Procesamiento del Lenguaje Natural, Revista nº 52 marzo de 2014, pp 53-60 recibido 15-11-13 revisado 06-02-14 aceptado 07-02-14

From constituents to syntax-oriented dependencies

De constituyentes a dependencias de base sintáctica

Benjamin Kolz, Toni Badia, Roser Saurí
Universitat Pompeu Fabra
c. Roc Boronat 138, Barcelona 08018
{benjamin.kolz, toni.badia, roser.sauri}@upf.edu

Resumen: El presente artículo describe el proceso automático de construir un corpus de dependencias basado en la estructura de constituyentes de Ancora. El corpus Ancora ya tiene una capa de información de dependencias sintácticas, pero la nueva anotación aplica criterios puramente sintácticos y ofrece de este modo un nuevo recurso a la comunidad investigadora en el campo del procesamiento del lenguaje. El artículo detalla el proceso de reanotación del corpus, los criterios lingüísticos empleados y los resultados que se han obtenido.
Palabras clave: análisis de dependencias, etiquetario de funciones sintácticas, anotación de corpus, conversión de constituyentes a dependencias

Abstract: This paper describes the automatic process of building a dependency annotated corpus based on Ancora constituent structures. The Ancora corpus already has a dependency structure information layer, but the new annotated data applies a purely syntactic orientation and offers in this way a new resource to the linguistic research community. The paper details the process of reannotating the corpus, the linguistic criteria used and the obtained results.
Keywords: dependency parsing, syntactic function tagset, corpus annotation, conversion from constituents to dependencies

1 Introduction

Syntax information, which is crucial in many NLP tools, can be represented by means of constituent structures or dependency relations. While each of these formalisms has its advantages and disadvantages and there is an ongoing debate on preferred uses of them, it is worth noting that dependency-based representations can also vary depending on the linguistic criteria they are based upon (Kübler, McDonald and Nivre, 2009:5-6): from purely syntactically oriented to semantically motivated.

Most current approaches to dependency functions within NLP embrace an (at least partial) semantic orientation, e.g., most notably, the Stanford parser (De Marneffe and Manning, 2012) and, in the case of Spanish, the Ancora corpus (Taulé, Martí, Recasens, 2008) and any parser trained on that. By contrast, the current article focuses on the automatic creation of a corpus of dependency relations for Spanish based on purely syntactic criteria.

The paper is structured as follows. The next section motivates this project, section 3 reviews the related works, section 4 presents the corpus on which the experiment was run, section 5 discusses the linguistic criteria applied, and the automatic annotation process is detailed in section 6. Finally results are presented in section 7. The article ends with some final considerations and a look into future work (section 8).

2 Motivation

Dependency relations can be grounded on different criteria: from purely syntactic to semantically oriented. Take for example the noun phrase *el resto de los chicos* (‘the rest of the boys’). A syntactic view will consider *resto* as its head, whereas a semantic approach will take *chicos* as the main element. The same tension between syntactic and semantic heads can be found in other constructions throughout the language, e.g., verbal periphrases, modification relations, etc.

ISSN 1135-5948

Choosing a specific dependency analysis depends on the future use of the data. For instance, semantic-oriented trees may be preferable for certain information extraction tasks. By contrast, a purely syntactic analysis offers a neutral ground for any task. However, in many cases there are no corpus resources compliant to the specific approach that is needed. Then, one can just build the NLP tool based on the available data, or create a neutral, syntax-based resource so that future, more semantics oriented and task based, dependency annotations can be generated. We chose this latter path as in our opinion the linguistic criteria in the input to any NLP tool should be adequate to it and not the other way around.

For our research goals we worked with the corpus Ancora (Taulé, Martí, Recasens, 2008), which is annotated with both constituent and dependency structures. However, dependency relations in Ancora are semantics-oriented, and we wanted a purely syntax-based annotation. Thus, we decided to build a further layer of dependency relations based on this other approach. Considering the large size of Ancora, we proceeded by automatic means from the layer of constituent structure. The process consists of two individual tasks: dependency relation annotation and, afterwards, syntactic function labeling.

3 Related Work

The conversion from constituent to dependency structures is not new. Magerman (1994) made use of a head driven approach, which is still used and enhanced in newer works such as Collins (1999), Yamada and Matsumoto (2003) and Johansson and Nugues (2007). The approach has shown good results but there is still ongoing research.

As can be seen in such previous works, the resulting dependency tree structure depends highly on the focus of the annotation, which can apply either a syntactic or a semantic analysis. Johansson and Nugues (2007) mention the possibility to allow multiple-headed dependency structures to overcome this dichotomy.

In the particular case of the Ancora corpus, it is worth noting that its dependency relations annotation was carried out automatically by a conversion from constituents (Civit, Martí and Bufí, 2006). Only a head and a function table were written manually. In many constructions, implicit semantic criteria are assumed in the linguistic decisions informing the conversion.

Along similar lines, Mille et al. (2009) present a reannotation of Ancora dependencies, already heading towards a more syntax-oriented approach. Their reannotation has been carried out semiautomatically and currently covers only a section of Ancora (100,892 out of 517,269 tokens). Their function tagset consists of 69 tags and so is quite fine-grained for an automatic annotation. Given this and the fact that the resulting annotation is not yet available for the whole corpus, we decided to create our own tagset and proceed with an automatic annotation of the whole corpus.

4 Corpus

For our experiments, we used the Spanish part of Ancora (Taulé, Martí, Recasens, 2008), which contains 17,376 sentences split over 1,636 files gathering a total count of 517,269 tokens. Ancora is annotated for different linguistic levels, including constituent structures and dependency relations. All sentences are tokenized, and tokens have information on their lemma and part-of-speech. Other annotation layers include:

- syntactic constituents and functions
- argument structure and thematic roles
- verb semantic classes
- denotative type of deverbal nouns
- WordNet synsets for nouns
- named entities
- coreference relations

5 Linguistic Criteria

This section details the linguistic criteria we adopted for grounding the dependency relations in our automatic annotation. First we focus on the structure of the dependency relations and then on their function labeling.

5.1 Dependency relations

The goal of this annotation is to obtain pure syntax-oriented dependency trees. Thus, our linguistic decisions are compliant to that.

Periphrastic verbs. In our annotation, auxiliary and modal verbs are the head of the structure, as shown below. In this and the following examples, the upper graph shows the Ancora treatment and the lower one our decision. [1]

[1] The head of the arrow leads to the dependent.

(1) debía enviar los rollos

(he had to send the reels)

(2) ella ha estado viendo la exposición

(she has been seeing the exhibition)

Ancora applies here an approach based on semantic criteria, so that the head is the main verb, while the conjugated auxiliary verb is a dependent of former.

As the auxiliary verb is in agreement to the subject, we wanted subjects to depend on the auxiliary or modal (as marked by the agreement relation) and other complements, on the main verb.

(3) debía enviar los rollos

(he had to send the reels)

(4) ella ha estado viendo la exposición

(she has been seeing the exhibition)

Complex nominal phrases. The treatment of complex nominal phrases like *el resto de los chicos* ('the rest of the boys') illustrates the differences between a semantic and a syntactic approach.

(5) una docena de los participantes.

(a dozen of the participants)

Coordinations. A coordination structure contains at least two elements which are coordinated by one or more conjunctions. Head candidates are one of the coordinated items or one of the conjunctions. Ancora sees the first coordinated element as head, while we decided to identify as head the conjunction.

(6) Juan y María

(Juan and María)

In case of coordinations with paired conjunctions (e.g., *ni…ni…,* 'neither…nor…'*)*, we treated the last conjunction as the head of both the conjuncts and any former conjunction or comma.

(7) Ni ministro ni excelencia.

(Neither minister nor excellency.)

Our approach has the advantage that all coordinated elements depend on the same node and can be found at the same level within the dependency tree.

Subordinating conjunctions. The conjunction is the head of the subordinated clause, in full accordance to the surface syntactic structure. By contrast, Ancora identifies the verb of the subordinated clause as head and sees the conjunct as its dependent.

(8) Amo Boston, aunque ahora vivo en York.

(I love Boston, although I now live in York.)

Relative clauses. The verb of the relative clause is also its head, while the relative pronoun is its dependent. This case has been treated differently than other subordinating structures given the double role of the relative pronoun (as connector and as argument of the main predicate in the subordinated clause).

(9) una mirada que traspasaba el techo

(a view which penetrated the roof)

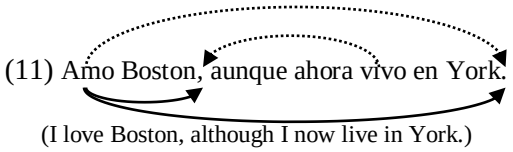
Our analysis corresponds to the same treatment as seen in Ancora.

Comparative Structures. The comparative element (e.g., *más* below) depends on the adjective (*correcta*) and at the same time is the head of the embedded phrase (*que la otra*).

(10) una decisión más correcta *que* la otra

(a more correct decision than the other)

Punctuation. Commas and full stops are seen as dependent of the higher constituent head. Brackets, quotation marks, etc. are seen as dependent of the head within their constituent range.



5.2 Function Tagset

The syntactic functions tagset has to fulfill two requirements. It has to be as informative as possible and must be of reasonable size in order to guarantee a successful automatic annotation.

The tagset used in Ancora has around 50 tags, thus being of a reasonable size. However, it has the problem of mixing dependency relations with part-of-speech and constituent structure tags. Some examples:

- **Dependency function tags**: suj (subject), cd (direct object), ci (indirect object).
- **Constituent structure tags:** sn (nominal phrase), s.a (adjectival phrase).
- **Part-of-speech tags:** v, n.

On the other hand, the Stanford tagset (de Marneffe and Manning, 2012) seems to be adequate for both requirements. The size of 53 tags is reasonable for an automatic annotation and the individual tags are a good choice to represent dependency relations information. In addition tags are structured in a hierarchical way, thus allowing underspecified tags when required. In our proposal, we adapted Stanford's tagset for Spanish (e.g., *reflec*, reflexive) and enhanced it with some tags already available in Ancora (e.g., *te*, textual element) in order to increase its informativeness.

Our tagset is presented in Table 1. It contains 42 function tags (including underspecified ones), which makes it fully adequate for automatic annotation (section 6.2). In the table, indentation shows the tagset hierarchical structure, conveying that general tags like *obj* or *mod* include more specific subclasses. In the annotation, the goal is obviously to be as specific as possible, as this leads to more informative data. Therefore the generic tags like *dep*, *comp*, *obj*, *mod* and *prep* are not expected to be of common use but only for cases where a more specific tag cannot be applied.

Tag	Full name
root	root
dep	dependent
arg	argument
comp	complement
attr	attributive
cpred	predicative complement
obj	object
cobj	complementizer object
dobj	direct object
iobj	indirect object
oobj	oblique object
pobj	object of a preposition
vobj	object of verb
crobj	object of comparative
subj	subject
nsubj	nominal subject
csubj	clausal subject
coord	coordination
conj	conjunct
agent	agent
reflec	reflexive ("se")
te	textual element
mod	modifier
abbrev	abbreviation modifier
amod	adjectival modifier
appos	appositional modifier
advcl	adverbial clause modifier
det	determiner
infmod	infinitival modifier
partmod	participial modifier
advmod	adverbial modifier
neg	negation modifier
rcmod	relative clause modifier
nn	noun compound modifier
tmod	temporal modifier
num	numeric modifier
prep	prepositional modifier
prepv	prep. mod. of a verb
prepn	prep. mod. of a noun
prepa	prep. mod. of adjective
poss	possession modifier
punct	punctuation

Table 1: Dependency function tagset

6 Automatic Dependency Annotation

6.1 Process

Our system takes the constituent structure layer in Ancora as input and builds the syntax-oriented dependency trees supported by linguistic rules.

The core of the process is identifying the head of each constituent, along the lines of Magerman (1994) and subsequent work. The dependent nodes can then be pointed to the identified head. One single main rule selects the head in all clearly headed constituents in

the corpus. However a remarkable number of constituent structures in Ancora are not clearly headed, because they are flat structures or conflate several nodes into one (e.g. the verbal group formed by the main verb and its auxiliaries or modals). To tackle these cases a set of nine finer grained rules are added (two for flat constructions and seven for divergence in head selection).

Once the dependency structures are obtained, the syntactic function of each head-dependent pair is determined.

The function labeling process is informed with data from two sources: the part-of-speech of both nodes in each pair, and the argument-structure function tags that had been manually annotated in the Ancora constituent structure layer (subject, direct and indirect object, oblique and textual element). Based on those two elements, rules can be established to automatically annotate the syntactic functions between head and dependent node.

6.2 Algorithm

The algorithm we applied is as shown in Figure 1.

```
1 function DEPENDENCY_ANNOTATION(parsed_text):
2    for sentence in constituents:
3        read_constituents_tree(sentence)
4        for constituent in constituents_tree:
5            identify_head_of_constituent(constituent)
6            # uses a preference list for possible candidates
7        for terminal_node in constituents_tree:
8            walk_constituents_tree(terminal_node)
9            # bottom-up
10           # walks tree until not head anymore and
11           #  connects there as dependent to head
12       for terminal_node in constituents_tree:
13           label_functions()
```

Figure 1: Algorithm

The procedure takes the parsed text as input (line 1), analyzes it sentence by sentence (line 2) and generates its dependency structures. In particular, the program reads the constituent tree of each sentence (line 3) and identifies the head of each constituent (line 5). The procedure then walks bottom-up from terminal nodes through the constituent structure and connects them to their head (line 8). Finally each relation between dependent and head is labeled according to the function tagset presented in Table 1 (line 13).

6.3 Issues

The conversion from constituent structures to dependency structures is highly dependent on the input that comes from the constituents. Thus inconsistencies in the constituent annotation may lead to problems when applying the general procedure.

Furthermore we encountered three specific issues: grouping of several lexical items as a single token (e.g., *la_mayoría_de*, 'the majority of'), in Ancora referred to as multiword, the depth of annotation in constituent trees (e.g., *debía haberlo resuelto*, 'should have solved it', as a flat structure), and the presence of empty tokens signaling subject ellipses.

Flat structures. Flat structures posed a problem for identifying heads and their dependents as they often contain several constituent heads: the head of the constituent and another head of what should have been a lower constituent, as underlined in (12).

(12) S=conj **S** **grup.verb** sa sn sp

In this example we would expect a deeper analysis grouping together also *grup.verb sa sn sp* to an S.

We tackled this problem by specific rules which detect flat structures and insert an intermediate structure introducing the different heads and their corresponding dependents. This way they can be treated as well-formed constituents.

Multiwords. In Ancora these include complex prepositions or conjunctions, verb groups, complex determiners and proper names. They are challenging because many of them are treated sometimes compositionally and sometimes as a single token:

(13) a. ya_que
b. ya que

For the moment, we have adapted our annotation to this multiword approach, but the deconstruction of them into individual tokens will be the next step in our project.

Empty elements. Another modification to the original Ancora annotation is the suppression of empty tokens which correspond to dropped subjects in Spanish. As these items do not appear in the text, we decided to not include them in the dependency tree.

7 Evaluation

7.1 Evaluation Corpus

The evaluation corpus was annotated manually for both dependency relations and syntactic functions. We annotated a total of 256 sentences which were chosen partially randomly; that is, we made sure that the selected files included all linguistic phenomena described in section 5.1 above. The evaluation corpus contains a total of 6,160 tokens (out of the 517,269 tokens in Ancora, which corresponds to a 1.5 % of the whole corpus in terms of number of files).

Figure 2 exemplifies the content and format of the evaluation corpus:

```
1#La #2#det
2#situación #10#nsubj
3#en #2#prepn
4#las #5#det
5#carreteras #6#coord
6#y #3#pobj
7#las #8#det
8#montañas #6#coord
9#se #10#reflec
10#normalizó #ROOT#root
11#en #10#prepv
12#todas #14#det
13#las #14#det
14#autonomías #11#pobj
15#afectadas #14#amod
16#. #10#punct
```

Figure 2: Evaluation corpus fragment

7.2 Results

The results obtained are highly satisfactory as the labeled attachment score (LAS) reached 0.85, the unlabeled attachment score (UAS) 0.92 and label accuracy (LA) a value of 0.89.

	Accuracy	**Kappa**
LAS	0.85	-
UAS	0.92	-
LA	0.89	0.88

Table 2: results

As syntactic function labels are likely to get an incorrect result if the corresponding node's head was not set correctly, we also calculated the label accuracy of the correctly identified attachments, which was 0.93.

The Kappa coefficient K for agreement between coders has been calculated in order to exclude the factor of agreement by chance. Among the two main ways of calculating Kappa we followed Cohen (1960) because it is better suited for cases where categories have significantly different distributions. In this case the coders were a human annotator and our system. The kappa value for syntactic function labels of 0.88 is in the range of almost perfect agreement according to Landis and Koch (1977).

Unfortunately, Civit, Martí and Bufí (2006) do not give results for their conversion from constituents to dependencies in their paper. These results would have been the best comparison for our results as they are based on the same corpus even if not tagged with the same function tagset.

7.3 Error Analysis

The error analysis splits into errors observed in the dependency relation identification task and errors in the labeling of the relation.

7.3.1 The dependency tree creation

Our data show that the system had problems with complex coordinated structures as, for example, citations which contain more than one sentence.

(14) He said: "Sentence 1. Sentence 2"

In addition, the rules which treated flat constituent structures were not always able to create the correct dependencies for deeper nodes.

7.3.2 Function labeling

The results and exact frequencies of agreement and disagreement between our manual annotation and the system's one are presented in a confusion matrix (table 3) which counts only the labels of correctly related dependencies.

As the matrix shows, the system had problems with some coordination structures. 72 out of 348 cases showed an incorrect label. Problems came up especially in cases of complex structures, particularly with correlative conjunctions (like *bien... bien*… 'either… or…').

In other cases the rules were too generic, as the one for labeling the function *attr*. The system looks at the head lemma and sets *attr* if it is *ser* ('to be'). Cases were found in which the label was wrongly used in passive contexts like *han sido absueltos* ('they were absolved'). The confusion matrix shows that in 10 out of

64 cases the system wrongly identifies the function as being *attr* instead of *vobj*. In this and similar cases, the rule needs to be written in a more specific way.

Furthermore, the system does not include rules for the use of generic labels like *obj*. Thus it always assigns a specific label and if this does not fit, it currently assigns the label *dep*.

Some not so frequently used labels like *nn* or *abbrev* could not be tested as they did not appear within the evaluation corpus.

8 Final Considerations

The approach presented in this work shows to work in a satisfactory way and the new annotation offers a further source of linguistic data for the research community.

There is still work left as we want to deconstruct Ancora multiwords into individual tokens and train a parser with the resulting data to work over unseen text.

Our new annotation adds value to the original Ancora annotation as dependency structures are now available according to two different points of views (semantic and now also syntactic) and can serve as basis for further research.

We plan to improve the results by adjusting some of the identified problems in the rules, testing the approach in corpora of different domains and make the data publicly available in the coming future (accessible on www.upf.edu/glicom/).

9 Bibliography

Carletta, J. 1996. Assessing agreement on classification task: the kappa statistic. *Computational Linguistics*, 22(2):249-254.

Civit, M., Marti M. A., and Bufi, N. 2006. Cat3LB and Cast3LB: From Constituents to Dependencies. In *Proceedings of the 5th International Conference on Natural Language Processing*, FinTAL, p. 143-151, Turku, Finland. Springer Verlag LNAI 4139.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37-46.

Collins, M. 1999. Head-driven statistical models for natural language parsing. Ph.D. thesis, University of Pennsylvania, Philadelphia.

De Marneffe, M. and Manning, C. D. 2012. Stanford typed dependencies manual. Technical report, Stanford University.

Johansson, R. and Nugues, P. 2007. Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conf. on Computational Linguistics (NODALIDA)*, p. 105-112.

Kübler, S., McDonald, R. and Nivre, J. 2009. Dependency Parsing. Morgan & Claypool.

Landis, J.R. and Koch, G.G. 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1): 159–174.

Magerman, D. 1994. Natural language parsing as statistical pattern recognition. Ph.D. thesis, Stanford University.

Mille, S., Burga, A., Vidal, V. and Wanner, L. 2009. Towards a Rich Dependency Annotation of Spanish Corpora. In *Proceedings of SELPN´09*, San Sebastian, p. 325-333.

Taulé, M., Martí, M. A., and Recasens, M. (2008). AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In ELRA (Ed.), *LREC*, Marrakech, Morocco, p. 96-101.

Yamada, H. and Matsumoto, Y. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of 8th InternationalWorkshop on Parsing Technologies*, p. 195-206.

manual | system	root	dep	arg	comp	attr	cpred	obj	cobj	dobj	iobj	oobj	pobj	vobj	crobj	subj	nsubj	csubj	coord	conj	agent	reflec	te	mod	abbrev	amod	appos	advcl	det	infmod	partmod	advmod	neg	rcmod	nn	tmod	num	prep	prepv	prepn	prepa	poss	punct
root	242																																								
dep		2							1																																
arg																																									
comp									8																																
attr				58	3		1	1								1														1					4						
cpred		4			17				1																	2						1				1	1				
obj		5																																								
cobj		2						94	4									1	1																						
dobj		2						4	294		2					3	2	2										1							2	1		7			7
iobj		1							6	30											1																1			1
oobj										1	37																										8			
pobj								19	1			863	1					1										6		2						4				
vobj					10			1				7	46																											
crobj														3																										
subj																																								
nsubj					1	1		1	4			1				312		5																						
csubj					3				1								7																								
coord		17						41								1		276	6									1		6										
conj								1											1									2												
agent																				8																				
reflec									2												65																			
te																						10																		
mod									2									1					1					1												
abbrev																																									
amod		5			3							1				1		1							306	2										2					
appos		7							1																1	43														
advcl		2						2	6																		24										1			
det		1										1						1										823								1				
infmod								1																													2	1	1	
partmod						1																																			
advmod		1							5							1														136	1									
neg																															34									
rcmod		7						1										2														82				4				
nn																																									
tmod		2							4																					3			8		4				
num									2			8				1																		36						
prep								1										1																	9					
prepv		7				1		1		5								1	2								5			2					212	1	2		
prepn		9																1	1						3									3	2	393	2		
prepa																		1									1										33		
poss																																						51		
punct																																								705

Table 3: confusion matrix for functions in evaluation corpus (only correct attachments)