IJCLA VOL. 3, NO. 2, JUL-DEC 2012, PP. 107–122
RECEIVED 31/10/11 ACCEPTED 09/12/11 FINAL 21/12/12

POS Taggers and Dependency Parsing

RAMADAN ALFARED AND DENIS BÉCHET

University of Nantes, France

ABSTRACT

A wide-coverage parser copes with the problem of the explosion of the number of combinations of sub-trees and the number of theoretically possible dependency trees, which in the majority give spurious analyses. We show that, by using a POS tagger for choosing the most probable grammatical classes of the lexical units, we can substantially improve the rate of spurious ambiguity in a categorial dependency grammar of French developed by the NLP team of LINA. The experimental results show that our models perform better than the model which do not use a POS tagger at the cost of losing some correct analyses especially when the model of the tagger is very different to the lexical model of the parser.

1 INTRODUCTION

In the last years, dependency parsing becomes very popular and has been a topic of active research in natural language processing. Many different algorithms were suggested and evaluated for this task. They achieve both, a reasonable time complexity and a high accuracy. Statistical parsers with high accuracy are generally trained on texts annotated with morphological and sometimes also some other features. In particular, the minimal necessary annotation is POS tags. In this paper, we show how the use of POS tags may improve the rate of spurious ambiguity of parsing with a wide scope categorial dependency grammar of French (CDG) which uses Le*fff* as its lexical base. In CDG, all lexical units (LU) are grouped into lexical classes (CDG classes). All units of a class share the same syntactic types. Le*fff* is a wide coverage lexicon of French representing a very large set of highly structured lexical information. Previously, a correspondence between CDG classes and Le*fff* classification was established and presented in [1].

The rest of the paper is structured as follows. Section 2 describes dependency grammars, Section 3 describes the parsing problem and our models. Section 4 presents the experimental evaluation, and Section 5 contains a comparative error analysis of the our different models. Finally, Section 6 concludes the paper.

2 DEPENDENCY GRAMMARS

Dependency-based representations have become increasingly popular in syntactic parsing, especially for languages that exhibit free or flexible word order, such as Czech (Collins et al., 1999), Bulgarian (Marinov Nivre, 2005), Turkish (Eryigit Oflazer, 2006), Russian (Boguslavsky et al., 2011). Many practical implementations of dependency parsing are restricted to projective structures, where the projection of a head word has to form a continuous substring of the sentence.

Dependency Grammars (DGs) are formal grammars assigning dependency trees (DT) to a sentence. A DT is a tree with words as nodes and dependencies, i.e. named syntactic binary relations between words, as arrows. In other words, if two words v_1 and v_2 are related by dependency d (denoted $v_1 \stackrel{d}{\to} v_2$) then v_1 is the governor and v_2 is the subordinate. Figure 1 illustrates the dependencies in the sentence "Au commencement était le Verbe."

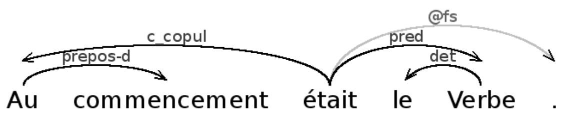

Fig. 1. *French: in the beginning was the Word.*

The relation $\acute{e}tait \xrightarrow{pred} Verbe$ represents the predicative dependency between the copula *était* and the subject *Verbe*. The head of this sentence is *était*.

2.1 *Categorials Dependency Grammars*

Categorial Dependency Grammars introduced by [2] are lexicalized in the same sense as the conventional categorial grammars. Here we briefly give basic information on CDG.

The CDG types are defined over a set C of elementary categories (types). A syntactic type may be repetitive or optional $C^* =_{df} \{X^* | X \in C\}$, $C^? =_{df} \{X^? | X \in C\}$. CDG use iteration to express all kinds of repetitive dependencies such as modifiers and coordination relations.

The non-projective dependencies are expressed using *polarized valencies*. Namely, the governor G which has a right distant subordinate D through a discontinuous dependency d has positive dependency $\nearrow d$, whereas its subordinate D has the negative valency $\searrow d$. Together these dual valencies define the discontinuous dependency d.

In CDG, the anchor types of the form $\#(\searrow d)$, $\#(\swarrow d)$ are used in the same way as local dependencies. More precisely, CDG define discontinuous dependencies using polarized valencies (left / right, positive / negative) and a simple valencies pairing principle First Available (FA). For every valency, the corresponding one is the closest dual valency in the indicated direction.

In order to define polarized categories, we distinguish between four dependency polarities: left and right positive $\nwarrow, \nearrow$ and left and right negative $\searrow, \swarrow$. For each polarity $v \in \{\nwarrow, \searrow, \nearrow, \swarrow\}$ there is a unique dual polarity $\breve{v}$: $\breve{\nwarrow} = \swarrow, \breve{\swarrow} = \nwarrow, \breve{\nearrow} = \searrow, \breve{\searrow} = \nearrow$. $\nearrow C, \nwarrow C, \searrow C$ and $\swarrow C$ denote the corresponding sets of polarized distant dependency categories.

The general form of a CDG type is $[l_1 \setminus l_2 \setminus \cdots \setminus H / \cdots / r_2 / r_1]^P$ where the head type H defines the incoming dependency on the word, l_1 and r_1 are elementary (iterated or optional) categories which correspond to left or right outgoing dependencies or anchors, P is a potential, a string of polarized valencies which defines the long distance dependencies (incoming or outgoing), see [3], [4] and [5] for more details. Figure 2 shows two discontinuous dependencies (non-projective) in the sentence “elle la lui a donnée.”.

Categorial dependency grammars which define this dependency tree affect the types which anchor the clitics *la, lui* on the auxiliary *a*. The discontinuous dependencies are represented by dotted arrows.

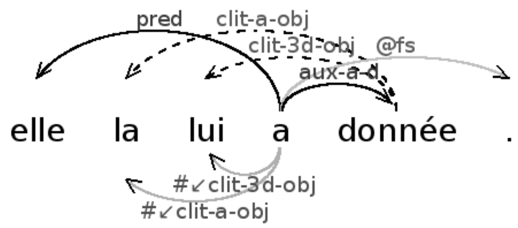

Fig. 2. Non-projective DS: "*she it[fem.] to him has given.".

$$
\begin{array}{ll}
elle^{PN(Lex=pers,C=n)} & \mapsto [pred] \\
la^{PN(Lex=pn,F=clit,C=a)} & \mapsto [\,\#(\swarrow clit-a-obj)]^{\swarrow clit-a-obj} \\
lui^{PN(Lex=pn,F=clit,P=3,C=d)} & \mapsto [\,\#(\swarrow clit-3d-obj)]^{\swarrow clit-3d-obj} \\
a^{Vaux(Lex=avoir,F=fin)} & \mapsto [\,\#(\swarrow clit-3d-obj) \\
 & \quad \backslash\#(\swarrow clit-a-obj) \\
 & \quad \backslash pred\backslash S/@fs/aux-a-d] \\
donnée^{V2t(F=pz,C1=a,C2=d|g|l,T=past)} & \\
 & \mapsto [aux-a-d]^{\nwarrow clit-3d-obj\nwarrow clit-a-obj} \\
.^{FullStop(Lex=".")} & \mapsto [@fs]
\end{array}
$$

The word *elle* is classified as a pronoun (PN), where *pers* and *n* correspond to person and noun. The word *la* is classified as a clitic at accusative case. The word *lui* is classified as a clitic for 3rd person with complement at dative case. The word *a* is classified as an auxiliary verb with a finite form "F=fin" while the word *donnée* is classified as a ditransitive verb where pz is "past participle" form and has two arguments (complement), the first complement is a direct complement (at accusative) and the second complement is a dative, a genitive or a locative.

The NLP team has developed a large scale CDG of French and a general purpose offline CDG parser. In this French CDG, the types are assigned to CDG classes (see [6] for details). The CDG parser is currently used to develop dependency tree corpora. The linguist's interface of this parser lets manually select for every LU one of its possible classes and one of the possible head dependencies. Then the parser finds all analyses compatible with the selection. Our goal in this paper is to automatically pre-fetch the most probable CDG classes per LU depending on its POS and to measure the impact of this selection on the ambiguity of the parser as applied to the CDG of French.

3 POS-BASED PARSING MODELS

Usually, the task of disambiguation of a dependency parser consists in deriving a single correct dependency tree τ for a given sentence S. The parsing problem consist in finding the mapping of an input sentence S, constituted of words $w_1 \cdots w_n$, to its dependency tree τ. More precisely, given a parsing model M and a sentence S, we derive the optimal dependency tree τ for S according to M. So the parsing problem is to construct the optimal dependencies for the input sentence, given the parsing model. Some parsers solve this problem by deriving a single analysis for each sentence. Our task is different: we should instead lower the ambiguity of the French CDG using POS tagging models and we evaluate the effect obtained by our method. Our POS-based parsing models first choose the most probable CDG classes through POS tags for the words in a sentence. Applying our method we should resolve a technical problem which arises from the nature of the lexical database of the CDG of French. In fact, this lexical database uses the (freely available) wide-coverage French lexicon Le*fff* [7]. It contains 110,477 lemmas (simple and compounds) and 536,375 inflected forms. The main part of the French CDG classes linked with Le*fff* is saved in a PostgreSQL Database. In this database, each LU of Le*fff* corresponds to one or several CDG classes. This correspondence is realized in the main table `lexicon`. Unfortunately, Le*fff* is not complete and contains errors. Therefore, in the lexical database there are several facilities for correction and complementation of Le*fff* definitions.

Before we describe our approach, we should explain that the CDG parser uses the following two strategies for lexicon (called below models):

Base model gives access to the forms contained in the classes of the French CDG (about 1500 forms), and also gives access to the original definitions of Le*fff* related with the CDG classes in the database.

The three other models use Le*fff* and the French CDG implicitly. First, a tagger is applied to the input sentence (Tree-Tagger [8] in T.T Model, MElt-Tagger [9] in M.T model and Brill tagger [10] in B.T model), Figure 4 presents this strategy.

Then, depending on the computed (composite in general) LU and their POS, a compatible lexical definition for every pair (LU, POS) and the corresponding CDG class is found in the database. If and when they exist, they are integrated to the input file that is sent to the parser.

Correspondence between POS tagging and Lefff: The correspondence between CDG classes and Le*fff* is established using the workspace dis-

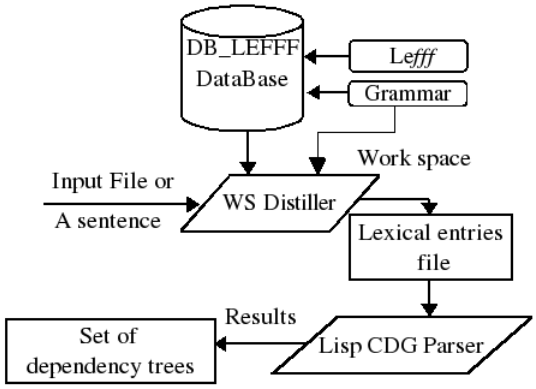

Fig. 3. General form of Base model.

tiller shown in Figure 4. We try to find the correspondence between the tags of POS-tagger and the syntactic categories of Le*fff*. This correspondence is approximate, because the lexical models of POS-tagger and of Le*fff* and the french CDG are different. Table 1 shows some examples of the correspondence.

Table 1. *Examples of correspondence between POS-tagger and* Le*fff*.

Le*fff*	T.T	M.T	B.T
np (noun phrases)	NAM	NPP	NAM, SBP
coo (coordination)	KON	CC, ET	COO
det (determiner)	DET:ART	DET	DTN
nc (commun nous)	NOM, NUM	NC	SBC, CAR

Some important information on POS-tagging e.g. `VER:futu` are very useful to determine both the *mood* and the *tense* of a verb. In this case, we also compare them to the *mood* and *tense* of the lexicon database. For instance, `VER:futu` means that *mood* is indicative and *tense* is future.

The WS distillers of the different models take an input file which contains the sentences with it POS (annotated sentence), and the output is a file with (lexical entries) annotated CDG classes and word features.

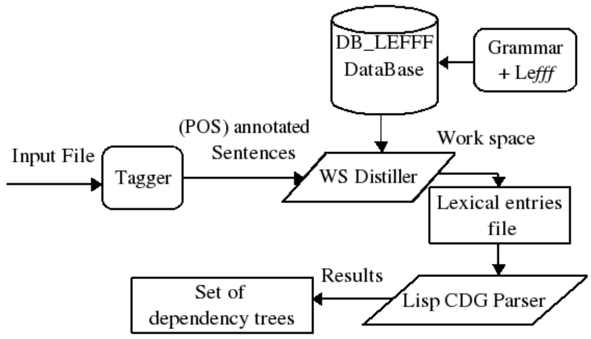

Fig. 4. General form of POS-bassed parsing models.

The algorithm chooses the most probable CDG classes for a LU by using POS tags and *cat* of Le*fff*.

This algorithm consists of the next three steps.

- First we search by the word of the sentence and its POS tag and compare them between the correspondence to *form* and it category that are found in the database, if it's equal, then we take it CDG class and it morphological features such as *mood*, *tense*, *person*, *gender*, *number*, *lemma* and saved all these information on file (lexical entry).
- If there is no result from the first step then we only search by the word of the sentence and compared it with *form* and take all the morphological information that correspond to this *form*.
- If also there is no result we classified this LU as `"UT(Lex=V|N|Adj|Adv)"`. This CDG class is assigned to unknown LU.

4 Experimental Results

In our experiments we use a corpus of sentences divided into two subsets. The first subset, serving as a test set, consists of 1443 French sentences that have been analyzed to build the French Gold Standard dependency corpus (DTB): a corpus with French sentences from various sources. These sentences have 14974 projective and non projective (discontinuous) dependencies.

The second subset of the corpus has 184 French sentences from the French treebank [11].

For the experiment with the first subset, we first run the parser with the maximum number of viewed dependency trees set to 2000. We can not request all the possible dependency trees per sentence. With the French CDG, it generates hundreds of spurious structures per sentence. So for long and complex sentences, it is practically impossible to know how many DS are produced. Till the final step where the DS are extracted from the chart, the parsing algorithm is polynomial. Given that the number of these DS may be exponential with respect to the size of the chart, the final step is exponential in space in the worst case. In this step, the DS are generated from the chart in a certain order. The parser generates a HTML report page, which includes various useful statistics. It can also produce an XML structure representation of every DS including all necessary information.

For our POS-based parsing models, we compute the ambiguity reduction of dependency trees using the formula $X^j = \sum_{i=1}^{N} A_i^j$ where A_i^j is the number of dependency trees that are found for model j, where j is Base model, T.T model, M.T model or B.T model and i=1,...,N. N represents the number of the sentences that have a 100% correct analysis in every model. For our experiments, N=325. The reduction of dependency trees of model j is $\frac{X^{Base}-X^j}{X^{Base}} \times 100$, where j is different from the Base model. 2

Table 2. *Experimental results (dependency structures) compared to four models*

	Base	T.T	M.T	B.T
# DS for 325 sentences	153938	42572	44056	46718
Reduction # DS	% wrt model j	72.34%	71.38%	69.65%
geometric mean # DS_j/# DS_{Base}	-	0,24	0,23	0,26

We do not compute the number of dependency trees of the sentences that have more or eqaul 2000 analyses and also we just take into account the sentences that have at least one analysis for each parsing model. Table 3 shows some cases (accepted or canceled cases).

For the second experiment with the first subset, we run the parser with the maximum number of viewed dependency trees set to 1 in order to obtain the maximal number of analyzed sentences, and also to know how many sentences have all dependencies correctly analyzed. We compute

Table 3. *Cas canceled or accepted for the # DS.*

Cas canceled or accepted	Base	T.T	M.T	B.T
$\times$ (because 0 analysis)	55	34	55	0
$\times$ (because $>=$2000 analyse)	$>=$2000	666	1000	867
$\times$ (because no analyse)	11	9	no analyse	8
$\surd$ (accepted)	67	13	16	33

the total number of composition trees [1] using the formula $Y^j = \sum_{i=1}^{M} B_i^j$, where B_i^j is the number of composition trees for sentences that are found using model j, where j is Base model, T.T model, M.T model or B.T model and i=1,...,M. M represents the number of sentences that have at least one analysis in every model. For our experiments M=780. The reduction of the composition trees for model j is $\frac{Y^{Base} - Y^j}{Y^{Base}} \times 100$, where j is different from Base model.

Table 4. *Experimental results (composition trees) compared to four parsing models*

	Base	T.T	M.T	B.T
# CT for 780 sentences	16330×10^8	27×10^8	34×10^8	28×10^8
Reduction de # CT	% wrt model j	99.83%	99.79%	99.82%
geometric mean # CT_j/# CT_{Base}	-	0,035	0,037	0,033

The results in Tables 4 and 2 show that the numbers of composition trees and dependency trees of the three POS-based parsing models are inferior that of Base model. Our models achieve high reduction of both, composition trees and dependency trees (over 99% and 70% respectively).

The evaluation of the parser uses classical measures. It uses the labeled attachment score AS_L for the mode on Figure 4, which is the

[1] For each dependency tree, there are several composition trees because each composition tree specifies also a set of word features, a class and a type. We use the number of composition trees rather than the number of dependency trees, because it's usually not possible to evaluate the total number of dependency trees.

proportion of tokens that are assigned the correct head and the correct dependency label. The labeled attachment score represents the percentage of tokens that have been assigned both the correct head and the correct dependency label. There are several sentences which have accuracy over 90% of correct dependencies, but we count only the sentences that have 100% correct analysis. The result in Table 5 shows that our models achieve between 88% and 95% accuracy for correct dependency relation labeling.

Table 5. *Experimental results of parsing accuracy compared to four parsing models.*

	Base	T.T	M.T	B.T
# Sentences that have at least one analysis (1)	1089	1125	1005	949
# Sentences have 100% correct dependencies	1089	874	892	667
Recall	75.46%	60.65%	61.81%	46.22%
Precision	100%	77.68%	88.75%	70.28%
# of dependencies (from (1))	8255	9571	7730	7603
# correct dependencies	8255	8465	7380	6838
Recall correct dependencies	55.12%	56.53%	49.28%	45.66%
Precision	100%	88.44%	95.47%	89.93%
Labeled accuracy average (on all 1443 sentences)	100%	82.27%	85%	69%

We do not need to use unlabeled attachment score AS_U, because we don't compare the result of several parsers, AS_U is used by [12], that compare between two parsing architectures for the high accuracy on unknown words. Indeed BKY+FLABELER [13] achieves only a 82.56% tagging accuracy for the unknown words in the development set (5.96% of the tokens), whereas MElt+MST [14] achieves 90.01%.

Comparing between the three POS-based parsing models, we note that M.T model performs better than T.T and B.T models in terms of parsing accuracy. But T.T model is better than the other models in terms of ambiguity reduction and parsing time.

Table 6 shows an example to explain the reduction for both dependency trees and composition trees of the four parsing models in the sentence : “il parle en courtes phrases”.

Table 6. *Reduction (dependency trees and composition trees) on the sentence "he speaks in short sentences".*

	Base	T.T	M.T	B.T
Reduction (# DS)	268	54	211	54
Reduction (# CT)	28732	3336	8559	3336

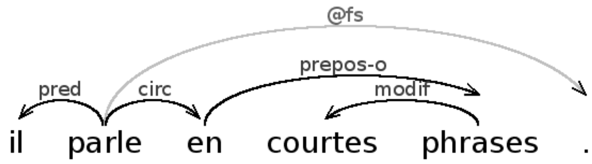

Fig. 5. "*he speaks in short sentences*"

$$\begin{aligned} il &\longmapsto PN(Lex = pers, C = n) \\ parle &\longmapsto Vt(F = fin, C = g) \\ en &\longmapsto PP(F = compl - obl, C = o) \\ courtes &\longmapsto Adj(F = modifier) \\ phrases &\longmapsto N(Lex = common) \end{aligned}$$

Table 7 shows comparative parsing times for each parsing model.

Table 7. *Comparation of the parsing times (four parsing models for 1443 sentences)*

	Base	T.T	M.T	B.T
Sentences that have at least one analysis	1089	1125	1005	949
Sentences that are analysed incorrect	0	141	127	314
Analyzed sentences total	1089	1266	1132	1263
Parsing time	03h 37mn	01h 32mn	02h 31mn	02h 8mn
Sentences that are not analyzed	354	177	311	180
Parsing time	05h 09mn	03h 35mn	05h 18mn	03h 00mn
Parsing time total	8h 46m	5h 07m	7h 49m	5h 08m

Table 8. *Effect of class pre-fetching (Paris 7 corpus).*

	Base	T.T
#CT	1097325498316350	7048627222816
#CT	10973254×10^8	70486×10^8
Total reduction #CT	% wrt Base model	99,9%
geometric mean of # $\mathrm{CT}_{T.T}$/# CT_{Base}	-	0.002
# CT of the sentence Figure 6	17284	241
# DT of the sentence Figure 6	1295	68

For the second subset of 184 French sentences, we use only the Base model and T.T model. We only compute the number of composition trees using the same formula of the first subset. The results show that pre-fetching of CDG classes reduces the ambiguity with respect to composition trees more than 99%.

Table 8 summarizes the experimental results for Base model and T.T model for the number of composition trees.

The results given in Table 8 show that pre-fetching of classes reduces the ambiguity in terms of composition trees more than 99%.

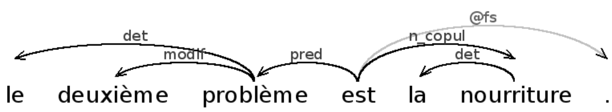

Fig. 6. *Paris 7 : the second problem is the food.*

5 DISCUSSION

This discussion provides a brief analysis of the errors made by the POS tagger for the first corpus, when we investigate the POS category of erroneous instances.

Each tagger tags this sentences by differnt way such as on the table 10.

In the CDG grammar, these tokens have different grammatical classes. As a result it gives different lexical classes for each token. Table 11 illustrates the lexical classes that correspond to the sentence “Ève, vas-t’en !”.

Table 9. *Errors make by the Parser for the parsing models.*

	memory exhansted	bad sentences	too complex sentences
Base	12	0	342
T.T	17	141	160
M.T	1	127	310
B.T	0	314	180

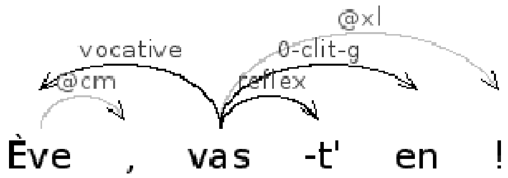

Fig. 7. *Structure de dépendances : Ève, vas-t'en !*

Table 10. Ťagset of differnt tagger

Tree Tagger	MElt Tagger	Bril Tagger
vas-t'en/NOM	vas-t'en/ADV	vas-t/VCJ en/PREP

Table 11. Classes assigned to the lexical unity

Frenche CDG	
lexical unity	Class
Éve	N(Lex=proper)
,	Comma(Lex='$CM')
vas	Vt(F=fin,C=l), Vt(F=fin,C=d), Vt(F=fin,C=a)
t'	PN(Lex=pn,F=refl)
en	PN(Lex=pn,F=clit,C=g—p), PN(Lex=attach-npers,C=g—p)
!	EmphatMark(Lex='!')

In the T.T model, there are 318 sentences that have no dependency tree, 177 sentences among them are not analyzed (time exceeded), which means there was not enough time to parse them, (the maximum number of seconds per sentence is set to 60 second), as we indicated above for ambiguous CDG. There are 141 sentences that are analysed as incorrect sentences. A first reason for this fact is that, there is at least one of the next compound words in the sentences : *à peu près*, *Hé bien*, *dès lors*, *de loin*, *au dessous*, *la-bas*, *des EU*, *de l'*. In these cases, Tree-Tagger tags these compound words as separate words: *à* as *prep*, *peu* as *adv*, *près* as *adv*, etc. But the database has only complete entries for them. The second main reason is that Tree tagger makes errors in tagging for some LU. Thus the distiller do not find a good CDG class for these LU. We have seen that the results of B.T model are worse than those for T.T and M.T models, because Brill-Tagger also makes many errors in tagging. For example, the sentence "Adam ne donne à Ève pas que les pommes." (Adam do not give to Eve only apple) is annotated as `Adam/SBC:sg ne/ADV donne/SBC:sg à/PREP Ève/SBC:sg pas/ADV que/SUB les/DTN:pl pommes/SBC:pl ./.`. The verb *donne* is tagged as common noun *SBC* and not as a verb. There are 17 sentences contain *donne* tagged as *SBC* and 28 sentences which contain the past participle "été" of the verbe "être" are also tagged as *SBC*. Errors like these lead to 314 sentences that have been analyzed as incorrect sentences. The example in Figure 6 shows the reason why we have obtained several analyses for this sentence. We note that the word "la", (*the*) is only tagged by T.T model as "determiner". Thus, there is only one CDG class corresponding to this LU: "`Det(Lex=art|pn)`", while Base model leaves all the CDG classes for this word. More precisely, the word *la* has in the grammar three different CDG classes, because this LU has different syntactic categories in Le*fff* such as *det*, *nc* and *pro* as illustrated in Table 12.

Table 12. *Some features and classes in the Database for LU "La".*

Form	Cat	Class
la	cla	PN(Lex=pers,C=a)
la	cla	PN(Lex=pn,F=clit,C=a)
la	det	Det(Lex=art—pn)
la	nc	N(Lex=common)

This lexical ambiguity in Base model leads to several analyses of this sentence. This example shows the importance of the assignment of proper POS tag to every word in a sentence which is to be parsed.

In the one hand, the POS tagging reduces the search space for the parser, and also reduces ambiguity, improving parsing by limiting the search space. The sentences are also more often completely analyzed by the parser, because the search space is smaller as compared to Base model.

On the other hand, using POS tagging, we lost some analyses for the reason of POS tagging errors. These sentences have been considered as incorrect sentences by the parser.

6 CONCLUSION

This paper evaluates the rate of improving dependency parsing through using different POS-tag models. These models choose the most probable grammatical classes for a word in a sentence based on POS tags, unfortunately at the cost of losing some correct analyses. Our experimental results have demonstrated the utility of POS-based parsing models. These models achieved substantial reductions of the number of dependency trees and of composition trees per sentence. Our experiments also show that to obtain an interesting system, the model used by the POS tagger must be compatible to the lexical model of the parser.

REFERENCES

1. Alfared, R., Béchet, D., Dikovsky, A.: "CDG LAB": a toolbox for dependency grammars and dependency treebanks development. In: K. Gerdes, E. Hajicova, and L. Wanner (Eds.), Proc. of the 1st Intern. Conf. on Dependency Linguistics (Depling 2011), Barcelona, Spain (September 2011)
2. Dikovsky, A.: Dependencies as categories. In: "Recent Advances in Dependency Grammars". COLING'04 Workshop. (2004) 90–97
3. Dekhtyar, M., Dikovsky, A.: Categorial dependency grammars. In Moortgat, M., ed.: Proceedings of Categorial Grammars 2004. (2004) 76–91
4. Béchet, D., Dikovsky, A., Foret, A.: Dependency structure grammar. In Philippe Blache, Edward Stabler, J.B., Moot, R., eds.: Proceedings of the 5th International Conference on Logical Aspects of Computational Linguistics, Bordeaux, France, April 2005. Volume 3492 of Lecture Notes in Artificial Intelligence (LNAI)., Springer-Verlag (April 2005) 18–34
5. Dekhtyar, M., Dikovsky, A.: Generalized categorial dependency grammars. In: Pillars of Computer Science'08. (2008) 230–255
6. Dikovsky, A.: Towards wide coverage categorical dependency grammars. In: Proceedings of the ESSLLI'2009 Workshop Parsing with Categorial Grammars - Parsing with Categorial Grammars Workshop ESSLLI 2009 Book of Abstracts. (2009) 230–255
7. Sagot, B.: The lefff, a freely available and large-coverage morphological and syntactic lexicon for french. In Calzolari (Conference Chair), N., Choukri,

K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)

8. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester, UK. (1994)
9. Denis, P., Sagot, B.: Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In: Pacific Asia Conference on Language, Information and Computation, Hong Kong, Chine (2009)
10. Brill, E.: Some advances in transformation-based part of speech tagging. In: Proceedings of the twelfth national conference on artificial intelligence. (1994) 722–727
11. Abeillé, A., Barrier, N.: Enriching a french treebank. In: Proc. of LREC'04, Lisbon, Portugal (2004)
12. Candito, M., Crabbé, B., Denis, P.: Statistical french dependency parsing: Treebank conversion and first results. In Calzolari (Conference Chair), N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)
13. Petrov, S., Barrett, L., Thibaux, R., Klein, D.: Learning accurate, compact, and interpretable tree annotation. In: ACL'06. (2006) 433–440
14. McDonald, R.: Discriminative Training and Spanning Tree Algorithms for Dependency Parsing. PhD thesis, University of Pennsylvania (July 2006)

RAMADAN ALFARED
LINA,
UNIVERSITY OF NANTES,
2, RUE DE LA HOUSSINIÈRE, 44000 NANTES, FRANCE
E-MAIL: <RAMADAN.ALFARED@ETU.UNIV-NANTES.FR>

DENIS BÉCHET
LINA,
UNIVERSITY OF NANTES,
2, RUE DE LA HOUSSINIÈRE, 44000 NANTES, FRANCE
E-MAIL: <DENIS.BECHET@UNIV-NANTES.FR>