first in the list. However, the annotator is free to use the criteria in any order; the output will always be the same.



J2d_agreement_with_sister_analyt
J2e_agreement_with_other
J3_variant_inflection
K_coordinate
L_governed_gramm_case_abl
L_governed_gramm_case_acc
L_governed_gramm_case_dat
L_governed_gramm_case_nom
L_governed_gramm_fin_fin
L_governed_gramm_fin_ger
L_governed_gramm_fin_inf
L_governed_gramm_fin_part
L_governed_prep
L_governed_prep_a_only
L_governed_prep_por_only
M1_presence_of_dash
M1_presence_of_semicolon
M2_presence_of_quotes
M3_presence_of_conjunction
N1_parenthetical
N2_abbreviation
N3_np_frontier_only
N4_projectivity
O_subj_coref_ii
O_subj_coref_iii
O_subj_is_i
O_subj_is_ii
Z_prot_dep_adj
Z_prot_dep_adv
Z_prot_dep_n
Z_prot_dep_v
Z_prot_dep_wrt_gov
relation_name

True
A_gov_n
B_dep_adj
E_fixed_lin
F_canonical_ord_r
H_dep_removable
J_agreement_involved

False
M1_presence_of_comma

Relations (7 criteria)	Relations (6 criteria)	Relations (5 criteria)	Relations (4 criteria)	Relations (3 criteria)
06-modif	27-elect_ii	02-det	09-coord_conj_i	01-prepos
	35-compar_ii	03-punc	12-restr	04-obl_obj
	37-modif_descr	04-obl_compl	16-copul_ii	07-subj
	47-abs_pred	05-adv	20-aux_refl_lex	08-dobj_i
	51-abbrev	09-coord_conj_ii	22-appos_descr	08-dobj_ii
	57-adv_mod	10-coord	23-dobj_clitic	15-sub_conj
		13-aux_phras	25-aux_refl_pass	21-analyt_perf
		14-attr	26-iobj_clitic	27-elect_i
		17-relat	28-relat_descr	29-modal
		18-adjunct	30-attr_descr	33-iobj
		19-quant	32-punc_init	44-compl_adnom
		24-appos	34-agent_ii	45-analyt_progr

Figure 3.3: Sample query in the DepRel identifier tool with seven criteria

in question. In this case, the annotator checks the next DepRel in the same or the following list.

In practice, our experience is that this tool is not used much after the annotators obtained some routine, since the vast majority of dependency relations is easily identifiable.[27] However, it is a considerable help when the annotator is confronted with a difficult case. Thus, even if the tool does not always give the correct DepRel, in the worst case, it directs the annotator towards a restricted subset of dependencies in the detailed guidelines which describe and illustrate every DepRel, in order to see which one seems to fit better. We believe that this method is an interesting way of achieving one of the general objectives of annotation guidelines a described in (Fort, 2012): categories—in our case DepRels—should be defined precisely enough, so as to reduce the stress of the annotators, but should at the same time leave room for doubts, so that the supervision is not too biased.

Finally, let us mention that the criteria we describe in this section do not represent in any way the exhaustive list of properties encoded by each relation. Some other properties are described in the complete guidelines, but their are usually not necessary in order for the annotator to get to the right label, so we do not mention them all here.

3.4 Multilayered annotation in practice

In the previous section, we explained what is behind the scenes of surface syntactic annotation, by giving details about the properties which are associated to the dependency relations and showing how to use them. This section reports on the actual annotation of a multilayered corpus. We first show how we transform a seed corpus, then explain how we obtained partially automatically the deeper annotations. Finally we present an evaluation of the annotation quality.

3.4.1 Annotation of the morphological and surface-syntactic layers: AnCora as a starting point

As already mentioned, we decided to start from an already existing dependency treebank, namely the AnCora-DEP-ES (Taulé et al., 2008), which

---

[27] 15 of the 48 relations occur more than a thousand times in our corpus; these are, from the most frequent to the less frequent: *prepos* (14,520), *det* (14,155), *punc* (10,853), *adv* (9,325), *modif* (6,373), *subj* (5,400),*obl_compl* (4,751), *dobj* (4,529), *coord* (3,814), *attr* (2,379), *aux_phras* (2,099), *obl_obj* (2,037), *conj* (2,017), *copul* (1,551), and *relat* (1,529).

comprised 3,510 sentences at the time we started this project, back in early 2008. The surface-syntactic and the morphological nodes have a one-to-one correspondence, which means that their annotation can be superimposed: the dependencies link pairs of governor-dependent, while the morphological feature/value pairs are associated to each node individually.

The surface-syntactic annotation procedure comprises two stages:

1. Automatic projection of the annotations of the 3,510 sentences from AnCora onto rudimentary surface-syntactic structures (2 steps).
2. Multiple manual revisions of the structures obtained in Stage 1. For the revision work carried out by a small team of trained annotators, the graph editor of the graph transducer MATE was used (Bohnet et al., 2000).

During the first mapping of Stage 1, the first goal is thus to simply convert all labels—attribute/value pairs and arcs—into labels used in our own scheme. 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. 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. Figure 3.4.

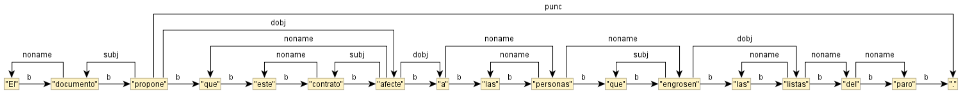


Figure 3.4: Sample AnCora structure visualized in the MATE workbench (*El documento propone que este contrato afecte a las personas que engrosen las listas del paro* 'The document suggests that this contract affect the persons who make the unemployment lists swell.')

The second mapping of Stage 1 is performed automatically using a small graph transformation grammar of 55 rules in the MATE workbench. Most of the rules check in the AnCora structure the nature of two nodes linked by unlabeled arcs (*noname* in MATE). Similarly to what was done for the

ISST, but unlike AnCora, we split bound clitic pronouns such as different kinds of non lexical reflexives (a), direct and indirect objects (b), as well as concatenations of preposition and determiner (c), so as to annotate all syntactic relations between them:

(a) *referirse*→ *referir+se [a]*(lit. 'refer+oneself [to]'); *mirarse*→ *mirar+se* (lit. 'look+each-other'); but *irse* 'go' and *burlarse* 'make-fun-of' for instance, are not split, since they are considered separate lexical entries: *irse* is slightly different from *ir*, and *burlarse* is totally different from its non-reflexive counterpart.

(b) *pegarlo*→ *pegar+lo* ('hit+him'), *pedirle*→ *pedir+le* ('ask-him');

(c) *del*→*de+el* ('of+the'); *al*→*a+el* ('to+the').[28]

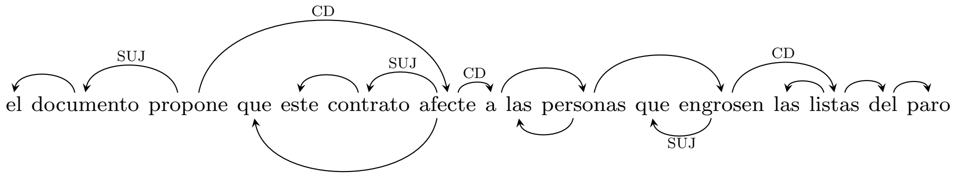
In addition, in AnCora, multi-word units are considered one single word, e.g. Barack_Obama. We preferred to split these so as to represent their internal dependencies. As mentioned in Section 2.2, the original AnCora corpus contains 95,028 tokens, but according to Section 3.2, ours contains 100,892 tokens. The fact that we separate these types of tokens accounts for the difference.

Figure 3.5, shows the different steps of the SSynt annotation, with the original AnCora annotation, the output of the automatic mapping, and the SSyntS after manual revision. We can see in Figure 3.5(b) that the *del* node has been split and all dependency relations labeled. In addition, some dependencies are changed, for instance, the pure subordinating conjunctions, which were dependents of the verb they introduce, become their governor in our annotation, as explained in Section 3.2 (e.g. the first direct object of the sentence). During the automatic mapping, some errors can be introduced (see the two edges pointing to the seventh node in Figure 3.5(b)). This happens because the rule system that produces such structures is quite simple, and it takes into account the fact that there are posterior manual revisions (together with automatic checks that specifically point to that kind of mistake).

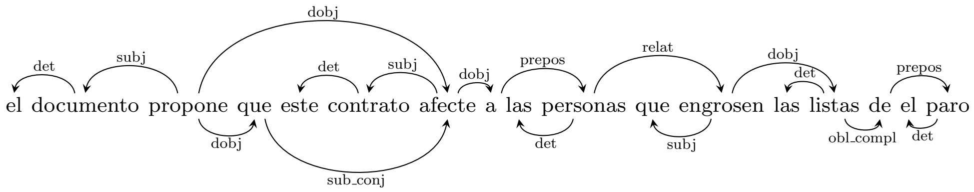
The manual revision of Stage 2 was performed in accordance with the detailed guidelines described in Section 3.3. But there is one important difference with what has been described so far: in order to facilitate the annotation of the deeper levels, we split 14 of the relations shown in Tables

---

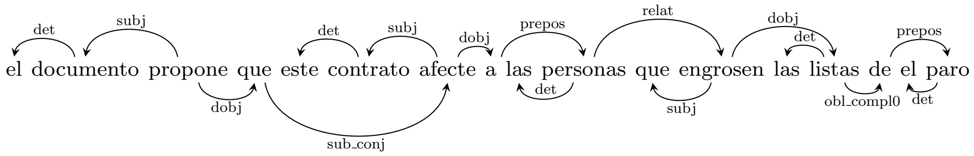
[28] Note that *al salir* (lit. 'at the moment of going out') is not considered a concatenation of preposition and determiner.



(a) Original AnCora annotation



(b) Structure after the automatic mapping



(c) Structure after the manual revision

Figure 3.5: Sentence *El documento propone que este contrato afecte a las personas que engrosen las listas del paro* 'The document suggests that this contract affect the persons who make the unemployment lists swell' at different steps of the annotation process

3.3 and 3.4 into more fine-grained relations which also encode predicate-argument information. Those labels are used to derive automatically rather complete deep-syntactic structures (see Section 3.4.2), but are not retained in the surface-syntactic annotation, which only includes the 48 original labels. That is, in order to label the dependencies, the annotator has to follow the syntactic guidelines, and when annotating some of the relations in the DepRel column of Tables 3.3 and 3.4, add or not a suffix to the label, based on the three following criteria.

(1) What is the configuration of the underlying predicate-argument structure? (5 DepRel→ 25)

For the DepRel *iobj*, *iobj_clitic*, *obl_compl*, *obl_obj*, the goal is to associate to the dependent a slot in the valency frame of its governor: by convention, we number the argument slots from 0 to 5, although they correspond to the first to the sixth arguments. For this, we asked the annotators to (i) consider the definition of the predicate, which can only be complete if all its arguments are mentioned, and (ii) evaluate the importance of each argument with respect to this predicate, which allows for assigning them a slot in its valency. At the first glance, the task may appear subjective and thus difficult. However, the very large majority of predicates have between one and three arguments. This makes the task easier, especially for verbs, for which the subject (in active voice) is always considered the first argument,[29] and the direct object the second. In case of oblique or indirect objects or oblique complements, the decision can be harder to make. But the high inter-annotator agreement rate obtained for the task (see evaluation at the end of this subsection) indicates that the intuition of the annotators coincides to a large extent. Consider, for example, the predicate *proponer* 'suggest': its definition would be something like "an entity E1 giving an idea I to another entity E2 for E2 to consider I". In other words, *proponer* has three arguments, E1, I, E2; E1 and I are almost never omitted, which makes them higher in the argument hierarchy than E2, and the entity "who does" is considered more important than what is done. As a result, we have E1=Arg1 (subject), I=Arg2 (direct object), and E2=Arg3 (oblique object 2).

In addition to object and complement DepRel, the reflexive auxiliary *aux_refl* tag is subdivided into four groups: direct (the pronoun is the second argument of the verb and has a coreference link with its subject), indirect (same as direct but the pronoun is third argument), passive (the pronoun is not an argument but triggers an inversion of first and second arguments in the DSyntS), and lexical (the pronoun is just a part of the verb's lemma).

(2) Is the dependent parenthetical? (6 DepRel→ 12)
This criterion is used in order to distinguish between two levels of modification for basic modifiers, one being closer to the governor than the other. For instance, the *adv* DepRel below a verb indicates the presence of a circumstancial element related to the verb itself, while the *adjunct* DepRel indicates that the circumstancial operates at the sentence level: (normalmente←*adjunct*-corre-*adv*→[cada dia] 'usually he runs every day').

---

[29] This is why there is no extension 0 for verbal relations (iobj, iobj_clitic, obl_obj), and also why by default we start numbering the arguments from the second.

For nominal governors (*appos*, *attr*, *modif*, *quant*, *relat*), the descriptive extension is usually granted to groups separated by a comma from their head.

	SSynt DepRel	**Split (SSynt DepRel$_A$)**
(a) Actancial	aux_refl	aux_refl_dir | aux_refl_indir | aux_refl_lex | aux_refl_pass
	iobj	iobj1 | iobj2 | iobj3 | iobj4 | iobj5
	iobj_clitic	iobj_clitic1 | iobj_clitic2 | iobj_clitic3 | iobj_clitic4 | iobj_clitic5
	obl_compl	obl_compl0 | obl_compl1 | obl_compl2 | obl_compl3 | obl_compl4 | obl_compl5
	obl_obj	obl_obj1 | obl_obj2 | obl_obj3 | obl_obj4 | obl_obj5
(b) Backgrounded	adv	adjunct | adv
	appos	appos | appos_descr
	attr	attr | attr_descr
	modif	modif | modif_descr
	quant	quant | quant_descr
	relat	relat | relat_descr
(c) Quotative	copul	copul | copul_quot
	dobj	dobj | dobj_quot
	prepos	prepos | prepos_quot

Table 3.12: Splitting of some syntactic labels into semantics-oriented labels

(3) Is the dependent quoted? (3 DepRel→ 6)
In simple terms, it is the group formed by the dependent and all its dependents surrounded by quotation marks, which indicate an actual quotation. Consider, for illustration, the difference between *dijo "me voy"* 'he said "I'm going"' (quote), and *¡Mira, el "presidente" llega!* 'Look, the "president" is arriving!', in which the quotation marks are a stylistic way of making fun of someone. Three DepRel are concerned: *subj*, *dobj* and *prepos*.

As a result, instead of these 14 DepRel, the annotator has to consider 43 (25+12+6), that is, 29 more (see Table 3.12). So far, this gives us 77 different tags (48+29). In addition, we further split for various testing reasons the label *conj* into *sub_conj* and *compar_conj*, and added a third label *restr* when splitting the DepRel *adv*. Thus, the total tagset which represents the base of our annotation process comprises **79 different tags**. We refer to this tagset as the "Annotation SSynt DepRel" tagset (SSynt DepRel$_A$).

As for the annotation at the morphological layer, it was mostly derived automatically from the AnCora annotation; only in a few cases the annotators had to manually revise the values of some features. Morpho-syntactic features are associated to each node of the structure.

Table 3.13 shows the number of occurrences of each feature in the corpus and their distribution over elements of different PoS tags.

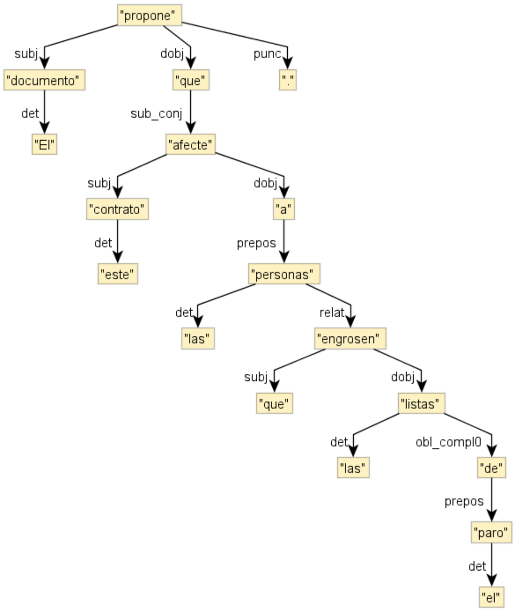
FEAT	#occurrences	**V**	**N**	**Adj**	**Det**	**Pro**	Other
fin	11,776	99.91	0.01	0.06	0	0	0.02
gen	41,735	2.02	46.72	14.31	32.33	4.37	0.25
moo	8,116	99.95	0.01	0	0	0	0.04
num	53,608	16.74	36.57	15.15	27.1	4.25	0.19
per	8,132	99.98	0.01	0	0	0	0.01
ten	8,070	99.98	0	0	0	0	0.02

Table 3.13: Distribution of features over elements of different generic Part-of-Speech (%)

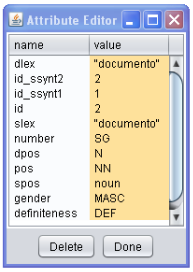
3.4.2 Annotation of the deep-syntactic layer

As mentioned in Section 3.2, the deep-syntactic structure has the form of an unordered dependency tree. The edges encode explicit valency relations, and also coordination and modifications, while only meaning-bearing units are accepted as nodes. Multi-word expressions are fused into single nodes. Sentence-internal coreferential links are superimposed on the annotation. Thanks to the splitting of 14 of the 48 SSyntRels, all surface-syntactic relations from the SSynt DepRel$_A$ tagset (but *compl_adnom* and *det*) have a direct correlation with deep-syntactic configurations.

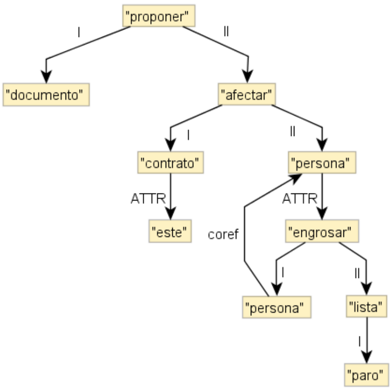
Taking this into account, together with the syntactic properties of each DepRel (e.g., *obl_obj* points to a governed preposition, i.e., to a functional node which does not carry any meaning on its own), the mapping between SSynt and DSynt can be largely automatic (for instance, the DSyntS shown in Figure 3.6 has required no manual modification, although this is not always the case). The workload of the annotator is reduced to (i) addition of coreferences between nodes of the same sentence, (ii) definition of the argument slot of possessive determiners and *compl_adnom* dependency when necessary, and (iii) repair of possible erroneous rule applications. There are currently 129 rules in the SSynt-DSynt mapping grammar, and its coverage is not yet complete, as some very specific configurations are still not taken into account. According to some informal evaluations, an average-length



[SSyntS]



name	value
dlex	"documento"
id_ssynt2	2
id_ssynt1	1
id	2
slex	"documento"
number	SG
dpos	N
pos	NN
spos	noun
gender	MASC
definiteness	DEF



[DSyntS]

El documento propone que este contrato afecte a las personas que engrosen las listas del paro 'The document suggests that this contract affect the persons who make the unemployment lists swell.'

Figure 3.6: Unordered SSyntS (as in Figure 3.5(c) and automatically derived DSynt annotation)

sentence (around 30 nodes) takes an annotator around one and a half minutes to process, while without the automatic annotation derivation it takes about 10 minutes.

For making the manual control of a DSyntS and its comparison with the corresponding SSyntS easier, an intermediate unordered SSyntS is provided to the annotator, as in Figure 3.6.

Table 3.14 exhibits all the correspondences between SSynt DepRel_A and DepRel. It indicates that some SSynt DepRel_A are not mapped to any DSynt DepRel: this is due to the fact that some nodes are removed from the deep-syntactic structure, namely, functional nodes (see details in Section 3.2.3):

(a) Governed elements: "empty" prepositions required by their governor and subordinating conjunctions *que* 'that' when they introduce an argument of a predicate. It is also to be noted that the mapping procedure covers coordinated governed elements, that is, elements that are dependents of the *coord_conj* DepRel which are required by a higher node in the tree.

(b) Auxiliaries: elements which carry tense (past: *haber* 'have' + past participle; future: *ir* 'go' + preposition *a* 'to' + infinitive), aspect (progressive: *estar* 'be' + present participle) or voice (passive: *ser* 'be' + past participle).

(c) Determiners: only definite *el* 'the' and indefinite *un* 'a' determiners are removed.[30]

Table 3.15 completes Table 3.14, by summarizing all the mappings of SSynt DepRel_A to something else than a single DSynt DepRel.[31]

Mapping rules implemented in the MATE workbench (Bohnet et al., 2000) perform all the transformations: mapping of regular and unusual edges,

---

[30] Relative pronouns with antecedent are not removed but substituted by their antecedent, and a coreference link is added between them. Given how we annotate relative clauses (see, e.g. Figure 3.6), we can always find the antecedent of the pronoun as the governor of the *relat* DepRel.

[31] The Meaning-Text Theory's deep-syntactic layer also contains abstract lexical units called *Lexical Functions* (Mel'čuk, 1996); even though we performed experiments on their annotation, we do not handle them in the framework of this thesis.

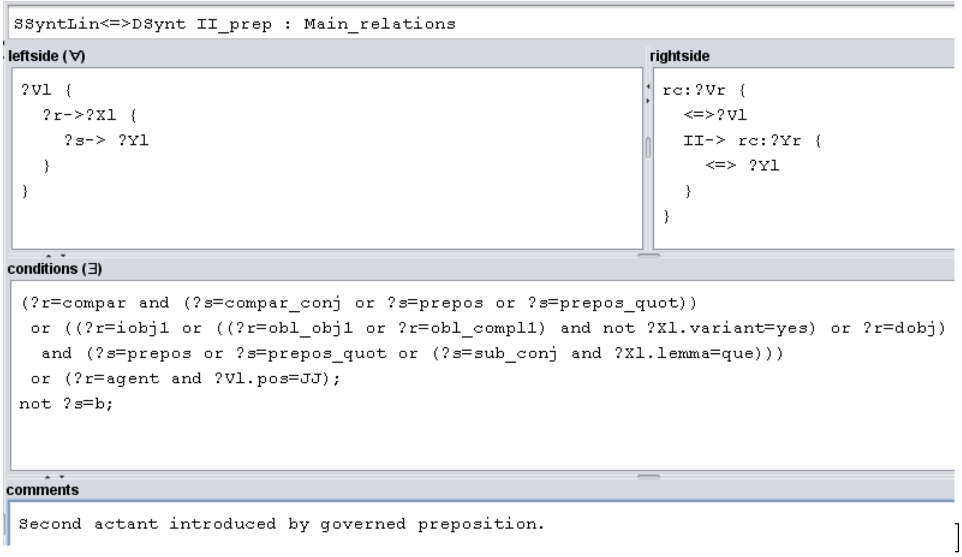
SSynt	DSynt
abbrev	ATTR
abs_pred	ATTR
adjunct	APPEND
adv	ATTR
adv_mod	ATTR
agent	I
analyt_fut	-
analyt_pass	-
analyt_perf	-
analyt_progr	-
appos	ATTR
appos_descr	APPEND
attr	ATTR
attr_descr	APPEND
aux_phras	-
aux_refl_dir	II
aux_refl_indir	III
aux_refl_lex	-
aux_refl_pass	-
bin_junct	ATTR
compar	II
compar_conj	II
compl1	II
compl2	III
compl_adnom	any
coord	COORD
coord_conj	II
copul	II
copul_clitic	II
copul_quot	II
det	any
dobj	II
dobj_clitic	II
dobj_quot	II
elect	ATTR
iobj1	II
iobj2	III
iobj3	IV
iobj4	V
iobj5	VI
iobj_clitic1	II
iobj_clitic2	III
iobj_clitic3	IV
iobj_clitic4	V
iobj_clitic5	VI
juxtapos	APPEND
modal	II
modif	ATTR
modif_descr	APPEND
num_junct	COORD
obj_copred	ATTR
obl_compl0	I
obl_compl1	II
obl_compl2	III
obl_compl3	IV
obl_compl4	V
obl_compl5	VI
obl_obj1	II
obl_obj2	III
obl_obj3	IV
obl_obj4	V
obl_obj5	VI
prepos	II
prepos_quot	II
prolep	APPEND
punc	-
punc_init	-
quant	ATTR
quant_descr	APPEND
quasi_coord	COORD
quasi_subj	I
relat	ATTR
relat_descr	APPEND
relat_expl	APPEND
restr	ATTR
sequent	ATTR
sub_conj	II
subj	I
subj_copred	ATTR

Table 3.14: Mapping of the 79 SSynt DepRel$_A$ onto DSynt DepRel

SSynt DepRel$_A$		**Changes in DSynt**
agent compar compar_conj coord_conj dobj iobj1-5 obl_compl0-5 obl_obj1-5 sub_conj		remove Dep if governed preposition
analyt_fut		remove Gov and Dep add tense=FUT
analyt_pass		remove Gov invert I and II add voice=PASS
analyt_perf		remove Gov add tense=PAST
analyt_progr		remove Gov add tem_constituency=PROGR
aux_refl_dir		replace node label with antecedent's add coreference between I and II
aux_refl_indir		replace node label with antecedent's add coreference between I and III
aux_refl_lex		remove Dep add *se* at the end of Gov's lemma
aux_refl_pass		remove Dep invert I and II add voice=PASS
compl_adnom		edit DepRel
det	*el*‖*un*	remove Dep add definiteness=DEF/INDEF
	possessive	replace node label with antecedent's add coreference link with antecedent edit DSynt DepRel
	other	map *det* to *ATTR*
relat relat_descr		replace node label with antecedent's add coreference link with antecedent

Table 3.15: More complex SSynt to DSynt mappings

adding coreference relations and attribute/value pairs, removing nodes. Figure 3.7 is a sample mapping rule from the mapping grammar; it maps the concerned edges to *II* while removing governed prepositions (dependent of



```
SSyntLin<=>DSynt II_prep : Main_relations
```

```
?Vl {
  ?r->?Xl {
    ?s-> ?Yl
  }
}
```

```
rc:?Vr {
  <=>?Vl
  II-> rc:?Yr {
    <=> ?Yl
  }
}
```

```
(?r=compar and (?s=compar_conj or ?s=prepos or ?s=prepos_quot))
 or ((?r=iobj1 or ((?r=obl_obj1 or ?r=obl_compl1) and not ?Xl.variant=yes) or ?r=dobj)
  and (?s=prepos or ?s=prepos_quot or (?s=sub_conj and ?Xl.lemma=que)))
 or (?r=agent and ?Vl.pos=JJ);
not ?s=b;
```

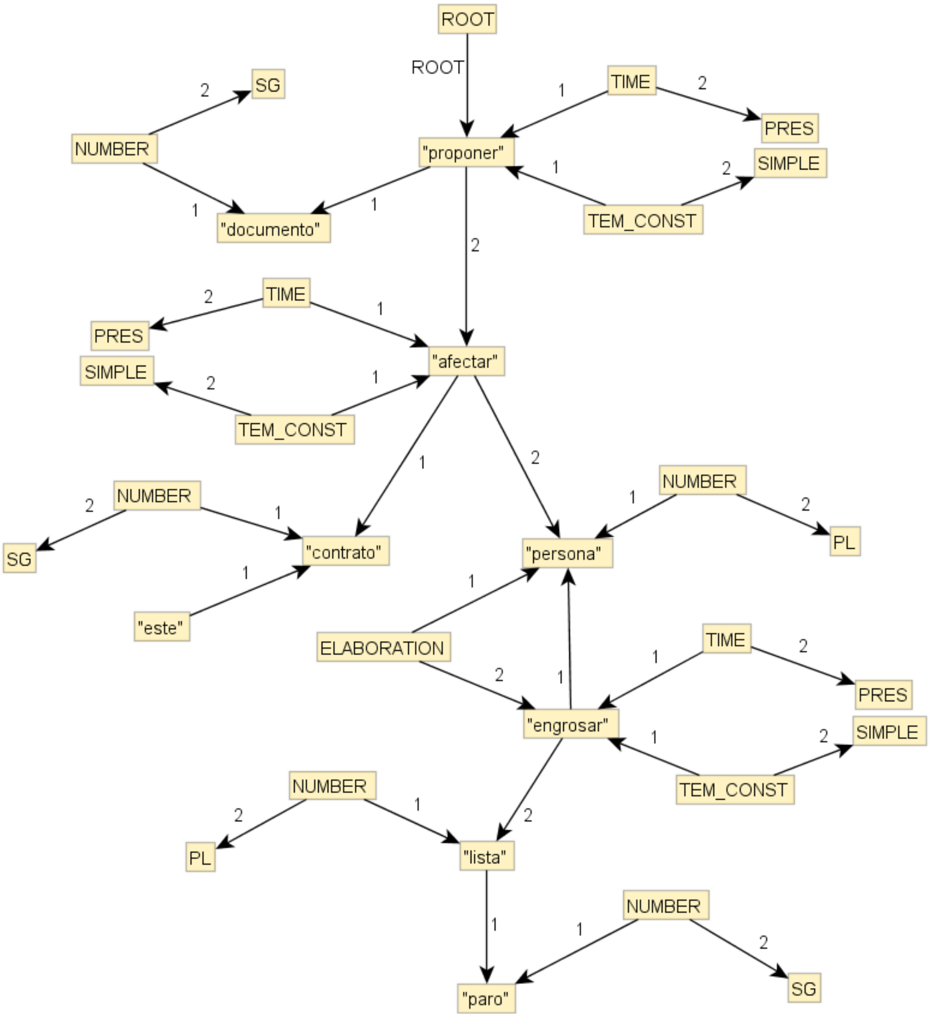
```
Second actant introduced by governed preposition.
```

Figure 3.7: Sample mapping rule for graph transducer

agent, *compar*, *dobj*, *iobj1*, *obl_compl1*, *obl_obj1*) or other functional elements (governor of *sub_conj*). In the rule, variables are indicated by the presence of a question mark, nodes are delimited by round brackets, dependency relation names are followed by right arrows, and correspondences between a node in the source structure and another one in the target structure are marked with a double arrow. In the leftside, this rule matches a subtree in the surface-syntactic structure; it looks for a node *?Vl* that has a dependency *?r* (specified in the *conditions* field) to a functional element *?Xl*, itself having another dependency to another node *?Yl*. On the rightside, the rule modifies the target deep-syntactic structure, in this case, it links the nodes corresponding to *?Vl* and *?Yl* with an edge *II*; in this mapping, *?Xl* does not appear on the rightside, and since no other rule transfers it, it simply does not appear in the deep-syntactic structure. In Figure 3.6, this rule matches the subtrees 'propone-*dobj*→que-*sub_conj*→ afecte' and 'afecte-*dobj*→a-*prepos*→ personas', while a similar rule (but mapping the edge to *I*) matches listas-*obl_compl0*→de-*prepos*→ paro.

3.4.3 Annotation of the semantic layer

In DSyntS, since all grammatical units have been removed from the structure, the mapping to a connected acyclic graph made of pure predicate-argument relations connecting any meaning-bearing unit used in the sen-



El documento propone que este contrato afecte a las personas que engrosen las listas del paro 'The document suggests that this contract affect the persons who make the unemployment lists swell.'

Figure 3.8: An automatically derived semantic annotation

tence (which includes DSynt nodes and some additional meta-nodes) is much easier. Another mapping grammar transforms the deep-syntactic structure in Figure 3.6 into a semantic structure, shown in Figure 3.8.

During this mapping, all nodes from the deep-syntactic structure are transferred, except nodes which have a coreference relation with another one.

Only one node that stands for all coreferring nodes appears in the semantic structure; all edges that point to a node which is removed are transferred to that one node.[32] Most relations can be derived in a straightforward way: Roman numerals map to Arabic numerals, and ATTR, APPEND and COORD edges are inverted and relabeled with *1* when the DSynt dependent is a predicate. Otherwise, we introduce the meta-predicates like *ELABORATION* or *POSSESS* in order to connect them to the graph.[33]

The attributes *tense*, *number* and *tem_constituency* are simply mapped to the corresponding binary semantemes (see Section 3.2). Only *definiteness*, as part of the communicative structure, is kept as an attribute on the concerned nodes, together with the IDs.

3.4.4 Correspondences of nodes between the layers

In this section, the possible configurations of correspondences between two adjacent strata of the annotation are detailed.

Surface-syntax⇔Morphology
There is one-to-one correspondence between those two levels.

Deep-syntax⇔Surface-syntax

- 1 DSynt node ↔ 1 SSynt node; this is the most frequent configuration; it concerns meaning-bearing units, such as *gato* 'cat' in Figure 3.9b.
- 1 DSynt node ↔ 2 to n SSynt nodes; this configuration concerns all grammatical units, which are removed at the deep-syntactic level: determiners (*cf* Figure 3.9c), auxiliaries (*cf* Figure 3.9d), functional prepositions. It also concerns phraseological units, which are split at the surface-syntactic level, but merged in deep-syntax, so that the internal syntactic structure is not shown anymore.
- 1 DSynt node ↔ nothing; this occurs only with empty subject pronouns, a frequent phenomenon in Spanish: in this case, we introduce

---

[32] Our mapping grammar actually has a parameter that allows for keeping the coreferring nodes separated in the SemS. This can be useful for experiments in which each instance of the referent has a distinct role with respect to some of the predicates it is related to, as it can be the case with communicative structure.

[33] Meta-nodes are shown in upper case in Figure 3.8, while regular nodes are in lower case.

a node in deep-syntax but is has not correlate in the superficial annotation; see, e.g., in Figure 3.9a, the node *3PL*, which stands for '3^{rd} person plural.

DSynt	**SSynt**	
3PL	∅	(Persiguen gatos.)
3PL	*They*	(*persecute cats.*)

(a) 1 to 0

DSynt	**SSynt**	
gato	Gatos	(duermen.)
cat	*Cats*	(*sleep.*)

(b) 1 to 1

DSynt	**SSynt**			
gato	Los	gatos	(han sido	perseguidos.)
cat	*The*	*cats*	(*have been*	*persecuted.*)

(c) 1 to 2

DSynt	**SSynt**				
perseguir	(Los	gatos)	han	sido	perseguidos (.)
persecute	(*The*	*cats*)	*have*	*been*	*persecuted* (.)

(d) 1 to 3

Figure 3.9: Sample DSynt-SSynt node correpondences

Semantics⇔Deep-syntax
Figure 3.10 shows the correspondences between the nodes of substructures of Figures 3.8 (on the left) and 3.1b (on the right).

- 1 Sem node ↔ 1 DSynt node; this is the most frequent configuration; it concerns simple meaning-bearing units, see e.g. the nodes *afectar* 'affect' connected by the long dashes in Figure 3.10.

- 1 Sem node ↔ nothing; it only happens with the node *ROOT*, added at the semantic layer in order to point to the main node of the graph (see Figure 3.8).

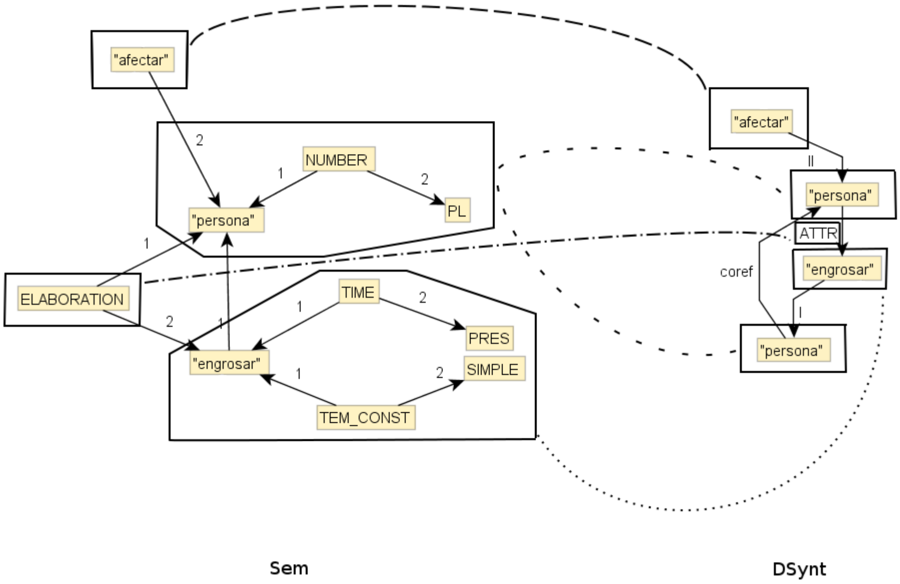


Figure 3.10: Sample Sem-DSynt node correpondences

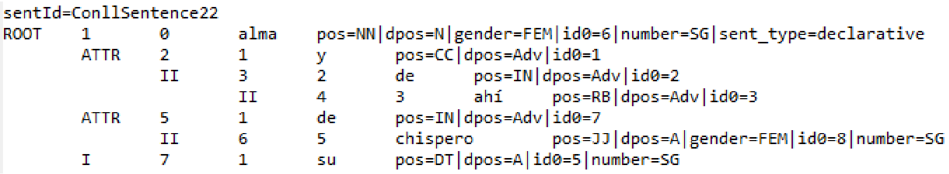
- 1 Sem node ↔ 1 DSynt relation; the *ELABORATION* and *POSSESS* meta-nodes are introduced based on the deep-syntactic dependency relation between the two nodes they have as arguments at the semantic level—ATTR for *ELABORATION*, *cf* the dot-and-dash line in Figure 3.10, and I for *POSSESS*.

- 1 to 3 Sem node(s) ↔ 2 to n DSynt nodes; this configuration concerns coreferring nodes, which are split at the deep-syntactic layer (each instance has its own syntactic function) but are merged in the semantic annotation; see the short-dash lines in Figure 3.10.

- n Sem nodes ↔ 1 DSynt node; this configuration concerns three of the meta-nodes added at the semantic level, when they are not in correspondence with coreferring elements: *TIME* (n=3), *TEM_CONSTITUENCY* (n=3), and *NUMBER* (n=3) (see the dotted line in Figure 3.10).

3.4.5 Format

In order to facilitate the processing of the superficial layers of the annotation, the sentence, morphological and surface-syntactic layers are presented in a single standard 14-column CoNLL file (Hajič et al., 2009). The deep-syntactic layer can be provided in the CoNLL and the HFG format, used in the first Surface-Realization Shared Task ((Belz et al., 2011), see Figure 3.11). Both layers are also available in the MATE graph format (.str), which is the only one available so far for the semantic layer. The different layers are connected thanks to the IDs of the nodes.



```
sentId=ConllSentence22
ROOT    1       0       alma    pos=NN|dpos=N|gender=FEM|id0=6|number=SG|sent_type=declarative
        ATTR    2       1       y       pos=CC|dpos=Adv|id0=1
                II      3       2       de      pos=IN|dpos=Adv|id0=2
                        II      4       3       ahí     pos=RB|dpos=Adv|id0=3
        ATTR    5       1       de      pos=IN|dpos=Adv|id0=7
                II      6       5       chispero        pos=JJ|dpos=A|gender=FEM|id0=8|number=SG
        I       7       1       su      pos=DT|dpos=A|id0=5|number=SG
```

Figure 3.11: A sample DSyntS in the HFG format *Y de ahí, su alma de chispera*, lit. 'And from there, her/his soul of gossip'.

Figure 3.11 reads as follows. Each line has indentations which show visually the dependencies. The first element is the dependency, the second the ID of the node, the third its governor, the fourth its lemma, and the final group of elements contains the grammemes and correspondence with other layers. In the column below the ID of the first node (ID=1, *alma* 'soul'), there are three dependencies, indicating that this node has three dependents, namely two *ATTR*, *y* 'and' (ID=2) and *de* 'of' (ID=5) and a *I* (*su* 'her/his'). In its turn, *y* 'and' (ID=2), has a dependent *de* 'of' (ID=3), and so on. For a sample MATE structure, see Figure 3.29 in Section 3.5.

3.4.6 Evaluation

First of all, it is important to point out that the evaluation of the annotation depends on the specific task the annotation implies. Although evaluation coefficients such as Kappa or Alpha are very useful for some tasks, some other tasks are not well evaluated with them (for a detailed description of the appropriateness of those coefficients depending on the task, see among others (Artstein and Poesio, 2008) and (Fort, 2012)). The evaluation of the annotation of dependency relations is a task that still needs a lot of exploration (Fort, 2012). We have reviewed the inter-annotator agreement methods used in some important corpora, such as the Penn TreeBank, the Prague Dependency Treebank, AnCora, PropBank/NomBank, Turku Dependency Treebank (TDT), Talbanken, Ontonotes (Hovy et al., 2006), and the Italian Syntactic-Semantic Treebank (ISST). Some do not report interannotator agreement figures (Talbanken, first version of TDT, ISST), some report figures without giving detailed explanations, (annotation of dependencies in AnCora), some use an evaluation coefficient such as Kappa (PropBank/NomBank) or Alpha (coreference annotation in AnCora), and finally some compare the tags of each annotator against a gold standard (PTB, last versions of TDT) or simply compare each annotator's annotation against each other (Penn Discourse Treebank, Ontonotes). In these last two works, the figures are not presented in terms of an evaluation coefficient but of similarity percentages, which is suitable for the task we want to evaluate, i.e., the inter-annotator agreement on several different layers.

Our methodology consists in using the surface-syntactic annotation with different granularities of tags. Indeed, as already mentioned, in the deep-syntactic structures, manual revisions are necessary in three cases only: (i) coreferences, (ii) mapping grammar mistakes, and (iii) the dependencies from noun to possessive determiners. First of all, we do not aim at evaluating coreferences, because coreferences have no impact on the syntactic structures. If two (or more) nodes are in a coreference relation, we simply add a common ID to these nodes in DSyntS, and if one of them is a pronoun, we substitute the node label by the one of the antecedent, but we do not remove any, so the structure remains the same whether two nodes have a coreference relation or not. Second, the SSyntS–DSyntS mapping grammars are not error-free, which means that annotators have to make corrections that they should not have to make (for instance, it can happen that one node is not transferred to the DSyntS, and that the annotator does not see that it is missing). For the evaluation, we do not want to take into account how good an annotator is at "repairing" mapping errors. Also, what is annotated as multi-word units can vary from one annotator to another; this means that the number of nodes in deep-syntax and semantics can be different amongst annotators, which makes its evaluation very difficult to perform automatically.

The most accurate way to calculate the inter-annotator agreement is then to use structures *before* they are mapped to the deep-syntactic layer, that is, surface-syntactic structures. In order to evaluate the superficial inter-annotator agreement, we use the SSynt DepRel tagset (48 tags), and in order to evaluate the deep inter-annotator agreement, we use SSynt DepRel$_A$

tagset (79 tags), which allows for getting automatically the deeper levels (see Section 3.4). The only problem is the case of possessive determiners, for which we do not know if they are correctly annotated at the deep layers with the 79-tag tagset. We decided not to take them into account in the deep evaluation (they represent only about 1% of the total number of dependencies in the corpus).

We actually also made other evaluations with the more coarse-grained tagsets, using the hierarchy presented in Section 3.3. We decided to have the two persons involved in the annotation label the dependencies of 72 sentences (2,443 tokens). Then, considering one annotation as gold standard and the other one as predicted, we run the CoNLL'08 evaluation in order to calculate the Labeled and Unlabeled Attachment Scores (respectively UAS and LAS). The UAS is the percentage of dependencies which are assigned the same governor and dependent in the two annotations, while the LAS is the percentage of dependencies which are assigned the same governor, dependent, and label in the two annotations. We trained a Spanish parser using the parser developed by Bohnet (2010), and then parsed the *linguistica* ('linguistics') wikipedia page after cleaning it. The two annotators then post-edited every sentence, using the SSynt DepRel$_A$ tagset.

	79 DepRels	**48 DepRels**	**31 DepRels**	**15 DepRels**
UAS (%)	96.15	96.15	96.15	96.15
LAS (%)	89.40	92.26	92.51	92.80

Table 3.16: Inter-annotator agreement

Since the successive mappings from 79 to 15 DepRel only concern the edge labels, it is normal that the UAS, which does not take into account the labels, remains the same for all tagsets. As expected, the least the tags in the annotation, the more the agreement between annotators: we reached 89.4% including predicate-argument identification, 92.26% with the 48 DepRel given in Tables 3.3 and 3.4, and until 92.8% when reducing the tagset to 15 DepRel. All inter-annotator agreement figures are close to the 90% threshold recommended in the OntoNotes project (Hovy et al., 2006).

In order to see whether those numbers actually can have a possible correlation with an application, we performed a quick experiment: we trained the MaltParser (Nivre et al., 2007b)(a dependency parser) with its default configuration, on the 2006 version of the AnCora corpus, and on our corpus. We obtained LAS=78.26% / UAS=82.23% on AnCora and LAS=80.18% / UAS=86.89% on our annotation. Taking into account that this version of AnCora has a set of deprels of approximately one third of the size of ours, and that many edges were not labeled, those figures confirm that our annotation is fairly consistent.

We believe that the high inter-annotator agreement obtained for this task is largely due to the fact that the criteria defining each dependency relation have been carefully selected. However, the 3-point difference between semantic and syntactic tagsets show that our predicate-argument annotation guidelines are not as clear as the syntactic guidelines. It is also to be noted that there is no actual criterion that allows for identifying easily the number of arguments of a predicate or their order; the interpretation of semantic phenomena is always prone to subjectivity.

3.5 Automatic mapping of the PTB

Even if the annotation process can be speeded up, it remains a time-consuming task, and for this, it is not always possible to annotate a corpus manually on a large scale. Thus, we carried out some experiments on the transformation of existing resources into the multilayered annotation presented in this chapter. Most existing annotation schemes are not thought for NLG, but they represent some very valuable data, and should be exploited as much as possible. For instance, as seen in Chapter 2, some corpora combine syntactic and semantic annotation, but do not clearly draw a line between the two, and/or are incomplete at the semantic layer. Thus, for our purposes, it is a matter of managing to “clean” and complete the annotations. In this section, we give an example of the use of the Penn TreeBank/PropBank/NomBank as a seed annotation suitable for NLG purposes. But first of all, let us describe briefly one of the first large-scale efforts that have been made in this direction.

3.5.1 Previous attempt

The data provided to the participants for the First Surface Realization Shared Task (*SRST*) in 2011 (Belz et al., 2010, 2011) included the dependency conversion of the Penn TreeBank as such (that is, morphological, syntactic and topological information), and a separate “deep” input built from the combination of Penn TreeBank, PropBank, NomBank, and the BBC Entity Type corpus (Weischedel and Brunstein, 2005). A great step towards a common-ground deep annotation was made with the organization of this first task. However, as became clear during the discussions with the

SRST working group since then, more efforts have to be done in order to make the deep input more "semantic". For example, the deep input was provided in the Human-Friendly Graph (HFG) format, which is actually a tree-like representation. This was made possible by creating a special type of edge called "inverted argument" (*AINV*), which means that the governor of this type of edge is actually an argument of the dependent: *car*-AINV→ *this* actually stands for the more semantic-oriented annotation *this*-A1→ *car*. This choice may be considered an anticipation of some syntactic decisions: the direction of non-argumental syntactic edges should not be resolved before getting to the syntactic structure. Another issue was that of governed prepositions, which were not distinguished from semantically loaded prepositions in the CoNLL annotation. As a consequence, in the SRST, only some easily indentifiable function words were removed from the deep input, namely *to* infinitive markers and *that* complementizers. As discussed in this chapter, more prepositions and elements should be removed from an abstract representation.

As done in the SRST, we use the Penn TreeBank annotation as such as our morphological and surface-syntactic layers. In the following, we show how to obtain a deep annotation which excludes as many syntactic features as possible, in order to get closer to what we consider a truly multilayered corpus. In other words, we aim at deriving *semantic structures* as defined in Section 3.2.4. For this, we keep or modify some edges, remove or replace functional nodes, and connect the semantic structure.[34]

3.5.2 Managing the edges already in the PropBank/NomBank annotation

In this section, we show that parts of the PB/NB annotation can be kept as they are (in particular, argumental and continuation edges), but that some edges such as circumstancial relations have to be inverted and renamed.

3.5.2.1 Keeping the argumental edges ***A0/A1/A2/A3/A4/A5***

Verbal and nominal predicates are annotated with high quality in PB/NB, and should be maintained in the annotation. However, in the current PropBank annotation, when the argument is introduced by a preposition in the

---

[34] Deep-syntactic structures can be easily obtained with the same methodology, the only differences being that we do not need to introduce meta-nodes or invert some edges; instead, we keep non-argumental edges like the *AINV* in SRST (namely, *ATTR*, *APPEND* and *COORD*).

sentence, this preposition receives the argumental edge independently from the fact that it is governed or not.

3.5.2.2 Leaving the edges of continuation structures

The mapping of the continuation constructions to actual semantic representations is not systematic. Consider the two following examples:

Example 3.7. *Labor costs continued to rise more rapidly in service industries than in goods-producing industries:*
continued-A1→costs
continued-C-A1→to [rise]

Example 3.8. *This enabled them to set prices for which goods may be sold:*
enabled-A1→them
enabled-C-A1→to [set]

In Example 3.7, *rise* would be the first semantic argument of *continue*,
continue-A1→*rise* & *rise*-A1→*cost*
while in Example 3.8, *set* would be the second argument of *enabled*:
enabled-A2→*set* & *set*-A1→*them* & *set*-A2→*prices*

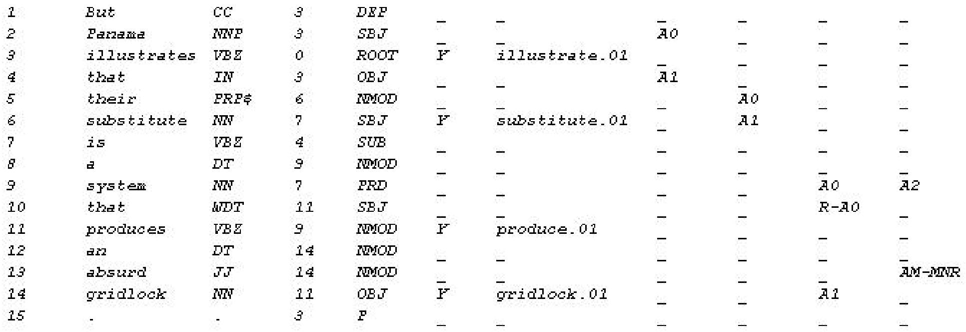
As a result, in order to introduce mistakes in the deep annotation, we leave the *C-Ax* edges in the deep representation for out experiments; *C-Ax* edges are handled exactly as the *Ax* edges.

3.5.2.3 Inverting and renaming edges to modals, adverbials and other circumstancials

The "modifiers" (or *attributes* since we speak about semantic notions) involved in PB/NB as governed elements in relations such as *AM-DIR*, *AM-LOC*, *AM-MNR*, *AM-TMP*, *AM-EXT*, *AM-PRD*, *AM-PNC*, *AM-CAU*, *AM-ADV*, and *AM-NEG* are, in fact, predicative semantemes that take as arguments nodes that govern them in the syntactic structure. For instance, *gridlock*−AM-MNR→*absurd* in Figure 3.12[35] reads in semantic terms as *absurd*−A1→*gridlock*. Therefore, the PB/NB annotation should be rectified.

---

[35] In all CoNLL structures shown in this section, the *lemma* and the different *predicted* columns are removed so as to make the figures clearer.



1	But	CC	3	DEP	_	_	_	_	_	_
2	Panama	NNP	3	SBJ	_	_	A0	_	_	_
3	illustrates	VBZ	0	ROOT	Y	illustrate.01	_	_	_	_
4	that	IN	3	OBJ	_	_	A1	_	_	_
5	their	PRP$	6	NMOD	_	_	_	A0	_	_
6	substitute	NN	7	SBJ	Y	substitute.01	_	A1	_	_
7	is	VBZ	4	SUB	_	_	_	_	_	_
8	a	DT	9	NMOD	_	_	_	_	_	_
9	system	NN	7	PRD	_	_	_	_	A0	A2
10	that	WDT	11	SBJ	_	_	_	_	R-A0	_
11	produces	VBZ	9	NMOD	Y	produce.01	_	_	_	_
12	an	DT	14	NMOD	_	_	_	_	_	_
13	absurd	JJ	14	NMOD	_	_	_	_	_	AM-MNR
14	gridlock	NN	11	OBJ	Y	gridlock.01	_	_	A1	_
15	.	.	3	P	_	_	_	_	_	_

Figure 3.12: PTB/PB/NB-structure for the sentence "*But Panama illustrates that their substitute is a system that produces an absurd gridlock.*" (CoNLL format)

It is most intuitive to interpret temporals, locatives, aspectuals, etc. as two place semantemes. That is, the original PB/NB annotation can be modified in this respect as follows: *next*–A1→*monday*←A2–*time*–A1→*begin*.

There are actually two distinct cases: (i) if the attribute already encodes the meaning expressed by the *DIR*, *LOC*, *MNR*, etc parts of the relations, and (ii) if it does not. We believe that in the first case, prepositions and adverbs are concerned (e.g. *John goes shopping during his break*), and in the second case all other categories (e.g. *John goes shopping every weekend*). For prepositions and adverbs, no additional semanteme is needed, since their valency foresees one or more argument(s) for this particular meaning ('time' in the forementioned example). However, nouns and verbs, for instance, have their own internal valency unrelated with the circumstancial meaning. As a consequence, a semanteme should be introduced in order to link the circumstancial group and the element it specifies the circumstance of. This implies:
(a) for the modifier edges *AM-DIR*, *AM-LOC*, *AM-MAN*, *AM-TMP*, *AM-EXT*, *AM-PRD*, *AM-PNC*, *AM-CAU*, *AM-ADV*, *AM-NEG*, *AM-DIS* with a target node of the PoS *preposition* or *adverb*: the original governor becomes the governed and the label of the inverted edges is by default set to *A1*;
(b) for the modifier edges *AM-DIR*, *AM-LOC*, *AM-MAN*, *AM-TMP*, *AM-EXT*, *AM-PRD*, *AM-PNC*, *AM-CAU*, *AM-ADV*, *AM-NEG*, *AM-DIS* with a target node of the Part-of-Speech which is not preposition or adverb: a semanteme corresponding to the circumstance ('time', 'cause', 'manner',

'location', etc.) is created, the original governor becomes the *A1* of this semanteme and the target node of the PB/NB edge is its *A2*;
(c) for the edges *R-AM-...*, whether they point to a relative (*pos=WP/ WP$/ WDT*) or an interrogative pronoun (*pos=WRB*), they should be handled like the *AM-x* edges (see below for the specificities of relative pronouns).

Some conjunctions, for instance, ambiguous (non-disambiguated) conjunctions, such as 'while', cannot be handled this way; if we replace the *AM-TM*P (for example) edge by an inverted *A1* edge, we lose the temporal meaning and cannot see if 'while' is temporal or contrastive anymore. It is not reasonable to think that we can obtain 'while.01' VS 'while.02', so we agree that we have to encode this meaning, either (1) by maintaining the original PB/NB edge, (2) by adding a meta-node above relational nodes as well, or (3) by adding an attribute on the node, based on the PB/NB *AM-x* edge. We suggest to use the third option on all cases in which we do not introduce a meta-node.

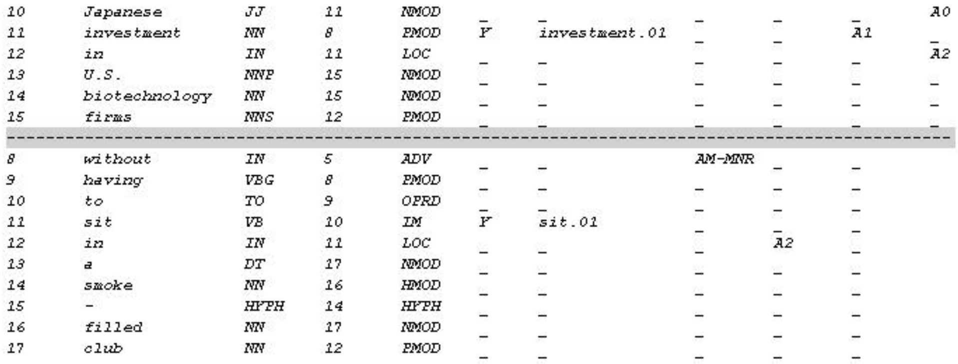
3.5.3 Removing or replacing functional nodes

The PB/NB annotation includes a number of syntactic nodes which should not be included in a deep annotation. In this section, we take a close look at what nodes to remove or replace, which includes governed prepositions and conjunctions, relative pronouns, auxiliaries and logical subjects.

3.5.3.1 Removing governed elements

Governed prepositions and conjunctions are recognizable because in the semantic annotation they receive an argumental relation (*A1*, *A2*, *A3*, etc.) and they have a particular PoS (*IN* or *TO*). In PB/NB, no distinction is made between void and semantically loaded prepositions. For instance, the prepositions *in* in *...the Japanese investment **in** U.S. biotechnology firms without having to sit **in** a smoke filled club* are annotated as *A2*, while the first of them is governed and the second is semantic. This is a problem when it comes to the removal of purely syntactic prepositions from the semantic representation.

An illustration of governed prepositions is given in Figure 3.13. One possible strategy for removing governed elements is the following: (i) rank all disambiguated predicates from PB/NB based on frequency, and (ii) take the predicates that appear at least *n* times through the corpus and check



```
10   Japanese        JJ    11   NMOD   _   _                _        _    _    A0
11   investment      NN    8    PMOD   Y   investment.01    _        _    A1   _
12   in              IN    11   LOC    _   _                _        _    _    A2
13   U.S.            NNP   15   NMOD   _   _                _        _    _    _
14   biotechnology   NN    15   NMOD   _   _                _        _    _    _
15   firms           NNS   12   PMOD   _   _                _        _    _    _
------------------------------------------------------------------------------------
8    without         IN    5    ADV    _   _                AM-MNR   _    _
9    having          VBG   8    PMOD   _   _                _        _    _
10   to              TO    9    OPRD   _   _                _        _    _
11   sit             VB    10   IM     Y   sit.01           _        _    _
12   in              IN    11   LOC    _   _                _        A2   _
13   a               DT    17   NMOD   _   _                _        _    _
14   smoke           NN    16   HMOD   _   _                _        _    _
15   -               HYPH  14   HYPH   _   _                _        _    _
16   filled          NN    17   NMOD   _   _                _        _    _
17   club            NN    12   PMOD   _   _                _        _    _
```

Figure 3.13: Two governed prepositions *in* annotated in Propbank

manually if they have some governed elements in the corresponding frameset of the PB/NB lexicon.[36] n should be high enough for the learning to actually take place. For example, for n=150, we found 152 different predicates which govern one or more prepositions. The predicates have from one to four slots which can require a preposition, and from one to four different prepositions per slot. (iii) build up a list of prepositions to remove based on the governing predicate, the argument slot and the name of the preposition. For this, we only consider the slots which merely have one possible preposition. That is, do not remove a preposition if there are several possible prepositions in one slot. It seems like it is the most promising method to us, since it ensures that only the targeted prepositions are removed. But this task can get quite tedious and was not carried out in the framework of our experiments. Instead, we "arbitrarily" targeted a list of prepositions that we systematically kept, while removing all others: *above*, *across*, *after*, *along*, *although*, *amid*, *among*, *around*, *atop*, *because*, *before*, *behind*, *below*, *beside*, *between*, *beyond*, *down*, *in*, *into*, *like*, *near*, *off*, *onto*, *out*, *outside*, *over*, *past*, *per*, *though*, *through*, *throughout*, *toward*, *towards*, *under*, *until*, *up*, *upon*, *whatever*, *whether*, *within*, *without*, *worth*.

Figure 3.13 also shows that some syntactic prepositions such as *to* are not annotated. However, they are removed based on their PoS tag *TO*, since *TO* is always void of meaning (it stands for infinitive markers or governed prepositions). We also suggest to remove all 'that' nodes with PoS tag *IN* which are arguments (there are 3590 of these subordinating conjunctions).

---

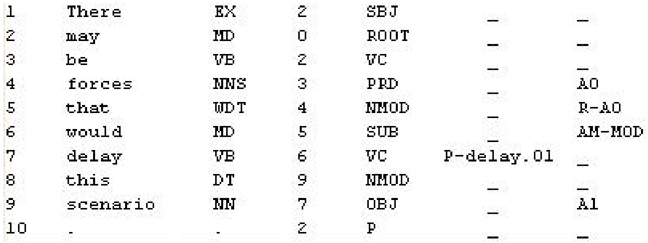
[36] http://verbs.colorado.edu/verb-index/index/L.php, or http://verbs.colorado.edu/propbank/framesets-english/

It is also to be noted that the mapping procedure should cover coordinated governed prepositions, in order to avoid that the conjunction in those cases remains disconnected; fortunately this phenomenon is not very common.

3.5.3.2 Removing relative pronouns

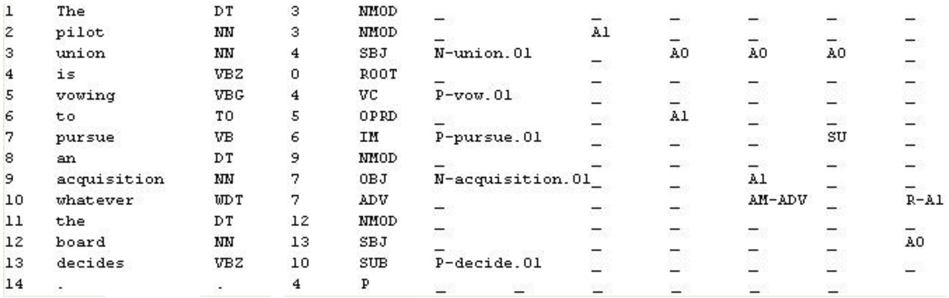
Relative pronouns with antecedent such as *that* in Figure 3.14 are semantically empty and should equally be discarded from the semantic structures.

The antecedent of a relative pronoun can be found in PB/NB by looking



1	There	EX	2	SBJ	_	_
2	may	MD	0	ROOT	_	_
3	be	VB	2	VC	_	_
4	forces	NNS	3	PRD	_	A0
5	that	WDT	4	NMOD	_	R-A0
6	would	MD	5	SUB	_	AM-MOD
7	delay	VB	6	VC	P-delay.01	_
8	this	DT	9	NMOD	_	_
9	scenario	NN	7	OBJ	_	A1
10	.	.	2	P	_	_

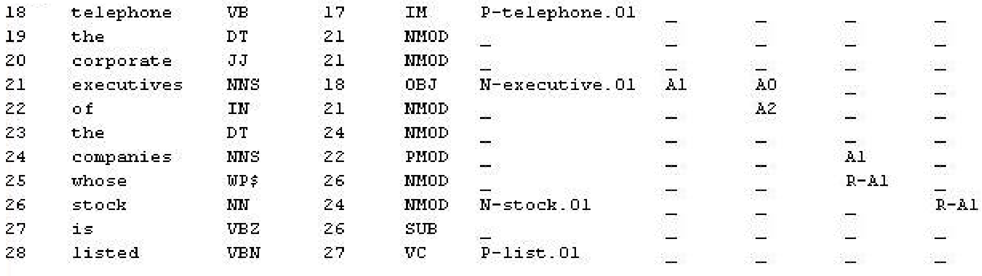
Figure 3.14: Simple relative pronoun with antecedent: *that*, *who*, *which*, *whom*



1	The	DT	3	NMOD	_	_	_	_	_	_
2	pilot	NN	3	NMOD	_	A1	_	_	_	_
3	union	NN	4	SBJ	N-union.01	_	A0	A0	A0	_
4	is	VBZ	0	ROOT	_	_	_	_	_	_
5	vowing	VBG	4	VC	P-vow.01	_	_	_	_	_
6	to	TO	5	OPRD	_	_	A1	_	_	_
7	pursue	VB	6	IM	P-pursue.01	_	_	_	SU	_
8	an	DT	9	NMOD	_	_	_	_	_	_
9	acquisition	NN	7	OBJ	N-acquisition.01	_	_	A1	_	_
10	whatever	WDT	7	ADV	_	_	_	AM-ADV	_	R-A1
11	the	DT	12	NMOD	_	_	_	_	_	_
12	board	NN	13	SBJ	_	_	_	_	_	A0
13	decides	VBZ	10	SUB	P-decide.01	_	_	_	_	_
14	.	.	4	P	_ _	_	_	_	_	

Figure 3.15: Simple relative pronoun without antecedent: *what*, *whatever*, *whoever*, *whichever*

at the column which contains the *R-Ax* edge label: the corresponding *Ax* edge in the same column points to this antecedent (e.g. ‘forces’, as an *A0* in the same column as the *R-A0* edge pointing to ‘that’ in Figure 3.14). In some cases, no antecedent can be found in the PropBank annotation; however, the antecedent can be retrieved by looking for the syntactic governor of the highest node of the relative construction in the primary syntactic annotation. Since the relative pronouns are removed, incoming semantic edges on (one part of) the relative pronoun in the PropBank annotation should be moved to the antecedent. In what follows, we detail the different configurations that allow for retrieving the antecedent and the relative pronoun itself, and go into more details about connecting the relative clause to the rest of the semantic structure. Relative pronouns can have different Parts-of-Speech: *WDT* (which, that, whatever, whichever), *WP* (who, what, whom, whoever), *WP$* (whose), *IN* or *DT* (that). The general idea is to identify the relative pronoun, its antecedent, its possible interesting "dependents" (in *all/some of which*, 'all/some' is annotated as a dependent of the relative pronoun; see last item in list below), and then move and modify the semantic edges as follows:

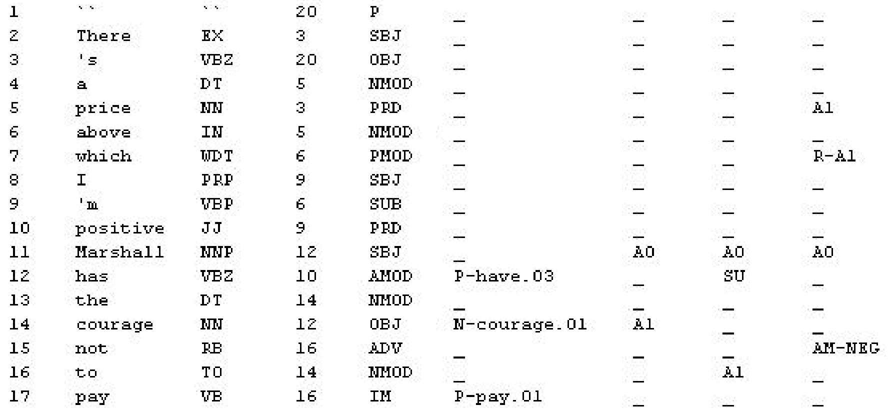


```
18   telephone    VB    17   IM     P-telephone.01   _    _    _    _
19   the          DT    21   NMOD   _                _    _    _    _
20   corporate    JJ    21   NMOD   _                _    _    _    _
21   executives   NNS   18   OBJ    N-executive.01   A1   A0   _    _
22   of           IN    21   NMOD   _                _    A2   _    _
23   the          DT    24   NMOD   _                _    _    _    _
24   companies    NNS   22   PMOD   _                _    _    A1   _
25   whose        WP$   26   NMOD   _                _    _    R-A1 _
26   stock        NN    24   NMOD   N-stock.01       _    _    _    R-A1
27   is           VBZ   26   SUB    _                _    _    _    _
28   listed       VBN   27   VC     P-list.01        _    _    _    _
```

Figure 3.16: Complex relative pronoun (i): nominal governor, *whose*

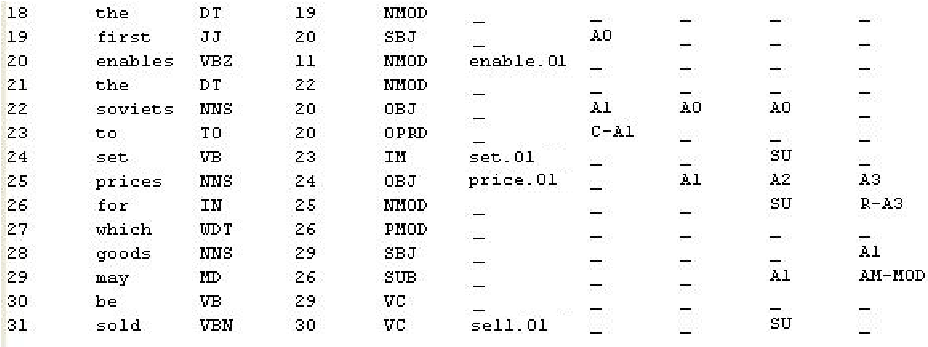
```
if there is a R-Ax edge
  then remove the R-prefix
  if the R-Ax word has pos=IN
    if the R-Ax word has no dependent NMOD which is NN or DT
      if the R-Ax is a governed preposition
        then connect PBGov and antecedent with the Ax edge if it does not already exist (it usually does)
      if the R-Ax is not a governed preposition
        then connect PBGov and preposition with the Ax edge, and connect preposition with antecedent with an A1 edge
    if the R-Ax word has a dependent NMOD which is NN or DT
      then connect PBGov with that NMOD, and connect R-Ax word with antecedent with an A1 edge
```

Due to the diversity of relative constructions in general, there are many different configurations of the annotation of relative clauses in PTB/PB/NB. Figures 3.14 to 3.20 show examples of different types of relative clauses.



```
1    ``          ``    20   P     _                _    _    _
2    There       EX    3    SBJ   _                _    _    _
3    's          VBZ   20   OBJ   _                _    _    _
4    a           DT    5    NMOD  _                _    _    _
5    price       NN    3    PRD   _                _    _    A1
6    above       IN    5    NMOD  _                _    _    _
7    which       WDT   6    PMOD  _                _    _    R-A1
8    I           PRP   9    SBJ   _                _    _    _
9    'm          VBP   6    SUB   _                _    _    _
10   positive    JJ    9    PRD   _                _    _    _
11   Marshall    NNP   12   SBJ   _                A0   A0   A0
12   has         VBZ   10   AMOD  P-have.03        _    SU   _
13   the         DT    14   NMOD  _                _    _    _
14   courage     NN    12   OBJ   N-courage.01     A1   _    _
15   not         RB    16   ADV   _                _    _    AM-NEG
16   to          TO    14   NMOD  _                _    A1   _
17   pay         VB    16   IM    P-pay.01         _    _    _
```

Figure 3.17: Complex relative pronoun (ii): non governed prepositional governor



```
18   the       DT    19   NMOD  _            _      _    _    _
19   first     JJ    20   SBJ   _            A0     _    _    _
20   enables   VBZ   11   NMOD  enable.01    _      _    _    _
21   the       DT    22   NMOD  _            _      _    _    _
22   soviets   NNS   20   OBJ   _            A1     A0   A0   _
23   to        TO    20   OPRD  _            C-A1   _    _    _
24   set       VB    23   IM    set.01       _      _    SU   _
25   prices    NNS   24   OBJ   price.01     _      A1   A2   A3
26   for       IN    25   NMOD  _            _      _    SU   R-A3
27   which     WDT   26   PMOD  _            _      _    _    _
28   goods     NNS   29   SBJ   _            _      _    _    A1
29   may       MD    26   SUB   _            _      _    A1   AM-MOD
30   be        VB    29   VC    _            _      _    _    _
31   sold      VBN   30   VC    sell.01      _      _    SU   _
```

Figure 3.18: Complex relative pronoun (iii): governed prepositional governor

The following rules are supposed to cover all seen configurations in order to find the relative pronoun and its antecedent:

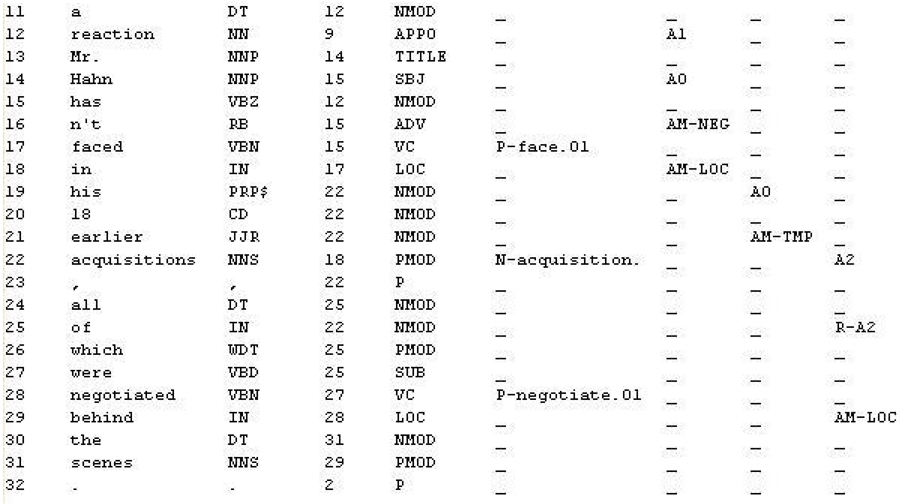
- first of all, we should not remove pronouns which have no clear antecedent, or no antecedent at all (free relatives), that is, leave in the semantic structure ‘what’, ‘whatever’, ‘whichever’, ‘whoever’; hence we should exclude the members of that list from the mapping rules that look for antecedents;

- for the rest of relative pronouns, look for *R-Ax* edges in PB and consider the word on that line; this part of the mapping covers numerous cases and is quite complex; for this reason it is presented a pseudo-code.[37]

```
if word.pos=WDT|WP$|WP OR (word.pos=DT|IN and word.lemma=that)
  then word is the relative pronoun
  then look for word.SyntGov
       if not (word.SyntGov).PB=R-Ax
         then (word.SyntGov).PB is the antecedent
       if (word.SyntGov).PB=R-Ax: look for SyntGov.SyntGov
         if not (SyntGov.SyntGov).PB=R-Ax
           then (SyntGov.SyntGov).PB is the antecedent
         if (SyntGov.SyntGov).PB=R-Ax
           then look for (SyntGov.SyntGov).SyntGov until finding a non-
           R-Ax edge
if not word.pos=WDT|WP$|WP AND not (word.pos=DT|IN and word.lem-
ma=that)
  if not word.SyntGov.PB=R-Ax
    then word.SyntGov is the antecedent
    then look for word.SyntDep
         if SyntDep.pos=WDT|WP$|WP
           then SyntDep is the relative pro
         if not SyntDep.pos=WDT|WP$|WP
           then look for SyntDep.SyntDep until finding a pos=WDT|WP$
           |WP
  if word.SyntGov.PB=R-Ax
    then look for SyntGov.SyntGov
         if not (SyntGov.SyntGov).PB=R-Ax
           then (SyntGov.SyntGov).PB is the antecedent
         if (SyntGov.SyntGov).PB=R-Ax
           then look for (SyntGov.SyntGov).SyntGov until finding a non-
           R-Ax edge
    then look for word.SyntDep
         if SyntDep.pos=WDT|WP$|WP
           then SyntDep is the relative pronoun
         if not SyntDep.pos=WDT|WP$|WP
           then look for SyntDep.SyntDep until finding a pos=WDT|WP$
           |WP
```

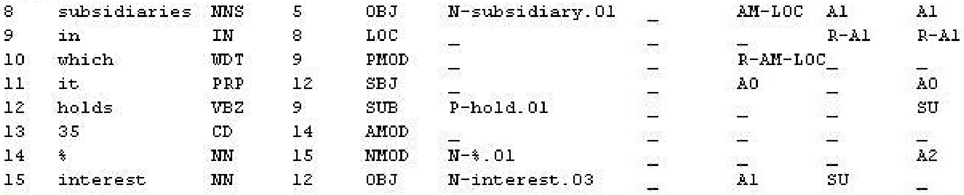
---

[37] In the psuedo-code, *word.SyntDep* means “dependent of the word in the primary syntactic annotation”;*(SyntGov.SyntGov).PB* means “the ProbBank edge on the line of the governor of the governor (in the primary syntactic annotation) of the word”.



11	a	DT	12	NMOD	_	_	_	_
12	reaction	NN	9	APPO	_	A1	_	_
13	Mr.	NNP	14	TITLE	_	_	_	_
14	Hahn	NNP	15	SBJ	_	A0	_	_
15	has	VBZ	12	NMOD	_	_	_	_
16	n't	RB	15	ADV	_	AM-NEG	_	_
17	faced	VBN	15	VC	P-face.01	_	_	_
18	in	IN	17	LOC	_	AM-LOC	_	_
19	his	PRP$	22	NMOD	_	_	A0	_
20	18	CD	22	NMOD	_	_	_	_
21	earlier	JJR	22	NMOD	_	_	AM-TMP	_
22	acquisitions	NNS	18	PMOD	N-acquisition.	_	_	A2
23	,	,	22	P	_	_	_	_
24	all	DT	25	NMOD	_	_	_	_
25	of	IN	22	NMOD	_	_	_	R-A2
26	which	WDT	25	PMOD	_	_	_	_
27	were	VBD	25	SUB	_	_	_	_
28	negotiated	VBN	27	VC	P-negotiate.01	_	_	_
29	behind	IN	28	LOC	_	_	_	AM-LOC
30	the	DT	31	NMOD	_	_	_	_
31	scenes	NNS	29	PMOD	_	_	_	_
32	.	.	2	P	_	_	_	_

Figure 3.19: Complex relative pronoun (iv): partitive governor



8	subsidiaries	NNS	5	OBJ	N-subsidiary.01	_	AM-LOC	A1	A1
9	in	IN	8	LOC	_	_	_	R-A1	R-A1
10	which	WDT	9	PMOD	_	_	R-AM-LOC	_	_
11	it	PRP	12	SBJ	_	_	A0	_	A0
12	holds	VBZ	9	SUB	P-hold.01	_	_	_	SU
13	35	CD	14	AMOD	_	_	_	_	_
14	%	NN	15	NMOD	N-%.01	_	_	_	A2
15	interest	NN	12	OBJ	N-interest.03	_	A1	SU	_

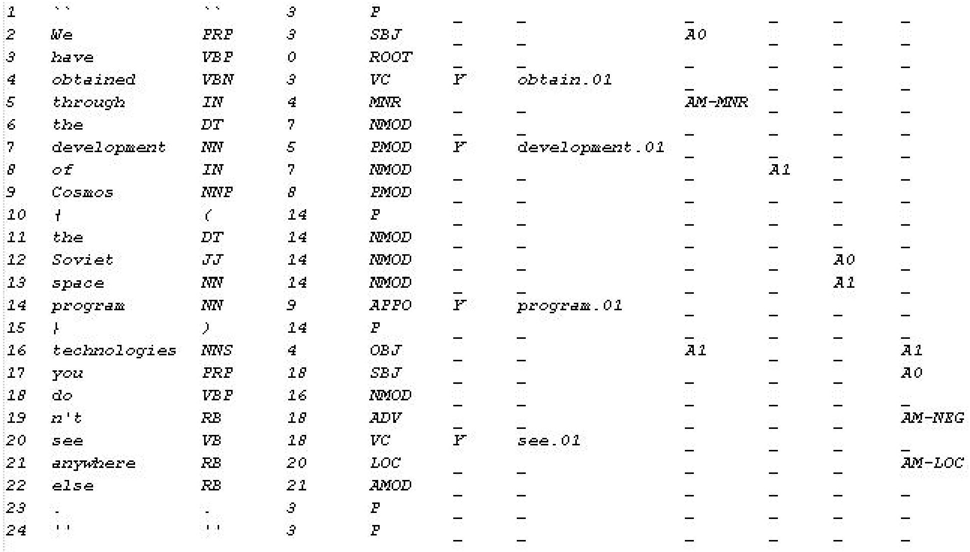
Figure 3.20: Complex relative pronoun (v): various PB edges

3.5.3.3 Replacing auxiliaries

As it was already the case for Spanish, English auxiliaries express semantic grammatical significations, namely tense (past: have+past participle), aspect (progressive: be+present participle) or voice (passive: be+past participle). In order to capture to capture tense and aspect, what is represented as meta-nodes in the Spanish SemS appears under an equivalent form, that is, attributes: *time* for tense (with as possible values *present*, *future* and *past*) and tem_constituency for aspect (with as possible values *simple*, *progressive*, *perfect*, *perfect progressive*). The grammeme of voice is motivated by the communicative structure: it is not the argumental structure of a verb which varies, but the theme/rheme opposition (see dedicated subsection below).

The verb *do* could be handled very easily: when it forms part of negation, it should simply be removed, and when not, it should be replaced by a *foregrounded* feature assigned to the governed verb, or kept in the annotation.

Auxiliaries can be spotted easily in PTB since they are the only nodes that govern verbal predicates with the relation *VC* (e.g., *have obtained* in lines 3-4 of Figure 3.21).

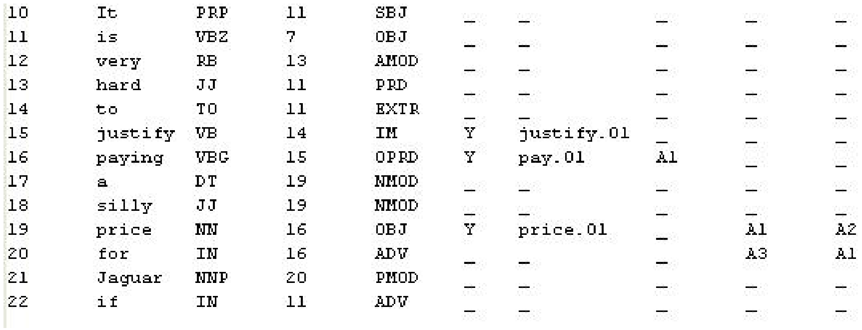


1	``	``	3	P	_	_	_	_	_	_
2	We	PRP	3	SBJ	_	_	A0	_	_	_
3	have	VBP	0	ROOT	_	_	_	_	_	_
4	obtained	VBN	3	VC	Y	obtain.01	_	_	_	_
5	through	IN	4	MNR	_	_	AM-MNR	_	_	_
6	the	DT	7	NMOD	_	_	_	_	_	_
7	development	NN	5	PMOD	Y	development.01	_	_	_	_
8	of	IN	7	NMOD	_	_	_	A1	_	_
9	Cosmos	NNP	8	PMOD	_	_	_	_	_	_
10	(	(	14	P	_	_	_	_	_	_
11	the	DT	14	NMOD	_	_	_	_	_	_
12	Soviet	JJ	14	NMOD	_	_	_	_	A0	_
13	space	NN	14	NMOD	_	_	_	_	A1	_
14	program	NN	9	APPO	Y	program.01	_	_	_	_
15	)	)	14	P	_	_	_	_	_	_
16	technologies	NNS	4	OBJ	_	_	A1	_	_	A1
17	you	PRP	18	SBJ	_	_	_	_	_	A0
18	do	VBP	16	NMOD	_	_	_	_	_	_
19	n't	RB	18	ADV	_	_	_	_	_	AM-NEG
20	see	VB	18	VC	Y	see.01	_	_	_	_
21	anywhere	RB	20	LOC	_	_	_	_	_	AM-LOC
22	else	RB	21	AMOD	_	_	_	_	_	_
23	.	.	3	P	_	_	_	_	_	_
24	''	''	3	P	_	_	_	_	_	_

Figure 3.21: PTB annotation of an auxiliary

3.5.3.4 Removing logical subjects

Logical subjects should be removed, and the actual subject connected to the predicate as its first argument, based on the syntactic annotation: ***It*** *is very hard to justify ...*, see Figure 3.22). The *EXTR* relation in the syntactic annotation indicates the presence of a "real" subject when there is a grammatical subject in the sentence. The PB edge going to the logical subject should be transferred to the *EXTR* dependent, and the incoming PB edge on the latter (if any) should be removed.



```
10   It       PRP  11  SBJ   _  _           _   _   _
11   is       VBZ  7   OBJ   _  _           _   _   _
12   very     RB   13  AMOD  _  _           _   _   _
13   hard     JJ   11  PRD   _  _           _   _   _
14   to       TO   11  EXTR  _  _           _   _   _
15   justify  VB   14  IM    Y  justify.01  _   _   _
16   paying   VBG  15  OPRD  Y  pay.01      A1  _   _
17   a        DT   19  NMOD  _  _           _   _   _
18   silly    JJ   19  NMOD  _  _           _   _   _
19   price    NN   16  OBJ   Y  price.01    _   A1  A2
20   for      IN   16  ADV   _  _           _   A3  A1
21   Jaguar   NNP  20  PMOD  _  _           _   _   _
22   if       IN   11  ADV   _  _           _   _   _
```

Figure 3.22: PropBank-structure for a logical subject

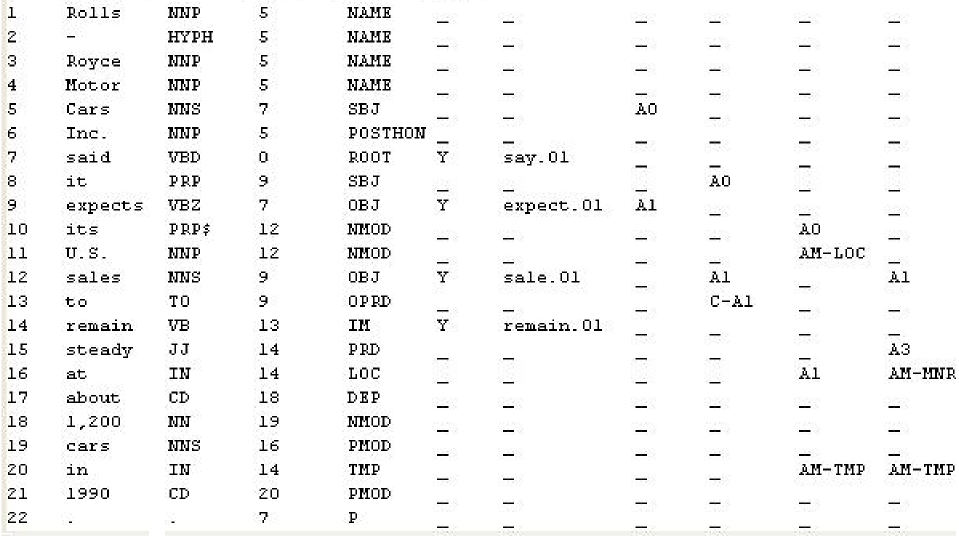
3.5.4 Connecting the semantic structure

In PB/NB, a number of meaning-bearing nodes of sentential semantic structures are not interconnected. This includes, for instance, the quantifiers, the governed NPs in the the case of argument PPs (while the prepositions are connected to the predicate), and often also modifiers (whatsoever the PoS of the governor is) and sentential adverbials. We need to connect them in order to obtain a connected graph. Particular attention has to be paid to governor-dependent pairs in which the governor is not a verb. A large number of noun modifiers, for instance, are not annotated at the semantic level. We use the syntactic annotation of the PTB to guide their connection. Below, we illustrate how this is done for a number of cases, starting with the PB/NB structure shown in the CoNLL format in Figure 3.23 and as graphic in Figure 3.24.

3.5.4.1 Connecting numbers

Numbers are identified in PTB by looking at several features: (1) the PoS of the node is CD, (2) the node has no PB/NB annotation, (3) the node is linked with the relation *NMOD*, *DEP* or *HMOD* to its syntactic governor, (4) which comes after the number in the linear order of the sentence.

We can trace this combination of features and introduce in the semantic graph a binary relation with the node 'QUANTITY' as head (see Figure 3.25), or consider such numbers as predicative semantemes with a single argument.



1	Rolls	NNP	5	NAME	_	_	_	_	_	_
2	-	HYPH	5	NAME	_	_	_	_	_	_
3	Royce	NNP	5	NAME	_	_	_	_	_	_
4	Motor	NNP	5	NAME	_	_	_	_	_	_
5	Cars	NNS	7	SBJ	_	_	A0	_	_	_
6	Inc.	NNP	5	POSTHON	_	_	_	_	_	_
7	said	VBD	0	ROOT	Y	say.01	_	_	_	_
8	it	PRP	9	SBJ	_	_	_	A0	_	_
9	expects	VBZ	7	OBJ	Y	expect.01	A1	_	_	_
10	its	PRP$	12	NMOD	_	_	_	_	A0	_
11	U.S.	NNP	12	NMOD	_	_	_	_	AM-LOC	_
12	sales	NNS	9	OBJ	Y	sale.01	_	A1	_	A1
13	to	TO	9	OPRD	_	_	_	C-A1	_	_
14	remain	VB	13	IM	Y	remain.01	_	_	_	_
15	steady	JJ	14	PRD	_	_	_	_	_	A3
16	at	IN	14	LOC	_	_	_	_	A1	AM-MNR
17	about	CD	18	DEP	_	_	_	_	_	_
18	1,200	NN	19	NMOD	_	_	_	_	_	_
19	cars	NNS	16	PMOD	_	_	_	_	_	_
20	in	IN	14	TMP	_	_	_	_	AM-TMP	AM-TMP
21	1990	CD	20	PMOD	_	_	_	_	_	_
22	.	.	7	P	_	_	_	_	_	_

Figure 3.23: Sample unconnected PB/NB semantic graph (CoNLL format)

3.5.4.2 Connecting adjectival modifiers

Adjectival modifiers are tagged in PTB as *NMOD* with a PoS *JJ*, and quantifiers, and determiners as *NMOD* having a PoS *DT*, with an anteposition to their governor. Once traced, they can be treated in the semantic graph as unary predicative semantemes and thus connected to their syntactic governor via the *A1* relation; see 'about'[38] in Figure 3.25.[39]

3.5.4.3 Connecting possessives

Possessives have the PoS *PRP$* in PTB. Some are already annotated at the semantic level (see 'its' in Figure 3.25), some are not. If latter is the case, the same strategy as for quantifiers and modifiers can be followed; that is, connect the possessive with an edge *A1* to the noun.

---

[38] 'about' is actually intensional here, which is not captured in our annotation. The sentence is not about a quantity x such that x is 1200 and x is "about", as the proposed semantic representation seems to state; rather, it is dealing with a quantity x that is "about 1200".

[39] Note that our solution is to handle adjectival predicates through default assignment rules is not very efficient since we perform the mapping without looking at other linguistic resources.

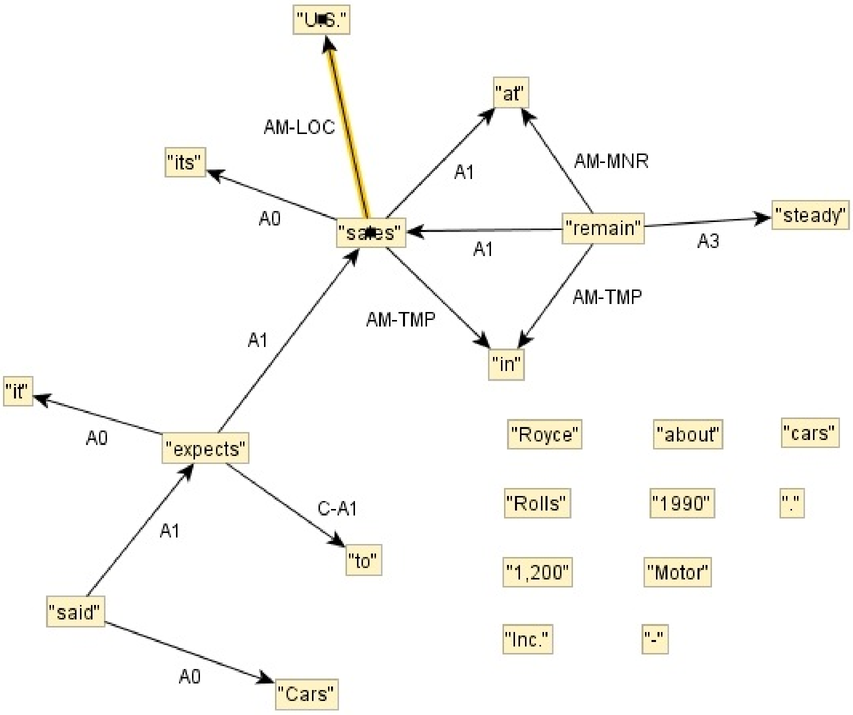


Figure 3.24: Sample unconnected PB/NB semantic graph (MATE format)

3.5.4.4 Connecting appositions

Appositions are tagged in PTB as *NMOD*; the dependent and the governor of the DepRel have a PoS *NN*, *NNS* or *NNP* (the three different types of nouns as annotated in PTB). Some dependents of the DepRel are already annotated at the semantic level. In this case, the edges are maintained if they are argumental; if they are not, the generic mapping of semantic edges as described in this section applies. If the apposed element is not annotated in PB/NB, in order to keep the structure connected, we can map this annotation to the binary predicative semanteme 'ELABORATION', with the apposed nouns as its arguments—as *the Soviet space program* in Figure 3.21 apposed to *Cosmos*: *cosmos*←A1-*ELABORATION*−A2→*program* and *soviet*←A1−*program*−A2→*space*.[40]

---

[40] See Section 3.2 for a discussion of the *ELABORATION* meta-node.

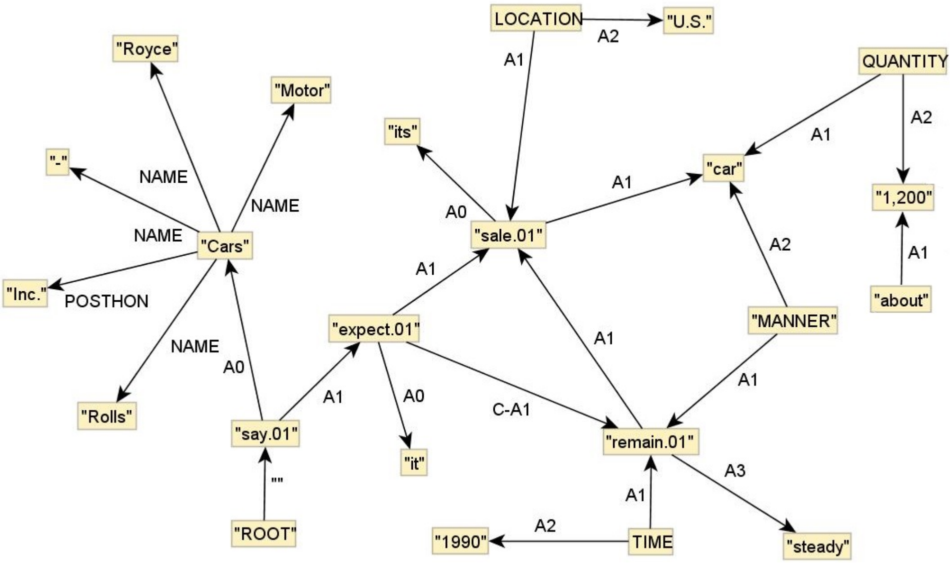


Figure 3.25: Construction of a connected semantic graph

3.5.4.5 Connecting adverbs and conjunctions

Adverbs and conjunctions are tagged as *IN* or *WRB* in PTB. Unconnected adverbs are interpreted as unary predicative semantemes and can thus be connected by the *A1* relation with their syntactic governor (see l.21-22 in Figure 3.21): else-*A1*→anywhere.

Unconnected subordinating conjunctions are considered binary predicative semantemes, with the governing element in syntax as its first argument and the head of the subordinated group as its second argument.Coordination conjunctions are also semantic predicates, with possibly unlimited number of arguments according to the Meaning-Text Theory tradition. If it is assumed to be $n-$ary with $n \geq 2$, the semantic representation should be as in Figure 3.26. However, after several unsuccessful intents to obtain this kind of representation with an acceptable quality, we decided to annotate the coordinating conjunctions as the subordinating ones, that is, as binary predicates, cf. Figure 3.27.

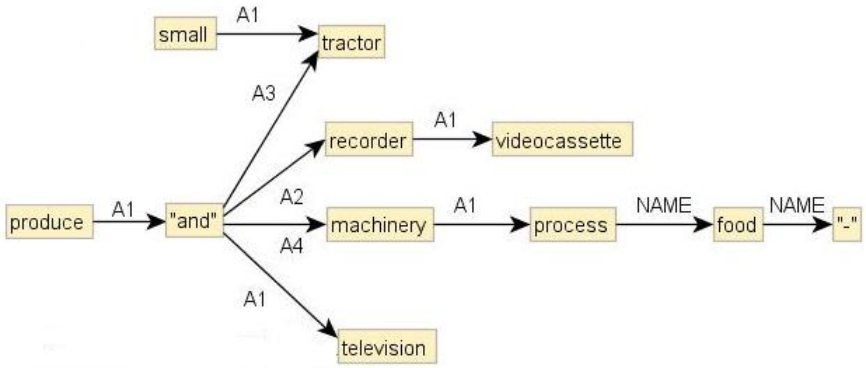


Figure 3.26: Coordination with conjunction as predicate with unlimited arguments (*produce [televisions, videocassette recorders, small tractors and food-processing machinery]*)

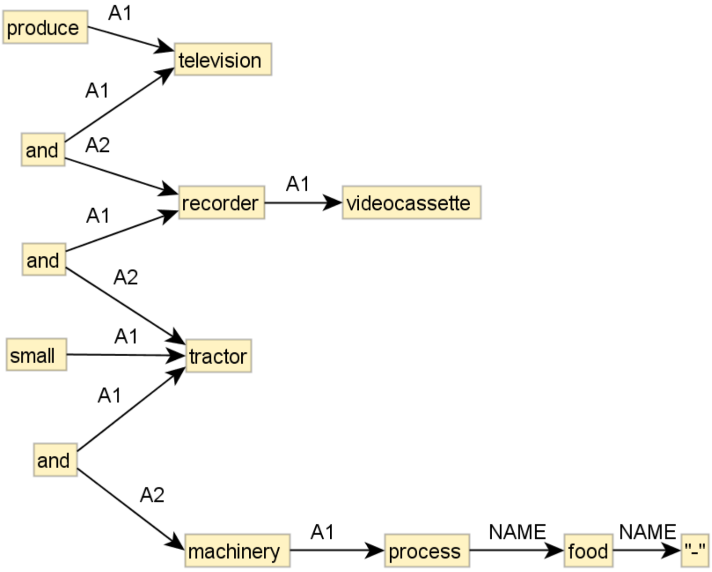


Figure 3.27: Coordination with conjunction as binary predicate (*produce [televisions, videocassette recorders, small tractors and food-processing machinery]*)

3.5.4.6 Connecting vocatives

In order to connect vocatives, we introduce the node ‘ADDRESSEE’, based on the *VOC* dependency relation: *John, come here!: John*←VOC−*come* gives *come*←A1−*ADDRESSEE*−A2→*John*

3.5.4.7 Connecting other elements

In the case of unconnected juxtapositions, parentheticals, sequentials, we can add a node (like ‘ELABORATION, or ‘SPECIFICATION’) to connect juxtaposed elements, as done for appositions (see above).

3.5.5 Adding basic communicative structure

The annotation must codify aspects of the information (or *communicative* in terms of Mel’čuk (2001)) structure (to be superimposed on the propositional semantic structure) without which the input to a generator is underspecified (and thus not complying with the basic requirement that the structures be self-contained). Thus, from an abstract semantic input structure a la *Minimal Recursion Semantics* (Copestake et al., 1997) such as *produce(h1:system, h2:gridlock)*, *absurd(h2)*, *be(h3:substitute,h1)*, a generator may namely produce a variety of sentences, among them *A substitute is a system that produces an absurd gridlock*, *The substitute is a system producing an absurd gridlock*, *An absurd gridlock is produced by the system of substitute*, etc.

At least those aspects must be introduced that predetermine the overall syntactic structure (paratactic, hypotactic or parenthetical), the internal syntactic structure (subject/object structure, clefted or not, any element fronted or not, etc.), and determiner distribution. These aspects concern at least *theme* and *rheme*, *foregrounded* and *backgrounded*, and *given* and *new* in the MTT model (Mel’čuk, 2001). *Theme* specifies what an utterance is about and *rheme* what is being stated about the theme; see, for instance, (Halliday, 1994) on this distinction. In a declarative sentence, the fragment of the semantic structure marked as the theme will, as a rule, be realized as subject and the fragment marked as the rheme will be realized as the verbal phrase of the sentence.[41] For instance, $[John]_{theme}$←A1−$[see$−A2→$Maria]_{rheme}$ will correspond to *John*←subject

---

[41] The distinction between *theme* and *rheme* is close to the distinction between *topic* and *focus* (Sgall et al., 1986), *topic* and *comment* (Gundel, 1988) and *ground* and *focus* (Vallduvi, 1990). For the discussion of some differences, see, e.g., (Hajíčova, 2007).

$-\mathit{see}-\text{dir.obj}\rightarrow Maria$ in the syntactic annotation and $[\mathit{John}\leftarrow\text{A1}-\mathit{see}]_{rheme}$ $\text{-A2}\rightarrow[Maria]_{theme}$ to $\mathit{John}\leftarrow\text{obj}-\mathit{see}_{pass}-\text{subject}\rightarrow Maria$.

For the generation of hypotactic sentences such as *John bought a car which was old and ugly*, we need to accomodate for a recursive definition of the theme/rheme dimension:
$[\mathit{John}]_{theme}\leftarrow\text{A1}-[\mathit{buy}-\text{A2}\rightarrow[\text{c1}:\mathit{car}]_{theme}\leftarrow\text{A1}-[\mathit{old}]_{rheme}\ \text{c1}:\leftarrow\text{A1}-[\mathit{ugly}]_{rheme}]_{rheme}$
With no recursive (or *secondary* in terms of Mel'čuk (2001)) theme/rheme, the generated sentence would be *John bought an old and ugly car.*

We mark a fragment of an utterance as *foregrounded* if it is to be presented as prominent and as *backgrounded* if it is to be presented as "secondary" (less prominent); otherwise, elements of the semantic structure are considered *neutral.* We thus fuse two communicative dimensions made by Mel'čuk (2001): *focalized* vs. *non-focalized* on the one side and *foregrounded* vs. *backgrounded* vs. *neutral* on the other side; see also Lambrecht (1994) for a similar distinction as Mel'čuk's. However, since we are interested in a straightforward correspondence between communicative and syntactic features, we think that this simplification can be justified.

The *foregrounded* feature of an *A1* element of a verbal semanteme will trigger a clefting construction. For instance, the communicative configuration $[\mathit{John}]_{foregr|theme}\leftarrow\text{A1}-[\mathit{see}-\text{A2}\rightarrow Maria]_{rheme}$ will lead to *It was John who saw Maria.* The *foregrounded* feature of an *A2* element of a verbal semanteme will trigger a clefting construction or a dislocation: *It was Maria, whom John saw.*

The *foregrounded* feature of an *A1* or *A2* element of a nominal (or nominalized) semanteme will trigger an *argument promotion*, as, e.g., *John's arrival* (instead of *arrival of John*).

The *foregrounded* feature of a circumstantial will lead to its fronting before the subject element: *Under this tree he used to rest.*

Marking a part of the semantic structure as *backgrounded* will lead to its realization as a parenthetical construction: *John (well known among the students and the professors alike) was invited as guest speaker.*

The necessity of a distinction between *given* and *new* as discussed, for instance, by Lambrecht (1994) is most evident: If an object node in the semantic structure is marked as *new*, its realization in the syntactic structure will be assigned an indefinite determiner (or no determiner at all): *A masked man was seen to enter the bank* (*man* is newly introduced into the discourse). If a node is marked as *given*, its syntactic realization will be assigned a definite determiner: *The masked man* (whom a passer-by noticed before) *was seen to enter the bank.*[42] To cope with the distinction between demonstratives and definite/indefinite articles, a gradation of givenness in the sense of Gundel et al. (1989) is necessary.

As far as communicative structure is concerned, the main problem is that there is no reliable way to identify automatically the exact thematicity, foregroundedness or givenness of the components of a sentence: there are no systematic cues that indicate a precise communicative status, be it words, grammemes or syntactic construction. As a result, for our experiments, we had to make simplified assumptions in order to superimpose the communicative structure onto the deep representation:

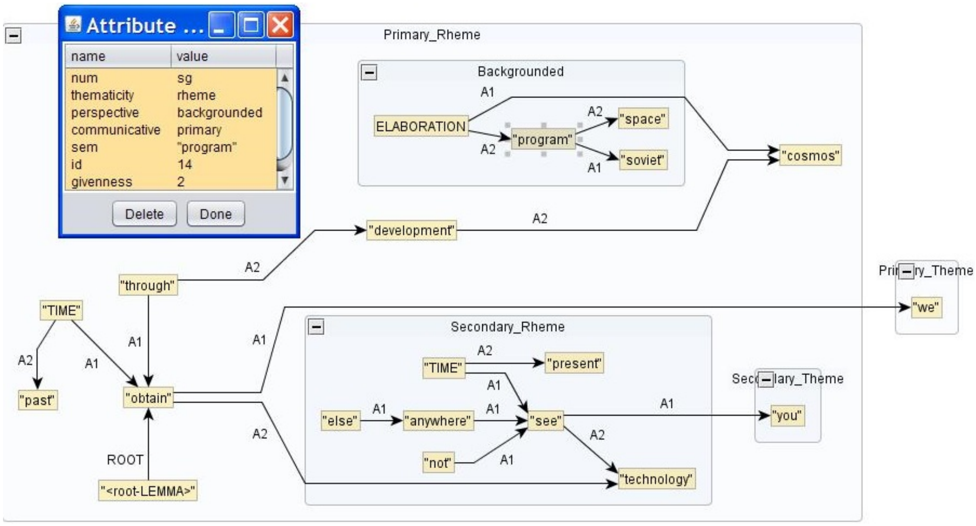
- A subject and its syntactic dependents represent the theme of a sentence, and the verb and it other dependents form the rheme. If the verb is the main verb of the sentence, theme and rheme are marked as *primary*; if the verb is embedded below a main verb, they are marked as *secondary*, and so on. We do not consider specifiers.
- The determiners "the" and "a" are respectively replaced by the attribute/value pairs *givenness=given* and *givenness=new* on the governing noun in syntax. Other types of givenness are not handled;
- When the governor is a verb, an adverbial group anteposed to the subject is marked as *foregrounded*, it is after the objects behind a comma, it is marked as *backgrounded*. When the governor is a noun, if there is comma between it and one of its modifiers, the latter is considered *backgrounded*. The rest is considered *neutral*.

Consider in Figure 3.28 an example of semantic annotation with its two structures.[43] All syntactic nodes have been removed, and all the remaining nodes are connected in terms of a predicate–argument structure, with no use of any syntactically motivated edge. Figure 3.28 illustrates the three main aspects of Informativity: (i) thematicity, with the two *theme/rheme* oppositions; (ii) foregroundedness, with the *backgrounded* part of the primary rheme; and (iii) givenness, with the attribute *givenness* and the value

---

[42] Actually, a generator has to be able to chose whether or not to introduce a determiner in a given context.

[43] Some meta-semantemes are not shown in the figure ('TEM_CONSTITUENCY', 'NUMBER', etc.).



name	value
num	sg
thematicity	rheme
perspective	backgrounded
communicative	primary
sem	"program"
id	14
givenness	2

Figure 3.28: Illustration of the semantic annotation of the sentence "*Through the development of Cosmos, the Soviet space program, we obtained technologies you do not see anywhere else.*"

2 on the node *program*. The communicative structure constrains the superficial realization of the sentence in that the primary theme will be the subject of the sentence, and the main node of the primary rheme pointing to it will be the main verb of the same sentence. The secondary theme and rheme will be realized as an embedded sentence in which *you* will be the subject, that is, forcing the realization of a relative clause. However, it does not constrain the appearance of a relative pronoun. For instance, *we obtained technologies you do not see anywhere else* and *we obtained technologies that you do not see anywhere else* are possible realizations of this structure. Leaving the relative pronoun in the semantic structure would force one realization to occur when it does not have to (both outputs are equally correct and meaning-equivalent to the other). Similarly, marking *the Soviet space program* as backgrounded leaves some doors open when it comes to surface realization: *Cosmos, the Soviet space program* vs. *Cosmos* (*the Soviet space program*) vs. *the Soviet space program Cosmos* (if *Cosmos* is backgrounded too) are possible realizations of this substructure.

3.5.6 Evaluation

All the transformations described in this subsection were implemented as a Java program and run on the PTB/PB/NB corpus. The goal of the following evaluation is to assess how well we can do with the mapping of the PTB/PB/NB annotation onto a well-formed syntax-void semantic structure. That is, it is not our aim to calculate the “absolute” quality of the obtained semantic annotation against a gold standard, but, rather, the “relative” quality achieved starting from the PTB/PB/NB annotation. Thus, if a node is not annotated and labeled *DEP* (meaning, “unknown dependency”) in PTB/PB/NB, it cannot take influence on the calculation of the accuracy of our mapping.

The preliminary evaluation consists of two parts: the evaluation of the predicate-argument structure, and the evaluation of the communicative structure. Both parts have been performed on the structures of 90 sentences from PB/NB.

Each structure obtained by the mapping procedure was manually examined and a minimal number of actions was performed, adding and removing nodes and relations to reach a correct (= purely semantic and well-formed) structure. Each action corresponds to an error. The following error typology was applied:

1. error in PTB/PropBank
2. error in the mapping procedure
    a. missing semantic node (added)
    b. surplus semantic node (removed)
    c. missing arc between two non-disconnected semantic nodes (added)
    d. wrongly established/directed/labeled arc (removed/inverted/relabeled)
    e. disconnected semantic node (connected)
    f. syntactic node (removed)
    g. syntactic arc (removed)

In the case of a removed node, all outgoing and incoming relations have been counted as removed arcs. In the case of an added node, all relations added at the same time have also been counted as added arcs. For the evaluation of the communicative structure, we simply counted how many times a node was assigned the correct communicative span.

The 1723 tokens in 90 test sentences correspond to 1367 semantic nodes (Total Nodes TN=1367) and 1380 semantic arcs (Total Arcs TA=1380) in the structures obtained by the mapping procedure. Table 3.17 displays the absolute error type count. The figures in the first line capture all errors, no matter whether they stem from the mapping itself or from the original PTB/PropBank annotation; the figures in the second line capture errors of the mapping only.

	(a) +SemN	(b) −SemN	(c) +SemA	(d) −SemA
total	25	5	90	157
mapping	23	1	69	109

	(e) −Discon.	(f) −NonSemN	(g) −NonSemA
total	0	43	1
mapping	0	39	1

Table 3.17: Error type count out of the automatic mapping

The structures obtained by applying manual corrective actions (a-g) to the automatically obtained structures served us as gold standard. Table 3.18 displays the evaluation of the mapping against this gold standard. The numbers in the first line reflect the performance of our mapping with the imperfect original PTB/PB/NB annotation; the numbers in the second line present the actual performance of our mapping, without billing the PTB/PB/NB annotation errors. "Connect." reflects the connectivity rate of the resulting structure (i.e., the ratio between the number of connected nodes and the total number of nodes in the resulting structure: (TA-e)/TA); "node p/r" stands for precision and recall of semantic node introduction: how many of the introduced nodes are semantic and required ((TN-b-f)/TN) and how many of the required nodes have been introduced ((TN-b-f)/(TN-b-f+a)); "arc dir/lab p/r" stands for precision ((TA-d-g)/TA) and recall ((TA-d-g)/(TA-d-g+c)) of semantic arc introduction and semantic arc labeling; "th-rh p/r" for precision and recall of theme/rheme introduction; "foregr/backgr. p/r" for precision and recall of the foregroundedness annotation; and "given p/r" for precision and recall of the givenness annotation.

The evaluation shows that the conversion experiment was successful with respect to the removal of syntactic nodes, introduction of semantic nodes and connecting the nodes to a connected graph. The introduction of communicative structure features also seems to have succeeded—except for the recall of the *foregr./backgr.* feature, which is low: 0.382.

	connect.	node p	node r	arc dir/lab. p	arc dir/lab. r
total	1	0.965	0.981	0.886	0.931
mapping	1	0.971	0.983	0.920	0.948

th-rh p	th-rh r	foregr./backgr. p	foregr./backgr.r	given p	given r
0.986	1.0	0.905	0.382	1.0	0.986

Table 3.18: Evaluation of the mapping of the PropBank annotation

The figures are somewhat better when the errors of the PTB/PB/NB annotation are ignored, but the difference is not striking. This shows the high quality and consistency of their annotation and underlines its suitability as starting annotation.

Quality checks

Manual and automatic reviewing of the structures would help control better the adequacy of the deep representation and improve the mapping.

- Automatic checks would include (i) connectivity and (ii) well-formedness (in particular: no duplicated arguments for any predicate, no erroneous edges, correct marking of the root of each sentence);
- Manual checks would cover what cannot be done automatically, in particular, missing or incorrect edges or nodes; this part is rather tedious.

3.5.7 IDs and format

It is important to ensure that there are links between the semantic and syntactic IDs. It is possible to do as for the SRST, that is to provide a file with correspondences between superficial and deep IDs, or to do as we suggest for Spanish in this thesis, that is, to encode IDs as attribute/value pairs associated to the nodes of the structures, which is what we did for the experiments.

With respect to the format of the corpus, as was done with the SRST, we keep the PTB dependencies and morpho-syntactic annotation in the CoNLL format. As for the deep input, for our experiments we use the MATE graph format, displayed in Figure 3.29.

```
structure Sem S {
   and:0 {
       sem=and
       A1→ tractor:1{sem=tractor}
       A2→ machinery:2{
           sem=machinery
           A1→ process:3{
               sem=process
               NAME→ food:4{
                   sem=food
                   NAME→ “-”:5{sem=“-”}
               }
           }
       }
   }
   and:6{
       sem=and
       A1→ television:7{sem=television}
       A2→ recorder:8{
           sem=recorder
           A1→ videocassette:9{sem=videocassette}
       }
   }
   produce:10{
       sem=produce
       A1→ television:7
   }
   small:11{
       sem=small
       A1→ tractor:1
   }
   and:12{
       sem=and
       A1→ recorder:8
       A2→ tractor:1
   }
}
```

Figure 3.29: Figure 3.27 in the MATE format (*produce televisions, videocassette recorders, small tractors and food-processing machinery*)

3.5.8 Conclusion

Even if the automatic mapping proved feasible, the amount of workload involved remains quite important and the result is not perfect. Furthermore, the derivations do not result in a genuine semantic structure, since a number of surface-oriented, syntactic features remain (cf determiners and continuation structures for instance), but it allowed for performing a series of experiments with statistical NLG which imply most operation that a generator has to be able to perform: starting form a deep input, decide the syntactic structure of a sentence, introduce functional words and punctuation signs, order the words and manage the agreements between them. Such experiments are shown in the next chapter.

Chapter **4**

Experiments on deep stochastic text generation

In Chapter 1, we established the main objectives of the thesis: to design and apply an annotation scheme for producing data which is suitable for corpus-based Natural Language Generation, and to use the data for deep stochastic NLG experiments. Now that we have presented the annotation in Chapter 3, we can embark on Machine Learning (ML) techniques for NLG. From a general perspective, applying ML techniques to NLG means aligning two structures of different levels of abstraction, and find regularities in the mapping of one onto the other, based on a selection of features present in the annotated data. For example, when aligning DSyntSs and SSyntSs, it is easy (for a human and for a training algorithm) to notice that whenever a DSyntS noun is the first argument of a DSyntS verb that carries the attributes *finiteness=FINITE* and *voice=ACTIVE*, the corresponding noun in the SSyntS is the *subject* of the corresponding verb. The present chapter accounts for three experiments performed with different systems and datasets. First of all, in Section 4.1, we go beyond the current state of the art by presenting a fully statistical deep generator of Spanish which draws upon all levels of annotation (semantic, syntactic and topological) for sentence generation in a genuinely statistical manner. This implies the handling of non-isomorphic mappings. Then, we describe systems which are tuned for performing high quality deep NLG on automatically annotated data. In Section 4.2, we present a prototype of such a system, designed for handling only isomorphic mappings, and show that it works with languages as different as Spanish, English, German, and Chinese. In Section 4.3, we extend this system so as to obtain a more powerful generator which handles non-isomorphic transitions, thanks to corpus-learned rules. This deep generator has been presented at the 2011 Surface-Realization Shared Task.[1]

In this chapter, we use the same terminology as in Chapter 3 for the levels of representation:

- SemS: a predicate-argument graph without functional nodes;
- DSyntS: a non-ordered syntactic tree with abstract dependency labels and without functional nodes;
- SSyntS: a non-ordered syntactic tree with idiosyncratic dependency labels and all nodes;
- MorphS: a chain of non-inflected words which bear morphological features;
- Sentence: a chain of inflected words.

For some experiments, it happens that we introduce new level names, but in this case we explain the differences with what we have seen so far.

4.1 Non-isomorphic stochastic graph transduction

In Section 2.1, we pointed out that no currently existing deep generator is able to handle non-isomorphic mappings, i.e., mappings between two structures which do not contain the same amount of nodes, without using rules. Non-isomorphic mappings are necessary in the generation pipeline, in particular in order to map a deep-syntactic structure, which contains only meaningful words, onto a surface-syntactic structure, which contains all the words of a sentence. The generator presented in this section is trained on the multilayered annotation presented in Chapter 3.[2]

For this experiment, we designed a system based on classifiers which are able to produce the functional words and insert them into the syntactic

---

[1] The system of Section 4.1 has been implemented by Miguel Ballesteros and Bernd Bohnet, and those of Sections 4.2 and 4.3 by Bernd Bohnet.

[2] This experiment has been described in (Bohnet et al., 2011b) and (Ballesteros et al., 2014b).

structure. Then, the nodes are ordered and inflected. Two approaches based on a cascade of Support Vector Machine (SVM) classifiers are presented, showing that fragmenting the decisions significantly improves the quality of the projection. The generator starts from abstract structures, which we have been referring to as *deep-syntactic structures* so far.

In this section, we first describe briefly the task (Section 4.1.1), then the classifiers for the different transitions (DSyntS-SSyntS—Section 4.1.2—, SSyntS-MorphS and MorphS-Sentence—Section 4.1.3). Then, the experiment is described and the obtained results discussed (Section 4.1.4).

4.1.1 The Task

As shown in Chapter 3, a difference in the linguistic abstraction of deep- and surface-syntactic structures leads to divergences that impede the isomorphy between the two and make the mapping between them a challenge for statistical generation. In order to handle this isomorphy more easily, we introduce the notion of a *hypernode*:

Definition 4.1 (Hypernode)**.** *Given a SSyntS S_s with its index matrix I and a DSyntS S_d with its index matrix I', a node partition p (with $|p| \geq 1$) of I/I' is a hypernode h_{s_i} / h_{d_i} iff p corresponds to a partition p' (with $|p'| \geq 1$) of S_d/S_s.*

In other words, a SSyntS hypernode, known as *syntagm* in linguistics, is any SSyntS configuration with a cardinality ≥1 that corresponds to a single DSyntS node. For instance, the complex verb forms, which are analytical in Spanish, e.g., *ha sido invitado* ‘she-has been invited’, constitute a hypernode because they correspond to the single node *invitar* ‘invite’ in the DSyntS. In this way, the SSyntS–DSyntS correspondence boils down to a correspondence between individual (hyper)nodes and between individual arcs. Note that this notion of hypernode is somewhat similar to Stanford’s collapsed dependencies (henceforth *StDs* (de Marneffe and Manning, 2008)). The main differences between the latter and the DSyntSs (apart from the fact that StDs may be (sometimes disconnected) graphs) are: (i) StDs collapse only (but all) prepositions and some conjunctions, whereas DSyntSs omit all functional nodes (auxiliaries, determiners, and some prepositions); (ii) collapsed StDs do not involve any removal of (syntactic) information since the meaning of the preposition remains encoded in the label of the collapsed dependency, while DSyntSs omit or generalize the purely functional elements. That is, collapsed StDs keep the surface-syntactic information, representing it in a different format, while the DSyntSs keep only deep-syntactic information, such that the transition from SSyntS to DSyntS is to be realized by appropriate linguistic structure induction.

Let us, before we come to the presentation of the implementation, summarize the tasks involved in the projection of a DSyntS onto its corresponding sentence in the course of sentence generation. First, we define four tasks for the DSyntS-SSyntS generation:

1. Project each node in the DSyntS onto its SSynS-correspondence. This correspondence can be a single node, as, e.g., *successful* → *successful*, or a subtree (*hypernode*), as, e.g., *song* → *the song* [*DT NN*] (where *DT* is a determiner and *NN* a noun) or *be* → *that will be* [*IN* V_{AUX} *VB*] (where *IN* is a conjunction, V_{AUX} an auxiliary and *VB* a full verb).
2. Generate the correct lemma for the nodes in SSyntS that do not have a 1:1 correspondence in the SSyntS (as *DT*, *IN* and V_{AUX} above).
3. Establish the dependencies within the individual SSyntS-hypernodes.
4. Establish the dependencies between the SSyntS-hypernodes (more precisely, between the nodes of different SSyntS-hypernodes) to obtain a connected SSyntS-tree.

The mapping between SSyntS and a full fledged sentence is then realized in two steps:
5. Establish the order between all the SSyntS nodes.
6. Generate the final form of the words which need to be inflected.

4.1.2 Classifiers for the SSyntS-DSyntS transition

The realization of the actions 1.–4. from above can be approached either in terms of 4 generic classifiers or in terms of 4 sets of fine-grained (micro) classifiers that map between one representation to another. The idea behind these experiments is to find out whether it is sufficient to implement a small set of classifiers for the SSyntS-DSyntS transition, or if fragmenting the decision process will lead to significantly better results.

4.1.2.1 Generic classifier approach

Each of the generic classifiers deals with one of the following tasks.

a. Hypernode Identification: Given a deep syntactic node n_d from the DSyntS, the system must find the shape of the surface hypernode that corresponds to n_d in the SSyntS. The hypernode identification SVM uses the features in Table 4.1.

#	**features**
1	PoS of n_d
2	PoS of n_d's head
3	voice
4	tem_constituency
5	finiteness
6	tense
7	lemma of n_d
8	n_d's dependencies

Table 4.1: Feature schemas used for hypernode identification

In order to simplify the task, we define the shape of a surface hypernode as a list of surface PoS-tags. This list contains the PoS of each of the lemmas within the hypernode and a tag that signals the original deep node; for instance:

$$[\text{ VB(deep)}, \text{V}_{AUX}, \text{IN}]$$

b. Lemma Generation. Once the hypernodes of the SSyntS under construction have been produced, the functional nodes that have been newly introduced in the hypernodes must be assigned a lemma. The lemma generation SVM uses the features in Table 4.2 of the deep nodes n_d in the hypernodes to select the most likely lemma.

#	**features**
1	finiteness
2	definiteness
3	PoS of n_d
4	lemma of n_d
5	PoS of the head of n_d

Table 4.2: Feature schemas used for lemma generation

c. Intra-hypernode Dependency Generation. Given a hypernode and its lemmas provided by the two previous stages, the dependencies (i.e., the dependency attachments and dependency labels) between the elements of the hypernode must be determined (and thus also the governor of the hypernode). For this task, the intra-hypernode dependency generation SVM uses the features in Table 4.3.

#	**features**
1	lemmas included in the hypernode
2	PoS-tags of the lemmas in the hypernode
3	*voice* of the head h of the hypernode
4	deep dependency relation to h

Table 4.3: Feature schemas used for Intra-hypernode dependency generation

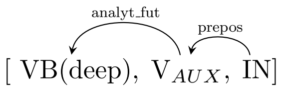


Figure 4.1: Internal dependency within a hypernode

d. Inter-hypernode Dependency Generation. Once the individual hypernodes have been converted into connected dependency subtrees, the hypernodes must be connected between each other, such that we obtain a complete SSyntS. The inter-hypernode dependency generation SVM uses the features of a hypernode s_s to determine for each hypernode its governor, see Table 4.4.[3]

#	**features**
1	internal dependencies of s_s
2	head of s_s
3	lemmas of s_s
4	PoS of the dependent of the head of s_s in DSyntS

Table 4.4: Feature schemas used for Inter-hypernode dependency generation

---

[3] The task is of the inter-hypernode dependency classifiers is the same as that of a dependency parser, only that its search space is very small.

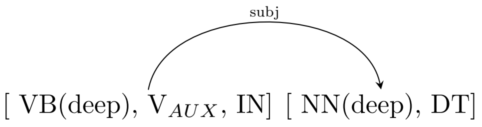


Figure 4.2: Surface dependencies between two hypernodes

4.1.2.2 Implementation of sets of micro classifiers

In this alternative approach, a single classifier is foreseen for each kind of input. Thus, for the **hypernode identification module**, for each deep PoS tag (which can be one of the following four: *N* (noun), *V* (verb), *Adv* (adverb), *A* (adjective)), a separate multi-class classifier is defined. For instance, in the case of *N*, the N-classifier will use the above features to assign to the a DSynt-node with PoS *N* the most appropriate (most likely) hypernode—in this case, [*NN(deep)*, *DT*]. In a similar way, in the case of the **lemma generation module**, for each surface PoS tag, a separate classifier is defined. Thus, the DT-classifier would pick for the hypernode [*NN(deep)*, *DT*] the most likely lemma for the DT-node (optimally, a determiner).

For the **intra-hypernode attachments module**, for each kind of hypernode, a separate classifier is generated dynamically.[4] In the case of the hypernode [*VB(deep)*, V_{AUX}, *IN*], the corresponding classifier will create a link between the conjunction and the auxiliary, and between the auxiliary and the verb, with respectively the conjunction and the auxiliary as heads because it is the best link that it can find; cf. Figure 4.1 for illustration.

Finally, for the **inter-hypernode attachments module**, for each hypernode with a distinct internal dependency pattern, a separate classifier is dynamically derived (for our treebank, we obtained 114 different SVM classifiers because it also takes into account hypernodes with just one token). For instance, the classifier for the hypernode [*NN(deep)*, *DT*] is most likely to identify as its governor V_{AUX} in the hypernode [*VB(deep)*, V_{AUX}, *IN*]; cf. Figure 4.2.

---

[4] This implies that the number of classifiers varies depending on the training set, in the intra-hypernode dependency generation there are 108 SVMs.

4.1.3 Decoders for the SSyntS-MorphS and MorphS-Sentence transitions

The SSyntS-MorphS decoder derives from a dependency tree a chain of lemmas, i.e., determines the word order within the sentence. The MorphS-Sentence decoder generates the inflected word form for each lemma in the chain. To compute the score of the alternative realizations by each decoder, MIRA (Margin Infused Relaxed Algorithm) has been applied to the features provided by the feature extractors.[5] Note that both the feature extractors and the decoders presented below are language-independent, which makes the realizer applicable to any language for which multilevel-annotated corpora are available (see Section 4.2).

4.1.3.1 SSyntS-MorphS transition

Since we use unordered dependency trees as syntactic structures, the realizer has to find the optimal linear order for the lexemes of each dependency tree. Algorithm 1 shows the linearization algorithm used for the experiment.

The algorithm is a beam search. It starts with an elementary list for each node of the dependency tree. Each elementary list is first extended by the children of the node in the list; then, the lists are extended stepwise by the children of the newly added nodes. If the number of lists during this procedure exceeds the threshold of 1000, the lists are sorted in accordance with their score, and the first 1000 are kept. The remaining lists are removed. Afterwards, the score of each list is adjusted according to a global score function which takes into account complex features such as the first word of a constituent, last word, the head, and the edge label to the head (cf. Table 4.5 for the list of the features). Finally, the nodes of the dependency tree are ordered with respect to the highest ranked lists. Only in a very rare case, the threshold of the beam search is exceeded. Even with a rich feature set, the procedure is very fast: the linearization takes about 3 milliseconds in average per dependency tree on a computer with a 2.8 Ghz CPU.

---

[5] MIRA is one of the most successful large-margin training techniques for structured data (Crammer et al., 2006). It has been used, e.g., for dependency parsing, semantic role labeling, chunking and tagging.

#	**word-pairs(w_1,w_2)**
1	label$_{w1}$+label$_{w2}$
2	label$_{w1}$+lemma$_1$
3	label$_{w1}$+lemma$_2$
4	label$_{w2}$+lemma$_1$
5	label$_{w2}$+lemma$_2$
6	PoS$_1$+PoS$_2$
7	PoS$_1$+PoS$_2$+head(w$_1$,w$_2$)
8	label$_{w1}$+label$_{w2}$+PoS$_1$+head(w$_1$,w$_2$)
9	label$_{w1}$+label$_{w2}$+PoS$_2$+head(w$_1$,w$_2$)
10	label$_{w1}$+label$_{w2}$+PoS$_1$+PoS$_2$+head(w$_1$,w$_2$)
11	label$_{w1}$+label$_{w2}$+PoS$_1$+#children$_2$+head(w$_1$,w$_2$)
12	label$_{w1}$+label$_{w2}$+PoS$_2$+#children$_1$+head(w$_1$,w$_2$)

#	**n-grams**
13	PoS$_1$+PoS$_2$+PoS$_3$
14	PoS$_1$+PoS$_2$+PoS$_3$+dist
15	lemma$_1$+lemma$_2$+lemma$_3$
16	lemma$_1$+lemma$_2$+lemma$_3$+dist
17	lemma$_1$+lemma$_3$+head(w1,w2,w3)
18	lemma$_1$+lemma$_3$+head(w1,w2,w3)+dist
19	label$_1$+label$_2$+label$_3$+head(w1,w2,w3)
20	label$_1$+label$_2$+label$_3$+head(w1,w2,w3)+dist
21	label$_1$+label$_2$+label$_3$+lemma$_1$+PoS$_2$+head(w1,w2,w3)
22	label$_1$+label$_2$+label$_3$+lemma$_1$+PoS$_2$+head(w1,w2,w3)+dist
23	label$_1$+label$_2$+label$_3$+lemma$_2$+PoS$_1$+head(w1,w2,w3)
24	label$_1$+label$_2$+label$_3$+lemma$_2$+PoS$_1$+head(w1,w2,w3)+dist

#	**global features for constituents**
25	**if** $\lvert$constituent$\rvert > 1$ **then** label$_{1st}$+label$_{last}$+label$_{last-1}$+PoS$_{first}$+PoS$_{last}$+PoS$_{head}$
26	**if** $\lvert$constituent$\rvert > 2$ **then** label$_{1st}$+label$_{2d}$+label$_{3d}$+PoS$_{last}$+PoS$_{last-1}$+PoS$_{head}$+contains-?
27	**if** $\lvert$constituent$\rvert > 2$ **then** label$_{1st}$+label$_{2d}$+label$_{3d}$+PoS$_{last}$+PoS$_{last-1}$+lemma$_{head}$+contains-?
28	**if** $\lvert$constituent$\rvert > 3$ **then** PoS$_{1st}$+PoS$_{2d}$+PoS$_{3d}$+PoS$_{4th}$+PoS$_{last}$+label$_{head}$ +contains-?+pos-head
29	**if** $\lvert$constituent$\rvert > 3$ **then** PoS$_{last}$+PoS$_{last-1}$+PoS$_{last-2}$+PoS$_{last-3}$+PoS$_{first}$ +label$_{head}$+contains-? +pos-head
30	PoS$_{first}$+PoS$_{last}$+lemma$_{first}$+lemma$_{last}$+lemma$_{head}$+contains-?+pos-head

Table 4.5: Feature schemas used for linearization ($label_w$ is the label of the in-going edge to a word w in the dependency tree; $lemma_w$ is the lemma of w, and PoS_w is the Part-of-Speech tag of w; *head*(w$_1$,w$_2$, ...) is a function which is 1 if w_1 is the head, 2 if w_2 is the head, etc. and else 0; *dist* is the position within the constituent; *contains-?* is a boolean value which is true if the sentence contains a question mark and false otherwise; *pos-head* is the position of the head in the constituent)

Algorithm 1: Dependency tree linearization

```
//$y_i$ a dependency tree
 for i ← 1  to $|I|$ // iteration over the training examples
  // iterate over all nodes of the dependency tree $y_i$
  for n ← 1 to $|y_i|$  do
    $subtree_n$ ← $children$(n) ∪ {n}
    ordered-lists$_n$ ← {} // initialize
    for all m ∈ $subtree_n$  do
      beam ← {}
      for all $l$ ∈ ordered-lists  do
        beam ← beam  ∪{ $append(clone(l)$,m)}
      for all $l$ ∈ ordered-lists  do
        score($l$) ←  compute-score-for-word-list($l$)
      sort-lists-descending-to-score(beam,score)
      if | beam | > beam-size  then
        beam ← sublist(0,1000,beam)
      ordered-lists$_n$ ← beam
    $score_g(l)$ ← $score(l)$ + $compute\text{-}global\text{-}score(l)$
    sort-lists-descending-in-score(beam,score$_g$)
```

4.1.3.2 MorphS-Sentence transition

The morphological realization uses the minimal string edit distance (Levenshtein, 1966) to map lemmas to word forms. As input to the MIRA-classifier, we use the lemmas of a sentence, its dependency tree and the already ordered sentence. The characters of the input strings are reversed since most of the changes occur at the end of the words and the string edit scripts work relatively to the beginning of the string. For example, to calculate the minimal string edit distance between the lemma *go* and the form *goes*, both are first reversed by the function **compute-edit-dist** and then the minimal string edit script between *og* and *seog* is computed. The resulting script is *Ie0Is0*. It translates into the operations 'insert *e* at the position 0 of the input string' and 'insert *s* at the position 0'.

Before MIRA starts, we compute all minimal edit distance scripts to be used as classes of MIRA. Only scripts that occur more often than twice are used.[6] The training algorithms typically perform 6 iterations (*epochs*) over the training examples. For each training example, a minimal edit script

---

[6] The number of the resulting edit scripts is language-dependent; e.g., we get about 1500 scripts for English and 2500 for German for the experiment described in Section 4.2.

Algorithm 2: Morphological realization training with MIRA

```
// $y_i, l_i$; $y_i$ is a dependency tree, $l_i$ lemmatized sentence
script-list ← {} //initialize the script-list
 for $i \leftarrow 1$ to $|I|$ // iteration over the training examples
   for $l \leftarrow 1$ to $|l_i|$ do//// iteration over the lemmas of $l_i$
     lemma$_l$ ← lower-case ($l_i$,$l$)
     //ensure that all lemmas start with a lower case letter
     script ← compute-edit-dist-script(lemma$_l$, form($l_i$,$l$))
      if script $\notin$ script-list
        script-list ← script-list ∪ {script }
 for $k \leftarrow 1$ to $E$ // $E$ = number of training epochs
   for $i \leftarrow 1$ to $|I|$ // iteration over the training examples
     for $l \leftarrow 1$ to $|l_i|$ do
        script$_p$ ← predict-script($l_i$,$y_i$,$l$)
        script$_g$ ← edit-dist-script(lemma$_l$, form($l_i$,$l$))
         if script$_p \neq$ script$_g$ then
        // update the weight vector $v$ and the vector $w$, which
        // averages over all collected weight vectors acc.
        // to diff. of the predicted and gold feature vector
         update $w, v$ according to $\Delta(\phi$(script$_p$), $\phi$(script$_g$))
        //with $\phi$(script$_p$), $\phi$(script$_g$) as feature vectors of
        //script$_p$ and script$_g$, respectively
```

Algorithm 3: Morphological realization

```
// $y_i$ a dependency tree, and $l_i$ an ordered list of lemmas
 for $l \leftarrow 1$ to $|l_i|$ do
   script$_p$ ← predict-script($l_i$,$y_i$,$l$)
   form$_l$ ← apply-edit-dist-script(lemma$_l$, script$_p$)
```

is selected. If this script is different from the gold script, the features of the gold script are calculated and the weight vector of the SVM is adjusted according to the difference between the predicted vector and the *gold feature vector*. The classification task consists then in finding the classification script that maps the lemma to the correct word form. For this purpose, the classifier scores each of the minimal edit scripts according to the input, choosing the one with the highest score.

The morphological realization algorithm selects the edit script in accordance with the highest score for each lemma of a sentence obtained during

#	**features**
1	es+lemma
2	es+lemma+m.feats
3	es+lemma+m.feats+POS
4	es+lemma+m.feats+POS+position
5	es+lemma+(lemma+1)+m.feats
6	es+lemma+(lemma+1)+POS
7	es+lemma+(m.feats-1)+(POS-1)
8	es+lemma+(m.feats-1)+(POS-1)+position
9	es+m.feats+(m.feats-1)
10	es+m.feats+(m.feats+1)
11	es+lemma+(m.feats-1)
12	es+m.feats+(m.feats-1)+(m.feats-2)
13	es+m.feats+POS
14	es+m.feats+(m.feats+1)
15	es+m.feats+(m.feats+1)+lemma
16	es+m.feats
17	es+e0+e1+m.feats
18	es+e0+e1+e2+m.feats
19	es+e0+e1+e2+e3+m.feats
20	es+e0+e1+e2+e3+e4+m.feats
21	es+e0+m.feats

Table 4.6: Feature schemas used for morphological realization

training (see Algorithm 2 above) and applies then the scripts to obtain the word forms; cf. Algorithm 3. Table 4.6 lists the feature schemas used for morphological realization.

4.1.4 Experiments and results

4.1.4.1 Setup and metrics

In the experiments, we want to calculate how good the system is at producing correct sentences from deep-syntactic structures. Following a classical machine learning set-up, we divide the treebank presented in Chapter 3 into: (i) a development set (219 sentences, 3271 tokens in the DSyntS treebank and 4953 tokens in the SSyntS treebank); (ii) a training set (3036 sentences, 57665 tokens in the DSyntS treebank and 86984 tokens in the SSyntS treebank); and a (iii) a held-out test for evaluation (258 sentences, 5641 tokens in the DSyntS treebank and 8955 tokens in the SSyntS treebank).

In order to see which granularity of surface-syntactic tag gives the best

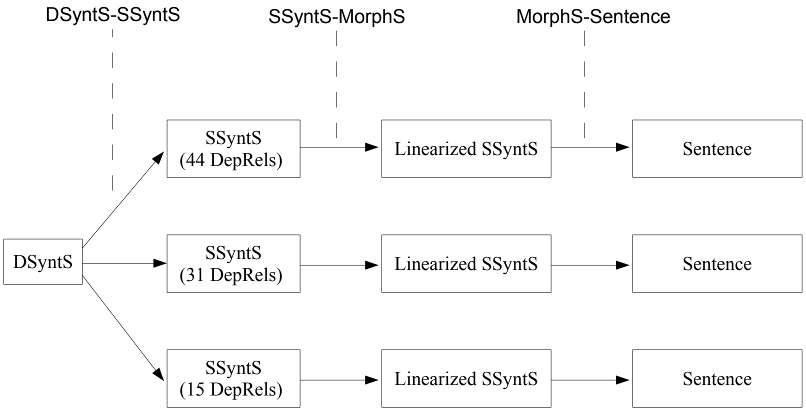


Figure 4.3: Setup of the experiments on non-isomorphic deep stochastic NLG

results, we run several times the experiment, once for each tagset of the hierarchy presented in Section 3.3.3.1.[7] Figure 4.3 gives an overview of the experiments.

To assess the quality of the DSyntS-SSyntS mapping, we simply compare each generated SSyntS to its corresponding gold SSyntS and count the differences in terms of nodes and dependencies. In order to compare these results with other systems, we assess the quality of the modules which include linearization via BLEU score, NIST and exactly matched sentences.

4.1.4.2 Evaluation of the DSyntS-Sentence pipeline

In this section, we first present the performance of the two approaches to DSyntS–SSyntS projection on the DSynt and SSynt layers of the treebank, and then the performance of the whole pipeline with the micro-classifier approach. Tables 4.7 to 4.9 display the results for both the generic classifier and the sets of micro classifiers for all SSyntS–DSyntS tasks on the test set, with different granularity of annotation at the surface-syntactic level. For each set of classifiers, we provide an overall measure (*ALL* at the bottom of each table). Since we simply add up the figures obtained for each submodule, this measure does not indicate how good is a generator performing,

---

[7] To be precise, we used the 44-, 31-, and 15-label tagsets shown in Section 5.1.2.2.

Generic classifiers, 15 SSyntRels

Hypernode identification	5166/5887	87.75
Lemma generation	1822/2084	87.43
Intra-hypernode dep. generation	1096/1699	64.51
Inter-hypernode dep. generation	4932/5385	91.59
ALL	**13016/15055**	**86.46**

Micro classifiers, 15 SSyntRels

Hypernode identification	5170/5887	87.82
Lemma generation	1913/2084	91.79
Intra-hypernode dep. generation	1691/1699	99.53
Inter-hypernode dep. generation	4921/5385	91.38
ALL	**13695/15055**	**90.97**

Table 4.7: Results of the evaluation on the test set of the different classifiers for the non-isomorphic transduction (15 SSyntRels)

Generic classifiers, 31 SSyntRels

Hypernode identification	5166/5887	87.75
Lemma generation	1822/2084	87.43
Intra-hypernode dep. generation	1096/1699	64.51
Inter-hypernode dep. generation	4844/5385	89.95
ALL	**12928/15055**	**85.87**

Micro classifiers, 31 SSyntRels

Hypernode identification	5169/5887	87.80
Lemma generation	1913/2084	91.79
Intra-hypernode dep. generation	1691/1699	99.53
Inter-hypernode dep. generation	4832/5385	89.73
ALL	**13605/15055**	**90.37**

Table 4.8: Results of the evaluation on the test set of the different classifiers for the non-isomorphic transduction (31 SSyntRels)

but rather the ratio of good decisions that it takes, be it on identifying hypernodes, generating lemmas or intra- and inter-hypernode dependencies. This simply gives a global view of each system and makes the comparison between them easier.

Table 4.10 shows the results of the whole pipeline as well as summarizes the figures obtained by each component. Note that we exclude all punctuation marks from the evaluation since in the corpus they are directly attached to the word they follow and hence they would distort the evaluation, and

Generic classifiers, 44 SSyntRels

Hypernode identification	5166/5887	87.75
Lemma generation	1822/2084	87.43
Intra-hypernode dep. generation	1093/1699	64.33
Inter-hypernode dep. generation	4768/5385	88.54
ALL	**12849/15055**	**85.35**

Micro classifiers, 44 SSyntRels

Hypernode identification	5170/5887	87.82
Lemma generation	1913/2084	91.79
Intra-hypernode dep. generation	1653/1699	97.29
Inter-hypernode dep. generation	4744/5385	88.10
ALL	**13480/15055**	**89.54**

Table 4.9: Results of the evaluation on the test set of the different classifiers for the non-isomorphic transduction (44 SSyntRels)

the evaluation of the linearizer is also performed on the lemmas to exclude effects of the word form generation (SSyntS-MorphS and DSyntS-MorphS).

The average sentence length in our original test set being very high (31 words), we also performed an evaluation on a subset of sentences of more common length (in particular, what we have for the evaluations with English described in the next sections), with 16 words per sentence, with an improvement of around 0.05 BLEU score.

Sample outputs of this system are provided in Appendix B.

4.1.4.3 Discussion

In general, most of the statistical state-of-the-art approaches to structure prediction use a single classifier model (Smith, 2011). But we are not the first to propose a multi-classifier solution either. For instance, Carreras et al. (2008) use different models to predict each part of the triplet for spinal model pruning, and for semantic role labeling, there are several systems that use a set of classifiers for predicate identification; cf., e.g., (Björkelund et al., 2010; Johansson and Nugues, 2008a).

The results show that for hypernode identification and inter-hypernode dependency generation, the results of both types of classifiers are comparable. However, thanks to the micro classifiers, with the same features, the lemma generation model improves by 4 points and the intra-hypernode dependency generation by around 30 points for all SSyntRel granularities. This means

15 SSyntRels

DSyntS→SSyntS	90.97 %		
	BLEU	NIST	Exact
SSyntS→MorphS	0.81	12.5	17.4%
SSyntS→Sent	0.80	11.6	9.25%
DSyntS→SSyntS→MorphS	**0.53**	**11.11**	**5.4%**
DSyntS→SSyntS→MorphS→Sent	**0.37**	**8.9**	**3.1%**

31 SSyntRels

DSyntS→SSyntS	90.37 %		
	BLEU	NIST	Exact
SSyntS→MorphS	0.80	12.5	16.7%
SSyntS→Sent	0.80	11.6	8.4%
DSyntS→SSyntS→MorphS	**0.52**	**11.0**	**5.4%**
DSyntS→SSyntS→MorphS→Sent	**0.38**	**9.0**	**3.5%**

44 SSyntRels

DSyntS→SSyntS	89.54 %		
	BLEU	NIST	Exact
SSyntS→MorphS	0.81	12.5	19.4%
SSyntS→Sent	0.80	11.6	7.8%
DSyntS→SSyntS→MorphS	**0.49**	**10.85**	**4.7%**
DSyntS→SSyntS→MorphS→Sent	**0.36**	**8.9**	**3.5%**

Table 4.10: Overview of the results on the test set with the different SSyntRel granularities (31 words per sentence on average)

that the intra-hypernode dependency generation task is too sparse to be realized as a single classifier. The micro classifiers are in this case binary, i.e., 2:1, or unary, i.e., 1:1 classifiers, which implies a tremendous reduction of the search space (and thus higher accuracy). In contrast, the single classifier is a multi-class classifier that must decide among more than 60 possible classes. Although most of these 60 classes are differentiated by features, the differentiation is not perfect. In the case of lemma generation, we observe a similar phenomenon. In this case, the micro-classifiers are multi-class classifiers that normally have to cope with 5 different classes (lemmas in this case), while the unique classifier has to cope with around 60 different classes (or lemmas). Hypernode identification and inter-hypernode dependency generation are completely guided by the input; thus, it seems that they do not err in the same way.

Although the micro classifier approach leads to significantly better results, we believe that it could still be improved. First, the introduction of prepositions causes most errors in hypernode detection and lemma generation: when a preposition should be introduced or not and which preposition should be introduced depends exclusively on the sub-categorization frame of the governor of the deep node. A treebank of a limited size as used in our experiments simply does not contain subcategorization patterns of *all* predicative lexical items (especially of nouns)—which would be crucial. Thus, in the test set evaluation of one of the experiments, out of the 171 lemma errors 147 are prepositions and out of the 717 errors on hypernode identification, more than 500 are due to nouns and preposition. The increase of the size of the treebank would therefore be an advantage.

In the case of inter-hypernode dependency, errors are due to the labels of the dependencies more than to the attachments, and are quite distributed over the different types of configurations. The generation of these dependencies suffers from the fact that the SSyntS tag-set can be fine-grained: there are up to 44 SSynt dependencies in total, to compare to the 7 dependencies in the DSyntS. For instance, there are up to 9 different types of verbal objects in SSyntS, which capture very specific syntactic properties of Spanish, such as "can the dependent can be replaced by a clitic pronoun? Can the dependent be moved away from its governor?" etc. (see Section 3.3.2). Reducing the granularity of the surface-syntactic annotation has a rather positive effect on the generation of dependencies: between the 44 SSyntRel and the 15 SSyntRel tagsets, there is a 3.28 points difference for the inter-hypernode dependency generation, and a 2.24 points difference for the intra-hypernode dependency generation. Since there is no noticeable impact of the granularity of SSyntS tags on the SSyntS–MorphS and MorphS–Sentence transitions (see Table 4.10), we can conclude that the coarse-grained annotation gives sufficient information for the system to linearize and inflect the words properly.

For the results obtained with the full pipeline, we provide two different figures in Table 4.10, one considering the inflection on the words, and one without. The reason is that the MorphS–Sentence transition gives very good results on its own (around 94% accuracy), but once it is coupled with the previous modules, the accuracy drops significantly and the evaluation of the pipeline is distorted. With 15 SSyntRels, the BLEU score drops from 0.53 (DSyntS–MorphS) to 0.37 (DSyntS–Sentence), with 31 SSyntRels from 0.52 to 0.38, and with 44 SSyntRels from 0.49 to 0.36. We have not been able to find a satisfying explanation for these drops so far. Table 4.10 shows that the system is very stable across different SSyntS tagset granularities.

There is no previous system that performs the DSyntS–SSyntS, SSyntS–MorphS or MorphS–Sentence transition in Spanish, so it is not possible to contrast our results with others. In Section 4.2, we provide such a comparison (for the SSyntS–MorphS and MorphS–Sentence transitions) for other languages, namely Chinese, English and German.

4.2 Isomorphic stochastic graph transduction

For this experiment, we use a deep input (which we call *shallow-semantic* annotation) which contains all the words of the final sentences, linked by predicate-argument relations. That is, we do not aim at introducing functional nodes. The input is mapped onto a surface-syntactic structure, which is then linearized and inflected. We present a Support Vector Machine (SVM)-based stochastic generator which is, in principle, language-independent in that it is trainable on any multilevel annotated corpus. We discuss its performance for Chinese, English, German, and Spanish, some of the languages for which the CoNLL'09 shared task (Hajič et al., 2009) data is available for training.[8]

In Section 4.2.1, we discuss the completion of the shallow-semantic annotation in the CoNLL'09 shared task corpora. Section 4.2.2 presents the training setup of our realizer. Section 4.2.3 shows the individual stages of sentence realization: from the shallow-semantic structure to the surface-syntactic structure, from the surface-syntactic structure to the linearized structure and from the linearized structure to a chain of inflected word forms (if applicable for the language in question). Section 4.2.4 outlines the experimental set up for the evaluation of our realizer and discusses the results of this evaluation.

4.2.1 Input to the generator

As mentioned above, we use the multilingual CoNLL'09 data as training and testing material. The sentences of the corpora are annotated with predicate–argument information, dependency trees, and lemmas; for some of the languages involved, they also contain morphological feature annotations. The input to our generator is based on the semantic annotation which follows the PropBank annotation guidelines (Palmer et al., 2005), detailed in Chapter 2. Problematic from the viewpoint of generation is that this annotation is not always a connected acyclic graph. As a consequence, in

---

[8] This experiment has been described in (Bohnet et al., 2010).

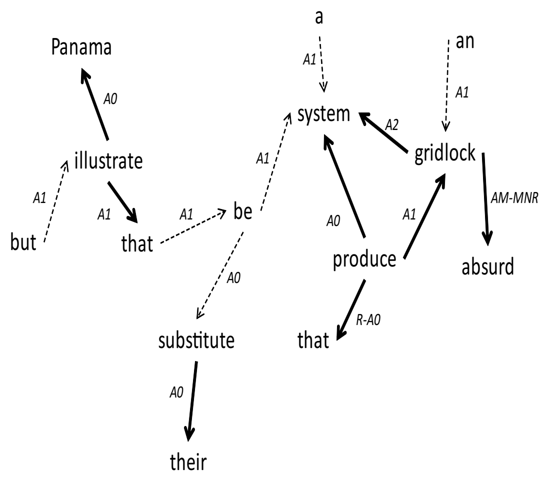


Figure 4.4: Shallow semantic representation of the sentence "*But Panama illustrates that their substitute is a system that produces an absurd gridlock.*" after completion

these cases no valid (connected) syntactic tree can be derived. The most frequent cases of violation of the connectivity principle are unattached adjectival modifiers, determiners, adverbs, and coordinations; sometimes, the verb is not connected with its argument(s). Therefore, the semantic annotation must be completed: non-connected adjectival modifiers must be annotated as predicates with their syntactic heads as arguments, determiners must be "translated" into quantifiers, detached verbal arguments must be connected with their head, etc.

Since we do not perform any other modification, we do not use the conversion described in Section 3.5. Instead, Algorithm 4 completes the semantic annotation of the corpus. Each sentence x_i of the corpus I, with $i = 1, \ldots, |I|$, is annotated with its surface-syntactic dependency tree y_i and its shallow semantic graph s_i. The algorithm traverses y_i breath-first, and examines for each node n in y_i whether n's corresponding node in s_i is connected with the node corresponding to the parent of n. If not, the algorithm connects both by a directed labeled edge. The direction and the label of the edge are selected consulting a look up table in which default labels and the orientation of the edges between different node categories are specified.

Figure 4.4 shows the shallow semantic representation of a sample English

Algorithm 4: Complete shallow semantic graph

```
//$s_i$ is a shallow semantic graph and $y_i$ a surface-syntactic dependency tree
// $s_i = \langle N_{s_i}, L_{s_i}, E_{s_i} \rangle$, where $N_{s_i}$ is the set of nodes
// $L_{s_i}$ the set of edge labels
// $E_{s_i} \subseteq N_s \times N_s \times L_s$ is the set of edges
 for $i \leftarrow 1$ to $|I|$ // iteration over the training examples
   let $r_y \in y_i$ be the root node of the dependency tree
   // initialization of the queue
   $nodeQueue \leftarrow children(r_y)$
    while $nodeQueue \neq \emptyset$ do
      $n_y \leftarrow removeFirst$(nodeQueue)
      // breath first: add nodes at the end of the queue
      nodeQueue $\leftarrow$ nodeQueue $\cup$ $children(n_y)$
      $n_{y_s} \leftarrow sem(n_y)$; $p_{y_s} \leftarrow sem(parent(n_y))$
      //get the shallow semantic equivalents of $n_y$ and of its parent
       if not exists path($n_{y_s}$, $p_{y_s}$) then
         $l \leftarrow label(n_y, parent(n_y))$
         $l_s \leftarrow look\text{-}up\text{-}sem\text{-}label(n_{y_s}, p_{y_s}, l)$
          if $look\text{-}up\text{-}sem\text{-}direction(n_{y_s}, p_{y_s}, l_s)$ = "→" then
            // add the shallow semantic edge
            $E_s \leftarrow E_s \cup (p_{y_s}, n_{y_s}, l_s)$
          else // direction of the edge "←"
            // add the shallow semantic edge
            $E_s \leftarrow E_s \cup (n_{y_s}, p_{y_s}, l_s)$
```

sentence obtained after the application of Algorithm 4. The solid edges are the edges available in the original annotation; the dashed edges have been introduced by the algorithm.[9] As can be seen, 6 out of the total of 14 edges in the complete representation of this example have been added by Algorithm 4.

4.2.2 Realizer training setup

Figure 4.5 shows the architecture of the realizer. For each level of annotation, an SVM feature extractor and for each pair of adjacent levels of annotation, an SVM decoder is defined. The ShallowSemS–SSyntS decoder constructs from a shallow semantic graph the corresponding dependency tree. The SSyntS–MorphS decoder derives from a dependency tree a chain of lemmas, i.e., determines the word order within the sentence. The

---

[9] See Section 2.2.2 for a description of edge label nomenclature.

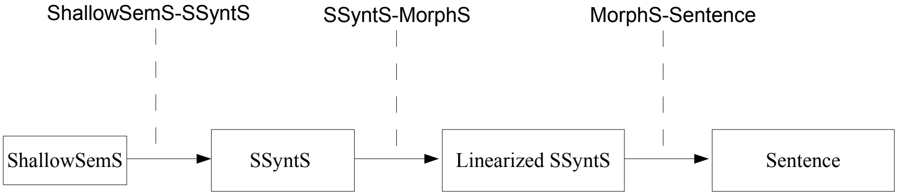


Figure 4.5: Architecture of the isomorphic realizer

MorphS–Sentence decoder generates the inflected word form for each lemma in the chain. Both the feature extractors and the decoders are language-independent, which makes the realizer applicable to any language for which multilevel-annotated corpora are available. To compute the score of the alternative realizations by each decoder, we apply MIRA, as in Section 4.1.3. The last two transitions are the same as described in Section 4.1.3.

4.2.3 Sentence generation

Sentence generation that starts from a given semantic structure as input consists in the application of the previously trained SVM decoders in sequence in order to realize the sequence of mappings shown in Figure 4.5.

4.2.3.1 Shallow semantic generation

Algorithm 5 shows the algorithm for semantic generation, i.e., the derivation of a surface-syntactic dependency tree from a shallow semantic structure. It is a beam search that creates a maximum spanning tree. In the first step, a seed tree consisting of one edge is built. In each of the subsequent steps, this tree is extended by one node. For the decision, which node is to be attached next and to which node, we consider the highest scoring options. This procedure works well since nodes that are close in the semantic structure are usually close in the syntactic tree as well. Therefore subtrees that contain those nodes are considered first. Unlike the traditional n-gram based stochastic realizers such as (Langkilde and Knight, 1998), we use for the score calculation structured features composed of the following elements: (i) the lemmas, (ii) the **dist**ance between the starting node s and the target node t, (iii) the **dir**ection of the path (if the path has a direction), (iv) the sorted **bag** of in-going edges labels without repetition, (v) the **path**

Algorithm 5: Shallow semantic generation

```
  //$s_i$, $y$ shallow semantic graph and its dependency tree
   for i ← 1  to $|I|$ // iteration over the training examples
     // build an initial tree
      for all $n_1 \in s_i$  do
     trees ← {} // initialize the constructed trees list
        for all $n_2 \in s_i$  do
          if $n_1 \neq n_2$  then
            for all $l \in$ dependency-labels  do
              trees = trees ∪ {($synt(n_1)$,$synt(n_2)$,$l$)}
     trees ← sort-trees-descending-to-score(trees)
     trees ← look-forward(1000,sublist(trees,20))
     //assess at most 1000 edges of the 20 best trees
     tree ← get-best-tree-due-to-score(trees)
     (s,t,l) ← first-added-edge(tree)
     // create the best tree
     best-tree ← (s,t,l)
     // compute the nodes that still need to be attached
     rest ← nodes($s_i$) - $\{s, t\}$
      while rest $\neq \emptyset$  do
       trees ← look-forward(1000,best-tree,rest)
       tree ← get-best-tree-due-to-score(trees)
       (s,t,l) ← first-added-edge(tree)
       best-tree ← best-tree ∪ {(s,t,l) }
        if (root(s,best-tree))  then rest ← rest - $\{s\}$
        else rest ← rest - $\{t\}$
```

of edge labels between source and target node. The composed structured features are shown in Table 4.11.

#	**features**
1	label+dist(s, t)+dir
2	label+dist(s, t)+lemma$_s$+dir
3	label+dist(s, t)+lemma$_t$+dir
4	label+dist(s, t)+lemma$_s$+lemma$_t$+dir
5	label+dist(s, t)+bag$_s$+dir
6	label+dist(s, t)+bag$_t$+dir
7	label+path(s, t)+dir

Table 4.11: Features for ShallowSemS–SSyntS mapping

4.2.3.2 Linearization and morphologization

Both transitions, i.e., SSyntS–MorphS and MorphS-Sentence, are described in Section 4.1.3. Note that for this system, to order the dependency tree, we use a one classifier-approach for all languages—in contrast to, e.g., (Filippova and Strube, 2009), who use a two-classifier approach for German.[10]

4.2.4 Experiments

To evaluate the performance of our realizer, we carried out experiments on deep generation of Chinese, English, German and Spanish. The size of the test sets is displayed in Table 4.12.[11]

Chinese	English	German	Spanish
2556	2400	2000	1725

Table 4.12: The number of sentences in the test sets used in the experiments

The performance of both the isolated stages and the realizer as a whole has been assessed.

4.2.4.1 Evaluation Metrics

In order to measure the correctness of the ShallowSemS–SSyntS mapping, we use the unlabeled and labeled attachment scores as commonly used in dependency parsing. The labeled attachment score (LAS) is the proportion of tokens that are assigned both the correct head and the correct edge label. The unlabeled attachment score (ULA) is the proportion of correct tokens that are assigned the correct head. To assess the quality of linearization, we use the per-phrase/per-clause accuracy (*acc snt.*):

$$acc = \frac{correct\ constituents}{all\ constituents}$$

As second evaluation metric, we use a metric related to the edit distance:

$$di = 1 - \frac{m}{total\ number\ of\ words}$$

---

[10] We decided to test at this stage of our work a uniform technology for all languages, even if the idiosyncrasies of some languages may be handled better by specific solutions.

[11] As in (Langkilde-Geary, 2002) and (Ringger et al., 2004), we used Section 23 of the WSJ corpus as test set for English.

(with m as the minimum number of deletions combined with insertions to obtain the correct order (Ringger et al., 2004)).

For the assessment of the quality of the word form generation, we use the accuracy score. The accuracy is the ratio between correctly generated word forms and the entire set of generated word forms. As in Section 4.1.4.1, we also provide the BLEU score for linearization and the whole pipeline.

4.2.4.2 Experimental Results

Table 4.13 displays the results obtained for the isolated stages of sentence realization and of the realization as a whole, with reference to a baseline and to some state-of-the-art works. The baseline is the deep sentence realization over all stages starting from the original semantic annotation in the CoNLL'09 shared task corpora.

Note, that our results are not fully comparable with (He et al., 2009), (Filippova and Strube, 2009) and (Ringger et al., 2004), respectively, since the data are different. Furthermore, Filippova and Strube (2009) linearize only English sentences that do not contain phrases that exceed 20,000 linearization options—which means that they filter out about 1% of the phrases.

For Spanish, to the best of our knowledge, no other linearization experiments have been carried out so far apart from the ones in this thesis, therefore, we cannot contrast our results with any reference work. If the results of linearization are significantly better than the results obtained in Section 4.1, it is due to the size of the corpus, which is 5 times bigger for this experiment.

As far as the MorphS–Sentence mapping is concerned, the performance achieved by our realizer for English is somewhat lower than in (Minnen et al., 2001) (97.8% vs. 99.8% of accuracy). Note, however, that Minnen et al. describe a combined analyzer-generator, in which the generator is directly derived from the analyzer, which makes both approaches not directly comparable. Sample outputs of this system are provided in Appendix B.

4.2.4.3 Discussion

The overall performance of our SVM-based deep sentence generator ranges between 0.611 (for German) and 0.688 (for Chinese) of the BLEU score. HALogen's (Langkilde-Geary, 2002) scores range between 0.514 and 0.924, depending on the completeness of the input. The figures are not directly

	Chinese	English	German	Spanish
ShallowSemS–SSyntS (ULA)	95.71	94.77	95.46	98.39
ShallowSemS–SSyntS (LAS)	86.29	89.76	82.99	93.00
SSyntS–MorphS (di)	0.88	0.91	0.82	0.83
SSyntS–MorphS (acc)	64.74	74.96	50.5	52.77
SSyntS–MorphS (BLEU)	0.85	0.894	0.735	0.78
MorphS–Sentence	–	97.8	97.49	98.48
(accuracy=correct words/all words)				
All stages (BLEU)	0.688	0.659	0.611	0.68
Baseline (BLEU)	0.12	0.18	0.11	0.14
(He et al., 2009)				
SSyntS–MorphS (di)	0.89	–	–	–
SSyntS–MorphS (acc)	–	–	–	–
SSyntS–MorphS (BLEU)	0.887	–	–	–
(Filippova and Strube, 2009)				
SSyntS–MorphS (di)	–	0.88	0.87	–
SSyntS–MorphS (acc)	–	67	61	–
(Ringger et al., 2004)				
SSyntS–MorphS (BLEU)	–	0.836	–	–

Table 4.13: Quality figures for the isolated stages of deep sentence realization and the complete process

comparable since HALogen takes as input surface-syntactic structures. However, it gives us an idea where this generator is situated.

Traditional linearization approaches are rule-based; cf., e.g., (Bröker, 1998; Gerdes and Kahane, 2001; Duchier and Debusmann, 2001; Bohnet, 2004). More recently, statistic language models have been used to derive word order, cf. (Ringger et al., 2004; Wan et al., 2009; Filippova and Strube, 2009). Because of its partially free order, which is more difficult to handle than fixed word order, German has often been worked with in the context of linearization. Filippova and Strube (2009) adapted their linearization model originally developed for German to English. They use two classifiers to determine the word order in a sentence. The first classifier uses a trigram language model to order words within constituents, and the second (which is a maximum entropy classifier) determines the order of constituents that depend on a finite verb. For English, we achieve with our SVM-based classifier a better performance. As mentioned above, for German, Filippova and Strube (2009)'s two classifier approach pays off because it allows them to handle non-projective structures for the *Vorfeld* within the field model.

It is certainly appropriate to optimize the performance of the realizer for the languages covered in a specific application. However, our goal has been so far different: to offer an off-the-shelf language-independent solution.

The linearization error analysis, first of all of German and Spanish, reveals that the annotation of coordinations in corpora of these languages as $X \leftarrow$ *and/or/...* $\rightarrow Y$ is a source of errors. The "linear" annotation used in the PropBank ($X \rightarrow$ *and/or/...* $\rightarrow Y$) appears to facilitate higher quality linearization. A pre-processing stage for automatic conversion of the annotation of coordinations in the corpora would have certainly contributed to a higher quality. We refrained from doing this because we did not want to distort the figures.

The morphologization error analysis indicates a number of error sources that we will address in the process of the improvement of the model. Among those sources are: quotes at the beginning of a sentence, acronyms, specific cases of starting capital letters of proper nouns (for English and Spanish), etc.

As far as the contrastive evaluation of the quality of our morphologization stage is concerned, it is hampered by the fact that for the traditional manually crafted morphological generators, it is difficult to find thorough quantitative evaluations, and stochastic morphological generators are rare.

As already pointed out above, so far we intentionally refrained from optimizing the individual realization stages for specific languages. Therefore, there is still quite a lot of room for improvement of this realizer when one concentrates on a selected set of languages.

4.3 Hybrid stochastic graph transduction

This generator is called hybrid because it uses a combination of classifiers and rules in order to perform the successive mappings. The rules are derived automatically from annotated data, and allow for introducing nodes during the DSyntS-SSyntS transition. Similar to the second experiment, we start from a CoNLL 2009 shared task corpus. However, unlike in Section 4.2, we extend the CoNLL 2009 annotation in two respects: (i) we map the original CoNLL 2009 annotation onto a more abstract semantic annotation, and (ii) we introduce a deep-syntactic annotation (as has already been used by (Walker et al., 2002), (Stent et al., 2004) and in Section 4.1), which provides intermediate linguistic structures which do not contain any superficial functional nodes, but rather only the grammatical function structures.[12]

In the next section, we introduce the two new levels of annotation of the CoNLL'09 corpus: the semantic and deep-syntactic annotations, and explain how we obtain them. In Section 4.3.2, we present the setup of the realizer. Section 4.3.3 outlines the individual stages of sentence realization: SemS → DSyntS → SSyntS → MorphS → Sentence. Section 4.3.4 describes the setup of the experiments for the evaluation of the realizer and discusses the results of the evaluation.

4.3.1 Adjusting the annotation

In order to get close to our ideal picture of NLG, we not only ensure that the starting semantic structure, i.e., the PropBank annotation, is a connected graph, but, furthermore, we make it truly semantic. Furthermore, we use the DSyntS as an intermediate structure. DSyntS links to the semantic structure (SemS) in that it does not contain any function words, and, at the same time, to the CoNLL syntactic structure (SSyntS) in that it contains the grammatical functions of the content words. DSyntS thus facilitates a two-step semantics-syntax projection, allowing for higher quality generation.

4.3.1.1 Deriving the semantic annotation

For turning the PropBank/NomBank-annotation as illustrated in Figure 4.6 into a genuine semantic input annotation that can serve as departure for stochastic sentence generation, we use the conversion detailed in Section 3.5.[13] In summary, it comprises mainly four steps:

1 : exclude the functional nodes from the annotation;

2 : substitute syntactically motivated arcs by semantic arcs;

3 : introduce missing semantic nodes;

4 : introduce minimal communicative structure (in particular, givenness and *theme/rheme* and *foregrounded/backgrounded* dimensions);

---

[12] This experiment has been described in (Bohnet et al., 2011b), (Bohnet et al., 2011a), and (Bohnet et al., 2014).

[13] Except for Section 3.5.3.2, since some information was missing in the corpus at the time in order to remove relative pronouns safely.

5 : ensure connectivity of the semantic annotation.

Figure 4.7 shows a sample SemS, as obtained from the original structure in Figure 4.6.

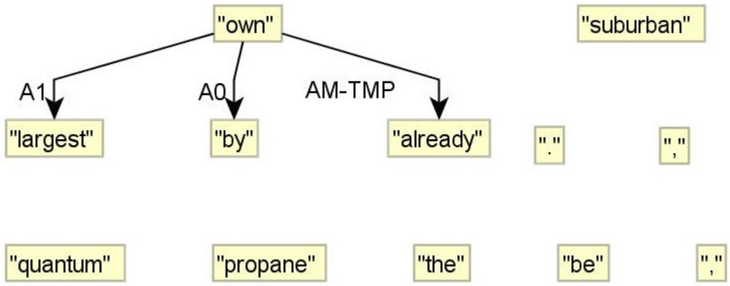


Figure 4.6: PropBank/NomBank annotation of the sentence "*The largest, Suburban Propane, was already owned by Quantum.*"

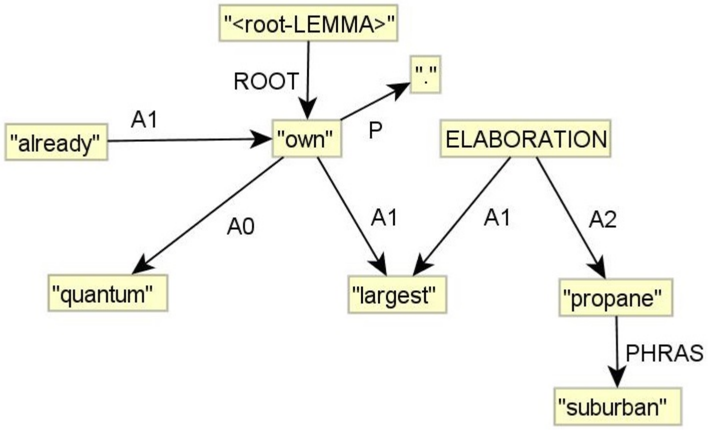


Figure 4.7: Semantic annotation of the sentence "*The largest, Suburban Propane, was already owned by Quantum.*"

4.3.1.2 Deriving the deep-syntactic annotation

As just pointed out, DSyntS is meant to facilitate the mapping between the abstract semantic structure obtained as described above and the CoNLL syntactic structure. It contains only content nodes, i.e., nodes of the semantic structure, and, at the same time, syntactic relations since the deep syntactic structure shows explicitly the structure of the sentence. That is, the governors and dependents are not organized based on predicate/argument relations, but rather on the notion of syntactic governor. The syntactic governor of a lexeme is the one that imposes syntactic constraints on its dependents: linearization and agreement constraints, case or governed prepositions assignments, etc. Hence, like the syntactic structure, the deep-syntactic structure representation is a tree, not a graph. Every node at this level contains Part-of-Speech tags. Figure 4.8 shows a sample DSyntS.

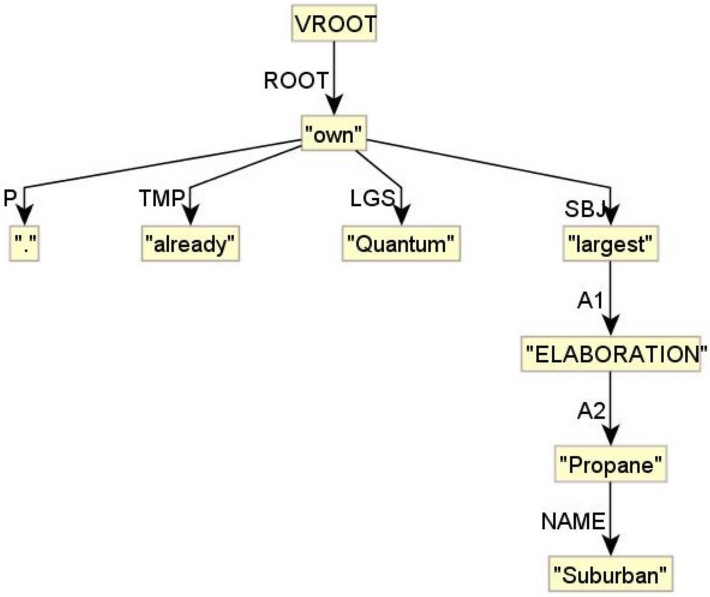


Figure 4.8: Deep-syntactic annotation of the sentence "*The largest, Suburban Propane, was already owned by Quantum.*"

There are differences between this DSyntS and the DSyntS which has been introduced in Chapter 3; this is due to the architecture of the generator, which is designed to tackle one task at a time. The first difference is that ALL the nodes of the semantic structures are in the DSyntS, which also include meta-nodes. Second of all, the labels connecting nodes which are not meta-nodes are superficial. This way, the first task of the generator is simply to redirect and relabel the edges when necessary; this task is an isomorphic mapping and can be handled through classifiers only. The second task aims at introducing the missing nodes and edges, and for this we use rules in this experiment. The result of these two steps gives a surface-syntactic structure. In the following sections, we detail each particular mapping.

4.3.2 Setup of the realizer

To generate a sentence for a given semantic input graph, our sentence realizer performs the mappings shown in Figure 4.9.

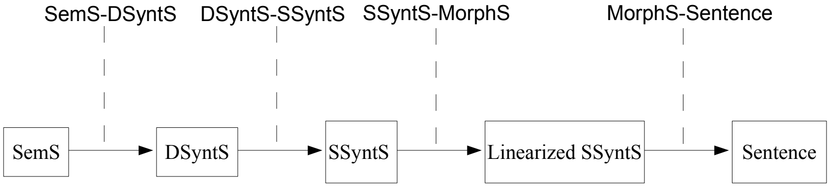


Figure 4.9: Architecture of the isomorphic realizer

Each of the steps is carried out by a decoder that uses a classifier to select the appropriate operations. As in our the experiment described in Section 4.1, we use MIRA (Crammer et al., 2006) for the realization of all classifiers. The goal is to obtain a function that separates correct realizations (or items) by a decoder from the incorrect realizations. The items are characterized by features provided by feature extractors. The features are used to obtain a weight vector that separates the correct and incorrect items. The features are represented as a vector $\phi(x_i)$, which can be multiplied with the weight vector w in order to obtain a score. The weight vector w can be obtained by an online learning algorithm, which considers a training example in each iteration of the training procedure. This has the advantage that we can process one example at a time, keeping only this example in the memory.

Algorithm 6 shows the outline of the training algorithm. The algorithm iterates I times over all training examples $\tau(x_i, y_i)_{i=1}^{n}$. A passive-aggressive weight vector update strategy updates at the beginning of the training procedure the weights more aggressively. To what extent is determined by the factor β. The weight vector v accumulates all weights, which are *averaged* at the end of the algorithm to avoid overfitting (Collins, 2002).

Algorithm 6: Online learning algorithm

```
Input: $\tau = \{(x_i, y_i)\}_{i=1}^{n}$
$w^{(0)} = 0; v = 0; i = 0;$
$\beta = I * N$
 for n = 1  to I // Training iterations
    for n = 1  to N // Training instances
      $w^{(i+1)}$ = update $w^{(i)}$ according to $(x_i, y_i)$
      v = v + $\beta$ $w^{i+1}$
      i = i + 1
      $\beta$ = $\beta$ - 1
$w = v/(I * N)$
```

4.3.3 Sentence generation

Sentence generation consists in the application of the previously trained decoders in the sequence outlined in the previous section.

4.3.3.1 Semantic generation

Our approach to semantic generation in this experiment, which consists of the derivation of the deep-syntactic tree from an input semantic graph, is analogous to graph-based parsing (Eisner, 1996; McDonald and Pereira, 2006). The derivation is defined as search for the highest scoring tree y from all possible trees given an input graph x:

$$F(x) = argmax\ Score(y), where\ y \in MAP(x)$$

(with $MAP(x)$ as the set of all trees spanning over the nodes of the semantic graph x).

The search is, again, a beam search which creates a maximum spanning tree.[14] Unlike in Section 4.2, however, we use “early update” as introduced for parsing by (Collins and Roark, 2004): when the correct beam element drops out of the beam, we stop and update the model using the best partial solution. The idea behind this is that when all items in the current beam are incorrect, further processing is obsolete since the correct solution cannot be reached extending any elements of the beam. When we reach a final state,

---

[14] The maximum spanning tree algorithm can be applied here thanks to the introduction of the isomorphic deep-syntactic structure.

Algorithm 7: Semantic generation

```
//$(x_i, y_i)$ semantic graph and the deep-syntactic tree
//beam-size $\leftarrow$ 80
// build an initial tree
for all $n_1 \in x_i$ do
  trees $\leftarrow$ {} // empty list of partial trees
  for all $n_2 \in x_i$ do
    if $n_1 \neq n_2$ then
      for all $l \in$ edge-labels do
        trees = trees $\cup$ {(synt($n_1$),synt($n_2$),$l$)}
trees $\leftarrow$ sort-trees-descending-to-score(trees)
trees $\leftarrow$ subset(0,beam-size,trees)
// extend the initial trees consisting of one edge
while rest $\neq \emptyset$ do
  trees $\leftarrow$ extend-trees(trees)
  trees $\leftarrow$ sort-trees-descending-to-score(trees)
  trees $\leftarrow$ subset(0,beam-size,trees)
  // training: if gold tree is not in the beam
  // then update weight vector and continue with next
return first element of trees
```

i.e., a tree spanning over all words and the correct solution is in the beam, but not ranked first, we perform an update as well since the correct element should have ranked first in the beam.

Algorithm 7 displays the algorithm for the generation of the deep-syntactic structure from the semantic structure. *extend-trees* is the central function of the algorithm. It expands a tree or a set of trees by one edge, selecting each time the highest scoring edge. Attachment point for an outgoing edge is any node; for an incoming edge only the top node of the built tree.

For score calculation, we use structured features composed of the following elements: (i) the lemmas, (ii) the **dist**ance between the starting node s and the target node t, (iii) the **dir**ection of the path (if the path has a direction), (iv) the sorted **bag** of in-going edges labels without repetition, (v) the **path** of edge labels between source and target node. The templates of the composed structured features are listed in Table 4.14. We obtain about 2.6 Million features in total. The features have binary values, meaning that a structure has a specific feature or it does not.

#	**features**
1	label+dist(s, t)+dir
2	label+dist(s, t)+lemma$_s$+dir
3	label+dist(s, t)+lemma$_t$+dir
4	label+dist(s, t)+lemma$_s$+lemma$_t$+dir
5	label+dist(s, t)+bag$_s$+dir
6	label+dist(s, t)+bag$_t$+dir
7	label+path(s, t)+dir

‘s’ means “source node” of an edge
‘t’ “target node” of an edge

Table 4.14: Feature templates for the SemS–DSyntS mapping

4.3.3.2 Deep-syntactic generation

Since the DSyntS contains by definition only content words, function words such as governed prepositions, auxiliaries, and determiners must be introduced during the DSyntS–SSyntS generation passage in order to obtain a fully spelled out syntactic tree. In this experiment, unlike in Section 4.1, we address this transition with rules instead of classifiers.

Tree transducers are suited for this task because of their capability to rewrite trees. Top down tree transducers have been independently introduced by Rounds (1970) and Thatcher (1970) as extensions of finite state transducers. Tree transducers have been already successfully applied in NLP—for instance, in machine translation (Knight and Graehl, 2005). Tree transducers traverse the input trees from the root to the leaves and rewrite the tree using rewriting rules. For DSyntS–SSyntS generation, we use around 280 rules derived automatically by comparing a gold standard set of deep-syntactic structures and surface-syntactic dependency trees. The rules are of the following three types:

1. Rules introducing an edge and a node:
   X $\Rightarrow$ X $label_s \rightarrow$ Y ,
   Example: X $\Rightarrow$ X *NMOD*$\rightarrow$ *‘the’*

2. Rules introducing a new node and edges between two nodes:
   X $label_d\rightarrow$ Y $\Rightarrow$ X $label^1_s \rightarrow$ N $label^2_s \rightarrow$ Y
   Example: X *OPRD*$\rightarrow$ Y $\Rightarrow$ X *OPRD*$\rightarrow$ ’to’ *IM*$\rightarrow$ Y

Algorithm 8: Deep-syntactic generation

```
//$(x_i, y_i^g)$ the deep syntactic tree
// and gold surface syntactic tree for training case only
// $R$ set of rules
// travers the tree top down depth first
$y_i$ ←clone($x_i$)
node-queue ← root($x_i$)
while node-queue $\neq \emptyset$ do
  //depth first traversal
  node ← remove-first-element(node-queue)
  node-queue ← children(node, $x_i$)∪ node-queue
  // select the rules, which insert a leaf node
  leaf-insert-rules ← select-leaf-rules(next-node,$x_i$,$R$)
  $y_i$ ← apply (leaf-insert-rules,$y_i$)
  // in the training, we update here the weight vector
  // if the rules are not equal to the gold rules
  //
  // select the rules, which insert a node in the tree
  // or a new node label
  node-insert-rules ← select-node-rules(node,$x_i$,$R$)
  // in the training, we update here the weight vector
  $y_i$ ← apply (edge-insert-rules,$y_i$)
```

3. Rules introducing a new node label:
   X ⇒ N
   Example: 'LOCATION' ⇒ 'on'

The restricted number of rules and rule types suggests the use of classifiers to select applicable rules in each stage of the DSyntS–SSyntS generation and thus consider more contextual information for the decision.

We train discriminative classifiers for each of three rule types that either select a specific rule or NONE (i.e., no rule is to be applied). Some parts do not need any changes. Therefore, on these parts there is no need to apply rules and the classifier has to select NONE. Algorithm 8 displays the algorithm for the generation of the surface-syntactic structure from the deep-syntactic structure. The algorithm uses the features listed in Table 4.15 for score calculation.

Table 4.16 shows the confusion matrix of the DSyntS → SSyntS transducer

#	**features**
1	pos(node)
2	pos(head(node))
3	pos(head(head(node)))
4	pos(node)+pos(head((node))
5	pos(node) + pos(head(node))+ edge-label(node)
6	feature-1(node)
7	feature-2(node)
8	feature-3(node)
9	feature-1(node)+feature-2(node)
10	lemma(node)
11	lemma(head(node))
12	lemma(node)+lemma(head(node))
13	bag-of-children-pos(node)
14	sorted-bag-of-children-pos(node)
15	sorted-bag-of-children-labels(node)

pos are coarse-grained Part-of-Speech tags
feature are the features attached to the nodes
lemma are node labels
edge-label labels of edges
feature-1 stands for “definite=yes”
feature-2 stands for “num=sg”
feature-3 stands for “tense=past”

Table 4.15: Feature templates for the DSyntS–SSyntS mapping

rules. The first column contains the number of the gold rule that should have been applied; the second the gold rule itself and the third the actually applied rule. *ie:* is the prefix of “insert-edge” rules, and *in:* the prefix of “insert-node” rules.

As we see, confusions occur, first of all, in the selection of the correct preposition in <nominal modifier>–<prepositional modifier> sequences in edge inserting rules. A possible solution to this problem that needs to be further explored is the inclusion of a larger context or/and consideration of semantic features. Note that with the classifiers on Section 4.1, confusions occurred in similar cases.

# rule	gold rule	wrongly applied rule
65	ie:NMOD:for:PMOD	ie:NMOD:of:PMOD
40	ie:LOC:in:PMOD	ie:NMOD:of:PMOD
34	ie:NMOD:to:PMOD	ie:NMOD:of:PMOD
23	ie:NMOD:on:PMOD	ie:NMOD:of:PMOD
26	ie:NMOD:with:PMOD	ie:NMOD:of:PMOD
18	ie:NMOD:from:PMOD	ie:NMOD:of:PMOD
16	ie:DIR:to:PMOD	ie:ADV:to:PMOD
12	ie:DIR:from:PMOD	ie:DIR:to:PMOD
11	in:NMOD:to	
11	ie:NMOD:of:PMOD	
10	ie:NMOD:of:PMOD	ie:LOC:in:PMOD
9	ie:ADV:at:PMOD	ie:ADV:for:PMOD
9	ie:DIR:from:PMOD	ie:ADV:from:PMOD
6	ie:PMOD:to:PMOD	
8	ie:OBJ:that:SUB	
8	ie:OPRD:to:IM	
8	ie:LOC:at:PMOD	ie:NMOD:with:PMOD

Table 4.16: Confusion matrix of the DSyntS → SSyntS rules

4.3.3.3 Linearization and morphologization

In this version of the realizer, we use the same implementation as in Section 4.1. The linearization is a beam search for an optimal linearization according to a local and global score functions. The morphological realization algorithm selects the edit script based on the minimal string edit distance (Levenshtein, 1966) in accordance with the highest score for each lemma of a sentence obtained during training and applies then the scripts to obtain the word forms.

4.3.4 Experiments

To evaluate the proposed realizer, we carried out a number of experiments, whose setup and results are presented in what follows.

4.3.4.1 Setup of the experiments

For this series of experiments, we use the usual training, development and test data split of the WSJ corpus (Langkilde-Geary, 2002; Ringger et al., 2004), the CoNLL'09 PTB/NB/PB corpus. Table 4.17 provides an overview of the used data.

set	section	# sentences
training	2 - 21	39218
development	24	1334
test	23	2400

Table 4.17: Data split of the used data in the WSJ Corpus

In order to measure the accuracy of the isolated components and of the realizer as a whole and to be able to compare their performance with previous works, we use measures already used in Sections 4.1 and 4.2. Thus, for the SemS–DSyntS mapping, we use the unlabeled and labeled attachment scores, as it is also commonly used in dependency parsing. For the assessment of the DSyntS–SSyntS mapping, we use the F-score of correctly/wrongly introduced nodes. For the evaluation of the sentence realizer as a whole, we use the BLEU metric.

To assess linearization and morphological realization, we also use the same metrics as in our first experiments (see Section 4.1.4.1).

4.3.4.2 Results of the experiments

Table 4.18 displays the figures obtained for both the isolated stages of the semantic sentence realization and the generation as a whole—with reference to some of the recent works on statistical generation.[15] We include the performance of the experiment of Section 4.2 in two stages that differ from this experiment: ShallowSemS→SSyntS, and SSyntS→MorphS, and its overall performance. We include (Filippova and Strube, 2009) and (Ringger et al., 2004) because these are reference works with which any new work on statistical generation has to compete (even though they are not fully comparable with our system, as mentioned in Section 4.2).

Sample outputs of this system are provided in Appendix B.

4.3.4.3 Discussion

The overall performance of this deep generator is comparable (although somewhat lower) to the performance of the one presented in Section 4.2.

---

[15] We do not compare here to (Wong and Mooney, 2007) and (Mairesse et al., 2010) because the the tasks of both are rather different from ours: both explore phrase-based generation.

Mapping	**Value**
SemS–DSyntS (ULA/LAS)	93.8/87.3
DSyntS–SSyntS (correct)	97.5
SSyntS–MorphS (BLEU)	0.89
MorphS–Sentence (accuracy)	97.8
All stages (BLEU)	0.64
All stages (BLEU) (as in Section 4.2)	0.659
ShallowSemS–SSyntS (ULA/LAS) (as in Section 4.2)	94.77/89.76
SSyntS–MorphS (di/acc)	
(as in Section 4.2)	0.91/74.96
(Filippova and Strube, 2009)	0.88/67
(Ringger et al., 2004) (BLEU)	0.836

Table 4.18: Performance of the individual stages of semantic sentence realization and of the realization as a whole

This is remarkable given that we start from a considerably more abstract semantic structure that does not contain any function words and that encodes some of the information (for instance, communicative structure features) in terms of node attributes instead of nodes/arcs. The performance of the SemS–DSyntS projection is slightly lower than our previous ShallowSemS–SSyntS projection. However, the quality of our present DSyntS–SSyntS projection is rather high—despite the fact that during this projection new nodes are introduced into the target structure (i.e., the projection is not isomorphic). A more detailed analysis of this projection shows that the precision of correctly introduced nodes is 0.79 and the recall is 0.74. As a result, we obtain an F-score of 0.765. The introduction of nodes affects only a relatively small part of the surface-syntactic structure. Before we apply the rules, the (gold) deep-syntactic tree has about 92% correct nodes and correctly attached edges of the (surface) syntactic tree. After the rule application this value improves to about 97.6%.

The performance during the SSyntS–MorphS mapping is slightly lower than in our first experiment. This is the effect of the (imperfect) introduction of function words (such as determiners and prepositions) into the surface-syntactic structure at the preceding stage. But it is still higher than the performance of the reference realizers such as (Ringger et al., 2004) and (Filippova and Strube, 2009) for this task.

4.3.5 Using different training data: the SRST and our Spanish corpus

With the implementation described in this section, we had submitted two systems to the deep track of the first SRST in 2011 (Belz et al., 2011). Since the input to the shared task is already a tree, the step corresponding to the semantic generation is not necessary. More precisely, the edges do not have to be redirected, but only relabeled. What is called ShallowSemS–DSyntS mapping for this experiment is simply this relabeling from predicate-argument relations to syntax oriented labels.

System 1	
Mapping	**Value**
ShallowSemS–DSyntS (ULA/LAS)	99.0/95.1
DSyntS–SSyntS (correct)	98.6
Tree-based PoS tagging	97.8
SSyntS–MorphS (% sent. eq. to reference)	54.2
MorphS–Sentence (accuracy)	98.2
All stages from deep representation	
BLEU	0.76
NIST	13.45
All stages from shallow representation	
BLEU	0.89
NIST	13.89
System 2	
Mapping	**Value**
ShallowSemS–DSyntS (ULA/LAS)	99.0/95.1
DSyntS–SSyntS (correct)	98.9
Tree-based PoS tagging	98.2
SSyntS–MorphS (% sent. eq. to reference)	57.7
MorphS–Sentence (accuracy)	98.2
All stages from deep representation	
BLEU	0.80
NIST	13.55
All stages from shallow representation	
BLEU	0.90
NIST	13.93

Table 4.19: Performance of our realizer on the development set

The differences between the first and the second system is that the latter is able to introduce more precise commas because of an improved feature set. In addition, it uses the word order of children as context to derive features for the linearization and it uses a language model to rerank output sentences. For the language model, we used a 5-gram model with Kneser-Ney smoothing derived from 11 million sentences, cf. (Kneser and Ney, 1995). Table 4.19 displays the figures obtained for both the realization stages in isolation and the entire pipeline.

This system was the only one to make use of an intermediate layer between the deep input and the surface-syntactic representation at the SRST'11, and got the best results for the task.

We then trained the same generator on the Spanish multilayered corpus presented in Chapter 3, taking the deep-syntactic layer as input, since it is the most similar to the SRST's deep representation. The results on the Spanish corpus are shown without morphology for the reasons detailed in Section 4.1 (see Table 4.20); the DSyntS–MorphS mapping is much worse than with English, which is mostly due to the fact that many more nodes are missing from the Spanish deep input. As a consequence, the simple rule system which introduces nodes in the surface-syntactic structure and works well for English does not give satisfying results. For the same reason, compared to the results obtained in Section 4.1, the BLEU score drops about 20 points.

System 1	
Mapping	**Value**
DSyntS–MorphS	
BLEU	0.30
NIST	7.5
Exact	1.5

Table 4.20: Overview of the results on the Spanish test set excluding punctuation marks after the linearization

4.4 Summary and conclusions

In Section 4.1, two alternative classifier approaches to deep generation have been presented that cope with the projection of non-isomorphic semantic and syntactic structures. We argued that the micro classifier approach is more adequate. Each set of micro classifiers achieves results above 86% on the Spanish test set. For intra-hypernode dependency generation, it even reaches 95.94%, which is very satisfying given the number of functional nodes which have to be introduced. Our experiments on varying the granularity of the surface-syntactic dependency tagset revealed a very limited impact on the accuracy of the whole system. Our generator achieves very stable performances with more or less fine-grained surface-syntactic relations, which shows that it will work for a large variety of syntactic annotations.

In Section 4.2, we presented an SVM-based stochastic deep multilingual sentence generator that is inspired by the state-of-the-art research in semantic parsing. It uses similar techniques and relies on the same resources. This intent shows that there is a potential for stochastic sentence realization to catch up with the level of progress recently achieved in parsing technologies. However, in these experiments, the result of the pre-processing stage on the input structures is still not a genuine semantic structure: it contains all nodes of a (surface-)syntactic structure (auxiliaries, governed prepositions, determiners, etc.), including the nodes of functional words, and the part of speech tags of the individual nodes. Furthermore, it maintains the syntactic traces of the PropBank annotation such as the orientation of modifier relations and annotation of relative constructions. Hence, the mappings between two consecutive intermediate structures are (i) all isomorphic, and (ii) not realistic from the perspective of deep NLG. As a consequence, we undertook another experiment in order to overcome these shortcomings.

In Section 4.3, we presented a decoder-based statistical semantic sentence realizer which goes significantly beyond the previous works in this area, while showing a similar or, in some aspects, even better performance. An important extension compared to what is presented in Section 4.2 is the mapping from the semantic graph to a deep syntactic structure that forms an intermediate structure between the semantic structure and the surface-syntactic structure. One other important improvement is that the input to the system is more semantic, in the sense that the deep representation does not contain syntactically motivated edges or nodes. The introduction of these functional nodes during the DSyntS-SSyntS mapping is performed thanks to rules that are obtained automatically from DSyntS-SSyntS parallel corpora. This strategy works well if the variety and quantity of nodes to introduce is not great, but as soon as it stops being the case, the system has a hard time producing the new nodes correctly. In spite of this, the system obtained excellent results at the Surface-Realization Shared Task, getting the best scores among all presented systems.

In Table 4.21, we briefly summarize the characteristics of the three different generators described in this chapter, following the model of Section 2.1.5.

		Non-isomorphic	Isomorphic	Hybrid
Corpus -based...	syntacticization	-/+	-/+	+
	lexicalization	-/+	-	-/+
	linearization	+	+	+
	morphologization	+	+	+
	ranking	-	-	+
Type of annotation	logical	+	-/+	+
	syntactic	+	+	+
	sentence	+	+	+
Statistical method	n-grams	-	-	+
	decision trees	-	-	-
	dynamic bayes	-	-	-
	maximum entropy	-	-	-
	SVM classifiers	+	+	+
Non-isomorphic mapping		+	-	-/+
Domain independent		+	+	+
Languages tested		ENG,CHN	ENG,CHN,GER,FRE	ENG

Table 4.21: Overview of features of statistical realizers presented in Sections 4.2, 4.3 and 4.1; “-” means “yes”, “+” means “no”, and “-/+” means “partially”

CHAPTER 5

Multilevel annotation and dependency parsing

In this chapter, we want to show that the resources we built with Natural Language Generation in mind can also be useful for other objectives, in particular for surface- and deep-syntactic parsing. First of all, we present a study on the impact of the granularity of our annotation scheme at the surface-syntactic layer on the results of various statistical dependency parsers (Section 5.1). Then, we report on experiments on making thorough use of the morphological features for optimizing the results of surface-syntactic parsing (Section 5.2). Finally, we explore deep-syntactic parsing, that is, the SSyntS–DSyntS transition (Section 5.3).

We show that separating the annotation of the different phenomena of language is equally justified for parsing, be it superficial or deep, as for NLG.

5.1 Tag granularity and dependency parsing performance

5.1.1 Introduction

As already pointed out by some researchers (see, e.g., Kübler (2005), Rehbein and van Genabith (2007), Bosco et al. (2010), Bosco and Lavelli (2010)), the use of a single annotation scheme for treebank creation leaves the question open to what extent the performance of an application trained on a treebank depends on the annotation scheme in question. Or, in other words, whether the annotation scheme in use is the best for a given application. To answer this question, Kübler (2005) and Rehbein and van Genabith (2007) compared the performance of a PCFG parser trained on two comparable corpora of German, annotated following different annotation schemes, while Bosco et al. (2010) trained three dependency parsers on two different Italian corpora. In contrast, we are interested in a comparison of the change of the performance of a dependency parser when trained on the same corpus, but annotated with gradually more fine-grained annotation schemes, that is, with gradually more arc labels in the tagset. We have seen that the results of micro-classifier-based stochastic generation got slightly better with a coarse-grained surface-syntactic annotation, but that globally the system was stable across granularities. In this section, we carry out a similar experiment with dependency parsing.

Our approach differs from (Bosco and Lavelli, 2010) in that the only information available in the tagset is syntactic (see Chapter 3). The background of our research is that standard annotation schemes such as the scheme underlying the dependency conversion from the Penn TreeBank tend to be minimal in order to facilitate the process of annotation and to improve the readability of the resulting annotation.[1] This tendency is reinforced by the general assumption that the less fine-grained the annotation, the better the parser performance. However, this has a major drawback, namely that the parsed structure is often too poor to serve well, e.g., semantic role labeling, deep summarization, content extraction, word sense disambiguation, etc.

To the best of our knowledge, no study actually compares the performance of a dependency parser trained on annotations of varying syntactic granularity, so there are no figures that would demonstrate that it is worth to sacrifice grammatical accuracy and detail for the sake of an acceptable parser accuracy. We carried out such a study on Spanish material, with a hierarchical syntactic dependency annotation scheme at hand that allows us to expand and contract syntactic relation branches into larger, more fine-grained, or smaller, more coarse-grained, annotation schemes (see Section 3.3.3.1). The results of parsing experiments demonstrate that it is possible to reach a good balance between the accuracy of a parser and the richness of the linguistic annotation. They also show that the principles that we applied when designing the hierarchical annotation scheme are valid and may be used for the design of other annotation schemes in the future.

---

[1] "Minimal" refers here not only to the number of tags, but also to the level of precision of the syntactic tags. Indeed, many corpora mix several levels of representation (e.g., syntax, semantics, lexicon, etc., see Section 2.3) such that the number of syntactic relations does not necessarily reflect the level of idiosyncracy of the annotation.

5.1.2 Experiments

5.1.2.1 Background

A number of experiments on the granularity of annotation and its impact on the performance of probabilistic parsers are known from the literature; see in particular Klein and Manning (2003) and Petrov et al. (2006), who show the benefits of splitting generic Part-of-Speech tags (e.g., *NP*, *VP*, etc.) into more precise subcategories for the derivation of accurate probabilistic context-free grammars (PCFG). Our proposal differs from these works in that they focus on constituency parsing and PoS tags, whereas we tackle dependency parsing and edge labels.[2] But more importantly, the goals are different. Thus, they target the improvement of parsing accuracy, and for that they infer, with simple rules, from the training data (categorial) information which is more specific than what is directly available. Bosco and Lavelli (2010) use an Italian corpus in which the dependency relations encode information on morphology, functional syntax and semantics. They discuss the influence of the annotation policies on the evaluation of the parsers and show that the precision and recall of hard-to-parse relations can be quite different, depending on the tag granularity in the annotation, that is, whether the annotation contains or not morphological and/or semantic information. In contrast, our goal is to provide evidence that the creation of annotations that capture significant fine-grained distinctive features of the grammar (and only the grammar) of a language does not need to harm significantly the performance of the parsers. Consider as two such fine-grained distinctive features the relations *modal* and *direct-object* in the following two sentences. As indicated, only the direct object can be pronominalized by a clitic pronoun and moved before the governing verb, without that a pro-verb is needed: *Juan puede*-modal→ *venir mañana*, lit. 'John might come tomorrow' (*Juan lo puede *(hacer)*, 'Juan it might *(do)'), and *Juan puede*-dobj→ *venir mañana*, lit. 'John is able to come tomorrow' (*Juan lo puede (hacer)* 'Juan it is-able (to do)'). If the annotation of the relations does not encode these phenomena, they are, in fact, lost.[3] Since this infor-

---

[2] Some other works present a hierarchical organization of grammatical relations (in particular (Bosco et al., 2000), (Briscoe et al., 2002), and (de Marneffe et al., 2006)), but those hierarchies are not used to test the impact of the tagset granularity on the results of a parser.

[3] One can always imagine some statistical "disambiguation" based on the context in which the construction is used, but the amount of data needed could be prohibitive—at least for Spanish—and eventually, the only way would probably be to imply human experts for the revision of the annotation.

mation is of primary relevance to applications related to natural language understanding, it would be an advantage to include it in the syntactic annotation. In the next sections, we show that its inclusion does not harm a parser's accuracy.

5.1.2.2 Setup of the experiments

In our experiments, we use the hierarchy introduced in Section 3.3.3.1; we add a very fine-grained tagset which contains 60 tags, and reduce the 48-tag column to 44 tags in order to obtain a better balance of the number of labels in each tagset; *cf.* Tables 5.1 and 5.2. Starting from the most fine-grained annotation, we derive automatically the other three, ending up with four different treebanks for the same corpus. Four reference parsers are used. Three of them are the top three parsers for Spanish in the CoNLL Shared Task 2009 (Hajič et al., 2009): Che's (Che et al., 2009), henceforth $Parser_{Che}$, Merlo's (Gesmundo et al., 2009), henceforth $Parser_{Merlo}$, and Bohnet's (Bohnet, 2009), henceforth $Parser_{Bohnet}$. The fourth, the Malt Parser (Nivre et al., 2007b), henceforth $Parser_{Malt}$, has been chosen because it is a very broadly used syntactic dependency parser. Parser$_{Malt}$ and Parser$_{Merlo}$ are transition–based, while Parser$_{Bohnet}$ and Parser$_{Che}$ are graph–based. In our experiments, all of them process non-projective dependency trees. Each parser contains its own configuration options, which depend on the parsing approach, the learning techniques, etc. Therefore, it is not possible to apply the same setup to all parsers. Instead, we use for each parser its own default configuration, which does not guarantee an optimal performance. However, as our objective is not to compare the results of the parsers, but rather the performance of the same parser with different tagsets, optimized configurations are not needed for our purpose.

To train the parsers, the corpus is divided randomly into a training set (3200 sentences) and a test set (313 sentences). Each parser is trained on each of the four annotations of the training set.[4] The obtained sixteen parsing models are applied to the corresponding test sets. Also, in order to see whether or not the performance improved with respect to the smallest tagset when training with more fine-grained tagsets, we map the output of each parser onto the smallest tagset. The training and the test sets are the same as in the base experiment.

---

[4] Bohnet's parser uses CoNLL'09 14-column format, while the other three need to be trained on the CoNLL'06 10-column format (Buchholz and Marsi, 2006), but the available information is exactly the same, whatever the format: word positions, word forms, PoS, lemmas, (all of which kept the same in our experiments), and dependencies.

60 Rels	**44 Rels**	**31 Rels**	**15 Rels**
abs_pred	abs_pred	abs_pred	NMOD
det	det	det	
quant	quant	quant	
compl_adnom	compl_adnom	compl_adnom	
appos	appos	modif	
abbrev	abbrev		
attr	attr		
modif	modif		
relat	relat		
adjunct	adv	adv	ADV
adv			
restr			
relat_expl	relat_expl		
prolep	prolep		
adv_mod	copred		
obj_copred			
subj_copred			
analyt_fut	analyt_fut	analyt_fut	AUX
analyt_pass	analyt_pass	analyt_pass	
analyt_perf	analyt_perf	analyt_perf	
analyt_prog	analyt_prog	analyt_prog	
modal	modal	modal	
dobj_clitic	dobj_clitic	dobj_clitic	DOBJ
dobj	dobj	dobj	
copul	copul	copul	COPUL
copul_clitic	copul_clitic	copul_clitic	
iobj1	iobj	iobj	IOBJ
iobj2			
iobj3			
iobj_clitic1	iobj_clitic	iobj_clitic	
iobj_clitic2			
iobj_clitic3			

Table 5.1: Tag groupings for a hierarchy of syntactic tags (1)

60 Rels	**44 Rels**	**31 Rels**	**15 Rels**
obl_obj1	obl_obj	obl_obj	OOBJ
obl_obj2			
obl_obj3			
obl_compl			
agent	agent		
compar	compar	compar	
compl1	compl	compl	
compl2			
elect	elect	elect	
subj	subj	subj	SUBJ
quasi_subj	quasi_subj	quasi_subj	QSUBJ
compar_conj	conj	prepos	PREPOS
sub_conj			
coord_conj	coord_conj		
prepos	prepos		
coord	coord	coord	COORD
num_junct	num_junct		
juxtapos	juxtapos	juxtapos	
quasi_coord	quasi_coord		
sequent	sequent	sequent	BIN
bin_junct	bin_junct	bin_junct	
aux_phras	aux_phras	aux_phras	NAME
aux_refl_lex	aux_refl	aux_refl	AUX_REFL
aux_refl_pass			
aux_refl_dir			
aux_refl_indir			
punc	punc	punc	PUNC
punc_init	punc_init		

Table 5.2: Tag groupings for a hierarchy of syntactic tags (2)

5.1.2.3 Results

For Malt, the assessment of the *Labeled Attachment Score* (LAS) (that is, the proportion of edges with correct governor and dependent and the right label on the edge) is carried out using the evaluation toolkit provided with the parser. For the other parsers, we use the official CoNLL'06 evaluation toolkit. The LAS figures for each parser and for each version of the annotation are shown in Table 5.3.

tags# >	**60**	**44**	**31**	**15**
Parser$_{Bohnet}$	81.95	84.11	84.28	84.69
Parser$_{Che}$	75.14	84.24	84.67	85.11
Parser$_{Malt}$	79.7	81.9	82.1	82.2
Parser$_{Merlo}$	82.32	84.53	84.05	84.52

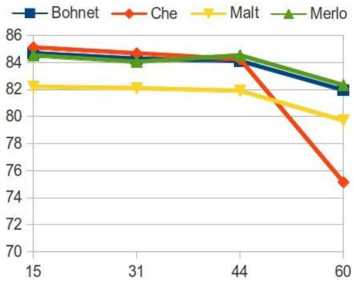


Table 5.3: LAS (%) of the parsers depending on tag granularity; right: graphical illustration

The graphic on the right of Table 5.3 shows how each parser reacts to and how its performance varies with the increasing number of relations in the tagset. We can observe that all four parsers behave similarly: their accuracy is very constant from 15 to 44 SSyntRels, and decreases with 60 SSyntRels. We also notice that there is a significant difference between Parser$_{Bohnet}$, Parser$_{Merlo}$ and Parser$_{Malt}$'s LAS progressions (which are rather parallel) and the progression of Parser$_{Che}$, which drops when trained with 60 relations (see Section 5.1.3). As expected, all parsers reach the highest accuracy with the smallest tagset (15 SSyntRels). But surprisingly, the LAS decreases only little with twice as many SSyntRels in the tagset (namely 31 SSyntRels): 0.1 for Malt, 0.41 for Parser$_{Bohnet}$, 0.44 for Parser$_{Che}$, and 0.47 for Parser$_{Merlo}$. Even more surprisingly, the drop is also rather small between 31 and 44 SSyntRels (0.2 for Parser$_{Malt}$, 0.17 for Parser$_{Bohnet}$, 0.43 for Parser$_{Che}$). Parser$_{Merlo}$ even gets better with 44 SSyntRels, obtaining a LAS of 84.53%, comparable to that with 15 SSyntRels and higher than that with 31 SSyntRels. As a result, the decrease of performance from 15 to 44 tags in the tagset is surprisingly small for Parser$_{Malt}$, Parser$_{Bohnet}$ and Parser$_{Che}$: 0.3 points for Parser$_{Malt}$, 0.6 points for Parser$_{Bohnet}$, 0.9 points for Parser$_{Che}$, and no decrease at all for Parser$_{Merlo}$. However, Parser$_{Bohnet}$, Parser$_{Malt}$ and Parser$_{Merlo}$ see their LAS drop significantly by around 2 points when trained with 60 SSyntRels. Parser$_{Che}$ drops by even more than 2 points. The in–depth analysis of the behavior of the parsers with respect to the groups of relations is presented in Section 5.1.3.

tags# >	**60**	**44**	**31**	**15**
Parser$_{Bohnet}$	90.49	90.39	90.31	90.27
Parser$_{Che}$	86.28	90.37	90.57	90.6
Parser$_{Malt}$	87.91	88	87.83	87.75
Parser$_{Merlo}$	90.11	90.67	90.39	-

Table 5.4: ULA of the parsers depending on tag granularity (%)

We also calculate the *Unlabeled Attachment* (ULA) score for all four parsers (see Table 5.4). For a reason beyond our control, we could not get the ULA for Parser$_{Merlo}$ with 15 relations (however, even if incomplete, the ULA figures for Parser$_{Merlo}$ are useful from the perspective of one of our experiments described below). For Parser$_{Bohnet}$, we observe that the ULA scores slightly but steadily increase in the range from 15 SSyntRels (90.27%) to 60 SSyntRels (90.49%). Opposite to this tendency, the scores for Parser$_{Che}$ slightly decrease in the range from 15 SSyntRels (90.6%) to 44 SSyntRels (90.37%), and drop then with 60 SSyntRels (86.28%). Parser$_{Malt}$ is as stable as Parser$_{Bohnet}$, but does not show a regular improvement when dealing with higher numbers of tags. Note that the observed slight variation of the performance numbers of the different parsers across tagsets of varying sizes (always lower than 0.25 points, except Parser$_{Che}$ with 60 relations) could be due to the small size of our training and test sets. In other words, it is possible that with more data, the parsers would give quite stable unlabeled attachment scores across tagsets of varying sizes.

In order to verify the effects of training a parser on a fine-grained tagset and using it then to parse with a coarse annotation, we take the test sets parsed with the models trained on 31, 44, and 60 relations, and map them to the coarse-grained tagset (15 different tags), following the hierarchy presented in Tables 5.1 and 5.2. Then, we run the evaluation of the resulting output against the gold standard of the 15-tag annotation; the results are presented in Table 5.5. In the first column, the figures obtained with the original 15-tag annotated test set for each parser are repeated in order to facilitate the comparison.

Table 5.5 shows that there does not seem to be a benefit in annotating with fine-grained arc labels if one wants a coarse annotation. The only case in

tags# >	**15**	**31→15**	**44→15**	**60→15**
Parser$_{Bohnet}$	84.69	84.56	84.51	84.54
Parser$_{Che}$	85.11	84.93	84.71	77.91
Parser$_{Malt}$	82.2	82.3	82.2	82.2
Parser$_{Merlo}$	84.52	84.33	84.92	84.12

Table 5.5: LAS of the parsers (with 15 SSyntRels) trained on fine-grained tagsets (%)

which a fine-grained annotation makes the parser improve significantly with 15 SSyntRels (0.4 points) is the 44 SSyntRel annotation for Parser$_{Merlo}$. Table 5.5 is actually very similar to Table 5.4, which contains the unlabeled attachment scores: all the figures for each parser are quite similar, with two exceptions: the fall of Parser$_{Che}$ trained with 60 SSyntRels, and a peak for Parser$_{Merlo}$ trained with 44 relations. The correlation between ULA and LAS is obvious, but unfortunately, we cannot explain so far those two deviations of ULA.

5.1.3 Evaluation of selected parsers with respect to specific SSyntRels

In the previous section, we saw that the figures of all four parsers drop when trained on the most fine-grained tagset. In this section, we try to identify which relations particularly affect the performance of the parsers and thus obtain information on how the composition of the tagset has an impact on the figures of the evaluation.[5]

5.1.3.1 Impact of distinctive properties of SSyntRels

Due to the relatively small amount of data we have at hand[6], there are only 8025 relation instances in the test set[7]. Some relations do not appear in it

---

[5] The problematic SSyntRels were the same for all four parsers. We choose to focus on the two graph-based parsers, since the graph-based approach becomes increasingly popular in parsing research.

[6] Still, we believe that our results are already quite reliable since the average accuracies (without tuning the parsers) get close to the accuracies obtained by the same parsers at the Shared Task 2009 with much larger data sets (http://ufal.mff.cuni.cz/conll2009-st/results/results.php).

[7] The dependencies to punctuation signs were not considered in the figures of the evaluation because they are parsed with the same (very high) accuracy whatever the tagset; considering them would boost the parser figures by 0.5% but it would not bring anything to our experiment.

at all: *prolep*, *adv-mod*, *copul-clitic*, *num-junct* and *aux-refl-indir*. On the other side, it is not possible to generalize along the lines that the less a relation appears in the training set, the worse the performance of the parser on this relation is. Some relations (*compl-adnom*, *analyt-fut*, *analyt-progr*, *analyt-perf*, *compar*, *compar-conj*, and *compl1*) are scarce in the training set (<200 instances) and in the test set (<20 instances) and, in spite of this, they are parsed with a high accuracy (78%–100%) at least by one of the parsers.

Interestingly, as opposed to the example about objects and modals in Section 5.1.2, either the governor or the dependent (or both) of these relations have very distinctive features:

- *compl-adnom* implies a determiner followed by a preposition; cf. *la*-compl-adnom→*del sombrero azul*, lit. 'the of-the hat blue', 'that one with the blue hat';
- *analyt-fut*, *analyt-progr* and *analyt-perf* always presuppose the same auxiliary as governor and a governed preposition or a non-finite verb as dependent; cf. *voy*-analyt-fut→*a cocinar*, lit. 'I-will [to] cook'; *estoy*-analyt-progr→*cocinando*, lit. 'I-am cooking'; *fue*-analyt-pass→ *cocinado*, lit. 'I-was cooked';
- *compar* and *compar-conj* require a comparative adjective governing a fixed conjunction, itself governing another element (*compar-conj*); cf. *mejor*-compar→*que*-compar-conj→*Juan*, lit. 'better than John';
- *compl1* requires an adjective on the right of a non-copular verb which undergoes agreement with the subject; cf. *la frase resultó*-compl1→ *buena*, lit. 'the sentence$_{FEM.SG}$ ended up correct$_{FEM.SG}$.

There are also some relations that are not parsed well by either of the parsers, even if the number of their instances in the training and test sets is significant (see Table 5.6). There are two main explanations of the poor figures for the SSyntRels in Table 5.6. First, the morpho-syntactic features of such relations (e.g., PoS of the head, PoS of the dependent) can vary a lot throughout the corpus: an *adverbial* or an *adjunctive* can be an adverb, a common noun, a non-finite verb, a prepositional group, etc. An *appositive* is usually a common or a proper noun, sometimes introduced by a preposition; an *attributive* can be a prepositional group or a gerund. Second, these relations also tend to share their basic syntactic configuration with

	Training Set (instances)	**Test Set** (instances)	**Parser**$_{Bohnet}$ (%)	**Parser**$_{Che}$ (%)
adjunct	830	87	37.93	31.03
adv	5751	549	62.3	56.83
appos	1060	100	54	34
attr	2165	213	37.56	41
obl-obj1	3551	384	50.78	26.82

Table 5.6: Poorly parsed frequent SSyntRels

other SSyntRels; consider, e.g., *casa*-attr→*de Barcelona*, lit. 'house from Barcelona' vs. *hermano*-obl-obj1→*de Juan* 'John's brother'. Thus, even if the two syntactic constructions seem to be the same (the governor is a noun, the dependent is a preposition, and the dependent of it is a proper noun), only the *attributive* dependent can be replaced by an adverb, and only the *oblique objectival* is introduced by a preposition which cannot be changed (i.e., a governed preposition; in this case, *de* 'of'). As far as the SSyntRels in Table 5.6 are concerned, an *appositive* (and even an *adverbial* in some cases) can also be confused with them: *nebulosa*-appos→*de Orion*, lit. 'nebula of Orion'. The other SSyntRels that share the same *N-Prep-N* configuration are: *abs-pred*, *obl-obj2*, *obl-obj3*, and *obl-compl*; all of these SSyntRels obtain poor scores in the evaluation of both parsers. Similarly, the only difference between *adverbials* and *adjunctives* is that *adjunctives* operate at a sentential level while the scope of *adverbials* is restricted to their governor: [*por ejemplo*]←adjunct-,-*funciona*-,-adv→ *con una silla*, lit. 'for instance, it-works, with a chair'. The two dependents of the verb are prepositional groups that could be found in any position of the sentence; in other words, there is no superficial clue that would differentiate one from the other.

This general absence of clear distinctive features for each particular SSyntRel makes it hard for the parsers to find patterns in their learning phases. Grouping the SSyntRels with similar configurations is the main factor that makes the parsers improve. In the next subsection, we give more details about the groupings made in the 60 label tagset.

5.1.3.2 Detailed analysis of the evaluations results

In this subsection, we take a close look at the SSyntRels which trigger the decrease of performance of the parsers between the tagsets containing 44 and 60 labels, respectively. In order to make an adequate comparison of the tagsets, we calculate the weighted average (*WA* in Tables 5.7 and 5.8) of the grouped relations and compare it with the score of the corresponding single edge label in the smaller tagset. We focus on the comparison between these two tagsets, given that the LAS variation of the parsers trained on them is higher than when trained on any other pair of tagsets.

SSyntRels 60	train #	test #	LAS %	WA %	**SSyntRels 44**	LAS %
iobj1	46	7	0			
iobj2	195	13	30.77	19.05	**iobj**	28.57
iobj3	1	1	0			
iobj-clitic1	81	5	20			
iobj-clitic2	262	21	76.19	62.96	**iobj-clitic**	81.48
iobj-clitic3	5	1	0			
obl-obj1	3551	384	50.78			
obl-obj2	662	62	20.97	52.24	**obl-obj**	71.1
obl-obj3	17	2	50			
obl-compl	1912	199	64.82			
compl1	141	9	66.67	50	**compl**	70
compl2	121	11	36.36			
aux-refl-pass	405	43	62.79			
aux-refl-lex	625	69	84.06	72.27	**aux-refl**	92.44
aux-refl-dir	102	7	14.29			
adjunct	830	87	37.93			
adv	5751	549	62.3	65.91	**adv**	69.64
restr	1913	194	88.66			
obj-copred	36	3	0	18.75	**copred**	16.67
subj-copred	76	9	25			

Table 5.7: Comparison between 60 and 44 SSyntRels for Parser$_{Bohnet}$

Table 5.7 does not show the results for the relations that have a one-to-one correspondence in both tagsets: *abs-pred*, *det*, *quant*, *compl-adnom*, *appos*, etc. This is because we observed that these relations show the same figures, or their figures only slightly improve or decrease from one tagset to another. In the end, these relations as a whole have almost no impact on the difference between the results obtained with the two tagsets. Instead, the two tables show the relations from the 60 relation tagset which are grouped together in the 44 relation tagset. Among them, only one grouping (*copred* for both parsers) does not lead to a better performance of the parser (16.67%, against 18.75% in average when separated into *obj-* and *subj-copred* for Parser$_{Bohnet}$, and 16.67% in both configurations for Parser$_{Che}$). The low number of occurrences of the relations grouped in *copred*, 25 in total, does not allow for a more profound analysis.

SSyntRels 60	train #	test #	LAS %	WA %	**SSyntRels 44**	LAS %
iobj1	46	7	0			
iobj2	195	13	15.38	5.13	**iobj**	57.14
iobj3	1	1	0			
iobj-clitic1	81	5	40			
iobj-clitic2	262	21	61.9	55.55	**iobj-clitic**	77.78
iobj-clitic3	5	1	0			
obl-obj1	3551	384	26.82			
obl-obj2	662	62	8.06	26.58	**obl-obj**	73.57
obl-obj3	17	2	0			
obl-compl	1912	199	32.16			
compl1	141	9	77.78	45	**compl**	65
compl2	121	11	18.18			
aux-refl-pass	405	43	62.79			
aux-refl-lex	625	68	42.03	49.64	**aux-refl**	91.6
aux-refl-dir	102	7	42.86			
adjunct	830	87	31.03			
adv	5751	549	56.83	59.51	**adv**	67.71
restr	1913	194	79.9			
obj-copred	36	3	66.67	16.67	**copred**	16.67
subj-copred	76	9	0			

Table 5.8: Comparison between 60 and 44 SSyntRels for Parser$_{Che}$

For all other relations in the 60 relation tagset, the weighted average in Parser$_{Bohnet}$ and Parser$_{Che}$ is significantly lower than the score of their corresponding group label in the 44 relation tagset:

- *iobj1*, *iobj2*, and *iobj3* give an average weighted LAS of 19.05% and 5.13% for the two parsers, whereas when they are grouped under one single label *iobj*, the LAS reaches 28.57% and 57.14%; in other words, the LAS drops 9.52 and 52.01 points respectively when training with the most fine-grained relations relations.

- The weighted average of *iobj-clitic1*, *iobj-clitic2*, and *iobj-clitic3* is 18.52 / 22.23 points lower than when these labels are grouped under the generic label *iobj-clitic*.

- The weighted average of *obl-obj1*, *obl-obj2*, *obl-obj3* and *obl-compl* is 18.86 / 46.99 points lower than when they are grouped under the label *obl-obj*. There are 647 instances of this relation in our test set, which means more than 8% of the total number of edges. This subset of SSyntRels is largely responsible for the bigger drop of Parser$_{Che}$ when trained with 60 relations.

- For *compl1* and *compl2*, the drop is also important compared to when they are grouped under *compl*: exactly 20 points for both parsers;

- The different types of reflexive auxiliaries that appear in the test set (passive, lexical, and direct) also work much better as one single label *aux-refl*: when they are separated, the LAS drops 20.17 and 41.96 points.

- Finally, for the other very important group by the number of instances in the test set (more than 10% of the edges), the comparison is similar, even if the amplitude is more reduced: *adjunct*, *adv* and *restr* see their LAS 3.73 and 8.2 points inferior to the LAS of the generic label *adv*, which includes them all in the 44 label tagset. Here too the drop is more important for Parser$_{Che}$ than for Parser$_{Bohnet}$ and largely accounts for the global LAS as seen in Table 5.3.

The performance drop of the 60 relation tagset when compared to the 44 relation tagset could, actually, be expected since some relations of the 60-tagset not only have superficially identical configurations (see Section 5.1.3.1), but the properties that differentiate them are closely related to semantics: the different kinds of oblique objects, completives, or reflexive auxiliaries actually behave among each other extremely similarly at the syntactic level, but reflect very distinct semantic realities. In fact, the number appended to the oblique object relation label not only stands for the order by default in a neutral sentence (with all the objects being present), but it also directly correlates with the slot in the valency pattern of the governor occupied by the corresponding dependent.[8] Although there is a relation between the default order of the objects and their (semantic) numbering, when several oblique objects of the same verb are used at the same time,

---

[8] This goes along the lines of Bosco et al. (2010), who mention that semantic distinctions are problematic in their experiments, and that merging locative and temporal complements under the same label, for example, increases the f-scores of the parsers.

there usually are communicative structure features that constrain their order. As a result, the objects are never instantiated in the same order, and the parser has almost no clue for guessing to which slot to assign an object.

From the bird's eye view of the composition of SSyntRel-tagsets, it seems that grouping together SSyntRels based on their syntactic properties helps the parsers. But not all relation groupings turn out to be beneficiary for the performance of the parsers. Consider the relations that connect two parallel clauses related by a coordination conjunction: *juxtapos*, *quasi-coord* and *coord*. In the 60 and 44 label tagsets, those three SSyntRels are kept separated, and the average weighted LAS is 71.5% and 72.58% for Parser$_{Bohnet}$, and 61.85% and 68.63% for Parser$_{Che}$ respectively. When *juxtapos* and *quasi-coord* are grouped in the 31 label tagset, Parser$_{Bohnet}$ drops by more than 2 points to 70.31%, while Parser$_{Che}$ slightly rises to 69.33%. However, when *coord* is also grouped with the other two under the label *COORD*, both parsers have more difficulties: Parser$_{Bohnet}$ drops by one point and Parser$_{Che}$ by more than six points. We believe that with these three SSyntRels, the syntactic constructions at stake are too different for the parsers to be able to find strong common features: a juxtaposition involves a punctuation sign (colon or semi-colon), while a coordination involves a conjunction or a comma, and a quasi-coordination nothing but the two coordinated elements (e.g., *Estoy aquí*-,-quasi-coord→*en mi cuarto!*, lit. 'I'm here, in my room!'). Therefore, we believe that even if it is tempting to annotate with a same label any coordinate structure, it is better to keep the different types annotated with different labels.

5.2 Morpho-syntactic annotation and dependency parsing

5.2.1 Introduction

As shown in NLP research, a careful selection of the linguistic information is relevant in order to produce an impact on the results. In this section, we want to look into different sets of morpho-syntactic features as we annotated them (see Section 3.2.1) in order to test their effect on the quality of surface-syntactic parsing for Spanish. To this end, we apply MaltParser (Nivre et al., 2007b), and MaltOptimizer (Ballesteros and Nivre, 2012a,b), which is a system capable of exploring and exploiting the different feature sets that can be extracted from the data and used over the models generated for MaltParser.

Starting from a corpus annotated with fine-grained language-specific information, we can use all or a part of the morpho-syntactic features to build different models and see the impact of each feature set on the Labeled Attachment Score (henceforth LAS) of the parser. We use MaltOptimizer in order to answer the following questions: (i) is the inclusion of all morphological features found in an annotation useful for Spanish parsing?; (ii) what are the optimal configurations of morphological features?; (iii) can we explain why different features are more or less important for the parser? For this purpose, the annotation presented in Chapter 3 is perfectly suitable: it includes features such as *number*, *gender*, *person*, *mood*, *tense*, *finiteness*, and coarse- and fine-grained *Part-of-Speech*. The impact of each feature or combination of features on subsets of dependency relations is also analyzed; for this, a fine-grained annotation of the syntactic layer is preferred since it allows for a more detailed analysis. We use a version of the corpus which contains 44 idiosyncratic syntactic tags (see Tables 5.1 and 5.2 in Section 5.1).

In the rest of the section, we situate our goals within the state of the art (Section 5.2.2), we describe the experimental setup, i.e. MaltParser, MaltOptimizer, the corpora used and the experiments that we carry out (Section 5.2.3), and we report and discuss the results of the experiments (Section 5.2.4).

5.2.2 Motivation and related work

Other researchers have already applied MaltOptimizer to their datasets, with different objectives in mind. Thus, the work of Seraji et al. (2012) shows that, for Persian, the parser results improve when following the model suggested by the optimizer. Tsarfaty et al. (2012a) work with Hebrew—a morphologically rich language—and incorporates the optimization offered by MaltOptimizer for presenting novel metrics that allow for jointly evaluating syntactic parsing and morphological segmentation. Mambrini and Passarotti (2012) use the optimizer not only to capture the feature model that fits best Ancient Greek, but also to evaluate how the genre used in the training set affects the parsing results. A step further is taken by Atutxa et al. (2012) for Basque: they want not only a good performance of the parser, but also a better disambiguation of the nominal phrases that can be either subjects or objects. In order to do that, they use the optimizer to detect the features (including morpho-syntactic ones) in the annotation that are useful for this task.

Even though the state-of-the-art results of parsing are very good when working with English, the results notoriously worsen when working with morphologically rich languages (MRLs). Tsarfaty et al. (2012b) present three different parsing challenges, broadly described as: (i) the architectural challenge, which focuses on how and when to introduce morphological segmentation; (ii) the modeling challenge, focused on how and where the morphological information should be encoded; and (iii) the lexical challenge, which faces the question of how to deal with morphological variants of a word that are not included in the corpus. The present experiment is directly related to the modeling challenge, given that we analyze in depth whether it is useful to incorporate morphological information as independent features.

Eryiğit et al. (2008) have already contributed to this topic by testing different morpho-syntactic combinations and their effect on MaltParser when applied to Turkish: they point out that some features do not make the dependency parser improve (in their case, *number* and *person*), and that Labeled and Unlabeled Attachment Scores (LAS/UAS) are unequally impacted by the feature variation (inflectional features affect more the labeled than the unlabeled accuracy). Bengoetxea and Gojenola (2009) and Agirre et al. (2011) have respectively tried to include semantic classes and feature propagation between different parsing models, with the intention of improving the parsing results for Basque.[9]

Spanish may not be as morphologically rich as other languages such as Hebrew, Turkish or Basque, but it involves enough morphological interactions to allow our research to contribute to such important discussion (Tsarfaty et al., 2010). For instance, determiners and adjectives agree in number and gender with the governing noun, finite verbs in number and person with their subjects; more complex types of agreement are (i) sibling interactions, such as copulative with subject, adjectival or past-participial with subject or object, (ii) dependents of siblings in the compound passive analytical construction, (iii) agreement of pronouns with their antecedent. (ii) and (iii) involve gender, number and sometimes person sharing; furthermore, some features are required on some verbs by their syntactic governor, such as a certain type of finiteness (gerund, participle, infinitive, finite) or mood. All those properties are encoded in the tagset used for the annotation of our corpus (see Section 3.3 for details about how the dependency tagset was designed), so we expect that the presence or absence of one or more of these

---

[9] Note that MaltOptimizer, which we use in this experiment, has been available since 2012, so previous works were realized with the basic version of MaltParser.

features in the training corpus will have a clear impact on the quality of the parsing.

5.2.3 Experimental setup

Here are the steps we follow in our experiments:

1. The corpus is divided into a training set (3263 sentences, 93803 tokens, 28.7 tokens/sentence) and a test set (250 sentences, 7089 tokens, 28.4 tokens/sentence);

2. 82 different versions of the training and test sets are created, based on different combinations of morpho-syntactic features;

3. The MaltParser is trained on a baseline model that does not include morphological features but uses the default feature models and parameters set in MaltOptimizer Phase 2, which provides general parameters and the best parsing algorithm for the data set.

4. We apply MaltOptimizer Phase 3, on each of the 82 training sets, and each configured model output is applied to the test set in order to obtain an evaluation;

5. We retain from the evaluation file LAS, UAS and LA (Labeled Accuracy) over all relations, as well as the recall of [*dependency relation + attachment*] for each of the 44 edge types.[10]

In the rest of this section, in order to understand better how morpho-syntactic features can impact the quality of parsing, we give more details about MaltParser and MaltOptimizer, before explaining the annotation that is used as the basis of this experiment.

5.2.3.1 MaltParser and MaltOptimizer

MaltParser (Nivre et al., 2007b) is a transition-based dependency parser generator that requires as an input a training set annotated in CoNLL-X data format,[11] and provides models capable of producing the dependency

---

[10] Because each training set contains different features, the test sets are obviously parsed differently and, in some cases, not all of the 44 dependency relations are predicted by the parser.

[11] http://ilk.uvt.nl/conll/#dataformat

Nivre's transition system:
$Initial = \langle [], [w_1 \ldots w_n], \emptyset, \emptyset \rangle \rightarrow Final = \{\langle \Pi, [], H, \Delta \rangle \in C\}$

Transitions:

Shift	$\langle \Pi, w_i \vert \beta, H, \Delta \rangle \Rightarrow \langle \Pi \vert w_i, \beta, H, \Delta \rangle$
Reduce	$\langle \Pi \vert w_i, \beta, H, \Delta \rangle \Rightarrow \langle \Pi, \beta, H, \Delta \rangle$
Left-Arc (dr)	$\langle \Pi \vert w_i, w_j \vert \beta, H, \Delta \rangle \Rightarrow \langle \Pi, w_j \vert \beta, H[w_i \rightarrow w_j], \Delta[w_i(dr)]\}\rangle$ if h(w_i) $\neq 0$.
Right-Arc (dr)	$\langle \Pi \vert w_i, w_j \vert \beta, H, \Delta \rangle \Rightarrow \langle \Pi \vert w_i \vert w_j, \beta, H[w_j \rightarrow w_i], \Delta[w_j(dr)]\}\rangle$ if h(w_j) $= 0$

Figure 5.1: Transition System for Nivre's algorithms with *reduce* transition (Nivre et al., 2007b)

parsing of new sentences. MaltParser implements four different transition-based parsers families and provides high and stable performance (see, e.g., Section 5.1). In the CoNLL Shared Tasks in 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007a), it was one of the best parsers, achieving either the first or the second place for most of the languages.

A transition-based parser is based on a state machine over mainly two data structures: (i) a buffer that stores the words to be processed and (ii) a stack that stores the ones that are being processed. The different transitions are shown in Figure 5.1; as can be observed, the state machine transitions manage the input words in order to assign dependencies between them. The transition-based parsers implemented in MaltParser use a model learned over a training corpus by using a classifier with the intention of selecting the best action (transition) in each state of the state-machine. The classifiers make their decisions according to the linguistic annotation included in the data, as shown in Figure 5.2. This basically means that the better the linguistic annotation is, the better the results are expected to be. The following attributes are the ones included in the CoNLL-X format which are used as features by the parser:

1. FORM: Word form.
2. LEMMA: Stemmed version of the word.
3. CPOSTAG: Coarse-grained PoS tag.

4. POSTAG: Fine-grained PoS tag.

5. **FEATS**: List of morpho-syntactic features (such as *number*, *gender*, *person*, *case*, *finiteness*, *tense*, *mood*, etc.)

6. DEPREL: Dependency relation to head.

A feature model is an option file in a MaltParser specific language based on XML that provides the linguistic annotation that the parser must take into account in order to produce the transitions. In each parsing state, the parser only knows the linguistic annotation included in the feature model. MaltParser includes a default feature model for each parsing algorithm. The

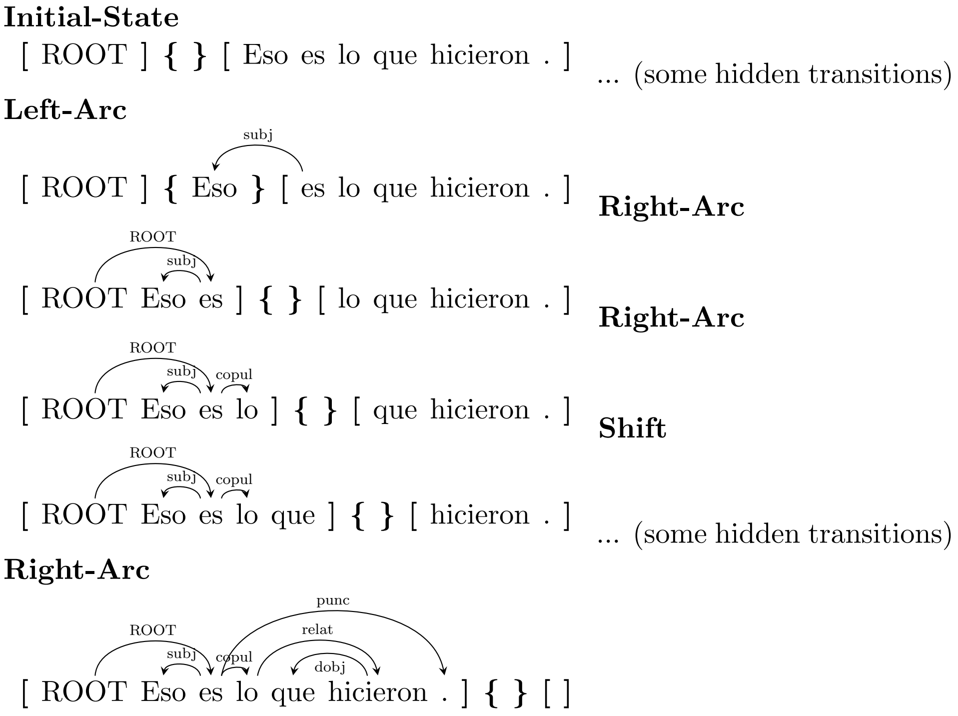
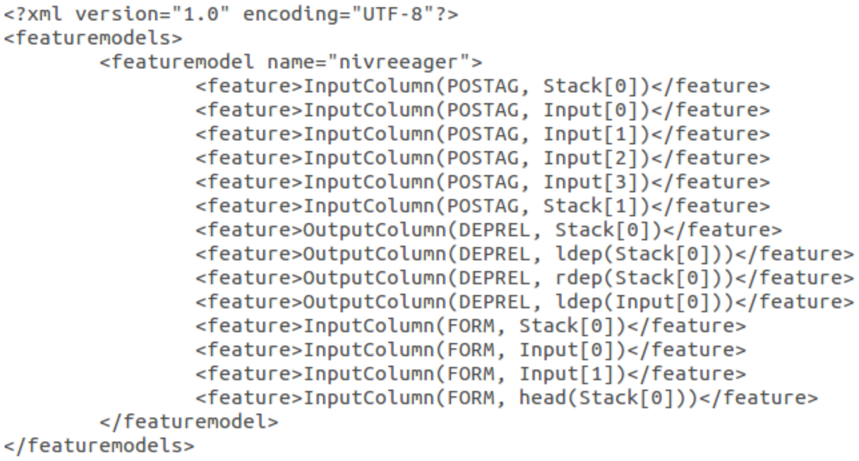


Figure 5.2: Some of the parsing transitions of a sentence taken from our data: *Eso es lo que hicieron* 'That's what they did'. The buffer is the structure that is represented to the right of the picture between '[' and ']', and the stack is the one to the left. Between each parsing state we show the transitions selected by the parser considering the features over the stack and the buffer.

default feature models, as we can see in Figure 5.3, only include features based on Part-of-Speech (*POSTAG*), the word form (*FORM*) and the partially built dependency structure (the output column, *DEPREL*) over the first positions of the stack and the buffer. Therefore, in order to let the parser know about the rest of the annotation (*LEMMA*, *CPOSTAG* and *FEATS*), if it exists, we need to perform a search of the different possible features.



```
<?xml version="1.0" encoding="UTF-8"?>
<featuremodels>
        <featuremodel name="nivreeager">
                <feature>InputColumn(POSTAG, Stack[0])</feature>
                <feature>InputColumn(POSTAG, Input[0])</feature>
                <feature>InputColumn(POSTAG, Input[1])</feature>
                <feature>InputColumn(POSTAG, Input[2])</feature>
                <feature>InputColumn(POSTAG, Input[3])</feature>
                <feature>InputColumn(POSTAG, Stack[1])</feature>
                <feature>OutputColumn(DEPREL, Stack[0])</feature>
                <feature>OutputColumn(DEPREL, ldep(Stack[0]))</feature>
                <feature>OutputColumn(DEPREL, rdep(Stack[0]))</feature>
                <feature>OutputColumn(DEPREL, ldep(Input[0]))</feature>
                <feature>InputColumn(FORM, Stack[0])</feature>
                <feature>InputColumn(FORM, Input[0])</feature>
                <feature>InputColumn(FORM, Input[1])</feature>
                <feature>InputColumn(FORM, head(Stack[0]))</feature>
        </featuremodel>
</featuremodels>
```

Figure 5.3: Default feature model for the Nivre arc-eager parsing algorithm

To this end, we use *MaltOptimizer* (Ballesteros and Nivre, 2012a,b), which is a system that not only implements a search of an optimal feature model, but also provides an optimal configuration based on the data set, exploring the parsing algorithms and the parameters within by performing a deep analysis of the data set. Thus, MaltOptimizer takes as an input a training set and it returns an options file and an optimal feature model. MaltOptimizer uses LAS as default evaluation measure and a threshold (>0.05) in order to select either the parameters, parsing algorithms or features. Due to the size of the training corpus, we run MaltOptimizer with 5 fold cross-validation in order to ensure the reliability of the produced outcome, and following the recommended settings of the system.

We are aware about the interactions between the features that are included in the default feature model and the ones selected or rejected by MaltOptimizer. However, our intention is to study the effect of the features included in the *FEATS* column, and the interaction with the other features is actually the real case scenario. By performing an automatic search of the linguistic annotation with MaltOptimizer, we are sure that all the morphosyntactic annotation included in the *FEATS* column is studied and tested by MaltOptimizer.

After running MaltOptimizer for Phase 1 and Phase 2, the best parser for (all) our data sets is Nivre arc-eager (Nivre, 2003), which behaves as shown in Figure 5.2; we are therefore ready to run the feature selection implemented in the Phase 3 of MaltOptimizer. Furthermore, the experiments performed by MaltOptimizer ensure that our features are tested in the last steps of the optimization process (Ballesteros and Nivre, 2012b).

5.2.3.2 Morphological features of our corpus

Table 5.9 shows the possible values that the features used in this experiment can take. In Chapter 3, Table 3.13 shows how these morpho-syntactic features are distributed through the corpus with respect to generic PoS. *gender* and *number* are the most frequent attributes, and they are annotated on elements of different PoS. The 2.02% of verbs that include *gender* are actually past participles. *gender=C* is not common; it stands for neutral elements, e.g., the dative pronoun *le* 'it' does not express masculine or feminine gender. *person* is only annotated on verbs, and not on nouns or pronouns. The other four attributes, (*finiteness*, *mood*, *person* and *tense*) are exclusively verbal features (except for the annotation errors). One can notice that there is some noise in the annotation of these verbal features (between 0.02% and 0.09% of elements not tagged as verbs carry them); however, as it happens in a reasonable proportion, it should not be a problem for our experiments. Also, not all the verbal elements carry all these features, given that some values of a specific feature impede the presence of another feature; e.g., *finiteness=INF* blocks *number* and *person*, since an infinitive verb cannot convey a number or a person.

5.2.3.3 Versions of the corpus

We prepared 82 different versions of the corpus in our experiments. The total number of possible combinations of the 7 features is 128 (0 features:1 combination; 1:7; 2:21; 3:35; 4:35; 5:21; 6:7; 7:1). However, after looking at figures with 1, 5, 6 and 7 features, we noticed that the combinations that excluded the *spos* feature were systematically making the parser unable to reach a certain score. As a result, for the rest of the experiments, we focused on combinations that do include *spos*.

FEAT	**Possible Values**	#Occurences
spos	adjective, adverb, auxiliary, conjunction, copula, determiner, foreign_word, formula, interjection, interrogative_pronoun, noun, number, percentage, preposition, pronoun, proper_noun, punctuation, relative_pronoun, roman_numeral, verb	100,892
pos	CC, CD, DT, IN, JJ, N, NN, NP, PP, RB, SYM, UH, VB, VH, VV, WP, formula	100,892
fin	finite, gerund, infinitive, past participle	11776
gen	neutral, feminine, masculine	41735
moo	imperative, indicative, subjunctive	8116
num	plural, singular	53608
per	1^{st}, 2^{nd}, 3^{rd}	8132
ten	future, past, present	8070

Table 5.9: Possible values and total number of occurrences of the 6 features

The 82 combinations are: 7 features (1 combination); 6 features (7); 5 features (21); 4 features, only those including *spos* (20); 3 features, only those including *spos* (15); 2 features, only those including *spos* (6); 1 feature (7); 0 feature (baseline, 1); 4 extra combinations in order to test the PoS/*spos* impact.

5.2.4 Results and discussion

First, we discuss the results of the first 78 experiments. In the last subsections, we will discuss the Part-of-Speech issues related to the other 4 experiments.

5.2.4.1 Feature combinations and general labeled accuracy

From a general perspective, as shown in Tables 5.10 and 5.11, 25 out of the 78 feature combinations make the LAS rise by at least 0.9 points; 14 of them make the LAS rise by more than 1 point. The biggest improvement, 1.33 points, is obtained with four features, namely [*finiteness gender number spos*]. Some similar improvements, between 1.28 and 1.3 points, have

	fin	**gen**	**moo**	**num**	**per**	**spo**	**ten**	**LAS**
0								82.25
1	**x**	**x**		**x**		**x**		+1.33
2	x			x	x	x		+1.3
3		x		x		x	x	+1.28
	x	**x**		**x**		**x**	**x**	
5	**x**	**x**	**x**	**x**	**x**	**x**		+1.22
6	x	x		x	x	x		+1.2
7		**x**		**x**		**x**		+1.14
	x	x		x	x	x	x	
9	x		x	x	x	x		+1.12
10	x			x	x	x	x	+1.1
		x	x	x	x	x	x	
12		x		x	x	x	x	+1.09
	x	x	x	x		x		
14	x		x	x		x		+1.02
15				x		x	x	+0.98
16			x	x	x	x		+0.94
	x		x	x	x	x	x	
18	x	x				x		+0.93
	x		x	x		x	x	
	x	x	x		x	x		
	x	**x**	**x**	**x**	**x**	**x**	**x**	
22	x	x	x		x	x	x	+0.91
23				**x**		**x**		+0.9
	x			x		x		
		x	x	x		x		
26	x	x	x	x		x	x	+0.88
27	x		x		x	x	x	+0.86
28	x					x		+0.82
	x	x		x			x	
			x	x		x	x	
31	x		x			x		+0.78
	x				x	x		
			x	x	x	x	x	
	x	x	x		x		x	
35		x	x	x		x	x	+0.77
	x		x	x	x		x	
	x	x	x	x			x	
38		x				x	x	+0.75
		x	x	x	x		x	

Table 5.10: Classification according to general LAS improvement of feature combinations (1st to 39th)

	fin	**gen**	**moo**	**num**	**per**	**spo**	**ten**	**LAS**
40			x	x		x		+0.73
	x				x	x	x	
		x	x	x	x	x		
43				x	x	x		+0.72
		x	x		x	x	x	
45	x		x		x	x		+0.7
	x			x		x	x	
		x		x	x	x		
48		x			x	x		+0.69
	x	x				x	x	
	x	x	x			x	x	
51		x				x		+0.67
			x		x	x		
	x	x			x	x	x	
54	x					x	x	+0.65
		x	x		x	x		
56	x	x	x			x		+0.62
	x	x			x	x		
	x		x			x	x	
59						x		+0.54
		x	x			x		
61		x	x			x	x	+0.53
			x		x	x	x	
63		x			x	x	x	+0.49
				x	x	x	x	
65	x							+0.46
66						x	x	+0.45
67			x			x		+0.43
68					x	x	x	+0.41
	x	x		x	x		x	
70	x	x	x	x	x		x	+0.4
71			x					+0.36
							x	
73					x			+0.35
74			x			x	x	+0.33
75					x	x		+0.3
	x	x	x	x	x			
77				x				+0.25
78		x						+0.09

Table 5.11: Classification according to general LAS improvement of feature combinations (40th to 78th)

been obtained with the following combinations: [*finiteness number person spos*], [*gender number spos tense*], [*finiteness gender number spos tense*]. Three out of the four biggest enhancements have been obtained with only 4 features.[12] This goes along the lines of Eryiğit et al. (2008), who report for Turkish the best results with only a subset of the morphological features present in the annotation.

What makes some features inefficient? In order to answer this question, we looked at Tables 5.10 and 5.11 from another perspective. For a given set of features, we wondered (1) if adding one particular feature make the LAS better or worse; and (2) which of the remaining features triggers the best LAS improvement. For instance, for the combination [*finiteness gender spos*]: (1) what happens to the LAS when we add one of the four remaining features? is it getting better or worse? and (2) which of these four features improves the most the LAS obtained while using only [*finiteness gender spos*]?

FEAT	**#Comb.**	**#better**	**#worse**	**#Best/Worst**
spo	6	6	0	6/0
num	31	30	1	22/3
fin	31	25	4	16/6
gen	31	21	10	9/11
per	31	16	15	7/9
moo	31	13	17	1/14
ten	31	12	19	1/22

Table 5.12: Contribution of each feature when enlarging the number of elements in a combination

Thus, based on the comparison between combinations that contain X elements and combinations that contain X+1 elements, we counted how many times each added feature made the LAS better, and how many times it made it worse. We also counted how many times each feature was involved in the best-scoring feature combination. The results obtained according to those lines are presented in Table 5.12. In the following, the detailed analysis for each feature is provided:

- ***spos*** was measured just when comparing the groups of five and six features (6 cases in total). It always improves the results (half of the

---

[12] In the table, the best combination for each size of feature set appears in bold, cf positions 1, 3, 5, 7, 18, 23, 59.

times with a percentage higher than 0.3 points). It never worsens and never belongs to the worst feature combination. See Section 5.2.4.3 for more details about this feature.

- ***number*** makes the LAS improve 30 out of 31 times (17 times the improvement is higher than 0.3 points), and is involved 22 times in the best scoring combination. It only worsens the results once (from 5 to 6 features, when combined with [*finiteness gender mood person tense*].[13] This feature is very useful in our experiments, and this could be explained by the following: (i) as shown in Table 5.9, this feature appears more frequently than any other feature (except *spos*), and it is distributed over elements of a great variety of PoS (see Table 3.13); (ii) many dependency relations in our annotation scheme use *number* directly or indirectly, on the head and/or the dependent: most verbal argumental relations (*subjectival*, *copulative*, *direct objectival*, *completive*, *clitic objectival*), verbal non-argumental relations (*passive analytical*, *copredicative*); nominal relations (*determinative*, *modificative*); etc.

- ***finiteness*** makes the LAS improve 25 times out of 31 (8 times the change is superior to 0.3 points). This feature is included in the optimal combination 16 times. On the other hand, it only worsens the results 4 times (and only once by more than 0.3 points, when combined with [*gender mood number person tense*], and it belongs to the worst combination 6 times. This feature often participates in improving the LAS, which could be due to the fact that it is the most important verbal feature, since it determines the presence or absence of other verbal features (e.g., it is only when *finiteness* has as value *finite* that other features such as *number*, *tense* or *person* can also be associated to the verb in question). In addition, this feature has a direct correlation with very frequent dependency relations as annotated in the corpus: only finite verbs can have a subject or be the head of a relative clause; only non-finite verbs can be governed by a preposition; in all *analytical* constructions (*perfect*, *progressive*, *passive*, *future*) the finiteness of the verb that depends on the auxiliary is always the same; etc.

- ***gender*** improves the results 21 times out of 31 (7 times the change is higher than 0.3 points), and belongs to the best combination 9 times.

---

[13] All the feature combinations improve the baseline results as shown in Tables 5.10 and 5.11, however, some of them do it in a more significant way.

However, it makes the LAS worsen 10 times (although just once—in combination with [*finiteness mood number person tense*] the variation is higher than 0.3 points), and belongs to the worst combination 11 times. Even though there are numerous relations that directly use this feature, most of the time it co-occurs with *number*, which possibly overshadows it. As a result, only in certain cases *gender* can bring new information that actually helps the parser.

- ***person*** improves the results 16 times out of 31 (4 times the change is higher than 0.3 points), and belongs to the best combination 7 times. On the other hand, it worsens the results 17 times (two times the change is higher than 0.3 points) and belongs to the worst combinations 14 times.

- ***mood*** improves the results 13 times out of 31 (only 2 times the variation is higher than 0.3 points), and belongs to the best combination just 1 time (with [*finiteness gender number person spos*]). It worsens the results 17 times (two times by more than 0.3 points) and belongs to the worst combination 14 times.

- ***tense*** is, according to this perspective, the "less useful" feature, in the sense that it improves the results just 12 times (and 2 times with a variation higher than 0.3 points). At the same time, *tense* makes the LAS drop 19 times out of 31, and it belongs to the worst combination 22 times. The only time that it belongs to the best combination (even if the results worsen) is with [*finiteness gender number spos*] (the "strongest" features).

We believe that *mood*, *tense*, and *person* are more redundant than informative for the parser, because (1) their presence on a node also indicates that a verb is finite, overlapping with the *finiteness* feature, and (2) no dependency relation uses the tense in the tagset, very few use the mood of a verb (only a subclass of the *conj* relation), and the person is only used in order to differentiate a subject from an object, since only the subject has to have the same *person* value as the verb. However, Spanish being an SVO (subject-verb-object) language, the order is most of the time sufficient in order to decide who is the subject and who is not (in addition, most of the nouns are 3rd person).

The first conclusion is that these observations coincide almost exactly with the ones made in Table 5.13: the features that individually tend to improve

	spo	**num**	**fin**	**gen**	**per**	**ten**	**moo**
14	14	14	10	10	8	6	5
25	25	22	17	15	13	11	12

Table 5.13: Occurrences of features in the 14 and 25 best scoring feature combinations

the LAS when added to other features are more likely to be in the best scoring combinations, while the features that often contribute to make the LAS drop are not. Interestingly, the four most frequent features in the 14 and 25 best combinations are also the four features that combine the best together, resulting in an increase of the baseline LAS of 1.33 points. This is not really a surprise, but it was a little less expected that this best scoring feature combination—[*finiteness gender number spos*]—comprises all (and only) the features that have a largely positive ratio of times they improve the LAS to times they make the LAS drop: respectively 25/4, 21/10, 30/1 and 6/0, as opposed the remaining three features that have 16/15, 13/17 and 12/19.

Second, the four best features according to our experiments are also the four most frequent in the corpus (see Table 5.9). The fact that a feature is productive in an annotation makes it obviously more likely to help a parser. However, it is not that straightforward: for instance, *finiteness* is four times less frequent than *gender*, but it triggers LAS improvements more often.

Third, it is not possible to get the best feature combination by simply looking at how each feature improves the LAS when being on its own: for instance, *number* and *gender* do not increase the LAS a lot by themselves (respectively ranks 77 and 78 out of 78 in Tables 5.10 and 5.11), but they do very well when they are combined to other attributes.

5.2.4.2 UAS, LA and specific dependency relations

Tables 5.10 and 5.11 are based on general LAS figures, because we are primarily interested in the general quality of the labeled parsing. However, depending on the type of application one is interested in, one may not be interested in labels, or may want to parse better some dependency relations in particular.

For this, we first compare the UAS and LA scores to the LAS, and as expected, they are behaving very similarly to the LAS results in that the same feature combinations work the best for all metrics. However, two

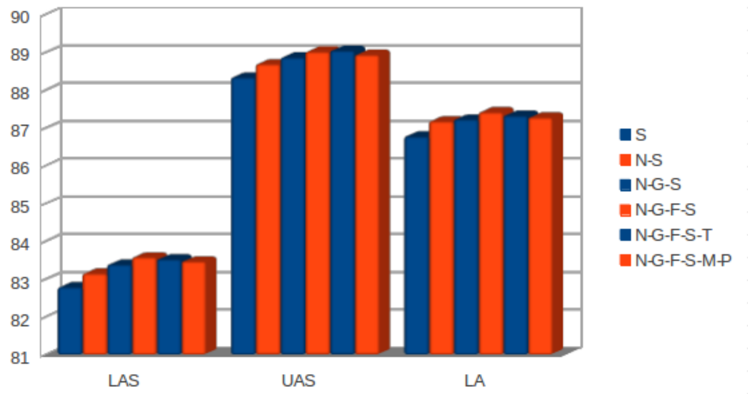


Figure 5.4: LAS, UAS, LA for the best feature combinations (S, *spos*), (N, *number*), (G, *gender*), (F, *finiteness*), (T, *tense*), (M, *mood*), (P, *person*)

differences can be pointed out: (1) the best LAS and LA are obtained with four features, while the best UAS is obtained with 5 features; (2) the LAS improves by up to 1.33 points (from 82.25 to 83.58), while the LA and UAS rise up to 1.04 and 1.06 points respectively (from 86.38 to 87.42 and from 87.99 to 89.05), corresponding to a reduction of errors of respectively 7.49%, 7.64% and 8.83%. Those observations are summed up in Figure 5.4, which shows the results according to each metric. The six columns for each metric represent, from left to right, the best results with one, two, three, four, five and six features.

If we try to find direct correlations between the presence or absence of a feature in the annotation and the improvement (or not) of the LAS figures for some relations in particular, it appears to be very hard to find such correlations by simply looking at the figures. For example, relations like subjects and different kind of objects are systematically parsed better with the introduction of any (combination of) feature(s), but some similar improvements are obtained with very different sets, which makes it hard to interpret. As pointed out recently by Schwartz et al. (2012) in a work about how to annotate some key dependencies in order to optimize parser results, annotating one dependency in a particular way will not only influence the parsing of this dependency, but also that of the surrounding dependencies. We believe that we fail in this particular task because one of the reasons is that there are a lot of indirect correlations that the human eye cannot see.

However, we wondered which feature combinations were the most efficient for specific applications, in particular, for the identification of verbal arguments and of the root of the sentences, and for the analysis of nominal groups and coordinated structures; interestingly, even if performing very well, the best general combination is never the best for any of those cases. For instance, for the identification of verbal arguments and sentence root, the best set is [*finiteness number person spos*]; for the internal NP structure, one should prefer [*gender mood number person spos tense*]; finally, for coordinated structures, one of [*finiteness gender number spos tense*], [*finiteness gender number person spos*] or [*gender number spos tense*]; see Table 5.14.

Task	**Best feature combination(s)**
Verbal argument identification	[*finiteness number person spos*]
NP structure definition	[*gender mood number person spos tense*]
Coordination parsing	[*finiteness gender number spos tense*] [*finiteness gender number person spos* [*gender number person tense*]

Table 5.14: Best morpho-syntactic feature combination according to particular parsing tasks

5.2.4.3 Some comments on Part-of-Speech

In this section, we detail shortly the last four experiments, that aim at finding out more about the importance of Part-of-Speech. The way we annotated the Part-of-Speech in our corpus slightly differs from the commonly used Tree Tagger tagset (see Table 3.2 in Section 3.2.1): the main difference is that while the tag *IN* gathers subordinating conjunctions and prepositions, we split it into two distinct tags *conjunction* (which also includes coordinating conjunctions) and *preposition*. Klein and Manning (2003) already showed that splitting the *IN* tag in this way improves constituency parsing accuracy with a PCFG parser. Our objective is to see if such a conclusion can be reached for dependency parsing.

We replaced the Tree Tagger PoS tags by the *spos* tags from our corpus in two feature combinations that did not include *spos*. Both times, the LAS was 0.5 points better. We also inverted PoS and *spos* in two other experiments, putting the latter in the POSTAG column of the CoNLL file, and the former in the FEATS column.[14] Again, the parser's LAS dropped half a point in both cases. We believe that this is due in particular to the

---

[14] Note that the default feature models include several feature specifications for the

fact that the *spos* tagset splits the IN tag into *conjunction* and *preposition* because this tag is way more frequent than the other mismatching tags. Therefore, when the *spos* is in the FEATS column, it specifies the POSTAG column and can be used in order to improve the parsing; however, the Tree Tagger tags in the FEATS column do not bring any new information and thus is ignored by MaltOptimizer. Also, MaltOptimizer starts with a higher baseline in this scenario and it is therefore more difficult to get improvements during the optimization steps, and thus the features are not selected. Splitting the *IN* tag does improve the accuracy of dependency parsing too.

5.3 Deep syntactic parsing

5.3.1 Introduction

Surface-syntactic structures as produced by data-driven syntactic dependency parsers (see previous sections) are *per force* idiosyncratic in that they contain governed prepositions, determiners, support verb constructions and language-specific (Johansson and Nugues, 2007). For many NLP-applications, including machine translation, paraphrasing, text simplification, etc., such a high idiosyncrasy is obstructive because of the recurrent divergence between the source and the target structures. Therefore, an increasing number of works suggest the use of more abstract “syntactico-semantic” structures; cf., among others, (Kittredge, 2002; Mel’čuk and Wanner, 2006; Siddharthan, 2011).

As *semantic role labeling* and *frame-semantic analysis*, deep-syntactic parsing has the goal to obtain more semantically oriented structures than those delivered by state-of-the-art syntactic parsing. Semantic role labeling received considerable attention in the CoNLL shared tasks for syntactic dependency parsing in 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007a), the CoNLL shared task for joint parsing of syntactic and semantic dependencies in 2008 (Surdeanu et al., 2008) and the shared task in 2009 (Hajič et al., 2009). The top ranked systems were pipelines that started with a syntactic analysis and continued with predicate identification, argument identification, argument labeling, and word sense disambiguation; cf. (Johansson and Nugues, 2008b; Che et al., 2009). At the end, a reranker that considers jointly all arguments to select the best combination

---

PoS column and the deepest experiments performed by MaltOptimizer are indeed in this feature window.

was applied. Some of the systems were based on integrated syntactic and semantic dependency analysis; cf., e.g., (Gesmundo et al., 2009); see also (Lluís et al., 2013) for a more recent proposal along similar lines. However, all of them lack the ability to perform structural changes—as, e.g., introduction of nodes or removal of nodes necessary to obtain a well-formed deep-syntactic structure. Furthermore, the resulting structures are usually not connected or complete, i.e., they do not capture all argumentative, attributive and coordinative dependencies between the meaningful lexical items of a sentence.

In Sections 5.1 and 5.2 above, we showed that it is possible to train surface-syntactic parsers on the annotation described in Chapter 3. Since we have a parallel deep-syntactic annotation at hand, we can go one step further and derive deep-syntactic structures from surface-syntactic structures, in a similar fashion to the pipeline implementation of (Johansson and Nugues, 2008b; Che et al., 2009). In the present experiment, the objective is then to learn how to remove functional node (and only functional nodes) from the SSyntS, and how to attach the remaining nodes together in the DSyntS. In other words, we address the same problem as the one of SSyntS–DSyntS transition between non-isomorphic structures (see Section 4.1), but in the other direction.

Therefore, we explore how a DSyntS is obtained from a SSyntS dependency parse using data-driven tree transduction in a pipeline with a syntactic parser. In Section 5.3.2, we discuss the fundamentals of SSyntS–DSyntS transduction. Section 5.3.3 describes the experiments that we carried out on Spanish material, and finally Section 5.3.4 discusses their outcome.[15]

5.3.2 Fleshing out the SSyntS–DSyntS transduction

As already pointed out in Chapter 4, it is clear that the SSyntS and DSyntS of the same sentence are not isomorphic. The following correspondences between the SSyntS S_{ss} and DSyntS S_{ds} of a sentence need to be taken into account during SSyntS–DSyntS transduction: **(i)** a node in S_{ss} is a node in S_{ds}; **(ii)** a relation in S_{ss} corresponds to a relation in S_{ds}; **(iii)** a fragment of the S_{ss} tree corresponds to a single node in S_{ds}; **(iv)** a relation with a dependent node in S_{ss} is a grammeme in S_{ds}; **(v)** a grammeme in S_{ss} is a grammeme in S_{ds}; (vi) a node in S_{ss} is conflated with another node in S_{ds}; and **(vii)** a node in S_{ds} has no correspondence in S_{ss}.

---

[15] The SSyntS–DSyntS transducer has been implemented by Miguel Ballesteros.

The grammeme correspondences (iv) and (v) and the "pseudo" correspondences in (vi) and (vii)[16] are few or idiosyncratic and are best handled in a rule-based post-processing stage. The main task of the SSyntS–DSyntS transducer is thus to cope with the correspondences (i)–(iii). For this purpose, we can view both SSyntS and DSyntS as vectors indexed in terms of two-dimensional matrices $I = N \times N$ (N being the set of nodes of a given tree $1, \ldots, m$), with $I(i,j) = \rho(n_i, n_j)$ if $n_i, n_j \in N$ and $(n_i, n_j) \in A$ ($i, j = 1, \ldots, m; i \neq j$) and $I(i,j) = 0$ otherwise (where '$\rho(n_i, n_j)$' is the function that assigns to an edge a relation label. That is, for a given SSyntS, $I(i,j)$ contains in the cells (i,j), $i,j = 1, \ldots, m$, the names of the SSynt-relations between the nodes n_i and n_j , and '0' otherwise, while for a given DSyntS, the cells contain DSyntS-relations.

For the reader's convenience, we give in see Figure 5.5 two more examples of SSyntSs and their corresponding DSyntSs.

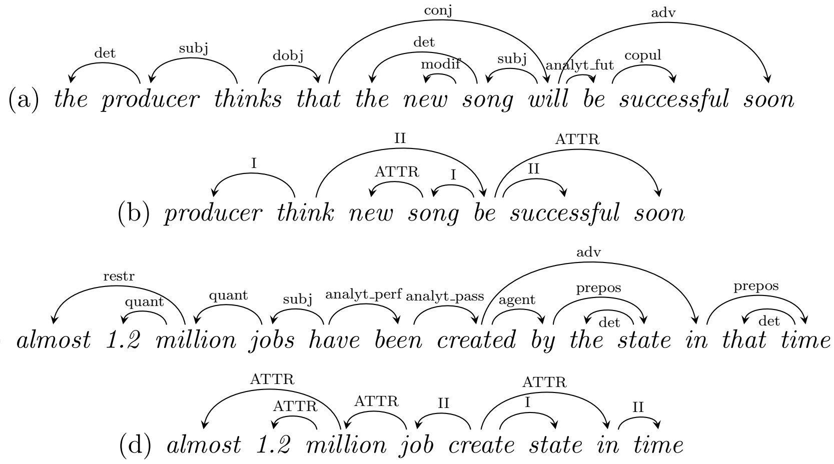


Figure 5.5: Two SSyntSs (a,c) and their corresponding DSyntSs (b,d)

Starting from I of a given SSyntS, the task is therefore to obtain I' of the corresponding DSyntS, or, in other words, to identify correspondences between i/j, (i,j) and groups of (i,j) of I with i'/j' and (i',j') of I'; see (i)–(iii) above. As 'token chain'→surface-syntactic tree' projection this task can be

---

[16] (vi) covers, e.g., reflexive verb particles such as *se* in Spanish, which are conflated in the DSyntS with the verb: *se*←aux_refl_dir-*conocer* vs. CONOCERSE 'know each other'; (vii) covers, e.g., the zero subject in pro-drop languages (which is absent in the SSyntS and present in the DSyntS).

viewed as a classification task. However, while the former is isomorphic, we know that the SSyntS–DSyntS projection is not. In order to approach it to an isomorphic projection (and thus simplify its modeling), it is convenient to interpret SSyntS and the targeted DSyntS as collections of *hypernodes*, as defined in Section 4.1.1. The fact that the SSyntS–DSyntS correspondence is reduced to a correspondence between individual nodes and between individual arcs has an interesting consequence from the perspective of this experiment. This way, the transduction embraces the following three (classification) subtasks: (i) Hypernode identification, (ii) DSynt tree reconstruction, and (iii) DSynt arc labeling, which are completed by a post-processing stage.

1. Hypernode identification. The hypernode identification consists of a binary classification of the nodes of a given SSyntS as nodes that form a hypernode of cardinality 1 (i.e., nodes that have a one-to-one correspondence to a node in the DSyntS) vs. nodes that form part of a hypernode of cardinality > 1. In practice, hypernodes of type one will be formed by: 1) noun nodes that do not govern determiner or functional preposition nodes, 2) full verb nodes that are not governed by any auxiliary verb nodes and that do not govern any functional preposition node, adjective nodes, adverbial nodes, and semantic preposition nodes. Hypernodes of type two will be formed by: 1) noun nodes + determiner / functional preposition nodes they govern, 2) verb nodes + auxiliary nodes they are governed by + functional preposition nodes they govern.

2. DSynt tree reconstruction. The outcome of the hypernode identification stage is thus the set $H_s = H_{s_{|p|=1}} \cup H_{s_{|p|>1}}$ of hypernodes of two types. With this set at hand, we can define an isomorphic function $\tau : H_s \to H_{d_{|p|=1}}$ (with $h_d \in H_{d_{|p|=1}}$ consisting of $n_d \in N_{ds}$, i.e., the set of nodes of the target DSyntS). τ is the identity function for $h_s \in H_{s_{|p|=1}}$. For $h_s \in H_{s_{|p|>1}}$, τ maps the functional nodes in h_s onto grammemes (attribute/value tags) of the lexically meaningful node in h_d and identifies the lexically meaningful node as head. Some of the dependencies of the obtained nodes $n_d \in N_{ds}$ can be recovered from the dependencies of their sources. Due to the projection of functional nodes to grammemes (which can be also seen as node removal), some dependencies will be also missing and must be introduced. Algorithm 9 recalculates the dependencies for the target DSyntS S_d, starting from the index matrix I of SSyntS S_s to obtain a connected tree. *BestHead* recursively ascends S_s from a given node n_i until it encounters one or several

Algorithm 9: DSyntS tree reconstruction

```
for $\forall n_i \in N_d$ do
  if $\exists n_j : (n_j, n_i) \in S_s \wedge \tau(n_j) \in N_d$ then
  $(n_j, n_i) \rightarrow S_d$ // the equivalent of the head node of $n_i$ is included in DSyntS
  else if $\exists n_j, n_a : (n_j, n_i) \in S_s \wedge \tau(n_j) \notin N_d \wedge$
        $\tau(n_a) \in N_d$ then
        //$n_a$ is the first ancestor of $n_j$ that has an equivalent in DSyntS
        //the equivalent of the head node of $n_i$ is not included in DSyntS,
        //but the ancestor $n_a$ is
        $(n_a, n_i) \rightarrow S_d$
      else
        //the equivalent of the head node of $n_i$ is not included in DSyntS,
        //but several ancestors of it are
        $n_b := BestHead(n_i, S_s, S_d)$
        $(n_b, n_i) \rightarrow S_d$
endfor
```

head nodes $n_d \in N_{ds}$. In case of several encountered head nodes, the one which governs the highest frequency dependency is returned.

3. Label Classification. The tree reconstruction stage produces a "hybrid" connected dependency tree $S_{s \rightarrow d}$ with DSynt nodes N_{ds}, and arcs A_s labeled by SSynt relation labels, i.e., an index matrix I^-, whose cells (i, j) contain SSynt labels for all $n_i, n_j \in N_{ds} : (n_i, n_j) \in A_s$ and '0' otherwise. The next and last stage of SSyntS–DSyntS transduction is thus the projection of SSynt relation labels of $S_{s \rightarrow d}$ to their corresponding DSynt labels, or, in other words, the mapping of I^- to I' of the target DSyntS.

4. Postprocessing. As mentioned above, there is a limited number of idiosyncratic correspondences between elements of SSyntS and DSyntS (the correspondences (iv–vii) which can be straightforwardly handled by a rule-based postprocessor because (a) they are non-ambiguous, i.e., $a \leftrightarrow b, c \leftrightarrow d \Rightarrow a = b \wedge c = d$, and (b) they are few. Thus, only determiners and auxiliaries at SSyntS level map onto a grammeme at DSyntS level, both SSyntS and DSyntS levels count with less than a dozen grammemes, etc.

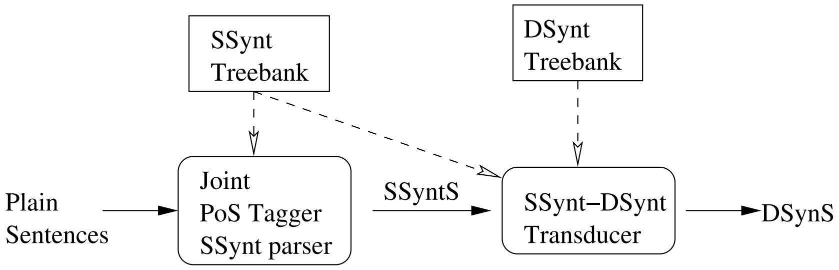


Figure 5.6: Setup of a deep-syntactic parser

5.3.3 Experiments

In order to validate the outlined SSyntS–DSyntS transduction and to assess its performance in combination with a surface dependency parser, i.e., starting from a plain sentence, we carry out a number of experiments in which we implement the transducer and integrate it into a pipeline shown in Figure 5.6.

For our experiments, we slightly adjust the surface-syntactic labels described in Tables 3.3 and 3.4 in order to include relations that can help the classifiers take decisions when deriving the deep-syntactic tree. We use 55 different syntactic labels: compared to the 48-label tagset, we add quotative and actantial *aux_refl* relations (*cf.* Table 3.12 on page 98), and leave the *sub_conj/compar_conj* and *adv/restr* distinctions from the original 79-tag annotation.

Our development set consists of 219 sentences (3271 tokens in the DSyntS treebank and 4953 tokens in the SSyntS treebank), the training set of 3036 sentences (57665 tokens in the DSyntS treebank and 86984 tokens in the SSyntS treebank), and the *test set* held-out for evaluation of 258 sentences (5641 tokens in the DSyntS treebank and 8955 tokens in the SSyntS treebank).

To obtain the SSyntS, we use Bohnet and Nivre (2012)'s transition-based parser, which combines lemmatization, PoS tagging, and syntactic dependency parsing—tuned and trained on the respective sets of the SSyntS treebank.

In what follows, we first present the realization of the SSyntS–DSyntS transducer and then the realization of the baseline.

5.3.3.1 SSyntS–DSyntS transducer

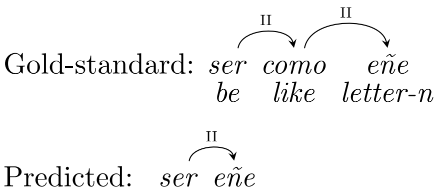
As outlined in Section 5.3.2, the SSyntS–DSyntS transducer is composed of three submodules:

1. Hypernode identification. For the hypernode identification, we train a binary polynomial (degree 2) SVM from LIBSVM (Chang and Lin, 2001). The SVM allows both features related to the processed node and higher-order features, which can be related to the head node of the processed node or to its sibling nodes. After several feature selection trials, we chose the following features for each node n:

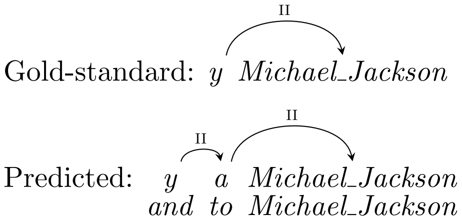
- lemma or stem of the label of n,
- label of the relation between n and its head,
- surface PoS of n's label (the SSynt and DSyntS treebanks distinguish between surface and deep PoS)
- label of the relation between n's head to its own head,
- surface PoS of the label of n' head node.

After an optimization round of the parameters available in the SVM implementation, the hypernode identification achieves over the gold development set 99.78% precision and 99.02% recall (and thus 99.4% F1). That is, only very few hypernodes are not identified correctly. The main error source are *governed prepositions*: the classifier has to learn when to assign a preposition an own hypernode (i.e., when it is lexically meaningful) and when it should be included into the hypernode of the verb/noun (i.e., when it is functional). Our interpretation is that the features we use for this task are appropriate, but that the training data set is too small. As a result, some prepositions are erroneously left out from or introduced in the DSyntS.

2. Tree reconstruction. The implementation of the tree reconstruction module shows an unlabeled dependency attachment precision of 98.18% and an unlabeled dependency attachment recall of 97.43% over the gold development set. Most of the errors produced by this module have their origin in the previous module, i.e., hypernode identification. When a node has been incorrectly removed, the module errs in the attachment because it cannot use the node in question as the destination or the origin of a dependency, as it is the case in the gold-standard annotation:



When a node has erroneously not been removed, no dependencies between its governor and its dependent can be established since DSyntS must remain a tree (which gives the same LAS and UAS errors as when a node has been erroneously removed):



3. Relation label classification. For relation label classification, we use a multiclass linear SVM. The label classification depends on the concrete annotation schemes of the SSyntS and DSyntS treebanks on which the parser is trained. Depending on the schemes, some DSynt relation labels may be easier to derive from the original SSyntS relation labels than others.[17]

The final set of features selected for label classification includes: (i) lemma of the dependent node, (ii) dependency relation to the head of the dependent node, (iii) dependency relation label of the head node to its own head, (iv) dependency relation to the head of the sibling nodes of the dependent node, if any.

After an optimization round of the parameter set of the SVM-model, relation labeling achieves 94.00% label precision and 93.28% label recall on the development set. The recall is calculated considering all the nodes that are included in the gold standard. The error sources for relation

---

[17] In Chapter 3, Table 3.14 (p.102) lists all SSynt relation labels that have a straightforward mapping to DSyntS relation labels in the used treebanks, i.e., neither their dependent nor their governor are removed, and the SSyntS label always maps to the same DSynt label; Table 3.15 (p.103) shows SSyntS relation–DSyntS relation label correspondences that are not straightforward.

labeling are mostly the dependencies which involve possessives and the various types of objects (see Table 3.15 p.103) due to their differing valency. For instance, the relation *det* in *su←det−coche* 'his/her car' and *su←det−llamada* 'his/her phone call' have different correspondences in DSyntS: *su←ATTR−coche* vs. *su←I−llamada.* That is, the DSyntS relation depends on the lexical properties of the governor. Once again, more training data is needed in order to classify better those cases.

4. Postprocessing In the postprocessing stage for Spanish, the following rules capture non-ambiguous correspondences between elements of the SSynt-index matrix $I = N_s \times N_s$ and DSyntS index matrix $I' = N_d \times N_d$, with $n_s \in N_s$ and $n_d \in N_d$, and n_s and n_d corresponding to each other (we do not list here identity correspondences such as between the *number* grammemes of n_s and n_d):

• if n_s is dependent of *analyt_pass* or governs *analyt_refl_pass* relation, then the *voice* grammeme in n_d is *PASS*;
• if n_s is dependent of *analyt_progr*, then the *tem_constituency* grammeme in n_d is *PROGR*;
• if n_s is dependent of *analyt_refl_lex*, then add the particle -SE as suffix of node label (word) of d_d;
• if any of the children of n_s is labeled by one of the tokens UN 'a$_{masc}$', UNA 'a$_{fem}$', UNOS 'some$_{masc}$' or UNAS 'some$_{fem}$', then the definiteness grammeme in n_d *INDEF*, otherwise it is *DEF*;
• if the n_s label is a finite verb and n_s does not govern a *subject* relation, then add to I' the relation $n_d - I \rightarrow n'_d$, with n'_d being a newly introduced node.

5.3.3.2 Baseline

As point of reference for the evaluation of the performance of our SSyntS–DSyntS transducer, we use a rule-based engine that carries out the most direct transformations extracted from Tables 3.14 and 3.15. The baseline detects hypernodes by directly removing all the nodes that we are sure need to be removed, i.e. punctuation and auxiliaries. The nodes that are only *potentially* to be removed, i.e., all dependents of DepRels that have a possibly governed preposition or conjunction in Table 3.15, are left in the DSyntS. The new relation labels in the DSyntS are obtained by selecting the label that is most likely to substitute the SSyntS relation label according to classical grammar studies. The rules of the rule-based baseline look as follows:

```
if (deprel==abbrev) then deep_deprel=ATTR
if (deprel==obl_obj) then deep_deprel=II
...
if (deprel==punc) then remove(current_node)
```

5.3.4 Results and discussion

The experiments give us performance figures of the SSyntS parser, the SSyntS–DSyntS transducer, and the sentence–DSyntS pipeline.

5.3.4.1 SSyntS–DSyntS transducer results

Hyper-Node Detection		
Measure	Rule-based Baseline	Tree Transducer
p	64.31 (5565/8653)	99.79 (5598/5610)
r	98.65 (5565/5641)	99.24 (5598/5641)
$F1$	77.86	99.51

Attachment and labeling		
Measure	Rule-based Baseline	Tree Transducer
LAP	50.02 (4328/8653)	91.07 (5109/5610)
UAP	53.05 (4590/8653)	98.32 (5516/5610)
LA-P	57.66 (4989/8653)	92.37 (5182/5610)
LAR	76.72 (4328/5641)	90.57 (5109/5641)
UAR	81.37 (4590/5641)	97.78 (5516/5641)
LA-R	88.44 (4989/5641)	91.86 (5182/5641)

LAP: labeled attachment precision
UAP: unlabeled attachment precision
LA-P: label assignment precision
LAR: labeled attachment recall
UAR: unlabeled attachment recall
LA-R: label assignment recall

Table 5.15: Performance of the SSyntS–DSyntS transducer and of the rule-based baseline over the gold-standard held-out test set (Spanish)

In Table 5.15, the performance of the subtasks of the SSyntS–DSyntS transducer is contrasted to the performance of the baseline; the evaluation of the postprocessing subtask is not included because the one-to-one projection of SSyntS elements to DSyntS guarantees an accuracy of 100%. The transducer is applied to the gold standard test set, which is the held-out test set, with gold standard PoS tags, lemmas and dependency trees. It outputs in total 5610 nodes; the rule-based baseline outputs 8653 nodes. As mentioned in Section 3, our gold standard includes 5641 nodes.

Hyper-Node Detection	
Measure	Tree Transducer
p	99.78 (44861/44961)
r	99.93 (44861/44892)
$F1$	99.85

Attachment and Labeling	
Measure	Tree Transducer
LAP	99.07 (44543/44961)
UAP	99.61 (44787/44961)
LA-P	99.08 (44549/44961)
LAR	99.22 (44543/44892)
UAR	99.77 (44787/44892)
LA-R	99.24 (44549/44892)

LAP: labeled attachment precision
UAP: unlabeled attachment precision
LA-P: label assignment precision
LAR: labeled attachment recall
UAR: unlabeled attachment recall
LA-R: label assignment recall

Table 5.16: Performance of the SSyntS–DSyntS transducer and of the rule-based baseline over the gold-standard held-out test set (Chinese)

Our data-driven SSyntS–DSyntS transducer is much better than the baseline with respect to all evaluation measures.[18] Also, while the rule-based baseline sometimes produces disconnected dependency trees, the transducer always delivers connected structures. The transducer relies on distributional patterns identified in the training data set, and makes thus use of information that is not available for the rule-based baseline, which studies one node at a time. However, the rule-based baseline results also show that transduction that would remove a few nodes would provide results close to a 100% recall for the hypernode detection because a DSynt tree is a subtree of the SSynt tree (if we ignore the nodes introduced by post-processing). This is also evidenced by the labeled and attachment recall scores. The results of the transducer on the test and development sets are quite comparable. The hypernode detection is even better on the test set. The label accuracy suffers most from using unseen data during the development of the system. The attachment figures are approximately equivalent on both sets. To validate our approach with languages other than Spanish, we carry out the exact same experiment on the Chinese Dependency Treebank (Chang et al., 2009), from which we automatically derived the deep-syntactic layer

---

[18] We also ran MaltParser by training it over the DSynt-treebank to parse the SSynt-test set; however, the outcome was too weak to be used as baseline.

thanks to a graph-transduction grammar. There are less functional nodes in Chinese than in Spanish, so the task is somehow easier, but the results are very good, as shown in Table 5.16.

5.3.4.2 Results of deep-syntactic parsing

POS	LEMMA	LAS	UAS
96.05	92.10	81.45	88.09

Table 5.17: Performance of Bohnet and Nivre's joint PoS-tagger+dependency parser trained on our annotation

Hyper-Node Detection		
Measure	Baseline	Tree Transducer
p	63.87 (5528/8655)	97.07 (5391/5554)
r	98.00 (5528/5641)	95.57 (5391/5641)
$F1$	77.33	96.31

Attachment and Labeling		
Measure	Baseline	Tree Transducer
LAP	38.75 (3354/8655)	68.31 (3794/5554)
UAP	44.69 (3868/8655)	77.31 (4294/5554)
LA-P	49.66 (4298/8655)	80.47 (4469/5554)
LAR	59.46 (3354/5641)	67.26 (3794/5641)
UAR	68.57 (3868/5641)	76.12 (4294/5641)
LA-R	76.19 (4298/5641)	79.22 (4469/5641)

LAP: labeled attachment precision
UAP: unlabeled attachment precision
LA-P: label assignment precision
LAR: labeled attachment recall
UAR: unlabeled attachment recall
LA-R: label assignment recall

Table 5.18: Performance of the deep-syntactic parsing pipeline on Spanish

We consider now the performance of the complete DSynt parsing pipeline (PoS-tagger+surface-dependency parser → SSyntS–DSyntS transducer) on the held-out test set. Table 5.17 displays the figures of the Bohnet and Nivre parser. The figures are in line with the performance of state-of-the-art parsers for Spanish. Table 5.18 shows the performance of the pipeline when we feed the output of the syntactic parser to the rule-based baseline SSyntS–DSyntS module and the tree transducer. We observe a clear error propagation from the dependency parser (which provides 81.45% LAS) to the SSyntS–DSyntS transducer, which loses in tree quality more than 18%.

We carry out the same experiment on the Chinese SSyntS-DSyntS Treebank, but with the MaltParser in its default settings. The results over predicted input show an accuracy of about 75%, i.e., an accuracy comparable to the one achieved for Spanish, see Table 5.20.

LAS	UAS	LAS
74.16	76.55	86.88

Table 5.19: Performance of MaltParser trained on the Chinese Dependency Treebank

Hyper-Node Detection	
Measure	Tree Transducer
p	99.33 (44849/45152)
r	99.90 (44849/44892)
$F1$	99.62

Attachment and Labeling	
Measure	Tree Transducer
LAP	75.45 (34068/45152)
UAP	77.81 (35135/45152)
LA-P	87.77 (39631/45152)
LAR	75.89 (34068/44892)
UAR	78.27 (35135/44892)
LA-R	88.28 (39631/44892)

LAP: labeled attachment precision
UAP: unlabeled attachment precision
LA-P: label assignment precision
LAR: labeled attachment recall
UAR: unlabeled attachment recall
LA-R: label assignment recall

Table 5.20: Performance of the deep-syntactic parsing pipeline on Chinese

5.4 Conclusions

In this chapter, we fulfilled the last objective of this thesis, i.e., showing that an annotation designed with NLG in mind can also be suitable for dependency parsing. In our first parsing experiment, the evaluation of the performance of four state-of-the-art parsers trained on the corpus annotated following schemes of different granularity revealed that the loss of accuracy as a consequence of the increase of the size of the tagset, in particular, from 15 to 44 tags, is surprisingly small. This outcome supports the claim that an annotation with more fine-grained syntactic relations does not necessarily imply a significant loss in accuracy. It also supports the argumentation that it is useful to compile a detailed annotation scheme, which then allows for the derivation of a variety of more or less detailed annotations. This study also suggests that there seems to be a limit with respect to the degree of detail of the tagset beyond which a parser's accuracy suffers significantly, and that there are some tags which provoke a drop of the LAS more than others. These are, in particular, the very fine-grained divisions which directly reflect semantic valency information. Another conclusion that can be drawn is that training a parser on a fine-grained annotation does not lead to a better performance of this parser when parsing with a coarse-grained tagset. However, it still remains unclear whether the unlabeled attachment score can improve when training on a fine-grained annotation.

In the second experiment, we found out that the best configuration for MaltParser and our annotation is [*finiteness gender number spos*]. For parsing purposes, then, it seems enough to enrich the morpho-syntactic annotation just with these features, at least in the case of Spanish. These features not only work well together, but they also very often improve the results when individually added to any combination of features. On the one hand, there is an almost perfect correlation between feature frequency and performance: those features that appear most frequently are the ones that provide best performance. On the other hand, we have observed that the interaction between features also influences significantly the results. So, in order to get the highest performance, frequency and linguistic knowledge should be both taken into account. However, it is important to see how features combine in practice, because when we look at how each feature makes the LAS improve individually, there is no way to say which combination is going to work the best. Another interesting conclusion is that it seems like separating the Part-of-Speech of prepositions and conjunctions has an important impact on the dependency parsing results, at least in the conditions of our experiments.

We believe that with this experiment opens many perspectives for further experiments. For instance, studying whether different levels of dependency relation granularity are affected by the combination of several features. It would also be interesting to study in depth the effect of different feature combinations for specific dependency relations, taking into account that the results for a specific dependency relation are deeply affected by the others that are interacting at the same time. For this, an automatic analysis of the results could allow for reaching conclusions that seem out of reach for the human eye.

A question that remains open is how to compare the effect of different morphological features on dependency parsing of different languages. It would be worth trying to create new CoNLL columns in the data format, one for each feature, and generate new feature models; we are actually doing a similar thing with the *split* MaltParser feature specification of the FEATS column, but we think that the features could be explored by the parser in a different way.[19] Finally, it would also interesting to try other parsers that use different parsing strategies, such as graph-based parsing (e.g., (McDonald et al., 2005)), other transition-based parsers (e.g., (Zhang and Clark, 2008a; Zhang and Nivre, 2011; Bohnet and Nivre, 2012)), joint systems (e.g., (Bohnet and Kuhn, 2012)) or even study the effect of the features in different algorithms included in MaltParser.

Finally, in our last experiment we have presented a novel deep-syntactic parsing pipeline which consists of a state-of-the-art dependency parser and a SSyntS–DSyntS transducer; the approach has been tested on two languages, namely Spanish and Chinese. The obtained DSyntSs can be used in different applications since they abstract from language-specific grammatical idiosyncrasies of the SSynt structures as produced by state-of-the art dependency parsers, but still avoid the complexities of genuine semantic analysis. DSyntS-treebanks needed for data-driven applications can be bootstrapped by the pipeline. If required, a SSyntS–DSyntS structure pair can be also mapped to a pure predicate-argument graph such as the DELPH-IN structure (Oepen, 2002) or to an approximation thereof (as the Enju conversion (Miyao, 2006), which keeps functional nodes), to an DRS (Kamp and Reyle, 1993) or to a PropBank structure.

One important limitation of our SSyntS–DSyntS mapping is that we do not perform lexical disambiguation, since it is not annotated in our corpus. It is clear that in order to provide good quality abstract structures, disambiguating the training data will be unavoidable. But this experiment also opens perspectives as for in-depth feature engineering for the task of DSyntS-parsing, which proved to be crucial in semantic role labeling and dependency parsing (Che et al., 2009; Ballesteros and Nivre, 2012b); we expect it be essential for this task as well. Furthermore, joining surface syntactic and deep-syntactic parsing we kept so far separate would the next natural step; see, e.g., (Zhang and Clark, 2008b; Lluís et al., 2013; Bohnet and Nivre, 2012) for analogous proposals. Further research is required here

---

[19] We did not do it for these experiments because this would make the use of the current version of MaltOptimizer impossible.

since although joined models avoid error propagation from the first stage to the second, overall, pipelined models still proved to be competitive; cf. the outcome of CoNLL shared tasks.

Chapter 6

Conclusions

In this Chapter, we present a summary and the final conclusions of the thesis. First, we summarize the contributions of our work (Section 6.1), then its limitations (Section 6.2), and finally the perspectives that it opens (Section 6.3).

6.1 Contributions of the thesis

The **first two contributions** of the thesis concern the methodology for multilevel corpus annotation and its application to a medium-size corpus of Spanish (100.892 tokens, 3.513 sentences).

In Chapter 3, we report on the elaboration of a methodology for annotating with good quality a Spanish corpus which separates morphological, surface-syntactic, deep-syntactic and semantic information, following the basic principles of the Meaning-Text Theory. We defined simple and easy-to-use syntactic criteria for the surface-syntactic annotation, completed by more semantics-oriented criteria which allow for automatically deriving the deeper layers, thanks to graph transduction grammars. We show that thanks to a sound theoretical framework and appropriate tools, it is possible to reduce the manual workload and, at the same time, achieve a high inter-annotator agreement rate on all evaluated levels (more than 92% for syntax and more than 89% for syntax and semantics). As shown in Chapter 5 on parsing, the corpus will make possible the development of tools for, e.g., NLG, parsing, and Machine Translation. Since the annotation is strongly linguistically motivated, it can also serve as educational material for Spanish learners. Furthermore, the developed annotation scheme is general and thus is easily applicable to other languages: the same methodology has been developed for the construction of a Finnish treebank, and following the same philosophy of automatic derivation of annotations at different layers, the annotation of the PropBank and the Chinese Dependency Treebank have been adapted for the needs of generation.

The following list displays the articles on corpus annotation published in conference proceedings, journals and books.

- Simon Mille, A Burga, V Vidal, and Leo Wanner. Creating an MTT treebank of Spanish. In *Proceedings of the 4th International Conference on Meaning-Text Theory (MTT)*, pages 287–298, Montreal, Canada, 2009a
- Simon Mille, Leo Wanner, Vanesa Vidal, and Alicia Burga. Towards a rich dependency annotation of Spanish corpora. *Procesamiento del Lenguaje Natural*, 43:325–333, 2009b
- Simon Mille and Leo Wanner. Syntactic dependencies for multilingual and multilevel corpus annotation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1889–1896, Valletta, Malta, 2010
- Alicia Burga, Simon Mille, and Leo Wanner. Looking behind the scenes of syntactic dependency corpus annotation: Towards a motivated annotation schema of surface-syntax in Spanish. In *Proceedings of the 1st International Conference on Dependency Linguistics (DepLing)*, pages 104–114, Barcelona, Spain, 2011
- Leo Wanner, Simon Mille, and Bernd Bohnet. Towards a surface realization-oriented corpus annotation. In *Proceedings of the 7th International Natural Language Generation Conference (INLG)*, pages 22–30, Utica, IL, USA, 2012
- Anja Belz, Bernd Bohnet, Simon Mille, Leo Wanner, and Michael White. The Surface Realisation Task: Recent developments and future plans. In *Proceedings of the 7th International Natural Language Generation Conference (INLG)*, pages 136–140, Utica, IL, USA, 2012
- Simon Mille, Leo Wanner, and Alicia Burga. Treebank annotation in the light of the Meaning-Text Theory. *Linguistic Issues in Language Technology*, 7: 1–12, 2012b
- Simon Mille, Alicia Burga, and Leo Wanner. AnCora-UPF: A multi-level annotation of Spanish. In *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pages 217–226, Prague, Czech Republic, 2013
- Alicia Burga, Simon Mille, and Leo Wanner. Looking behind the scenes of syntactic dependency corpus annotation: Towards a motivated annotation

schema of surface-syntax in Spanish. In *Computational Dependency Theory. Frontiers in Artificial Intelligence and Applications Series*, volume 258, pages 26–46. Amsterdam:IOS Press, 2014 (Long version)

The **third contribution** of the thesis concerns the development of deep stochastic text generators. Despite the increasing amount of work on statistical sentence generation, no one had addressed so far the problem of deep generation from semantic structures that are not isomorphic to syntactic structures as a stochastic problem. Thanks to our multilevel annotation, it has been possible to train for the first time classifiers which achieve this task without resorting to rules for any mapping. We show that the corpus developed in the framework of this thesis is perfectly suitable for training a deep-stochastic generator. In spite of the modest size of the corpus, classifiers trained on it manage to perform a very challenging task that no other classifiers had been able to achieve up to now: decide, in the course of generation from abstract structures, when nodes should be introduced and which ones. Other prominent features of the presented generators are that they are *per se* multilingual, and that they achieve a very broad coverage. The fact that we start from abstract structures allows us to cover a number of critical generation issues: sentence planning, linearization and morphological generation. The separation of the semantic, syntactic, linearization and morphological levels of annotation and their modular processing by separate SVM decoders does not only lead to good results, it also facilitates a subsequent integration of other generation tasks such as referring expression generation, ellipsis generation, and aggregation. Getting closer to large-coverage sentence generation from abstract structures will definitely benefit NLG in general as it will make it more usable, and reduce the gap between the popularity of state-of-the-art parsers and generators.

The following list displays the articles on deep stochastic generation published in conference proceedings and books.

- Bernd Bohnet, Leo Wanner, Simon Mille, and Alicia Burga. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 98–106, Beijing, China, 2010
- Bernd Bohnet, Simon Mille, and Leo Wanner. Statistical language generation from semantic structures. In *Proceedings of the 1st International Conference on Dependency Linguistics (DepLing)*, pages 251–261, Barcelona, Spain, 2011b

- Bernd Bohnet, Simon Mille, Benoît Favre, and Leo Wanner. StuMaBa: From deep representation to surface. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 232–235, Nancy, France, 2011a
- Bernd Bohnet, Simon Mille, and Leo Wanner. One step further towards stochastic semantic sentence generation. In *Computational Dependency Theory. Frontiers in Artificial Intelligence and Applications Series*, volume 258, pages 93–112. Amsterdam:IOS Press, 2014 (Long version)
- Miguel Ballesteros, Simon Mille, and Leo Wanner. Classifiers for data-driven deep sentence generation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, Philadelphia, PA, USA, 2014b

The **final contribution** of the thesis is to show that if obtaining NLG-suited resources from "classic" data annotated with analysis in mind is challenging, the opposite is not true. In our last chapter, we show that it is possible to use our data as it is for training statistical surface and deep syntactic parsers, and to perform a variety of experiments in this field. First of all, we revealed that there is a correlation between the granularity of the surface-syntactic annotation scheme and the accuracy of dependency parsers; we also showed a correlation between the quality of annotation and the accuracy of these parsers. Second, our rich morphological annotation has been used to investigate which features and combinations of features help improving surface-syntactic dependency parsers. Finally, we used the multilevel annotation to make experiments on deep-syntactic parsing and evaluated a Sentence-DSyntS pipeline which produces fully connected predicate–argument structures for any input sentence. Thanks to our experiments with a Chinese treebank, we also showed that such a system is easily portable to other languages.

The following list displays the articles on dependency parsing published in conference proceedings.

- Simon Mille, Alicia Burga, Gabriela Ferraro, and Leo Wanner. How does the granularity of an annotation scheme influence dependency parsing performance? In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 839–852, Mumbai, India, 2012a
- Miguel Ballesteros, Simon Mille, and Alicia Burga. Exploring morphosyntactic annotation over a Spanish corpus for dependency parsing. In *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pages 13–22, Prague, Czech Republic, 2013

- Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. Deep-syntactic parsing. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, 2014a

Annotated data, resources developed during the annotation (guidelines, software, etc.), stochastic realizers and parsers are available to the community; they are downloadable at `http://www.recerca.upf.edu/taln`.

6.2 Limitations

Despite the contributions depicted above, this thesis obviously also has its limitations due to some aspects that could not be handled during the time frame set for a PhD dissertation.

As far as corpus annotation is concerned, the major limitation is that we focused on the relations between words, and not between the lexical units. We did not disambiguate words in the corpus, and even though this did not imply any significant problem for our experiments, we believe that a disambiguated version of the data would significantly improve its value and increase the performance of the systems trained on it. At the deep-syntactic level, we did not annotate lexical functions, neither did we annotate manually the communicative structure (Mel'čuk, 2001) on top of the semantic structures.

The main limitation of our work on deep stochastic NLG is that we did not tackle one-step generation from abstract structures. From the perspective of MTT, a linguistically sound strategy consists in performing transitions between the different levels of abstraction, one at a time, and this is what has been done in our experiments. In the parsing world, there is sometimes a preference for one-step extraction of abstract relations: it turned out that in experimental settings, one-step approaches can perform better than multiple-step approaches, due to the fact that the errors made in one step propagate to the subsequent steps. But for generation we cannot confirm or refute that the MTT approach is the right one to follow and be it only for achieving higher scores.

Finally, the parsing experiments have been very informative, but we realized that many more of these experiments would be needed to draw solid conclusions, be it with the surface-syntactic tag granularity or with morpho-syntactic features (and the combination of both).

6.3 Future work

Some of the above mentioned limitations will be addressed in the future. Our work has also opened a broad range of other tasks that can be tackled next:

- Disambiguate the lexical units of the corpus: we already started working on the recovery of sense IDs from the original AnCora annotation and lexicons.
- Annotate MTT's lexical functions on the deep-syntactic layer: this task has already been started, with 3 classes of lexical functions (*Oper*, *Func*, *Caus*) on 1,500 sentences.
- Annotate the communicative structure on the semantic layer: we performed experiments on annotating it automatically from the DSyntS annotation; this work will continue on a broader scale.
- Adapt the surface-syntactic annotation scheme to other languages: we did it on Finnish already (2,000 sentences annotated with about 50 relations at the surface-syntactic layer). Further languages we are about to address are French, German and Bulgarian.
- Test alternative ways of annotating deep layers: on Finnish, government pattern dictionaries of all the predicates have been manually built, so the annotation of DSyntS can be fully non-supervised.
- Annotate automatically the deep layers of other languages: we made a deep version of English and Chinese data with the help of native speakers; the same work on French and German is planned.
- Implement a generator that performs the DSyntS–Sentence transition in one step in order to compare it with the multilevel generators presented in Chapter 4.
- Make experiments on Machine Translation, by aligning DSyntSs for different languages, learning how derive a DSyntS of a language from a DSyntS of another languages, and training deep parsers and generators on the multilevel data.
- Test the impact of the different combinations of tag granularity and morpho-syntactic features on the results of different types of dependency parsers.
- Work on a new method for the evaluation of dependency parsers, based on adapting the penalties to the distance between predicted and gold dependency relations in a hierarchical scheme.

Bibliography

Each reference indicates the pages where it appears.

Anne Abeillé and Nicolas Barrier. Enriching a French treebank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 2233–2236, Lisbon, Portugal, 2004. 38

Anne Abeillé, Lionel Clément, and François Toussenel. Building a treebank for French. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 165–187. Kluwer, 2003. 14, 38

Itziar Aduriz Agirre, Alicia María Ageno Pulido, Bertol Arrieta Cortajarena, José María Arriola Egurrola, Empar Bisbal Asensi, Nuria Castell Ariño, Montserrat Civit Torruella, Arantza Díaz de Ilarraza Sánchez, B Fernández, Koldobika Gojenola Galletebeitia, et al. 3lb: Construcción de una base de datos de árboles sintáctico semánticos. *Procesamiento del Lenguaje Natural*, 31:297–298, 2003. 14

Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos. Floresta Sintá (c) tica: A treebank for Portuguese. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 1698–1703, Las Palmas, Canary Islands, Spain, 2002. 14

E. Agirre, K. Bengoetxea, K. Gojenola, and J. Nivre. Improving dependency parsing with semantic classes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 699–703, Portland, Oregon, USA, 2011. 197

Juri Apresjan, Igor Boguslavsky, Boris Iomdin, Leonid Iomdin, Andrei Sannikov, and Victor Sizov. A syntactically and semantically tagged corpus of Russian: State of the art and prospects. In *Proceedings of the 5th In-*

ternational Conference on Language Resources and Evaluation (LREC), pages 1378–1381, Genoa, Italy, 2006. 14

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. 109

Jordi Atserias, Bernardino Casas, Elisabet Comelles, Meritxell Gonzlez, Llus Padr, and Muntsa Padr. FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2281–2286, Genoa, Italy, 2006. 68

Aitziber Atutxa, Eneko Agirre, and Kepa Sarasola. Contribution of complex lexical information to solve syntactic ambiguity in Basque. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 97–113, Mumbai, India, 2012. 196

Olga Babko-Malaya. *Propbank Annotation Guidelines*, 2005. URL `http://verbs.colorado.edu/~mpalmer/projects/ace/PBguidelines.pdf`. 46

M. Ballesteros and J. Nivre. MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of the System Demonstration Session of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 58–62, Avignon, France, 2012a. 195, 201

M. Ballesteros and J. Nivre. MaltOptimizer: A system for MaltParser optimization. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 2757–2763, Istambul, Turkey, 2012b. 195, 201, 202, 226

Miguel Ballesteros, Simon Mille, and Alicia Burga. Exploring morphosyntactic annotation over a Spanish corpus for dependency parsing. In *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pages 13–22, Prague, Czech Republic, 2013.

Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. Deep-syntactic parsing. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, 2014a.

Miguel Ballesteros, Simon Mille, and Leo Wanner. Classifiers for data-driven deep sentence generation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, Philadelphia, PA, USA, 2014b. 140

David Bamman and Gregory Crane. The design and use of a Latin dependency treebank. In *Proceedings of the 5th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 67–78, Prague, Czech Republic, 2006.

14

David Bamman, Francesco Mambrini, and Gregory Crane. An ownership model of annotation: The Ancient Greek dependency treebank. In *Proceedings of the 8th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 5–15, Milan, Italy, December 2009. 14

Srinivas Bangalore and Owen Rambow. Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 42–48, Saarbrücken, Germany, 2000. 11, 19

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. Developing a large semantically annotated corpus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 3196–3200, Istambul, Turkey, 2012. 52

J.A. Bateman, I. Kruijff-Korbayová, and G.-J. Kruijff. Multilingual resource sharing across both related and unrelated languages: An implemented, open-source framework for practical Natural Language Generation. *Research on Language and Computation*, 3(2-3):191–219, 2005. 12

John A Bateman. Enabling technology for multilingual Natural Language Generation: The KPML development environment. *Natural Language Engineering*, 3(1):15–55, 1997. 11

Anja Belz. Statistical generation: Three methods compared and evaluated. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG)*, pages 15–23, Aberdeen, Scotland, 2005. 11, 20

Anja Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Journal of Natural Language Engineering*, 14(4):431–455, 2008. 12, 20

Anja Belz, Mike White, Josef Van Genabith, Deirdre Hogan, and Amanda Stent. Finding common ground: Towards a Surface Realisation Shared Task. In *Proceedings of the 6th International Natural Language Generation Conference (INLG)*, pages 268–272, Dublin, Ireland, 2010. 112

Anja Belz, Mike White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. The first Surface Realisation Shared Task: Overview and evaluation results. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 217–226, Nancy, France, 2011. 1, 15, 32, 43, 109, 112, 177

Anja Belz, Bernd Bohnet, Simon Mille, Leo Wanner, and Michael White. The Surface Realisation Task: Recent developments and future plans. In *Proceedings of the 7th International Natural Language Generation Con-*

ference (INLG), pages 136–140, Utica, IL, USA, 2012. 2, 43

K. Bengoetxea and K. Gojenola. Application of feature propagation to dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 142–145, Paris, France, 2009. 197

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. A maximum entropy approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996. 7

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of the 3rd Linguistic Annotation Workshop*, pages 186–189, Suntec, Singapore, 2009. 14

Eckhard Bick, Heli Uibo, and Kaili Müürisep. Arborest–a VISL style treebank derived from an Estonian constraint grammar corpus. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 1–14, Tübingen, Germany, 2004. 14

A. Björkelund, B. Bohnet, L. Hafdell, and P. Nugues. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics : Demonstration Volume (COLING)*, pages 33–36, Beijing, China, 2010. 153

B. Bohnet. A graph grammar approach to map between dependency trees and topological models. In *Proceedings of the 1st International Joint Conference on Natural Language Processing (IJCNLP)*, pages 636–645, Hainan Island, China, 2004. 163

B. Bohnet and J. Kuhn. The best of both worlds - a graph-based completion model for transition-based parsers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 77–87, Avignon, France, 2012. 226

B. Bohnet and J. Nivre. A transition-based system for joint Part-of-Speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1455–1465, Jeju Island, Korea, 2012. 217, 226

Bernd Bohnet. Mapping phrase structures to dependency structures in the case of (partially) free word order languages. In *Proceedings of the 1st International Conference on Meaning-Text Theory (MTT)*, pages 239–249, Paris, France, 2003. 69

Bernd Bohnet. A graph grammar approach to map between dependency trees and topological models. In *Natural Language Processing–IJCNLP 2004*, pages 636–645. Springer, 2005. 21

Bernd Bohnet. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 67–72, Boulder, CO, USA, 2009. 184

Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 89–97, Beijing, China, 2010. 111

Bernd Bohnet and Halyna Seniv. Mapping dependency structures to phrase structures and the automatic acquisition of mapping rules. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 855–858, Lisbon, Portugal, 2004. 53

Bernd Bohnet, Andreas Langjahr, and Leo Wanner. A development environment for an MTT-based sentence generator. In *Proceedings of the 1st International Natural Language Generation Conference (INLG)*, pages 260–263, Mitzpe Ramon, Israel, 2000. 94, 101

Bernd Bohnet, Leo Wanner, Simon Mille, and Alicia Burga. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 98–106, Beijing, China, 2010. 11, 156

Bernd Bohnet, Simon Mille, Benoît Favre, and Leo Wanner. StuMaBa: From deep representation to surface. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 232–235, Nancy, France, 2011a. 165

Bernd Bohnet, Simon Mille, and Leo Wanner. Statistical language generation from semantic structures. In *Proceedings of the 1st International Conference on Dependency Linguistics (DepLing)*, pages 251–261, Barcelona, Spain, 2011b. 140, 165

Bernd Bohnet, Alicia Burga, and Leo Wanner. Towards the annotation of Penn TreeBank with information structure. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1250–1256, Nagoya, Japan, 2013. 50

Bernd Bohnet, Simon Mille, and Leo Wanner. One step further towards stochastic semantic sentence generation. In *Computational Dependency Theory. Frontiers in Artificial Intelligence and Applications Series*, volume 258, pages 93–112. Amsterdam:IOS Press, 2014. 165

Johan Bos, Edward Briscoe, Aoife Cahill, John Carroll, Stephen Clark, Ann Copestake, Dan Flickinger, Josef van Genabith, Julia Hockenmaier, Aravind Joshi, Ronald Kaplan, Tracy Holloway King, Sandra Kubler, Dekang Lin, Jan Tore Lnning, Christopher Manning, Yusuke Miyao, Joakim Nivre, Stephan Oepen, Kenji Sagae, Nianwen Xue, and Yi Zhang (Eds.). Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, 2008. 43

Johan Bos, Cristina Bosco, and Alessandro Mazzei. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Proceedings of the 8th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 27–38, Milan, Italy, December 2009. 52

C. Bosco and A. Lavelli. Annotation schema oriented evaluation for parsing validation. In *Proceedings of the 9th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 19–30, Tartu, Estonia, 2010. 181, 182, 183

C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, F. Dell'Orletta, A. Lenci, L. Lesmo, G. Attardi, M. Simi, A. Lavelli, J. Hall, J. Nilsson, and J. Nivre. Comparing the influence of different treebank annotations on dependency parsing. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1794–1801, Valletta, Malta, 2010. 181, 182, 194

Cristina Bosco. Multiple-step treebank conversion: From dependency to Penn format. In *Proceedings of the 1st Linguistic Annotation Workshop*, pages 164–167, Prague, Czech Republic, 2007. 52

Cristina Bosco, Vincenzo Lombardo, Daniela Vassallo, and Leonardo Lesmo. Building a treebank for Italian: A data-driven annotation schema. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, pages 99–105, Athens, Greece, 2000. 183

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, Marco Rospocher, Horacio Saggion, Luciano Serafini, and Leo Wanner. From ontology to NL: Generation of multilingual user-oriented environmental reports. In *Natural Language Processing and Information Systems*, pages 216–221. Springer, 2012a. 7

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, Marco Rospocher, Horacio Saggion, Luciano Serafini, and Leo Wanner. From Ontology to NL: Generation of multilingual user-oriented environmental reports. In

17th International conference on Applications of Natural Language Processing to Information Systems (NLDB), pages 216–221, Groningen, The Netherlands, 2012b. 62

Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, and Leo Wanner. Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Transactions on Speech and Language Processing*, 9(2):3:1–3:31, 2012c. 7, 62

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597–620, 2004. 14

Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. Syntactic annotation of a German newspaper corpus. In *Treebanks*, pages 73–87. Springer, 2003. 69

Joan Bresnan. *Lexical-Functional Syntax*. Blackwell Oxford, 2001. 23, 36

Ted Briscoe, J. Carroll, J. Graham, and A. Copestake. Relational evaluation schemes. In *Proceedings of the Workshop on Beyond PARSEVAL: Towards Improved Evaluation Measures for Parsing Systems, 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 4–6, Gran Canaria, Spain, 2002. 183

Norbert Bröker. Separating Surface Order and Syntactic Relations in a Dependency Grammar. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL)*, pages 174–180, Montreal, Canada, 1998. 163

Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164, New-York, NY, USA, 2006. 1, 32, 184, 199, 212

Alicia Burga, Simon Mille, and Leo Wanner. Looking behind the scenes of syntactic dependency corpus annotation: Towards a motivated annotation schema of surface-syntax in Spanish. In *Proceedings of the 1st International Conference on Dependency Linguistics (DepLing)*, pages 104–114, Barcelona, Spain, 2011.

Alicia Burga, Simon Mille, and Leo Wanner. Looking behind the scenes of syntactic dependency corpus annotation: Towards a motivated annotation schema of surface-syntax in Spanish. In *Computational Dependency Theory. Frontiers in Artificial Intelligence and Applications Series*, volume 258, pages 26–46. Amsterdam:IOS Press, 2014.

Stephan Busemann and Helmut Horacek. A flexible shallow approach to text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation (INLG)*, pages 238–247, Niagara-on-the-Lake, Canada, 1998. 8, 9

Aoife Cahill and Josef Van Genabith. Robust PCFG-based generation using automatically acquired LFG approximations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 1033–1040, Sydney, Australia, 2006. 21

Mihaela Călăcean. Data-driven dependency parsing for Romanian. Master's thesis, Uppsala University, 2008. 14

Marie Candito and Djamé Seddah. Le corpus Sequoia: Annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Proceedings of 19e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 331–334, Grenoble, France, 2012. 37

Marie Candito, Benoît Crabbé, Pascal Denis, et al. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1840–1847, Valletta, Malta, 2010. 38

Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, Éric De La Clergerie, et al. Deep syntax annotation of the Sequoia French treebank. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland, 2014. 39

Xavier Carreras, Michael Collins, and Terry Koo. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL)*, pages 9–16, Manchester, UK, 2008. 153

John Carroll, Ted Briscoe, and Antonio Sanfilippo. Parser evaluation: A survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*, pages 447–454, Granada, Spain, 1998. 36

Atanas Chanev, Kiril Simov, Petya Osenova, and Svetoslav Marinov. Dependency conversion and parsing of the BulTreeBank. In *Proceedings of the Merging and Layering Linguistic Information Workshop, 5th International Conference on Language Resources and Evaluation (LREC)*, pages 17–24, Genoa, Italy, 2006. 14

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm. 218

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3), Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL HLT)*, pages 51–59, Boulder, CO, USA, 2009. 14, 222

Wanxiang Che, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. Multilingual dependency-based syntactic and semantic parsing. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 49–54, Boulder, CO, USA, 2009. 184, 212, 213, 226

John Chen, Srinivas Bangalore, Owen Rambow, and Marilyn A Walker. Towards automatic generation of Natural Language Generation systems. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 1–7, Taipei, Taiwan, 2002. 19

Keh-Jiann Chen, C Luo, M Chang, F Chen, C Chen, C Huang, and Zhao-Ming Gao. Sinica treebank: Design criteria, representational issues and implementation. In (Anne Abeillé), editor, *Treebanks: Building and Using Parsed Corpora*, pages 231–248. Kluwer, 2003. 14

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965. 13

Noam Chomsky. *Lectures on Government and Binding: The Pisa lectures*, volume 9. Walter de Gruyter, 1993. 45

Montserrat Civit and Ma Antònia Martí. Building Cast3LB: A Spanish treebank. *Research on Language and Computation*, 2(4):549–574, 2004. 69

M. Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, Philadelphia, PA, USA, 2002. 168

M. Collins and B. Roark. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 111–118, Barcelona, Spain, 2004. 169

Bernard Comrie. *Aspect*. Cambridge University Press, Cambridge, 1976. 63

A. Copestake, D. Flickinger, and I. Sag. Minimal Recursion Semantics. Technical report, CSLI, Stanford University, Stanford, 1997. 129

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332, 2005. 42

Simon Corston-Oliver, Michael Gamon, Eric Ringger, and Robert Moore. An overview of Amalgam: A machine-learned generation module. In *Proceedings of the 2nd International Natural Language Generation Conference (INLG)*, pages 33–40, New-York, NY, USA, 2002. 11, 22

K. Crammer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006. 146, 168

Marie-Catherine de Marneffe and Christopher D. Manning. The Stanford typed dependencies representation. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, 22nd International Conference on Computational Linguistics (COLING)*, pages 1–8, Manchester, UK, 2008. 36, 37, 87, 141

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 449–454, Genoa, Italy, 2006. 36, 37, 87, 183

Ferdinand De Saussure. *Cours de linguistique générale*, volume 1. Otto Harrassowitz Verlag, 1989. 13

José Deulofeu, Lucie Duffort, Kim Gerdes, Sylvain Kahane, and Paola Pietrandrea. Depends on what the French say: Spoken corpus annotation with and beyond syntactic functions. In *Proceedings of the 4th Linguistic Annotation Workshop*, pages 274–281, Uppsala, Sweden, 2010. 14

Christy Doran, Dania Egedi, Beth Ann Hockey, Bangalore Srinivas, and Martin Zaidel. XTAG system: A wide coverage grammar for English. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*, pages 922–928, Kyoto, Japan, 1994. 19

D. Duchier and R. Debusmann. Topological dependency trees: A constraint-based account of linear precedence. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 180–187, Toulouse, France, 2001. 163

Kais Dukes, Eric Atwell, and Abdul-Baquee M Sharaf. Syntactic annotation guidelines for the Quranic Arabic Dependency Treebank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1822–1827, Valletta, Malta, 2010. 14

Sašo Džeroski, Tomaž Erjavec, Nina Ledinek, Petr Pajas, Zdenek Žabokrtsky, and Andreja Žele. Towards a Slovene dependency treebank. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1388–1391, Genova, Italy, 2006. 14

J. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 340–345, Copenhagen, Denmark, 1996. 169

Michael Elhadad and Jacques Robin. An overview of SURGE: A reusable comprehensive syntactic realization component. *Technical report 96-03, Dept of Mathematics and Computer Science, Ben Gurion University, Beer Shiva, Israel*, 1996. 11, 12

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389, 2008. 197, 206

K. Filippova and M. Strube. Tree linearization in English: Improving language model based approaches. In *Proceedings of the 2009 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, pages 225–228, Boulder, CO, USA, 2009. 161, 162, 163, 175, 176

Katja Filippova and Michael Strube. Generating constituent order in German clauses. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 45, pages 320–327, Prague, Czech Republic, 2007. 23

Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000. 41

Karen Fort. *Les ressources annotées, un enjeu pour l'analyse de contenu: Vers une méthodologie de l'annotation manuelle de corpus.* PhD thesis, Université Paris-Nord-Paris XIII, 2012. 93, 109

Haim Gaifman. Dependency systems and phrase-structure systems. *Information and control*, 8(3):304–337, 1965. 52

K. Gerdes and S. Kahane. Word order in German: A formal dependency grammar using a topological hierarchy. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 220–227, Toulouse, France, 2001. 163

Pablo Gervás. UCM submission to the Surface Realization Challenge.

In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 239–241, Nancy, France, 2011. 11

Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 37–42, Boulder, CO, USA, 2009. 184, 213

Yoav Goldberg and Michael Elhadad. Hebrew dependency parsing: Initial results. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 129–133, Paris, France, 2009. 14

J. Gundel, N. Hedberg, and R. Zacharski. Giveness, implicature and demonstrative expressions in English discourse. *Chicago Linguistics Society (Parasession on Language in Context)*, 25(2):89–103, 1989. 131

Jeanette Gundel. Universals of topic-comment structure. In Edith A. Moravcsik Michael Hammond and Jessica Wirth, editors, *Studies in Syntactic Typology*, pages 209–239. John Benjamins, Amsterdam, 1988. 129

Yuqing Guo, Deirdre Hogan, and Josef van Genabith. DCU at Generation Challenges 2011 Surface Realisation Track. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 227–229, Nancy, France, 2011a. 23

Yuqing Guo, Haifeng Wang, and Josef Van Genabith. Dependency-based n-gram models for general purpose sentence realisation. *Natural Language Engineering*, 17(04):455–483, 2011b. 23

Nizar Habash. The use of a structural n-gram language model in generation-heavy hybrid machine translation. In *Natural Language Generation*, pages 61–69. Springer, 2004. 19

Jan Hajič. Complex corpus annotation: The Prague Dependency Treebank. *Insight into the Slovak and Czech Corpus Linguistics*, pages 54–73, 2005. 14, 26

Jan Hajic, Otakar Smrz, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. Prague Arabic Dependency Treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117, Cairo, Egypt, 2004. 14

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang. The CoNLL-2009 Shared Task:

Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 1–18, Boulder, CO, USA, 2009. 1, 29, 32, 109, 156, 184, 212

Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, and Alla Bémová. A manual for analytic layer tagging of the Prague Dependency Treebank. CDROM CAT: LDC2001T10., ISBN 1-58563-212-0, 2001. URL `http://shadow.ms.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/aman_en.pdf`. English translation of the original Czech version. 28

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, and Zdenĕk Žabokrtský. Prague Dependency Treebank 2.0. Linguistic Data Consortium, Philadelphia, 2006. 14, 26

E. Hajíčova. The position of TFA (information structure) in a dependency based description of language. In *Proceedings of the 3rd International Conference on Meaning-Text Theory (MTT)*, pages 159–178, Klagenfurt, Austria, 2007. 129

M.A.K. Halliday. *An Introduction to Functional Grammar*. Edward Arnold, London, 1994. 129

Katri Haverinen, Filip Ginter, Veronika Laippala, Timo Viljanen, and Tapio Salakoski. Dependency annotation of Wikipedia: First steps towards a Finnish treebank. In *Proceedings of the 8th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 95–105, Milan, Italy, December 2009. 14

Wei He, Haifeng Wang, Yuqing Guo, and Ting Liu. Dependency based Chinese sentence realization. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 809–816, Singapore, 2009. 23, 162, 163

Jesús Herrera, Pablo Gervás, Pedro J Moriano, A Moreno, and Luis Romero. JBeaver: Un analizador de dependencias para el Español basado en aprendizaje. *Procesamiento del Lenguaje Natural*, 39:285–286, 2007a. 68

Jesús Herrera, Pablo Gervás, Pedro J Moriano, Alfonso Moreno, and Luis Romero. Building corpora for the development of a dependency parser for Spanish using MaltParser. *Procesamiento del Lenguaje Natural*, 39: 181–186, 2007b. 69

Deirdre Hogan, Conor Cafferkey, Aoife Cahill, and Josef Van Genabith. Exploiting multi-word units in history-based probabilistic generation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 267–276, Prague, Czech Republic, 2007. 21

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: The 90% solution. In *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 879–884, New York, NY, USA, 2006. 110, 111

Chu-Ren Huang, Feng-Yi Chen, Keh-Jiann Chen, Zhao-ming Gao, and Kuang-Yu Chen. Sinica Treebank: Design criteria, annotation guidelines, and on-line interface. In *Proceedings of the second workshop on Chinese language processing, 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 29–37, Hong Kong, China, 2000. 57

Lidija Iordanskaja and Igor Mel'čuk. Towards establishing an inventory of surface-syntactic relations: Valency-controled surface-syntactic dependents of verb in French. In Polguère & Mel'čuk, editor, *Dependency in Linguistic Description*, pages 151–234. John Benjamins Publishing Company, 2009. 26, 74

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea, 2012. 41, 43, 52

Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for English. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 105–112, Tartu, Estonia, 2007. 14, 29, 52, 69, 212

Richard Johansson and Pierre Nugues. Dependency-based Semantic Role Labeling of PropBank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 69–78, Honolulu, Hawaii, 2008a. 153

Richard Johansson and Pierre Nugues. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL)*, pages 183–187, Manchester, UK, 2008b. 212, 213

Aravind K Joshi, K Vijay Shanker, and David Weir. The convergence of mildly context-sensitive grammar formalisms. *Foundational Issues in Natural Language Processing*, pages 31–81, 1991. 23

Sylvain Kahane. The Meaning-Text Theory. In *Dependency and Valency. An International Handbook on Contemporary Research*, pages 546–570. De Gruyter, 2003. 53

Tuomo Kakkonen. Dependency treebanks: Methods, annotation schemes and tools. In *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 94–104, Joensuu, Finland, 2005. 68

Hans Kamp and Uwe Reyle. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory.* Springer, 1993. 52, 226

Min-Yen Kan and Kathleen McKeown. Corpus-trained text generation for summarization. In *Proceedings of the 2nd International Natural Language Generation Conference (INLG)*, pages 1–8, New-York, NY, USA, 2002. 21

Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. The PARC 700 dependency bank. In *Proceedings of the 4th international Workshop on Linguistically Interpreted Corpora (LINC), 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–8, Budapest, Hungary, 2003. 36

R. Kittredge. Paraphrasing for condensation in journal abstracting. *Journal of Biomedical Informatics*, 35(4):265–277, 2002. 212

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430, Sapporo, Japan, 2003. 183, 211

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics,Speech and Signal Processing (ICML)*, pages 181–184, Detroit, MI, USA, 1995. 178

K. Knight and J. Graehl. An overview of probabilistic tree transducers for Natural Language Processing. In *Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 1–24, Mexico City, Mexico, 2005. 171

Kevin Knight and Vasileios Hatzivassiloglou. Two-level, many-paths generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 252–260, Cambridge, MA, USA,

1995. 17, 18

Matthias Trautner Kromann. The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 217–220, Växjö, Sweden, 2003. 14

Sandra Kübler. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 293–300, Borovets, Bulgaria, 2005. 181, 182

Sadao Kurohashi and Makoto Nagao. Building a Japanese parsed corpus. In *Treebanks*, pages 249–260. Springer, 2003. 14

K. Lambrecht. *Information structure and sentence form: Topic, focus, and the mental representation of discourse referents.* Cambridge University Press, Cambridge, 1994. 130

I. Langkilde and K. Knight. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL)*, pages 704–710, Montreal, Canada, 1998. 11, 18, 19, 20, 159

Irene Langkilde. Forest-based statistical sentence generation. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 170–177, Seattle, WA, USA, 2000. 18

Irene Langkilde-Geary. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the 2nd International Natural Language Generation Conference (INLG)*, pages 17–24, New-York, NY, USA, 2002. Citeseer. 19, 161, 162, 174

Benoit Lavoie and Owen Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, pages 265–268, Washington, DC, USA, 1997. 11, 12, 19

Willem JM Levelt. *Speaking: From Intention to Articulation.* MIT Press, Cambridge, MA, 1989. 6

V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics*, 10:707–710, 1966. 148, 174

Xavier Lluís, Xavier Carreras, and Lluís Màrquez. Joint arc-factored parsing of syntactic and semantic dependencies. *Transactions of the Association for Computational Linguistics (TACL)*, 1:219–230, 2013. URL `http://www.transacl.org/wp-content/uploads/2013/05/`

`paper219.pdf`. 213, 226

François Mairesse and Marilyn A Walker. Trainable generation of big-five personality styles through data-driven parameter estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies (ACL:HLT)*, pages 165–173, Columbus, OH, USA, 2008. 21

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1552–1561, Uppsala, Sweden, 2010. 11, 23, 175

Robert Malouf et al. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Workshop on Computational Language Learning (CoNLL)*, pages 49–55, Taipei, Taiwan, 2002. 20

F. Mambrini and M.C. Passarotti. Will a parser overtake Achilles? First experiments on parsing the Ancient Greek Dependency Treebank. In *Proceedings of the 11th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 133–144, Lisbon, Portugal, December 2012. 196

Tomasz Marciniak and Michael Strube. Classification-based generation using TAG. In *Natural Language Generation*, pages 100–109. Springer, 2004. 22

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2):313–330, 1993. 29

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3. *Linguistic Data Consortium, Philadelphia*, 1999. 14

Montserrat Marimon. The Tibidabo Treebank. *Procesamiento del Lenguaje Natural*, 45:113–119, 2010. 41

R. McDonald and F. Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88, Trento, Italy, 2006. 169

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530, Vancouver, Canada, 2005. 226

Susan W McRoy, Songsak Channarukul, and Syed S Ali. An augmented template-based approach to text realization. *Natural Language Engineering*, 9(4):381–420, 2003. 9

Chris Mellish, Donia Scott, Lynne Cahill, Daniel Paiva, Roger Evans, and Mike Reape. A reference architecture for Natural Language Generation systems. *Natural language engineering*, 12(01):1–34, 2006. 7

Igor Mel'čuk. *Dependency Syntax: Theory and Practice.* State University of New York Press, Albany, 1988. 13, 19, 49, 70, 71, 72, 83, 84

Igor Mel'čuk. Lexical functions: A tool for the description of lexical relations in a lexicon. *Lexical functions in lexicography and natural language processing*, 31:37–102, 1996. 101

Igor Mel'čuk. *Communicative Organization in Natural Language: The Semantic-Communicative Structure of Sentences.* John Benjamins, Philadelphia, 2001. 47, 129, 130, 233

Igor Mel'čuk and Leo Wanner. Syntactic mismatches in machine translation. *Machine Translation*, 20(2):81–138, 2006. 53, 64, 212

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. The NomBank Project: An interim report. In *Proceedings of the Workshop on Frontiers in Corpus Annotation, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 24–31, Boston, MA, USA, 2004. 15, 29

Simon Mille and Leo Wanner. Syntactic dependencies for multilingual and multilevel corpus annotation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1889–1896, Valletta, Malta, 2010.

Simon Mille, A Burga, V Vidal, and Leo Wanner. Creating an MTT treebank of Spanish. In *Proceedings of the 4th International Conference on Meaning-Text Theory (MTT)*, pages 287–298, Montreal, Canada, 2009a.

Simon Mille, Leo Wanner, Vanesa Vidal, and Alicia Burga. Towards a rich dependency annotation of Spanish corpora. *Procesamiento del Lenguaje Natural*, 43:325–333, 2009b.

Simon Mille, Alicia Burga, Gabriela Ferraro, and Leo Wanner. How does the granularity of an annotation scheme influence dependency parsing performance? In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 839–852, Mumbai, India, 2012a.

Simon Mille, Leo Wanner, and Alicia Burga. Treebank annotation in the light of the Meaning-Text Theory. *Linguistic Issues in Language Technology*, 7:1–12, 2012b.

Simon Mille, Alicia Burga, and Leo Wanner. AnCora-UPF: A multi-level annotation of Spanish. In *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pages 217–226, Prague, Czech Republic, 2013.

G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing for English. *Natural Language Engineering*, 7(3):207–223, 2001. 162

Yusuke Miyao. *From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model.* PhD thesis, The University of Tokyo, 2006. 226

Simonetta Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Mana, F. Pianesi, and R. Delmonte. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 189–210. Kluwer, 2003. xv, xvi, 14, 39, 40, 41

Hiroko Nakanishi, Yusuke Miyao, and Jun'ichi Tsujii. Probabilistic models for disambiguation of an HPSG-based chart generator. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT)*, pages 93–102, Vancouver, Canada, 2005. 11, 21

Karthik Sankaran Narayan, Charles Lee Isbell Jr, and David L Roberts. DEXTOR: Reduced effort authoring for template-based Natural Language Generation. In *Proceedings of the 7th Annual International Artificial Intelligence and Interative Digital Entertainment Conference (AIIDE)*, pages 170–175, Palo Alto, CA, USA, 2011. 9

Jens Nilsson, Johan Hall, and Joakim Nivre. MAMBA meets TIGER: Reconstructing a Swedish treebank from Antiquity. In *Proceedings of the 15th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 119–132, Joensuu, Finland, 2005. 14

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 Shared Task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic, 2007a. 199, 212

Joakim Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies*

(IWPT), pages 149–160, Nancy, France, 2003. 202

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007b. xvii, 111, 184, 195, 198, 199

Jon Oberlander and Chris Brew. Stochastic text generation. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1769):1373–1387, 2000. 7

Stephan Oepen. *Collaborative Language Engineering: A Case Study in Efficient Grammar-based Processing.* Stanford Univ Center for the Study, 2002. 41, 226

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D Manning. Lingo Redwoods. *Research on Language and Computation*, 2 (4):575–596, 2004. 41

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. Building a Turkish treebank. In *Treebanks*, pages 261–277. Springer, 2003. 14

Alice H Oh and Alexander I Rudnicky. Stochastic language generation for spoken dialogue systems. In *Proceedings of the ANLP Workshop on Conversational Systems, 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 27–32, Seattle, WA, USA, 2000. 20

Alice H Oh and Alexander I Rudnicky. Stochastic Natural Language Generation for spoken dialog systems. *Computer Speech & Language*, 16(3): 387–407, 2002. 21

Daniel S Paiva and Roger Evans. Empirically-based control of Natural Language Generation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 58–65, Ann Harbour, Michigan, USA, 2005. 21

Martha Palmer, Paul Kingsbury, and Daniel Gildea. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31: 71–105, 2005. 15, 29, 30, 43, 156

Shimei Pan and James Shaw. SEGUE: A hybrid case-based surface natural language generator. In *Natural Language Generation*, pages 130–140. Springer, 2004. 21

Jarmila Panevová, Alena Böhmová, and Petr Sgall. Syntactic tagging: Procedure for the transition from the analytic to the tectogrammatical tree structure. In *Proceedings of the 2nd International Conference on Text, Speech and Dialogue (TSD)*, pages 34–38, Plzen, Czech Republic, 1999. 48

Penman. *The Penman documentation.* Technical report, USC/Information Sciences Institute, 1989. 18

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 433–440, Sydney, Australia, 2006. 183

Carl Pollard. *Head-driven Phrase Structure Grammar.* University of Chicago Press, 1994. 41, 52

Prokopis Prokopidis, Elina Desipri, Maria Koutsombogera, Harris Papageorgiou, and Stelios Piperidis. Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160, Barcelona, Spain, 2005. 14

Rajakrishnan Rajkumar, Dominic Espinosa, and Michael White. The OSU system for surface realization at Generation Challenges 2011. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 236–238, Nancy, France, 2011. 21

Loganathan Ramasamy and Zdenek Zabokrtskỳ. Prague dependency style treebank for Tamil. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 1888–1894, Istambul, Turkey, 2012. 14

Owen Rambow and Tanya Korelsky. Applied text generation. In *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP)*, pages 40–47, Trento, Italy, 1992. 6, 7

Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. Development of a Persian syntactic dependency treebank. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, pages 306–314, Atlanta, GA, USA, 2013. 14

Adwait Ratnaparkhi. Trainable methods for surface Natural Language Generation. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 194–201, Seattle, WA, USA, 2000. 19, 20

Ines Rehbein and Josef van Genabith. Treebank annotation schemes and parser evaluation for German. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 630–639, Prague, Czech Republic, 2007. 181, 182

Ehud Reiter. Has a consensus NL Generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation (INLG)*, pages 163–170, Kennebunkport, Maine, 1994. 2

Ehud Reiter and Robert Dale. Building applied Natural Language Generation systems. *Natural Language Engineering*, 3(1):57–87, 1997. 7

Eric Ringger, Michael Gamon, Robert C Moore, David Rojas, Martine Smets, and Simon Corston-Oliver. Linguistically informed statistical models of constituent structure for ordering in sentence realization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 673–679, Geneva, Switzerland, 2004. 22, 161, 162, 163, 174, 175, 176

W. Rounds. Mappings and grammars on trees. *Mathematical Systems Theory*, 2:257–287, 1970. 171

Beatrice Santorini. Part-of-Speech tagging guidelines for the Penn TreeBank Project. Technical report, Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania, 1990. 56

Roy Schwartz, Omri Abend, and Ari Rappoport. Learnability-based syntactic annotation design. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 2405–2421, Mumbai, India, 2012. 60, 210

N. Sebastián, M.A. Martí, M.F. Carreiras, and F. Cuetos. *LEXESP: Léxico informatizado del español*. Edicions Universitat Barcelona, 2000. 32

M. Seraji, B. Megyesi, and J. Nivre. Dependency parsers for Persian. In *Proceedings of 10th Workshop on Asian Language Resources, 24th International Conference on Computational Linguistics (COLING)*, pages 35–44, Mumbai, India, 2012. 196

P. Sgall, E. Hajičová, and J. Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel Publishing Company, Dordrecht, 1986. 129

A. Siddharthan. Text simplification using typed dependencies: A comparison of the robustness of different generation strategies. In *Proceedings of* *the 13th European Workshop on Natural Language Generation (ENLG)*, pages 2–11, Nancy, France, 2011. 212

Noah A. Smith. *Linguistic Structure Prediction.* Synthesis Lectures on Human Language Technologies. Morgan and Claypool, 2011. 153

John Sowa. *Knowledge representation: Logical, philosophical, and computational foundations*, volume 13. Brooks/Cole Publishing, Pacific Grove, CA, 2000. 4, 37

Mark Steedman. *Surface structure and interpretation*, volume 30. MIT Press Cambridge, MA, 1996. 52

Amanda Stent. ATT-0: Submission to Generation Challenges 2011 Surface Realization Shared Task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation (ENLG)*, pages 230–231, Nancy, France, 2011. 19

Amanda Stent, Rashmi Prasad, and Marilyn Walker. Trainable sentence planning for complex information presentation in spoken dialog systems. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 79–86, Barcelona, Spain, 2004. 19, 164

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluis Márquez, and Joakim Nivre. 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)*, pages 159–177, Manchester, UK, 2008. 13, 212

Marko Tadić. Building the Croatian Dependency Treebank: The initial stages. *Suvremena lingvistika*, 63(1):85–92, 2007. 14

Mariona Taulé, M. Antònia Martí, and Marta Recasens. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pages 96–101, Marrakech, Morocco, 2008. 14, 69, 93

Lucien Tesnière. *Eléments de syntaxe structurale.* Librairie C. Klincksieck, 1959. 13

J.W. Thatcher. Generalized^2 sequential machine maps. *Journal of Computer and System Sciences*, 4:339–367, 1970. 171

Mariët Theune, Esther Klabbers, Jan-Roelof de Pijper, Emiel Krahmer, and Jan Odijk. From data to speech: A general approach. *Natural Language Engineering*, 7(1):47–86, 2001. 9

R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kübler, M. Candito, J. Foster, Y. Versley, I. Rehbein, and L. Tounsi. Statistical Parsing of Morphologically Rich Languages (SPMRL): What, how and whither. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL), Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, pages 1–12, Los Angeles, California, 2010. 197

R. Tsarfaty, J. Nivre, and E. Andersson. Cross-framework evaluation for statistical parsing. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 44–54, Avignon, France, 2012a. 196

R. Tsarfaty, D. Seddah, S. Kübler, and J. Nivre. Parsing Morphologically Rich Languages: Introduction to the Special Issue. *Computational Linguistics*, 39(1):15–22, 2012b. 197

Reut Tsarfaty. A unified morpho-syntactic scheme of Stanford dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 578–584, Sofia, Bulgaria, 2013. 87

Enric Vallduvi. *The information component.* PhD thesis, University of Pennsylvania, Philadelphia, 1990. 129

Kees Van Deemter and Jan Odijk. Context modeling and the generation of spoken discourse. *Speech Communication*, 21(1):101–121, 1997. 9

Kees Van Deemter, Emiel Krahmer, and Mariët Theune. Real versus template-based Natural Language Generation: A false opposition? *Computational Linguistics*, 31(1):15–24, 2005. 9

Leonoor Van der Beek, Gosse Bouma, Rob Malouf, and Gertjan Van Noord. The Alpino dependency treebank. *Language and Computers - Studies in Practical Linguistics*, 45(1):8–22, 2002. 14

Sebastian Varges and Chris Mellish. Instance-based Natural Language Generation. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8, Pittsburgh, PA, USA, 2001. 21

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. Hungarian Dependency Treebank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1855–1862, Valletta, Malta, 2010. 14

Marilyn A Walker. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system. *Journal of Artificial Intelligence Research*, 12:387–416, 2000. 20

Marilyn A Walker, Owen C Rambow, and Monica Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3):409–433, 2002. 19, 164

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 852–860, Athens, Greece, 2009. 23, 163

Leo Wanner, Bernd Bohnet, Nadjet Bouayad-Agha, Francois Lareau, and Daniel Nicklaß. MARQUIS: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, 24(10):914–952, 2010. xv, 9, 10, 11

Leo Wanner, Simon Mille, and Bernd Bohnet. Towards a surface realization-oriented corpus annotation. In *Proceedings of the 7th International Natural Language Generation Conference (INLG)*, pages 22–30, Utica, IL, USA, 2012. 43

Ralph Weischedel and Ada Brunstein. BBN pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 2005. 112

Michael White and Ted Caldwell. A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation (INLG)*, pages 238–247, 1998. 9

Michael White, Rajakrishnan Rajkumar, and Scott Martin. Towards broad coverage surface realization with CCG. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation: Language Generation and Machine Translation (UCNLG+MT)*, pages 22–30, Copenhagen, Denmark, 2007. 11, 21

Yuk Wah Wong and Raymond J Mooney. Generation by inverting a semantic parser that uses statistical machine translation. In *Proceedings of the 2007 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT)*, pages 172–179, Rochester, NY, USA, 2007. 23, 175

Y. Zhang and S. Clark. A Tale of Two Parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 562–571, Honolulu, Hawaii, 2008a. 226

Y. Zhang and J. Nivre. Transition-based parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 188–193, Portland, Oregon, USA, 2011. 226

Yue Zhang and Stephen Clark. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies (ACL:HLT)*, pages 888–896, Columbus, OH, USA, 2008b. 226

Huayan Zhong and Amanda Stent. Building surface realizers automatically from corpora. *Proceedings of the Workshop on Using Corpora for Natural Language Generation (UCNLG), Corpus Linguistics*, 5:49–54, 2005. 21

APPENDIX A

SSyntRel properties and illustrations

In this Appendix, all 48 SSyntRels introduced in Chapter 3 are listed, together with their properties and some representative examples. Note that the illustrations for each SSyntRel includes examples of more fine-grained dependencies from the 79-label tagset which are subsumed by said SSyntRel: for instance, the SSyntRel *adv* includes cases of *adjunct* and *restr*; the SSyntRel *obl_obj* includes *obl_obj1*, *obl_obj2*, *obl_obj3*, etc. If one of the subsumed DepRels is different enough from the typical configuration, another column for possible values for this DepRel is added (*adv*, *conj*, *elect*, *obl_obj*). If it is just a matter of the type of dependent and it only affects the agreement criterion, i.e. depending if the dependent allows for an agreement or not, we keep only one column for the possible values, but specify various values in the same cell (*compl*, *conj*, *coord_conj*, *copul*, *obj_copred*, *subj_copred*). The same is applied for *dobj*, which can be introduced by a preposition *a* 'to' in certain cases.

Unless mentioned otherwise, DepRels with extensions *_descr* have only one difference with the corresponding extentionless DepRel, which is the presence of a comma between the governor and the dependent. DepRels with extensions *_quot* have only one difference with the corresponding extentionless DepRel, which is the presence of citation quotes around the dependent.

In the tables:

- N/A is "no answer", but it actually means "all answers": a criterion with this value does not help identifying the DepRel in question, because it is easy to find examples which would have different values.

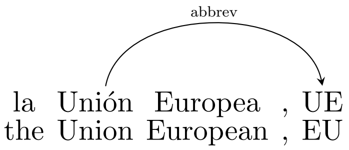
- - is “not used”, mainly in case of a subcriterion of a criterion the value of which is NO (e.g. agreement/agreement with), and in cases of absence of generic prototypical dependent.

For an explanation of the criteria, refer to Section 3.3.2.

Abbreviative	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N | Date
prototypical Dep	N
PoS Dep	Acronym
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	YES

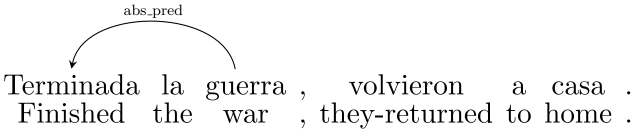
Table A.1: Distinctive properties of the *abbrev* SSynt DepRel

This relation has the same basic properties as the *appos* DepRel; however, we keep it separated because it easy for an annotator to differentiate an abbreviation from a classic apposition, and the difference in meaning is substantial.



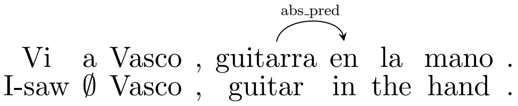
Absolute predicative	
Criterion	**Possible values**
PoS Gov	N | Date
prototypical Dep	Adj
PoS Dep	V_{Ger} | V_{Part} | Adj | Prep | Adv
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	N/A
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	Gov
variant inflection	YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.2: Distinctive properties of the *abs_pred* SSynt DepRel



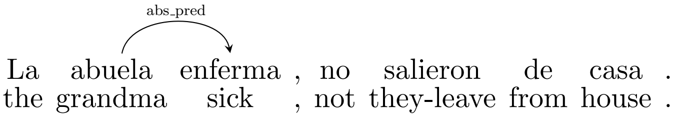
Terminada la guerra , volvieron a casa .
Finished the war , they-returned to home .

‘The war being over, they returned home.’



Vi a Vasco , guitarra en la mano .
I-saw ∅ Vasco , guitar in the hand .

‘I saw Vasco, with his guitar in his hand.’

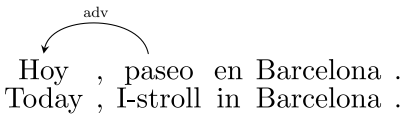


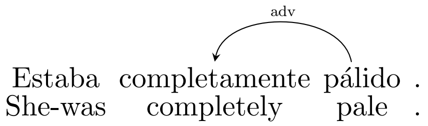
La abuela enferma , no salieron de casa .
the grandma sick , not they-leave from house .

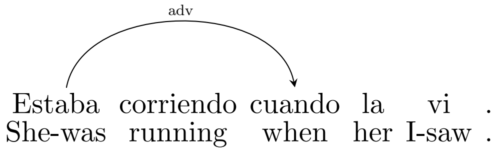
‘The grandma being sick, they did not leave their house.’

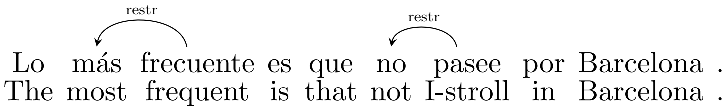
Adverbial		
Criterion	**Possible values**	
	typical	restr
PoS Gov	V_{Fin} | V_{NoFin} | N | Adj | Adj_{comp} | Adj_{sup} | Adv | Num | Prep | Conj | $Conj_{coord}$ | Date	
prototypical Dep	Adv	Adv
PoS Dep	Conj | V_{Ger} | Prep | Adv | N	Adv
governed preposition	NO	NO
governed grammeme	NO	NO
type of linearization	N/A	FIXED
canonical order	N/A	LEFT
adjacency to Gov	N/A	YES
cliticization	NO	NO
promotion	NO	NO
demotion	NO	NO
agreement	NO	NO
agreement with	-	-
variant inflection	-	-
Dep omissibility	YES	YES
dependency	SUBORD	SUBORD
left disloc = strong focus	NO	NO
punctuation	N/A	N/A

Table A.3: Distinctive properties of the *adv* SSynt DepRel



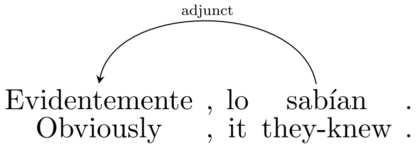




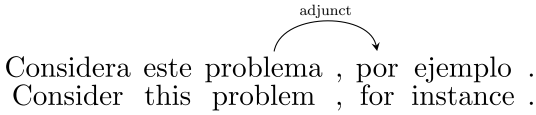


Lo	más	frecuente	es	que	no	pasee	por	Barcelona	.
The	most	frequent	is	that	not	I-stroll	in	Barcelona	.

The *adjunctives* are backgrounded adverbials.



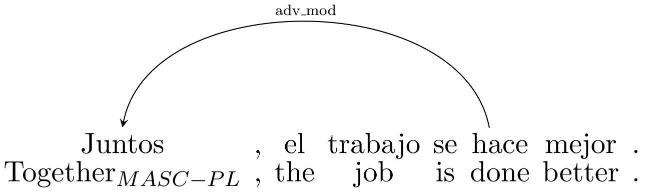
Evidentemente	,	lo	sabían	.
Obviously	,	it	they-knew	.



Considera	este	problema	,	por	ejemplo	.
Consider	this	problem	,	for	instance	.

Adverbial modificative	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	V_{Part} | Adj
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	LEFT
adjacency to Gov	NO
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	External element
variant inflection	YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	N/A

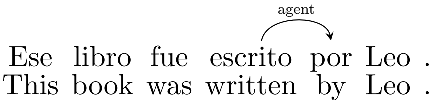
Table A.4: Distinctive properties of the *adv_mod* SSynt DepRel



Juntos	,	el	trabajo	se	hace	mejor	.
Together$_{MASC-PL}$	,	the	job	is	done	better	.

Agentive	
Criterion	**Possible values**
PoS Gov	V_{NoFin}
prototypical Dep	N
PoS Dep	Prep
governed preposition	YES (por)
governed grammeme	NO
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	YES
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	YES
punctuation	N/A

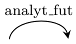
Table A.5: Distinctive properties of the *agent* SSynt DepRel



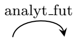
Ese libro fue escrito por Leo .
This book was written by Leo .

Analytical future	
Criterion	**Possible values**
PoS Gov	$V_{Fin} \mid V_{NoFin}$
prototypical Dep	V
PoS Dep	Prep
governed preposition	YES (a)
governed grammeme	fin=INF
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.6: Distinctive properties of the *analyt_fut* SSynt DepRel



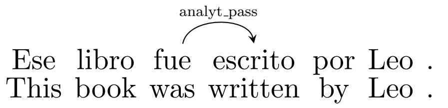
Va	a	conducir	hasta	Burdeos	.
She-will	∅	drive	until	Bordeaux	.



Cuando	Joan	estaba	yendo	a	cenar	,	lo	llamaron	.
When	Joan	was	going	to	eat	,	him	they-called	.

Analytical passive	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	V
PoS Dep	V_{Part}
governed preposition	NO
governed grammeme	fin=PART
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	Subject
variant inflection	YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.7: Distinctive properties of the *analyt_pass* SSynt DepRel

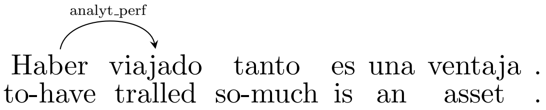


Ese libro fue escrito por Leo .
This book was written by Leo .

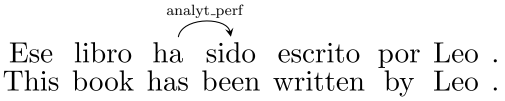
Ser derrotado por Barcelona les occurre mucho .
Be beaten by Barcelona to-them happens a-lot .

Analytical perfective	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	V
PoS Dep	V_{Part}
governed preposition	NO
governed grammeme	fin=PART
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.8: Distinctive properties of the *analyt_perf* SSynt DepRel



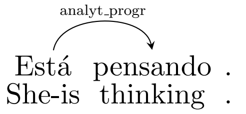
Haber	viajado	tanto	es	una	ventaja	.
to-have	tralled	so-much	is	an	asset	.

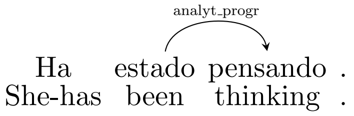


Ese	libro	ha	sido	escrito	por	Leo	.
This	book	has	been	written	by	Leo	.

Analytical progressive	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	V
PoS Dep	V_{Ger}
governed preposition	NO
governed grammeme	fin=GER
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

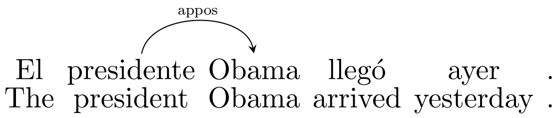
Table A.9: Distinctive properties of the *analyt_progr* SSynt DepRel

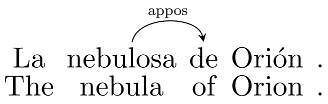




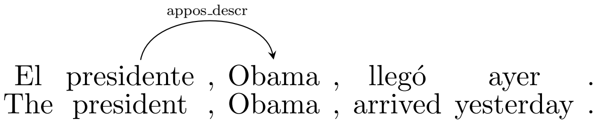
Appositive	
Criterion	**Possible values**
PoS Gov	N
prototypical Dep	N
PoS Dep	Prep | N
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

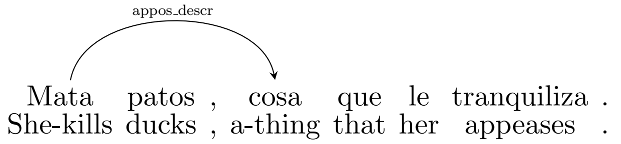
Table A.10: Distinctive properties of the *appos* SSynt DepRel





The *appos_descr* DepRel is backgrounded (there are alomost always commas), and can also be used with governors such as finite and non-finite verbs or dates.

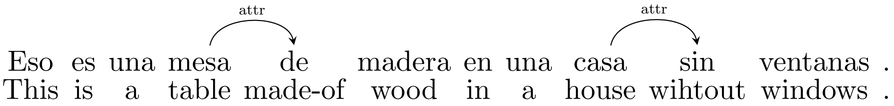




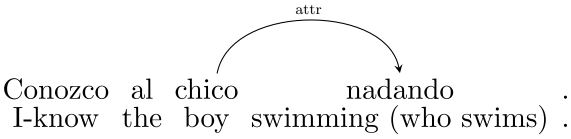
Mata patos , cosa que le tranquiliza .
She-kills ducks , a-thing that her appeases .

Attributive	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N
prototypical Dep	Adv
PoS Dep	V_{Ger} | Prep | Adv
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.11: Distinctive properties of the *attr* SSynt DepRel

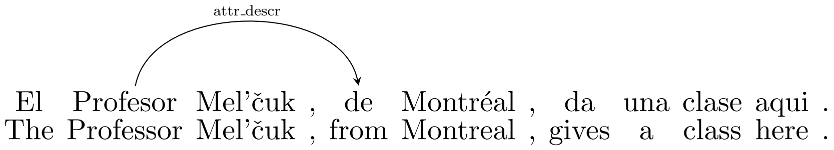


Eso es una mesa de madera en una casa sin ventanas .
This is a table made-of wood in a house wihtout windows .



Conozco al chico nadando .
I-know the boy swimming (who swims) .

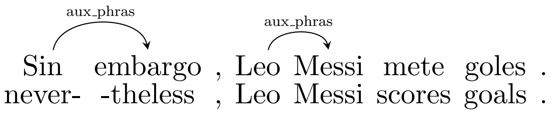
The backgrounded variant of this DepRel is *attr_descr*.

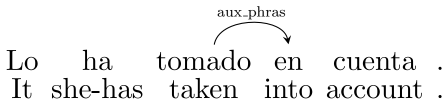


El Profesor Mel'čuk , de Montréal , da una clase aqui .
The Professor Mel'čuk , from Montreal , gives a class here .

Auxiliary phraseological	
Criterion	**Possible values**
PoS Gov	Any
prototypical Dep	Any
PoS Dep	VPart—A
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.12: Distinctive properties of the *aux_phras* SSynt DepRel

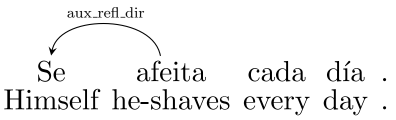




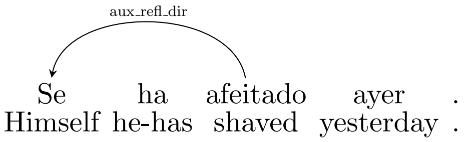
Auxiliary reflective	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	-
PoS Dep	Clitic_{se}
governed preposition	NO
governed grammeme	∅ | case=ACC | case=DAT
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.13: Distinctive properties of the *aux_refl* SSynt DepRel

aux_refl_dir requires *case=ACC*.

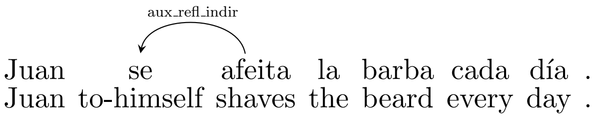


Se afeita cada día .
Himself he-shaves every day .

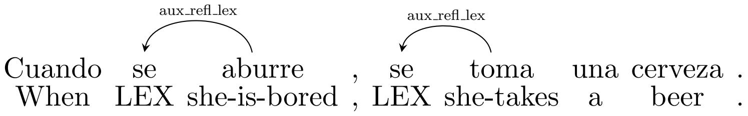


Se ha afeitado ayer .
Himself he-has shaved yesterday .

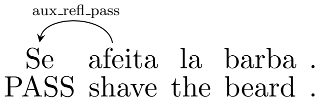
aux_refl_indir requires *case=DAT*.



Juan se afeita la barba cada día .
Juan to-himself shaves the beard every day .

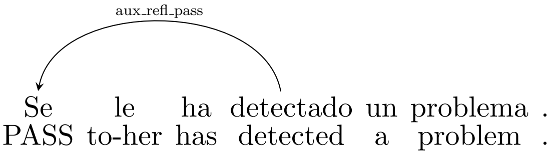


Cuando	se	aburre	,	se	toma	una	cerveza	.
When	LEX	she-is-bored	,	LEX	she-takes	a	beer	.



Se	afeita	la	barba	.
PASS	shave	the	beard	.

‘The beard is shaved.’

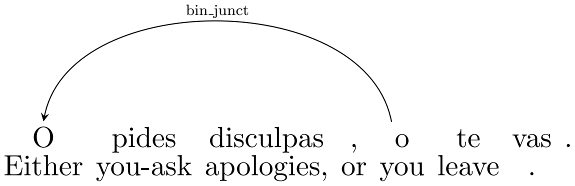


Se	le	ha	detectado	un	problema	.
PASS	to-her	has	detected	a	problem	.

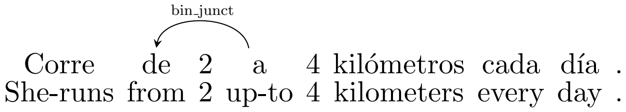
‘A problem was detected on her’

Binary junctive	
Criterion	**Possible values**
PoS Gov	Prep | Conj
prototypical Dep	Adv
PoS Dep	Conj | Conj_{coord} | Prep | Adv
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	NO
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	COORD
left disloc = strong focus	NO
punctuation	N/A

Table A.14: Distinctive properties of the *bin_junct* SSynt DepRel



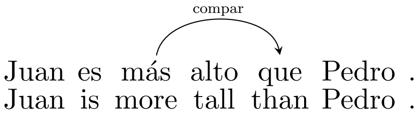
O pides disculpas , o te vas .
Either you-ask apologies, or you leave .



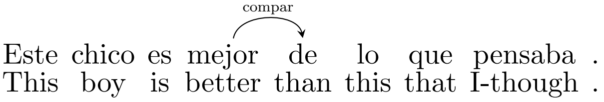
Corre de 2 a 4 kilómetros cada día .
She-runs from 2 up-to 4 kilometers every day .

Comparative	
Criterion	**Possible values**
PoS Gov	Adj $\mid$ Adj$_{comp}$ $\mid$ Adv
prototypical Dep	N/A
PoS Dep	Conj $\mid$ Prep
governed preposition	YES
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

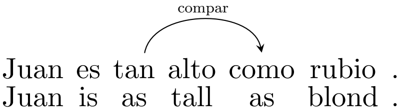
Table A.15: Distinctive properties of the *compar* SSynt DepRel



Juan es más alto que Pedro .
Juan is more tall than Pedro .



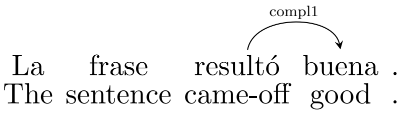
Este chico es mejor de lo que pensaba .
This boy is better than this that I-though .

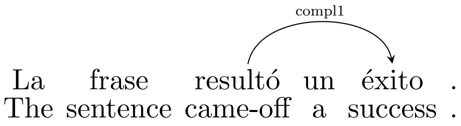


Juan es tan alto como rubio .
Juan is as tall as blond .

Completive 1	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep | Adv | N
governed preposition	NO
governed grammeme	fin=INF | fin=PART | fin=GER
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | Subject
variant inflection	- | YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	YES
punctuation	NO

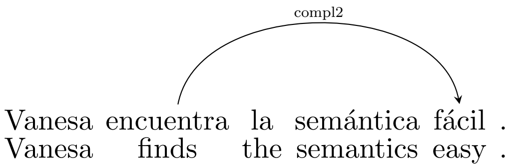
Table A.16: Distinctive properties of the *compl1* SSynt DepRel



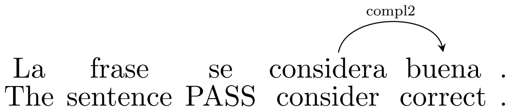


Completive 2	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep | Adv | N
governed preposition	NO
governed grammeme	fin=INF | fin=PART | fin=GER
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | Direct object
variant inflection	- | YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	YES
punctuation	NO

Table A.17: Distinctive properties of the *compl2* SSynt DepRel

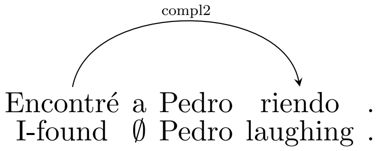


Vanesa encuentra la semántica fácil .
Vanesa finds the semantics easy .

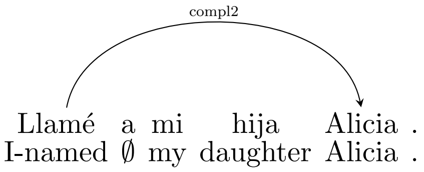


La frase se considera buena .
The sentence PASS consider correct .

‘The sentence is considered correct.’



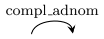
Encontré a Pedro riendo .
I-found ∅ Pedro laughing .



Llamé a mi hija Alicia .
I-named ∅ my daughter Alicia .

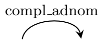
Completive adnominal	
Criterion	**Possible values**
PoS Gov	Det
prototypical Dep	Adv
PoS Dep	Prep
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.18: Distinctive properties of the *compl_adnom* SSynt DepRel



Los	de	Barcelona	no	han	llegado	.
The	of	Barcelona	not	have	arrived	.

‘The ones from Barcelona have not arrived.’

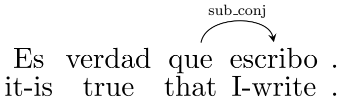


Veo	a	la	del	sombrero	rojo	.
I-see	∅	the	of-the	hat	red	.

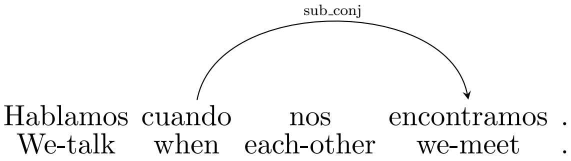
‘I see the one with the red had.’

Conjunctive		
Criterion	**Possible values**	
	sub	compar
PoS Gov	Conj	Conj
prototypical Dep	N/A	N/A
PoS Dep	V	Conj | $V_{FinRelatNoAnt}$ | V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep | Adv | N | Num
governed preposition	NO	NO
governed grammeme	fin=FIN	case=NOM
type of linearization	FIXED	FIXED
canonical order	RIGHT	RIGHT
adjacency to Gov	N/A	N/A
cliticization	NO	NO
promotion	NO	NO
demotion	NO	NO
agreement	NO	NO | dep=TARGET
agreement with	-	- | External element
variant inflection	-	- | YES
Dep omissibility	NO	NO
dependency	SUBORD	SUBORD
left disloc = strong focus	NO	NO
punctuation	NO	NO

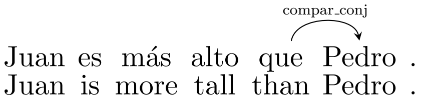
Table A.19: Distinctive properties of the *conj* SSynt DepRel



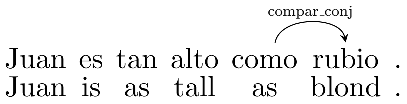
Es verdad que escribo .
it-is true that I-write .



Hablamos cuando nos encontramos .
We-talk when each-other we-meet .



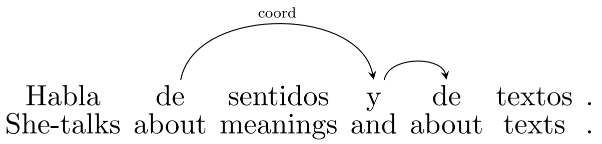
Juan es más alto que Pedro .
Juan is more tall than Pedro .



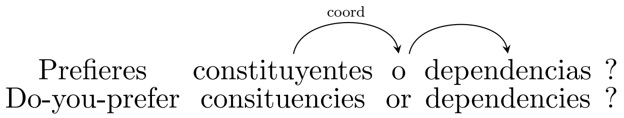
Juan es tan alto como rubio .
Juan is as tall as blond .

Coordinative	
Criterion	**Possible values**
PoS Gov	V$_{Fin}$ | V$_{NoFin}$ | N | Adj | Adv | Num | Prep | Conj | Date | Det
prototypical Dep	-
PoS Dep	Conj$_{coord}$ | Punc
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	COORD
left disloc = strong focus	NO
punctuation	N/A

Table A.20: Distinctive properties of the *coord* SSynt DepRel



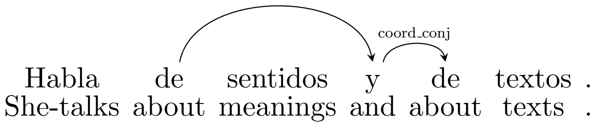
Habla de sentidos y de textos .
She-talks about meanings and about texts .



Prefieres constituyentes o dependencias ?
Do-you-prefer consituencies or dependencies ?

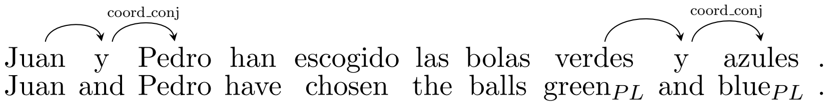
Coordinative conjunctive	
Criterion	**Possible values**
PoS Gov	Conj_{coord}
prototypical Dep	N/A
PoS Dep	Clitic | Num | Det | Conj | V_{Fin} | $V_{FinRelatNoAnt}$ | V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep | Adv | N
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | External element
variant inflection	- | YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.21: Distinctive properties of the *coord_conj* SSynt DepRel



Habla de sentidos y de textos .
She-talks about meanings and about texts .

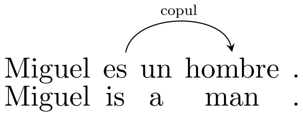
Prefieres constituyentes o dependencias ?
Do-you-prefer consituencies or dependencies ?



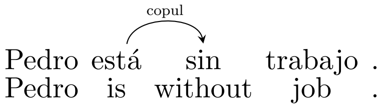
Juan y Pedro han escogido las bolas verdes y azules .
Juan and Pedro have chosen the balls green_{PL} and blue_{PL} .

Copulative	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	Conj | $V_{FinRelatNoAnt}$ | V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep | Adv | Num| N
governed preposition	NO
governed grammeme	case=ACC
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	YES
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | Subject
variant inflection	- | YES
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	YES
punctuation	N/A

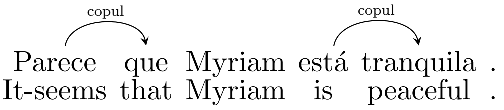
Table A.22: Distinctive properties of the *copul* SSynt DepRel



Miguel es un hombre .
Miguel is a man .

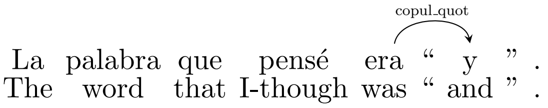


Pedro está sin trabajo .
Pedro is without job .



Parece que Myriam está tranquila .
It-seems that Myriam is peaceful .

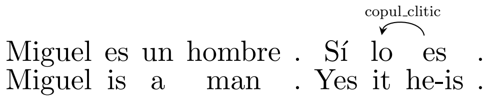
The quotative variant can have a dependent of almost any PoS.



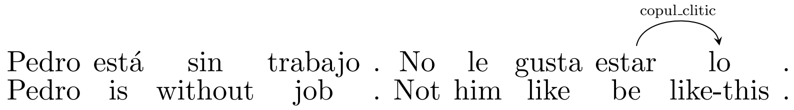
La palabra que pensé era “ y ” .
The word that I-though was “ and ” .

Copulative clitic	
Criterion	**Possible values**
PoS Gov	$V_{Fin} \mid V_{NoFin}$
prototypical Dep	-
PoS Dep	Clitic$_{lo}$
governed preposition	NO
governed grammeme	case=ACC
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

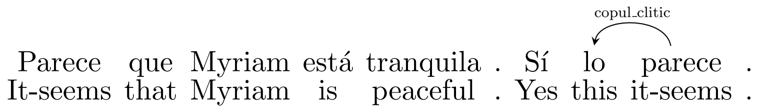
Table A.23: Distinctive properties of the *copul_clitic* SSynt DepRel



Miguel	es	un	hombre	.	Sí	lo	es	.
Miguel	is	a	man	.	Yes	it	he-is	.



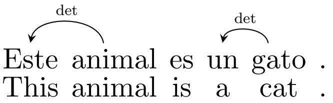
Pedro	está	sin	trabajo	.	No	le	gusta	estar	lo	.
Pedro	is	without	job	.	Not	him	like	be	like-this	.

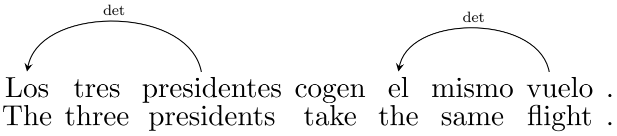


Parece	que	Myriam	está	tranquila	.	Sí	lo	parece	.
It-seems	that	Myriam	is	peaceful	.	Yes	this	it-seems	.

Determinative	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N | Date
prototypical Dep	-
PoS Dep	Det
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	Gov
variant inflection	YES
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

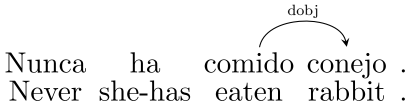
Table A.24: Distinctive properties of the *det* SSynt DepRel



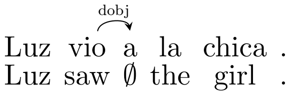


Direct objectival	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	N
PoS Dep	Conj | $V_{FinRelatNoAnt}$ | V_{Inf} | Prep | N
governed preposition	NO | $Prep_a$
governed grammeme	case=ACC
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	YES
promotion	N/A
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	YES
punctuation	N/A

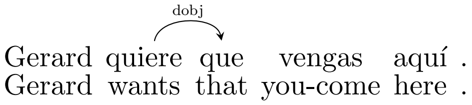
Table A.25: Distinctive properties of the *dobj* SSynt DepRel



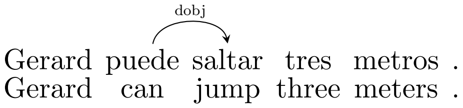
Nunca ha comido conejo .
Never she-has eaten rabbit .



Luz vio a la chica .
Luz saw ∅ the girl .

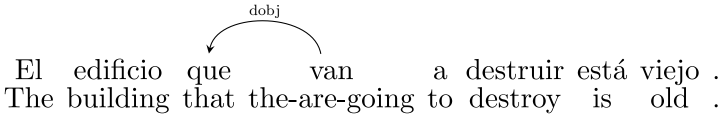


Gerard quiere que vengas aquí .
Gerard wants that you-come here .



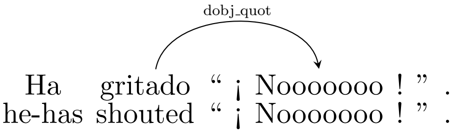
Gerard puede saltar tres metros .
Gerard can jump three meters .

‘Gerard is capable of jumping three meters.’

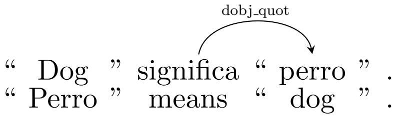


El	edificio	que	van	a	destruir	está	viejo	.
The	building	that	the-are-going	to	destroy	is	old	.

The quotative variant can have a dependent of almost any PoS.



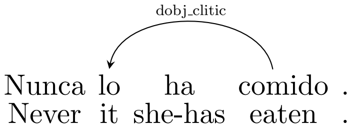
Ha	gritado	“	¡	Nooooooo	!	”	.
he-has	shouted	“	¡	Noooooooo	!	”	.



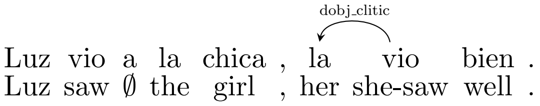
“	Dog	”	significa	“	perro	”	.
“	Perro	”	means	“	dog	”	.

Direct objectival clitic	
Criterion	**Possible values**
PoS Gov	$\mathrm{V}_{Fin} \mid \mathrm{V}_{NoFin}$
prototypical Dep	-
PoS Dep	Clitic
governed preposition	NO
governed grammeme	case=ACC
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

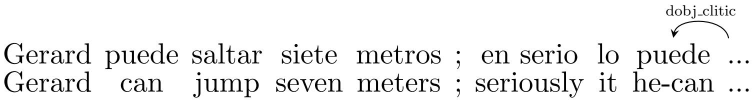
Table A.26: Distinctive properties of the *dobj_clitic* SSynt DepRel



Nunca lo ha comido .
Never it she-has eaten .



Luz vio a la chica , la vio bien .
Luz saw ∅ the girl , her she-saw well .

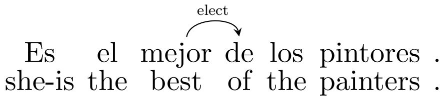


Gerard puede saltar siete metros ; en serio lo puede ...
Gerard can jump seven meters ; seriously it he-can ...

‘Gerard is capable of jumping three meters.’

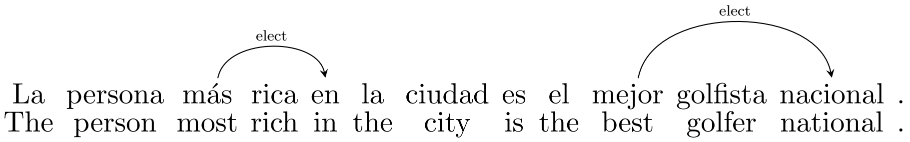
Elective		
Criterion	**Possible values**	
	typical	variant Agree
PoS Gov	Adj$_{comp}$ | Adj$_{sup}$ | Adv | Num	
prototypical Dep	Adj	Adj
PoS Dep	V$_{FinRelatNoAnt}$ | Prep	V$_{Part}$ | Adj
governed preposition	NO	NO
governed grammeme	NO	NO
type of linearization	FREE	FIXED
canonical order	RIGHT	RIGHT
adjacency to Gov	NO	NO
cliticization	NO	NO
promotion	NO	NO
demotion	NO	NO
agreement	NO	dep=TARGET
agreement with	-	Gov
variant inflection	-	YES
Dep omissibility	YES	YES
dependency	SUBORD	SUBORD
left disloc = strong focus	YES	NO
punctuation	N/A	N/A

Table A.27: Distinctive properties of the *elect* SSynt DepRel

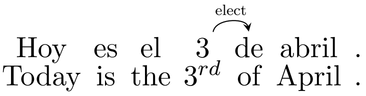


Es el mejor de los pintores .
she-is the best of the painters .

Una de las chicas no llegó .
One of tjhe girls not arrived .



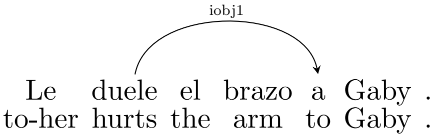
La persona más rica en la ciudad es el mejor golfista nacional .
The person most rich in the city is the best golfer national .



Hoy es el 3 de abril .
Today is the 3^{rd} of April .

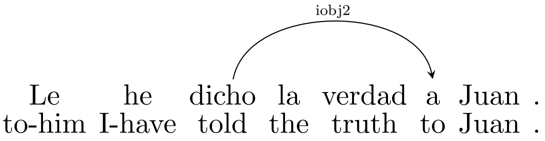
Indirect objectival	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	N
PoS Dep	$Prep_a$
governed preposition	YES (a)
governed grammeme	case=DAT
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	YES
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	YES
punctuation	N/A

Table A.28: Distinctive properties of the *iobj* SSynt DepRel



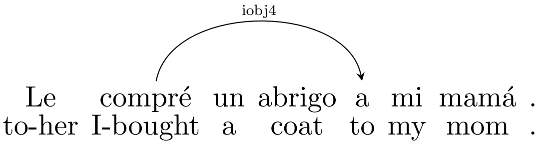
Le duele el brazo a Gaby .
to-her hurts the arm to Gaby .

‘Gaby’s arm hurts.’



Le he dicho la verdad a Juan .
to-him I-have told the truth to Juan .

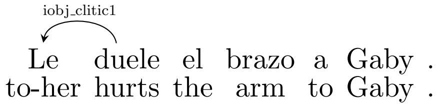
‘I told the truth to Juan.’



Le compré un abrigo a mi mamá .
to-her I-bought a coat to my mom .

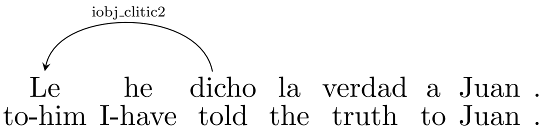
Indirect objectival clitic	
Criterion	**Possible values**
PoS Gov	$V_{Fin} \mid V_{NoFin}$
prototypical Dep	-
PoS Dep	Clitic
governed preposition	NO
governed grammeme	case=DAT
type of linearization	FIXED
canonical order	N/A
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.29: Distinctive properties of the *iobj_clitic* SSynt DepRel



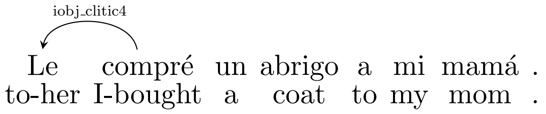
Le duele el brazo a Gaby .
to-her hurts the arm to Gaby .

‘Gaby’s arm hurts.’



Le he dicho la verdad a Juan .
to-him I-have told the truth to Juan .

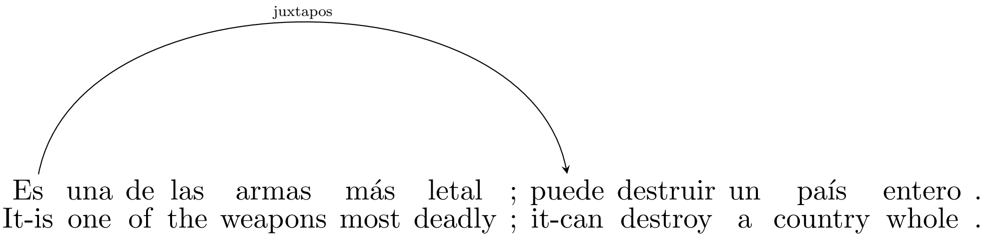
‘I told the truth to Juan.’



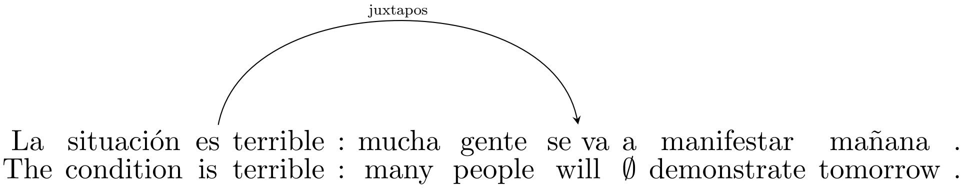
Le compré un abrigo a mi mamá .
to-her I-bought a coat to my mom .

Juxtapositive	
Criterion	**Possible values**
PoS Gov	$V_{Fin} \mid V_{NoFin} \mid$ N $\mid$ Adj $\mid$ Adv $\mid$ Num $\mid$ Prep $\mid$ Conj
prototypical Dep	-
PoS Dep	Conj $\mid \mid V_{Fin} \mid V_{FinRelatNoAnt} \mid V_{Inf} \mid V_{Ger} \mid V_{Part} \mid$ Adj $\mid$ Prep $\mid$ Adv $\mid$ Num $\mid$ N
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	NO
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	COORD
left disloc = strong focus	NO
punctuation	YES

Table A.30: Distinctive properties of the *juxtapos* SSynt DepRel



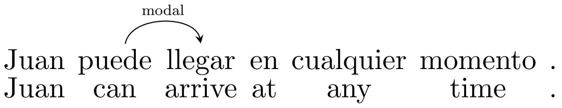
Es una de las armas más letal ; puede destruir un país entero .
It-is one of the weapons most deadly ; it-can destroy a country whole .



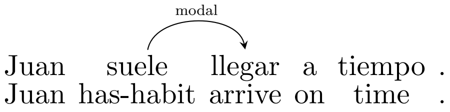
La situación es terrible : mucha gente se va a manifestar mañana .
The condition is terrible : many people will ∅ demonstrate tomorrow .

Modal	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	V
PoS Dep	V_{Inf} | Prep
governed preposition	N/A
governed grammeme	fin=INF
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	YES
punctuation	NO

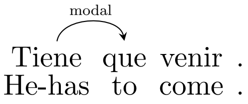
Table A.31: Distinctive properties of the *modal* SSynt DepRel



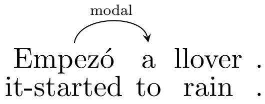
Juan puede llegar en cualquier momento .
Juan can arrive at any time .



Juan suele llegar a tiempo .
Juan has-habit arrive on time .



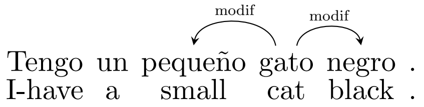
Tiene que venir .
He-has to come .



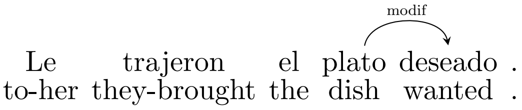
Empezó a llover .
it-started to rain .

Modificative	
Criterion	**Possible values**
PoS Gov	N | Date
prototypical Dep	Adj
PoS Dep	V_{Part} | Adj
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=TARGET
agreement with	Gov
variant inflection	YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.32: Distinctive properties of the *modif* SSynt DepRel

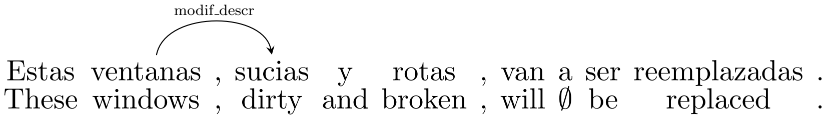


Tengo un pequeño gato negro .
I-have a small cat black .



Le trajeron el plato deseado .
to-her they-brought the dish wanted .

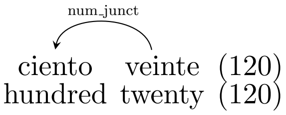
The backgrounded variant of this DepRel is *modif_descr*.

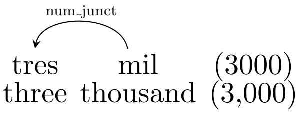


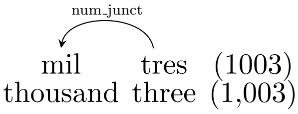
Estas ventanas , sucias y rotas , van a ser reemplazadas .
These windows , dirty and broken , will ∅ be replaced .

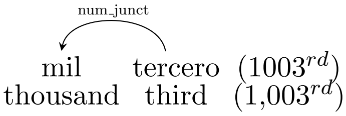
Numeral junctive	
Criterion	**Possible values**
PoS Gov	Num
prototypical Dep	-
PoS Dep	Num
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.33: Distinctive properties of the *num_junct* SSynt DepRel



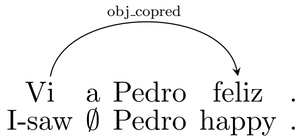


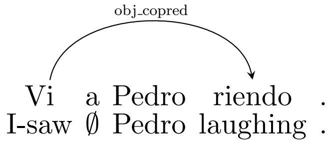


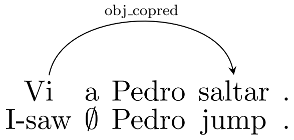


Object copredicative	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	Conj | V_{Inf} | V_{Ger} | V_{Part} | Adj | Prep
governed preposition	NO
governed grammeme	NO
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | Direct object
variant inflection	- | YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	N/A
punctuation	NO

Table A.34: Distinctive properties of the *obj_copred* SSynt DepRel

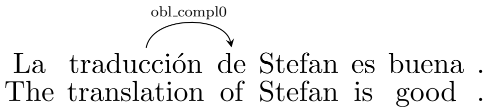


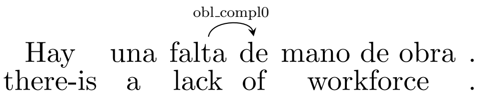


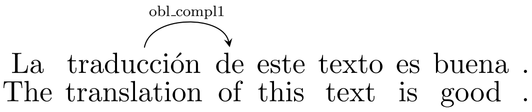


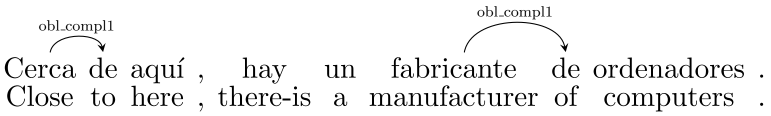
Oblique completive	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N | Adj | Adv | Prep | Conj
prototypical Dep	N
PoS Dep	Prep
governed preposition	YES
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

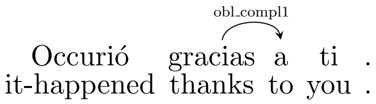
Table A.35: Distinctive properties of the *obl_compl* SSynt DepRel



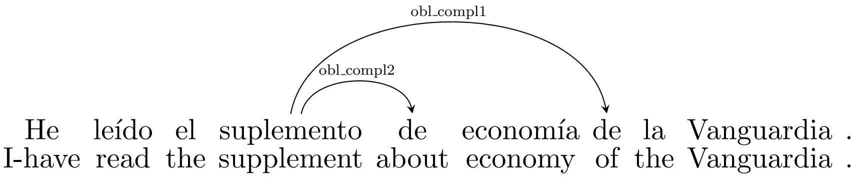








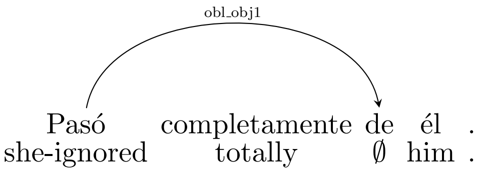
Occurió gracias a ti .
it-happened thanks to you .



He leído el suplemento de economía de la Vanguardia .
I-have read the supplement about economy of the Vanguardia .

Oblique objectival			
Criterion	**Possible values**		
	typical	variant Adv	variant V
PoS Gov	V_{Fin} | V_{NoFin}		
prototypical Dep	N	N	V
PoS Dep	Prep | Conj	Adv	V_{Inf}
governed preposition	YES	NO	NO
governed grammeme	NO	circum	fin=INF
type of linearization	FREE	FREE	FREE
canonical order	RIGHT	RIGHT	RIGHT
adjacency to Gov	N/A	N/A	N/A
cliticization	NO	NO	NO
promotion	NO	NO	NO
demotion	NO	NO	NO
agreement	NO	NO	NO
agreement with	-	-	-
variant inflection	-	-	-
Dep omissibility	N/A	N/A	N/A
dependency	SUBORD	SUBORD	SUBORD
left disloc = strong focus	YES	YES	YES
punctuation	N/A	N/A	N/A

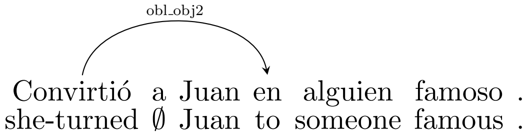
Table A.36: Distinctive properties of the *obl_obj* SSynt DepRel

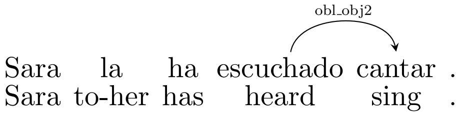


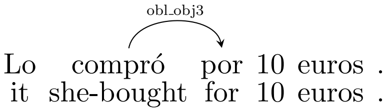
Pasó completamente de él .
she-ignored totally ∅ him .

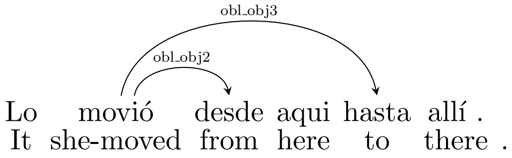
Note that when the dependent is an adverb, the type of adverb is not free; it has to be circumstancial (location, time, etc.). For instance, movement verbs require a locative adverb. If an adverb is not circumstancial, it is more probable that the concerned DepRel is a completive or a copredicative one.

Cuando voy allí , me siento en casa .
When I-go there , REFL I-feel at home .


obl_obj2
Convirtió a Juan en alguien famoso .
she-turned ∅ Juan to someone famous .


obl_obj2
Sara la ha escuchado cantar .
Sara to-her has heard sing .

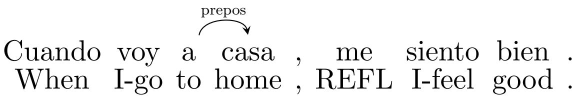

obl_obj3
Lo compró por 10 euros .
it she-bought for 10 euros .


obl_obj3
obl_obj2
Lo movió desde aqui hasta allí .
It she-moved from here to there .

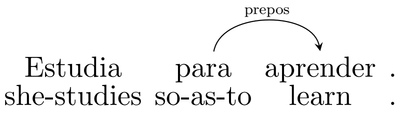
Prepositional	
Criterion	**Possible values**
PoS Gov	Prep
prototypical Dep	N
PoS Dep	Conj | $V_{FinRelatNoAnt}$ | V_{Inf} | Prep | Adv | Num | N
governed preposition	NO
governed grammeme	fin=INF | case=ABL
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.37: Distinctive properties of the *prepos* SSynt DepRel

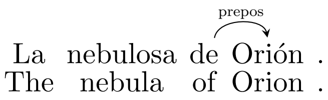
Pasó completamente de él .
she-ignored totally ∅ him .



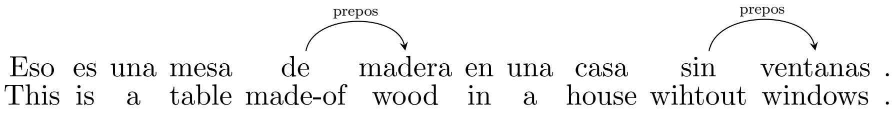
Cuando voy a casa , me siento bien .
When I-go to home , REFL I-feel good .

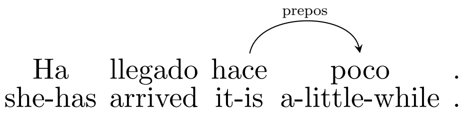


Estudia para aprender .
she-studies so-as-to learn .

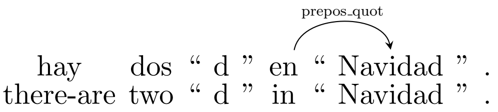


La nebulosa de Orión .
The nebula of Orion .



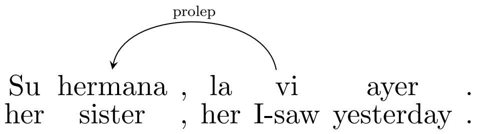


The quotative variant can have a dependent of almost any PoS.

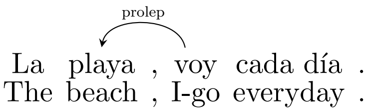


Prolepsis	
Criterion	**Possible values**
PoS Gov	Any
prototypical Dep	N
PoS Dep	Conj | V_{Inf} | V_{Ger} | Prep | Adv | N
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	N/A
punctuation	YES

Table A.38: Distinctive properties of the *prolep* SSynt DepRel



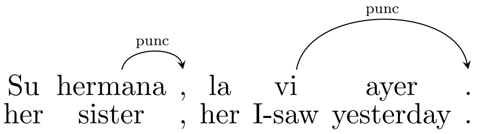
Su hermana , la vi ayer .
her sister , her I-saw yesterday .



La playa , voy cada día .
The beach , I-go everyday .

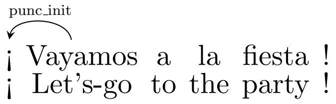
Punctuational	
Criterion	**Possible values**
PoS Gov	Any
prototypical Dep	-
PoS Dep	Punc
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.39: Distinctive properties of the *punc* SSynt DepRel



Punctuational initial	
Criterion	**Possible values**
PoS Gov	Any
prototypical Dep	-
PoS Dep	Punc
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

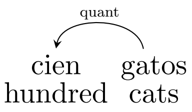
Table A.40: Distinctive properties of the *punc_init* SSynt DepRel

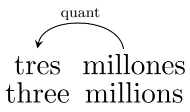


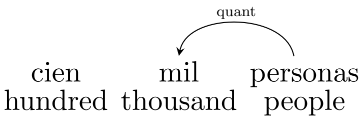
¡ Vayamos a la fiesta !
¡ Let's-go to the party !

Quantitative	
Criterion	**Possible values**
PoS Gov	N
prototypical Dep	-
PoS Dep	Num | Adj
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	dep=CONTROL
agreement with	Gov
variant inflection	YES
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

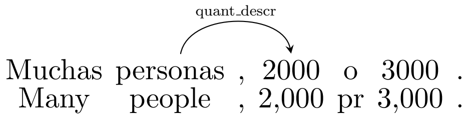
Table A.41: Distinctive properties of the *quant* SSynt DepRel







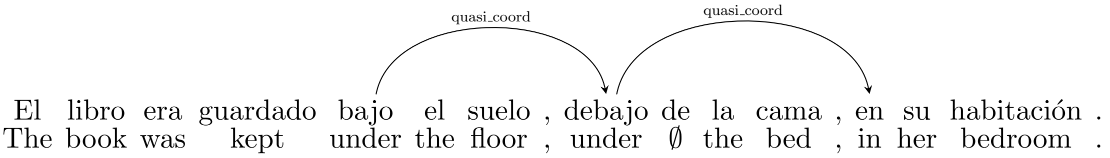
The backgrounded variant of this DepRel is *quant_descr*. One notable difference with *quant* is that *quant_descr* is usually on the right of its governor.



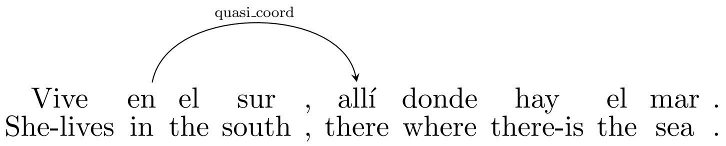
Muchas personas , 2000 o 3000 .
Many people , 2,000 pr 3,000 .

Quasi-coordinative	
Criterion	**Possible values**
PoS Gov	V_{Fin} $\mid$ $V_{FinRelatNoAnt}$ $\mid$ V_{NoFin} $\mid$ N $\mid$ Adj $\mid$ Adv $\mid$ Prep $\mid$ Conj $\mid$ Date
prototypical Dep	-
PoS Dep	V_{Fin} $\mid$ $V_{FinRelatNoAnt}$ $\mid$ V_{Ger} $\mid$ V_{Inf} $\mid$ V_{Part} $\mid$ N $\mid$ Adj $\mid$ Adv $\mid$ Prep $\mid$ Conj $\mid$ Date
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	COORD
left disloc = strong focus	NO
punctuation	YES

Table A.42: Distinctive properties of the *quasi_coord* SSynt DepRel



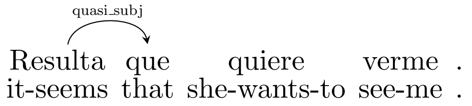
El libro era guardado bajo el suelo , debajo de la cama , en su habitación .
The book was kept under the floor , under $\emptyset$ the bed , in her bedroom .

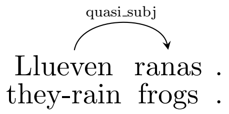


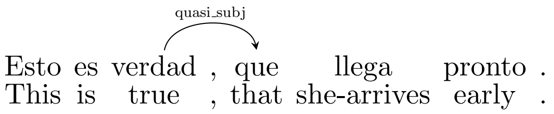
Vive en el sur , allí donde hay el mar .
She-lives in the south , there where there-is the sea .

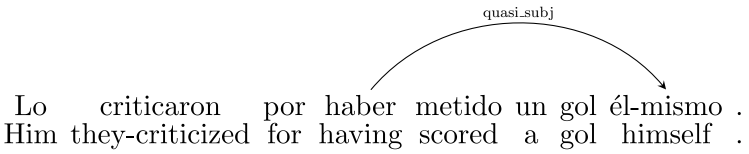
Quasi-subjectival	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin} | Adv
prototypical Dep	N
PoS Dep	Conj | Prep | N
governed preposition	NO
governed grammeme	NO
type of linearization	N/A
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	N/A
dependency	SUBORD
left disloc = strong focus	YES
punctuation	N/A

Table A.43: Distinctive properties of the *quasi_subj* SSynt DepRel



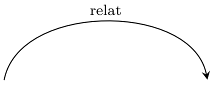




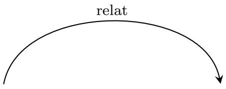


Relative	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N | Adv | Date
prototypical Dep	V
PoS Dep	V_{Fin}
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	NO

Table A.44: Distinctive properties of the *relat* SSynt DepRel

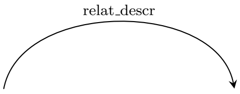


El edificio que van a destruir está viejo .
The building that the-are-going to destroy is old .



El edificio al que vamos es viejo .
The building to which we-go is old .

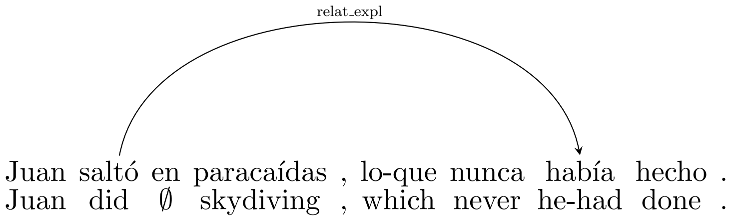
The backgrounded variant of this DepRel is *relat_descr*.



Este artículo , que mandé el més pasado , ha sido rechazado .
This paper , which I-submitted the month before , has been rejected .

Relative explicative	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin} | N | Adj | Adv | Num | Prep | Conj
prototypical Dep	V
PoS Dep	V_{Fin}
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	NO
punctuation	YES

Table A.45: Distinctive properties of the *relat_expl* SSynt DepRel

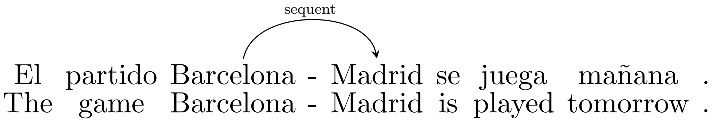


Juan saltó en paracaídas , lo-que nunca había hecho .
Juan did ∅ skydiving , which never he-had done .

Sequential	
Criterion	**Possible values**
PoS Gov	V_{NoFin} | N | Adj | Adv | Num
prototypical Dep	-
PoS Dep	V_{Inf} | V_{Part} | Adj | Adv | N | Num
governed preposition	NO
governed grammeme	NO
type of linearization	FIXED
canonical order	RIGHT
adjacency to Gov	YES
cliticization	NO
promotion	NO
demotion	NO
agreement	NO
agreement with	-
variant inflection	-
Dep omissibility	NO
dependency	SUBORD
left disloc = strong focus	NO
punctuation	YES

Table A.46: Distinctive properties of the *sequent* SSynt DepRel

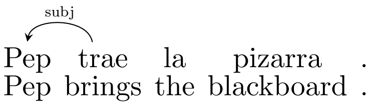
Trabaja en la interacción hombre - máquina .
She-works on the interaction man - machine .

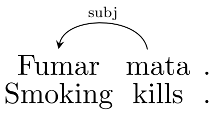


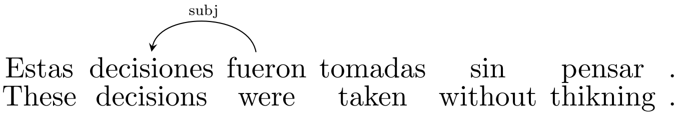
El partido Barcelona - Madrid se juega mañana .
The game Barcelona - Madrid is played tomorrow .

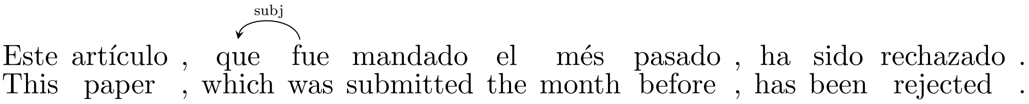
Subjectival	
Criterion	**Possible values**
PoS Gov	V_{Fin}
prototypical Dep	N
PoS Dep	Conj $\mid$ $V_{FinRelatNoAnt}$ $\mid$ V_{Inf} $\mid$ N
governed preposition	NO
governed grammeme	NO
type of linearization	FREE
canonical order	LEFT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	N/A
agreement	dep=CONTROL
agreement with	Gov
variant inflection	YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	N/A
punctuation	NO

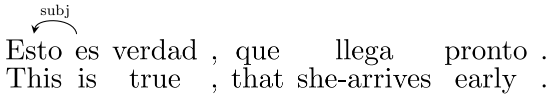
Table A.47: Distinctive properties of the *subj* SSynt DepRel



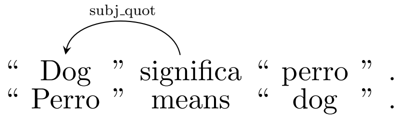








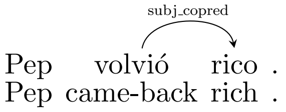
Esto es verdad , que llega pronto .
This is true , that she-arrives early .



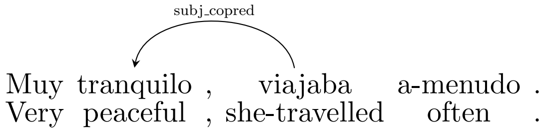
“ Dog ” significa “ perro ” .
“ Perro ” means “ dog ” .

Subject copredicative	
Criterion	**Possible values**
PoS Gov	V_{Fin} | V_{NoFin}
prototypical Dep	Adj
PoS Dep	V_{Ger} | V_{Part} | Adj | Prep
governed preposition	NO
governed grammeme	NO
type of linearization	FREE
canonical order	RIGHT
adjacency to Gov	N/A
cliticization	NO
promotion	NO
demotion	NO
agreement	NO | dep=TARGET
agreement with	- | Subject
variant inflection	- | YES
Dep omissibility	YES
dependency	SUBORD
left disloc = strong focus	N/A
punctuation	N/A

Table A.48: Distinctive properties of the *subj_copred* SSynt DepRel



Pep volvió rico .
Pep came-back rich .



Muy tranquilo , viajaba a-menudo .
Very peaceful , she-travelled often .

Appendix B

Sample outputs of the deep generators

The outputs presented here have not been post-processed; they are shown as they are returned by the deep generator. They have not been selected ono by one; rather, there were picked ramdomly in the output of the test sets of each experiment.

Non-isomorphic generation: all funtional words have been removed from the input (Spanish)

- Está previsto que el gabinete de Mori decidieron formalmente el 25 de junio como la fecha para los comicios que es los primeras elecciones general desde octubre de 1996 en los los de día transcurso .
- A lo mejor existen de verdad de esas peces abisal minúsculo dentro de nosotros y lo que ocurre es que podrá ver lo sólo en raro ocasiones .
- Pese a ello la 74 por ciento la israelí opinar Weizmanndo ha sido un buen presidente .
- Que ocho meses se tardan para dar un permiso de trabajo de residencia un a ciudadanos polaca no puede ser en un momento de falta de mano de obra que actuar como freno el crecimiento económico en Pujol opiniones según Pujol .
- Toledo señaló que el de gobierno en este elecciones quería llevar nos un a trampa y quiso repetir el misma fraude de los primera vuelta .

- El Kremlin oficialmente anunció hoy la próximo viaje España el ruso presidente invitado el rey Juan Carlos I Vladímir Puti .
- El 13 de junio la campaña oficial de doce días comenzó .
- Se llama fosfeno y sean divertido seguir la mientras sobreandar en la inestable marea de los ojos .
- Según mi noticia eso de los tinnitus podía deber se un simple tapón de cera o la inflamación un membrana.

Isomorphic generation: all lemmas and punctuations in the input (Spanish)

- entró en silencio absoluto Desde entonces .
- Nadie sabe cuál es la nueva fecha que propone para las votaciones ni si las quiere juntas o separadas , , ni cuando va a reanudar la campaña .
- “ El pueblo puede estar seguro de que , no existe aquí ni por esos motivos tampoco nadie está preso nada de conspiración ” , declaró Hurtado en una rueda de prensa .
- La amnistía favorece a los catorce coroneles detenidos y al más de un centenar de oficiales de menor rango procesados por participar en la asonada golpista contra Mahuad que facilitó la sucesin en la presidencia de GustavoNoboa .
- Noboa , que fue vicepresidente en el gobierno de Mahuad y le sucedió en el cargo tras su caída , considera que la amnistía permitirá la pacificación de la nación y la creación de un ambiente propicio para el diálogo y la concertación .
- Y es que los coroneles rebeldes gozan entre la población de una amplia simpatía , pues la mayoría considera positivo , según las encuestas , elque hayan apoyado a los movimientos sociales que exigían la salida de Mahuad , acusado de haber ahondado la crisis económica que afecta al país.
- El cabecilla del movimiento militar fue el coronel Lucio Gutiérrez , quien apoyó a los miles de indígenas que ocuparon el Palacio Legislativo el 21 de enero y luego marcharon hacia el centro de Quito para tomar posesión de la Casa Presidencial .

- Gutiérrez no se arrepiente de haber participado en la insurrección contra Mahuad y está seguro que la actitud de los oficiales se debió al elevado grado de corrupción que hubo durante la administración anterior dice .
- El coronel quiere concluir su carrera militar brillante , aunque aún debe esperar las posibles sanciones disciplinarias que le podría imponer el mando militar .
- , pues el recurso político sólo establece la suspensión de procesos civiles penales y los seguidos en la Corte de Justicia Militar La amnistía , según opiniones de diputados , no impide que las autoridades militares impongan sanciones disciplinarias a los oficiales involucrados .

Hybrid generation: a few fuctional words have been removed from the input; nodes are introduced with rules (English)
(a) Outputs obtained on our automatic annotation of the PTB/PB/NB

- The economy 's temperature will be taken this week from several vantage points , with readings on trade , output , housing and inflation .
- The most troublesome report may be the August merchandise trade deficit out due tomorrow .
- The trade gap is expected to widen from $ 7.6 billion July 's to about $ 9 billion , according to a survey by MMS International , a unit of McGraw - Hill Inc. New York , .
- Thursday 's report on the September consumer price index is expected to rise sharply as not as the 0.9 % gain reported Friday in the producer price index although , .
- That gain was being cited as a reason early in Friday 's session , the stock market was down before it got started on its reckless 190 - point plunge .
- Views on manufacturing strength are split between economists and those who use the total comforting more somewhat employment figures in their calculations who read as a sign of a slowdown September 's low level of factory job growth .

(b) Output provided for the human evaluation of the SRST 2011
But re - exports mainly from China jumped 75 % to HK$ 15.92 billion . Domestic exports fell 29 % in 1989 's first seven months to HK$ 3.87 billion , while re - exports rose 56 % to HK$ 11.28 billion . Manufacturers say there is no immediate substitute for southern China , an estimated 120,000 people are where employed by the toy industry . " For the next few years , China like it or not is going to be the main supplier , " says Edmund Young , vice president of Perfecta Enterprises Ltd. , one of the biggest Hong Kong first toy makers , move across the border . In the meantime , as manufacturers and buyers seek new sites they are focusing mainly on Southeast Asia . Junk 's collapse helped stoke the panicky selling of stocks , that produced the deep one - day dive in the Dow Jones Industrial Average since the Oct. 1987 19 crash . It also helped trigger this year 's big rally in the U.S. government bond market simultaneously , as investors rushed to move capital into the highest - quality securities they could find . But " Friday an eerie silence pervaded the junk market as prices tumbled on hundreds of high - yield bonds despite no " active trading , " , " says John Lonski , an economist at Moody 's Investors Service Inc .