

Exploring Morphosyntactic Annotation Over a Spanish Corpus for Dependency Parsing

Miguel Ballesteros Simon Mille Alicia Burga

Natural Language Processing Group

Pompeu Fabra University

Barcelona, Spain

{firstname.lastname}@upf.edu

Abstract

It has been observed that the inclusion of morphosyntactic information in dependency treebanks is crucial to obtain high results in dependency parsing for some languages. In this paper we explore in depth to what extent it is useful to include morphological features, and the impact of diverse morphosyntactic annotations on statistical dependency parsing of Spanish. For this, we give a detailed analysis of the results of over 80 experiments performed with MaltParser through the application of MaltOptimizer. Our goal is to isolate configurations of morphosyntactic features which would allow for optimizing the parsing of Spanish texts, and to evaluate the impact that each feature has, independently and in combination with others.

1 Introduction

As shown in natural language processing (NLP) research, a careful selection of the linguistic information is relevant in order to produce an impact on the results. In this paper, we want to look into different sets of morphosyntactic features in order to test their effect on the quality of parsing for Spanish. To this end, we apply MaltParser (Nivre et al., 2007b), and MaltOptimizer (Ballesteros and Nivre, 2012b; Ballesteros and Nivre, 2012a), 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 morphosyntactic features to build different models and see the impact of each feature set on the Labeled Attachment Score (henceforth LAS) of the parser.

We decided to 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, we used the UPF version of a subsection of the AnCora corpus (Mille et al., 2013) (see also Section 3.2), which includes features such as number, gender, person, mood, tense, finiteness, and coarse- and fine-grained part-of-speech (PoS). 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. The version of the AnCora-UPF corpus that we use contains 41 language-specific syntactic tags and thus is perfectly suitable for our task.

In the rest of the paper, we situate our goals within the state-of-the-art (Section 2), we describe the experimental setup, i.e. MaltParser, MaltOptimizer, the corpus used and the experiments that we carried out (Section 3), we report and discuss the results of the experiments (Section 4), and finally present the conclusions and some suggestions for further work (Section 5).

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 incorporate 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 op-

timizer 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 those nominal phrases that can be either subjects or objects. In order to do that, they use the optimizer to detect the features (including morphosyntactic 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). In this way, 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. Our work is directly related to the modeling challenge, given that we analyze in depth whether it is useful to incorporate morphological information as independent features.

Eryigit et al. (2008) have already contributed to this topic by testing different morphosyntactic 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). We also find interesting the work of Bengoetxea and Gonenola (2009) and Atutxa et al. (2012), which have respectively tried to include semantic classes and feature propagation between different parsing models, with the intention of improving the parsing results for Basque. However, none of these works made use of MaltOptimizer in their experiments, for the simple reason that it was not available at the time.

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) involving 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 the AnCora-UPF corpus (see (Burga et al., 2011; Mille et al., 2013) for details about how the tagset was designed), so we expect that the presence or absence of one or more of these features in the training corpus will have a clear impact on the quality of the parsing.

In this way, the work of (Cowan and Collins, 2005) makes a step exploring how specific morphologic features (encoded as different PoS) affect the parsing results in Spanish. Even though the authors use a constituent-based treebank and not a dependency-based one, they find that *number* and *mood* (verbal feature that overlaps our *mood* and *finiteness*) are the features that most affect the parser’s behaviour.

3 Experimental Setup

Here are the five steps we followed:

1. The corpus was 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 were created, based on different combinations of morphosyntactic features;
3. MaltParser was 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 applied MaltOptimizer Phase 3, on each of the 82 training sets, and each configured model output was applied to the test set in order to obtain an evaluation;

5. We retained 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 41 edge types.¹

In the rest of this section, we give more details about MaltParser and MaltOptimizer, before explaining the annotation that is used as the basis of this experiment.

3.1 MaltParser, MalOptimizer and the CoNLL Data Format

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,² and provides models capable of producing the dependency parsing of new sentences. MaltParser implements four different transition-based parsers families and provides high and stable performance (see, e.g., (Mille et al., 2012)). 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 (see Figure 1 for details). The different transitions are shown in Figure 2; 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, shown in Figure 3. This basically means that the better the linguistic annotation is, the better the results are expected to be.

The CoNLL data format is now a standard for dependency parsers; the following attributes are the ones included in the CoNLL-X format that are used as features by the parser:

1. **FORM**: Word form.
2. **LEMMA**: Stemmed version of the word.

¹Because each training set contains different features, the test sets are obviously parsed differently and, in some cases, not all of the 41 dependency relations were predicted by the parser.

²<http://ilk.uvt.nl/conll/#dataformat>

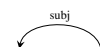
INITIAL-STATE

[ROOT] { } [Eso es lo que hicieron .]

... (some hidden transitions)

LEFT-ARC

[ROOT] { Eso } [es lo que hicieron .]



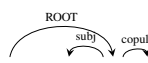
RIGHT-ARC

[ROOT Eso es] { } [lo que hicieron .]



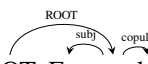
RIGHT-ARC

[ROOT Eso es lo] { } [que hicieron .]



SHIFT

[ROOT Eso es lo que] { } [hicieron .]



... (some hidden transitions)

RIGHT-ARC

[ROOT Eso es lo que hicieron .] { } []

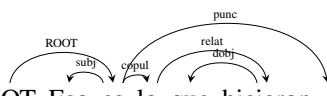


Figure 1: Some of the parsing transitions of the sentence included in the AnCora-UPF corpus: *Eso es lo que hicieron - That's what they did*. The buffer is the structure that is represented to the right of the picture between '[' and ']', and the stack is the one to the left. Between each parsing state we show the transitions selected by the parser considering the features over the stack and the buffer.

3. **CPOSTAG**: Coarse-grained part-of-speech tag.
4. **POSTAG**: Fine-grained part-of-speech tag.
5. **FEATS**: List of morphosyntactic features (such as number, gender, person, case, finiteness, tense, mood, etc.)
6. **DEPREL**: Dependency relation to head.³

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

³These six attributes are located in columns 2, 3, 4, 5, 6, 8 respectively in Figure 3.

Nivre’s transition system:

$$Initial = \langle [], [w_1 \dots w_n], \emptyset, \emptyset \rangle \rightarrow Final = \{ \langle \Pi, [], H, \Delta \rangle \in C \}$$

Transitions:

Shift	$\langle \Pi, w_i \beta, H, \Delta \rangle \Rightarrow \langle \Pi w_i, \beta, H, \Delta \rangle$
Reduce	$\langle \Pi w_i, \beta, H, \Delta \rangle \Rightarrow \langle \Pi, \beta, H, \Delta \rangle$
Left-Arc (<i>dr</i>)	$\langle \Pi w_i, w_j \beta, H, \Delta \rangle \Rightarrow \langle \Pi, w_j \beta, H[w_i \rightarrow w_j], \Delta[w_i(dr)] \rangle$ if $h(w_i) \neq 0$.
Right-Arc (<i>dr</i>)	$\langle \Pi w_i, w_j \beta, H, \Delta \rangle \Rightarrow \langle \Pi w_i w_j, \beta, H[w_j \rightarrow w_i], \Delta[w_j(dr)] \rangle$ if $h(w_j) = 0$

Figure 2: Transition System for Nivre’s algorithms with *reduce* transition (Nivre et al., 2007b).

1	Los	e1	A	DT	gender=MASC number=PL spos=determiner	2	det	-	-
2	Mbitis	mbitis	N	NN	gender=MASC number=PL spos=noun	4	subj	-	-
3	también	también	Adv	RB	spos=adverb	4	adv	-	-
4	mueren	morir	V	VV	finiteness=FIN mood=IND number=PL person=3 spos=verb tense=PRES	0	ROOT	-	-
5	.	.	SYM	SYM	spos=punctuation	4	punc	-	-

Figure 3: Sample AnCora-UPF annotated sentence in the 10-column CoNLL format: *Los Mbitis también mueren* (lit. ‘the Mbitis also die’).

feature model. MaltParser includes a default feature model for each parsing algorithm. The default feature models, as we can see in Figure 4, 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 4: Default feature model for the Nivre arc-eager parsing algorithm.

To this end, we used *MaltOptimizer* (Ballesteros and Nivre, 2012b; Ballesteros and Nivre, 2012a) 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 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. Note also that *MaltOptimizer* sets a held-out development set during the optimization process (actually, 5 different development sets, one for each fold cross-validation), thus the evaluation results provided over the test set are actually using unseen data during the optimization process.

We are aware of the interactions between the features that are included in the feature model –the ones 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 1; we were 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, 2012a).

3.2 The AnCora-UPF dependency corpus

This corpus, presented by Mille et al. (2013), consists of a small section (3513 sentences, 100892 tokens) of the Spanish dependency corpus AnCora (Taulé et al., 2008). Mille et al. (2009) explain the partially automatic mapping between the two corpora, and Burga et al. (2011) detail what kind of information is encoded in the syntactic tags.⁴ The annotation is theoretically based on the Meaning-Text Theory (MTT, (Mel’čuk, 1988)), according to which the set of surface syntactic (SSynt) relations is unique to each language, and should cover as many syntactic idiosyncrasies of the given language. Lexical and morphologic features are not directly encoded into the syntactic relations, but rather into attribute/value pairs associated to each node. The authors manually revised the syntactic annotation, but no manual revision was performed on the morphosyntactic features.

The AnCora-UPF corpus is released in the CoNLL’08 format⁵; hence, it contains all the information that a CoNLL file as described in Section 3.1 can contain. We took a close look at the annotation, and in particular at the FEAT column, in which there are **7 features**: *finiteness*, *gender*, *mood*, *number*, *person*, *spos*, *tense*. Unlike in the source AnCora corpus, the authors did not annotate cases. One explanation could be that there are no very clear case markers in Spanish apart from on personal pronouns. However, there is a new feature, *spos*, which is another feature for part-of-speech. The possible values for this attribute are very similar to those of the POSTAG column⁶, but the few differences between the two tagsets have noticeable consequences on the results of the evaluation, as discussed in Section 4.3. Table 1 shows these discrepancies: four POSTAGs have been split into two (*IN*, *SYM*, *VB*, *WP*), while two *spos* tags (in bold) correspond to twice as many POSTAGs.

Table 2 shows the possible values that the re-

⁴For downloading the corpus, see <http://www.taln.upf.edu/content/resources/495>.

⁵We transformed it into the 10-column CoNLL-X format for our experiments.

⁶This column contains a subset of the Tree Tagger PoS tagset, widely used in corpus annotation nowadays.

POSTAG	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 1: Correspondences between PoS and *spos* tagsets.

maining six features can take, and Table 3 how these morphosyntactic features are distributed through the corpus with respect to generic part-of-speech. We can see that *gender* and *number* are the most frequent attributes, and that they are annotated on elements of different parts-of-speech. The 2.02% of verbs that include *gender* are actually past participles. *gender=C* is not common; it stands for elements that do not express masculine or feminine gender, e.g. the dative pronoun “le”. The other four attributes, (*finiteness*, *mood*, *person* and *tense*) are exclusively verbal features (except for the annotation errors).

FEAT	Possible Values	#Occurrences
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 2: Possible values and total number of occurrences of the 6 features.

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

FEAT	V	N	Adj	Det	Pro	Other
fin	99.91	0.01	0.06	0	0	0.02
gen	2.02	46.72	14.31	32.33	4.37	0.25
moo	99.95	0.01	0	0	0	0.04
num	16.74	36.57	15.15	27.1	4.25	0.19
per	99.98	0.01	0	0	0	0.01
ten	99.98	0	0	0	0	0.02

Table 3: Distribution of features over elements of different generic part-of-speech (%).

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*.

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 features (baseline, 1); 4 extra combinations in order to test the PoS/*spos* impact.

4 Results and Discussion

First, we discuss the results of the first 78 experiments. In the last subsection, we will discuss the part-of-speech issues related to the other 4 experiments.

4.1 Feature combinations and general labeled accuracy

The LAS recall provided by the baseline model (no features) is 82.25%.⁷ From a general perspective, 25 out of the 78 feature combinations make the baseline 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 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. This goes along the lines of Eryigit 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 that question, we looked at the re-

⁷The full set of results can be checked at http://www.taln.upf.edu/system/files/resources_files/table.pdf

	spo	num	fin	gen	per	ten	moo
14	14	14	10	10	8	6	5
25	25	22	17	15	13	11	12

Table 4: Occurrences of features in the 14 and 25 best scoring feature combinations.

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: Contribution of each feature when enlarging the number of elements in a combination.

sults from another perspective. For a given set of features, we wondered (1) if adding one particular feature makes 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*]?

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. 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 times with a percentage higher than 0.3 points). It never worsens and never belongs to the worst feature combination. See Section 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*]).⁸ This feature is very useful

⁸All the feature combinations improve the baseline; how-

in our experiments, and this could be explained by the following: (i) as shown in Table 2, 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); (ii) many dependency relations in the AnCora-UPF corpus use *number* directly or indirectly, on the head and/or the dependent: most verbal argumental relations (subjects, copulatives, direct objects, compleatives, clitic objects), verbal non-argumental relations (passive analytical, copredicatives); 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. 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

ever, some of them do it in a more significant way.

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, being Spanish an SVO (subject-verb-object) language, it is possible that the linear order—which is also taken into account by the parser—is sufficient to decide who is the subject and who is not; in addition, most nouns are 3rd person, thus, it is not surprising that this feature does not help much. This redundancy is reflected in McNemar’s test for $p < 0.05$, which indicates that there is a statistically significant difference between the best model with 4 features and another model that has the same number of features, but includes, for instance, *mood* instead of *gender*.⁹

The first conclusion is that the observations of

⁹McNemar’s test shows no statistically significant differ-

this section coincide almost exactly with the ones made in Table 4: the features that individually tend to improve 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 2). 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 combinations), but they do very well when they are combined to other attributes.

4.2 UAS, LA and specific dependency relations

We look first at 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 compared 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 differences can be pointed out: (1) the best LAS and LA are obtained with

ence between the best 14 feature combinations, but we consider that the differences can be interpreted anyway; in the rest of the section we look at the results taking into account both perspectives.

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%.

Then, we tried 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. The task was maybe too ambitious: 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 failed in our 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*].

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. In two feature combinations that did not include *spos*, we filled the POSTAG column –which normally contains the Tree Tagger PoS tags– with the *spos* tags from the

AnCora-UPF corpus. 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.¹⁰ Again, the parser’s LAS dropped half a point in both cases. It is obviously due to the tagsets differences between *PoS* and *spos* pointed out in Section 3.2, and we believe that in particular to the fact that the *spos* tagset splits the *IN* tag into *conjunction* and *preposition*, since this tag is way more frequent than the other mismatching tags.

Therefore, when the more fine-grained tagset *spos* is in the FEATS column, it specifies the POSTAG column and can be used in order to improve the parsing; however, it does not work the other way around: the Tree Tagger PoS tags in the FEATS column do not bring any new information to that one already introduced in the POSTAG column, and thus are ignored by MaltOptimizer. Also, MaltOptimizer follows a stepwise procedure, under this scenario it starts with a higher baseline and it is therefore difficult to get improvements during the optimization steps by testing new features, and thus the features are not selected. There is therefore less room for improvement.

Klein and Manning (2003) present similar improvements when splitting the *IN* tag during their experiments on constituency parsing with a PCFG; we can see now that it is probably the case for dependency parsing too.

5 Conclusions and Future Work

The best configuration for MaltParser and AnCora-UPF corpus is [*finiteness gender number spos*]. For parsing purposes, then, it seems enough to enrich the morphosyntactic annotation just with these features, at least in the case of Spanish. These features not only work well together, but also very often improve the results when are 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 (see Section 4.1). On the other hand, we have observed that the interaction between features also influences significantly the results. So, in order

¹⁰Note that the default feature models include several feature specifications for the POSTAG column and the deepest experiments performed by MaltOptimizer are indeed in this feature window.

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 (1FEAT), 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 this paper opens many perspectives for further experiments. The next step will be to study whether different levels of dependency relation granularity are affected by the combination of several features and the analysis of the results presented in this paper, following the same idea as presented by Mille et al. (2012). It will 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. Moreover, another interesting experiment would be to make use of an automatic morphological-analyzer/tagger that could show the accuracy provided by the parser when it does not use gold morphosyntactic tags coming from the treebank.

We could 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.¹¹ Finally, we could also 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, 2008; 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.

¹¹We did not do it for these experiments because this would make the use of the current version of MaltOptimizer impossible; however, we are planning to modify the MaltOptimizer source code in order to make it possible.

References

- A. Atutxa, E. Agirre, and K. Sarasola. 2012. Contribution of Complex Lexical Information to Solve Syntactic Ambiguity in Basque. In *Proceedings of COLING*, pages 97–114.
- M. Ballesteros and J. Nivre. 2012a. MaltOptimizer: A System for MaltParser Optimization. In *Proceedings of the 8th LREC*, pages 2757–2763.
- M. Ballesteros and J. Nivre. 2012b. MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of the System Demonstration Session of the 13th EACL*, pages 58–62.
- K. Bengoetxea and K. Gojenola. 2009. Application of feature propagation to dependency parsing. In *Proceedings of IWPT*, pages 142–145.
- B. Bohnet and J. Kuhn. 2012. The Best of Both Worlds - A Graph-based Completion Model for transition-based parsers. In *Proceedings of EACL*, pages 77–87.
- B. Bohnet and J. Nivre. 2012. A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing. In *Proceedings of EMNLP-CoNLL*, pages 1455–1465.
- S. Buchholz and E. Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-06*, pages 149–164.
- A. Burga, S. Mille, and L. Wanner. 2011. Looking Behind the Scenes of Syntactic Dependency Corpus Annotation: Towards a Motivated Annotation Schema of Surface-Syntax in Spanish. In *Proceedings of DepLing '11*, pages 104–114.
- B. Cowan and M. Collins. 2005. Morphology and reranking for the statistical parsing of Spanish. In *Proceedings of EMNLP*, pages 795–802.
- G. Eryigit, J. Nivre, and K. Ofazer. 2008. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.
- D. Klein and Ch. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of ACL-03*, pages 423–430.
- F. Mambrini and M.C. Passarotti. 2012. Will a Parser Overtake Achilles? First experiments on parsing the Ancient Greek Dependency Treebank. In *Proceedings of the 11th TLT*, pages 133–144.
- R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *HLT-EMNLP*, pages 523–530.
- I.A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany.
- S. Mille, L. Wanner, V. Vidal, and A. Burga. 2009. Towards a rich dependency annotation of Spanish corpora. *Procesamiento del Lenguaje Natural*, 1(43):325–333.
- S. Mille, A. Burga, G. Ferraro, and L. Wanner. 2012. How Does the Granularity of an Annotation Scheme Influence Dependency Parsing Performance? In *Proceedings of COLING 2012*, pages 839–852.
- S. Mille, A. Burga, and L. Wanner. 2013. Ancora-UPF: A Multi-Level Annotation of Spanish. In *Proceedings of DepLing*.
- J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007a. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of CoNLL-ST-07*, pages 915–932.
- J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007b. Maltparser: A Language-Independent System for Data-Driven Dependency Parsing. *Natural Language Engineering*, 13:95–135.
- J. Nivre. 2003. An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of IWPT-03*, pages 149–160.
- R. Schwartz, O. Abend, and A. Rappoport. 2012. Learnability-based Syntactic Annotation Design. In *Proceedings of COLING 2012*, pages 2405–2422.
- M. Seraji, B. Megyesi, and J. Nivre. 2012. Dependency parsers for Persian. In *Proceedings of 10th Workshop on Asian Language Resources, COLING 2012*, pages 35–44.
- M. Taulé, M. A. Martí, and M. Recasens. 2008. Ancora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the 6th LREC*, pages 96–101.
- R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kübler, M. Candito, J. Foster, Y. Versley, I. Rehbein, and L. Tounsi. 2010. Statistical Parsing of Morphologically Rich Languages (SPMRL): What, How and Whither. In *Proceedings of the NAACL HLT 2010 First Workshop on SPMRL*, pages 1–12.
- R. Tsarfaty, J. Nivre, and E. Andersson. 2012a. Cross-Framework Evaluation for Statistical Parsing. In *Proceedings of EACL*, pages 44–54.
- R. Tsarfaty, D. Seddah, S. Kübler, and J. Nivre. 2012b. Parsing Morphologically Rich Languages: Introduction to the Special Issue. *Computational Linguistics*, 39(1):15–22.
- Y. Zhang and S. Clark. 2008. A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing. In *Proceedings of EMNLP*, pages 562–571.
- Y. Zhang and J. Nivre. 2011. Transition-Based Parsing with Rich Non-Local Features. In *Proceedings of ACL*, pages 188–193.