French parsing enhanced with a word clustering method based on a syntactic lexicon

Anthony Sigogne
Université Paris-Est, LIGM
sigogne@univ-mlv.fr

Matthieu Constant
Université Paris-Est, LIGM
mconstan@univ-mlv.fr

Éric Laporte
Université Paris-Est, LIGM
laporte@univ-mlv.fr

Abstract

This article evaluates the integration of data extracted from a French syntactic lexicon, the Lexicon-Grammar (Gross, 1994), into a probabilistic parser. We show that by applying clustering methods on verbs of the French Treebank (Abeillé et al., 2003), we obtain accurate performances on French with a parser based on a Probabilistic Context-Free Grammar (Petrov et al., 2006).

1 Introduction

Syntactic lexicons are rich language resources that may contain useful data for parsers like subcategorisation frames, as it provides, for each lexical entry, information about its syntactic behaviors. Many works on probabilistic parsing studied the use of a syntactic lexicon. We can cite Lexical-Functional Grammar [LFG] (O'Donovan et al., 2005; Schluter and Genabith, 2008), Head-Driven Phrase Structure Grammar [HPSG] (Carroll and Fang, 2004) and Probabilistic Context-Free Grammars [PCFG] (Briscoe and Carroll, 1997; Deoskar, 2008). The latter has incorporated valence features of verbs to PCFGs and observe slight improvements on global performances. However, the incorporation of syntactic data on part-of-speech tags increases the effect of data sparseness, especially when the PCFG grammar is extracted from a small treebank[1]. (Deoskar, 2008) was forced to reestimate parameters of his grammar with an unsupervised algorithm applied on a large raw corpus. In the case of French, this observation can be linked to experiments described in (Crabbé and Candito, 2008) where POS tags are augmented with some syntactic functions[2]. Results have shown a huge decrease on performances.

The problem of data sparseness for PCFG is also lexical. The richer the morphology of a language, the sparser the lexicons built from a treebank will be for that language. Nevertheless, the effect of lexical data sparseness can be reduced by word clustering algorithms. Inspired by the clustering method of (Koo et al., 2008), (Candito and Crabbé, 2009; Candito et al., 2010) have shown that by replacing each word of the corpus by automatically obtained clusters of words, they can improve a PCFG parser on French. They also created two other clustering methods. A first method consists in a step of *desinflection* that removes some inflexional marks of words which are considered less important for parsing. Another method consists in replacing each word by the combination of its POS tag and lemma. Both methods improve significantly performances.

In this article, we propose a clustering method based on data extracted from a syntactic lexicon, the Lexicon-Grammar. This lexicon offers a classification of lexical items into tables, each table being identifiable by its unique identifier. A lexical item is a lemmatized form which can be present in one or more tables depending on its meaning and its syntactic behaviors. The clustering method consists in replacing a verb by the combination of its POS tag and its tables identifiers. The goal of this article is to show that a syntactic lexicon, like the Lexicon-

[1] Data sparseness implies the difficulty of estimating probabilities of rare rules extracted from the corpus.

[2] There were 28 original POS tags and each can be combined with one of the 8 syntactic functions.

Proceedings of the 2nd Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL 2011), pages 22–27,
Dublin, Ireland, October 6, 2011. ©2011 Association for Computational Linguistics

Grammar, which is not originally developed for parsing algorithms, is able to improve performances of a probabilistic parser.
In section 2 and 3, we describe the probabilistic parser and the treebank, namely the French Treebank, used in our experiments. In section 4, we describe more precisely previous work on clustering methods. Section 5 introduces the Lexicon-Grammar. We detail information contained in this lexicon that can be used in the parsing process. Then, in section 6, we present methods to integrate this information into parsers and, in section 7, we describe our experiments and discuss the obtained results.

2 Non-lexicalized PCFG parser

The probabilistic parser, used into our experiments, is the Berkeley Parser[3] (called BKY thereafter) (Petrov et al., 2006). This parser is based on a PCFG model which is non-lexicalized. The main problem of non-lexicalized context-free grammars is that nonterminal symbols encode too general information which weakly discriminates syntactic ambiguities. The benefit of BKY is to try to solve the problem by generating a grammar containing complex symbols. It follows the principle of latent annotations introduced by (Matsuzaki et al., 2005). It consists in iteratively creating several grammars, which have a tagset increasingly complex. For each iteration, a symbol of the grammar is splitted in several symbols according to the different syntactic behaviors of the symbol that occur into a treebank. Parameters of the latent grammar are estimated with an algorithm based on Expectation-Maximisation (EM). In the case of French, (Seddah et al., 2009) have shown that BKY produces *state-of-the-art* performances.

3 French Treebank

For our experiments, we used the French Treebank[4] (Abeillé et al., 2003) [FTB]. It is composed of articles from the newspaper *Le Monde* where each sentence is annotated with a constituent tree. Currently, most of papers about parsing of French use a specific variant of the FTB, namely the FTB-UC described for the first time in (Candito and Crabbé, 2009). It is a partially corrected version of the FTB which contains 12351 sentences and 350931 tokens. This version is smaller[5] and has specific characteristics. First, the tagset takes into account the rich original annotation containing morphological and syntactic information. It results in a tagset of 28 part-of-speech tags. Some compounds with regular syntax schemas are undone into phrases containing simple words. Remaining compounds are merged into a single token, whose components are separated with an underscore.

4 Previous work on word clustering

Numerous works used clustering methods in order to reduce the size of the corpus lexicon and therefore reducing the impact of lexical data sparseness on treebank grammars. A method, described in (Candito and Seddah, 2010) and called *CatLemma*, consists in replacing a word by the combination of its POS tag and its lemma. In the case of a raw text to analyze (notably during evaluations), they used a statistical tagger in order to assign to each word both POS tag and lemma[6].
Instead of reducing each word to the lemmatized form, (Candito and Crabbé, 2009; Candito and Seddah, 2010) have done a morphological clustering, called *desinflection* [DFL], which consists in removing morphological marks that are *less important* for determining syntactic projections in constituents. The mood of verbs is, for example, very helpful. On the other hand, some marks, like gender or number for nouns or the person of verbs, are not so crucial. Moreover, original ambiguities on words are kept in order to delegate the task of POS tags desambiguation to the parser. This algorithm is done with the help of a morpho-syntactic lexicon.
The last clustering method, called *Clust*, consists in replacing each word by a cluster id. Cluster ids are automatically obtained thanks to an unsupervi-

[3] The Berkeley Parser is freely available at http ://code.google.com/p/berkeleyparser/

[4] The French Treebank is freely available under licence at http ://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php

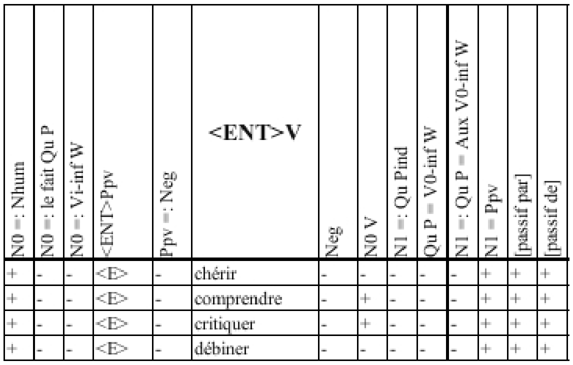
[5] The original FTB contains 20,648 sentences and 580,945 tokens.

[6] They used the tagger MORFETTE (Chrupala et al., 2008; Seddah et al., 2010) which is based on two statistical models, one for tagging and the other for lemmatization. Both models were trained thanks to the *Average Sequence Perceptron* algorithm.

sed statistical algorithm (Brown et al., 1992) applied on a large raw corpus. They are computed by taking account of co-occurrence information of words. The main advantage of this method is the possibility of combining it to *DFL* or *CatLemma*. First, the raw corpus is preprocessed with one of these two methods and then, clusters are computed on this modified corpus. Currently, this method permits to obtain the best results on the FTB-UC.

5 Lexicon-Grammar

The Lexicon-Grammar [LG] is the richest source of syntactic and lexical information for French[7] that focuses not only on verbs but also on verbal nouns, adjectives, adverbs and frozen (or fixed) sentences. Its development started in the 70's by Maurice Gross and his team (Gross, 1994). It is a syntactic lexicon represented in the form of tables. Each table encodes lexical items of a particular category sharing several syntactic properties (e.g. subcategorization information). A lexical item is a lemmatized form which can be present in one or more tables depending on its meaning and its syntactic properties. Each table row corresponds to a lexical item and a column corresponds to a property (e.g. syntactic constructions, argument distribution, and so on). A cell encodes whether a lexical item accepts a given property. Figure 1 shows a sample of verb table *12*. In this table, we can see that the verb *chérir* (*to cherish*) accepts a human subject (pointed out by a + in the property *N0 = : Nhum*) but this verb cannot be intransitive (pointed out by a − in the property *N0 V*). Recently, these tables have been made consistent and explicit (Tolone, 2011) in order to be exploitable for NLP. They also have been transformed in a XML-structured format (Constant and Tolone, 2008)[8]. Each lexical entry is associated with its table identifier, its possible arguments and its syntactic constructions.



N0 =: Nhum	N0 =: le fait Qu P	N0 =: Vi-inf W	<ENT>Ppv	Ppv =: Neg	<ENT>V	Neg	N0 V	N1 =: Qu Pind	Qu P = V0-inf W	N1 =: Qu P = Aux V0-inf W	N1 = Ppv	[passif par]	[passif de]
+	-	-	<E>	-	chérir	-	-	-	-	-	+	+	+
+	-	-	<E>	-	comprendre	-	+	-	-	-	+	+	+
+	-	-	<E>	-	critiquer	-	+	-	-	-	+	+	+
+	-	-	<E>	-	débiner	-	-	-	-	-	+	+	+

FIG. 1: Sample of verb table *12*.

For the verbs, we manually constructed a hierarchy of the tables on several levels. Each level contains classes which group LG tables which may not share all their defining properties but have a relatively similar syntactic behavior. Figure 2 shows a sample of the hierarchy. The tables *4, 6* and *12* are grouped into a class called *QTD2* (transitive sentence with two arguments and sentential complements). Then, this class is grouped with other classes at the superior level of the hierarchy to form a class called *TD2* (transitive sentence with two arguments). The characteristics of each level are given in the Table 1 (level 0 represents the set of tables of the LG). We can state that there are 5,923 distinct verbal forms for 13,862 resulting entries in tables of verbs[9]. The column *#classes* specifies the number of distinct classes. The columns *AVG_1* and *AVG_2* respectively indicate the average number of entries per class and the average number of classes per distinct verbal form.

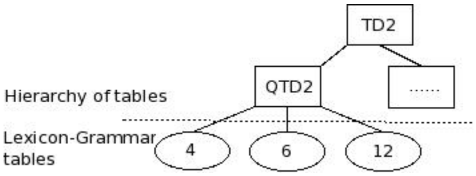


FIG. 2: Sample of the hierarchy of verb tables.

Level	#classes	AVG_1	AVG_2
0	67	207	2.15
1	13	1,066	1.82
2	10	1,386	1.75
3	4	3,465	1.44

TAB. 1: Characteristics of the hierarchy of verb tables.

The hierarchy of tables has the advantage of reducing the number of classes associated with each verb

[7] We can also cite lexicons like LVF (Dubois and Dubois-Charlier, 1997), Dicovalence (Eynde and Piet, 2003) and Lefff (Sagot, 2010).

[8] These resources are freely available at http ://infolingu.univ-mlv.fr/

[9] Note that 3,121 verb forms (3,195 entries) are unambiguous. This means that all their entries occur in a single table.

of the tables. We will see that this ambiguity reduction is crucial in our experiments.

6 Word clustering based on the Lexicon-Grammar

The LG contains a lot of useful information that could be used into the parsing process. But such information is not easily manipulable. We will focus on table identifiers of the verb entries which are important hints about their syntactic behaviors. For example, the table *31R* indicates that all verbs belonging to this table are intransitive. Therefore, we followed the principle of the clustering method *CatLemma*, except that here, we replace each verb of a text by the combination of its POS tag and its table ids associated with this verb in the LG tables[10]. We will call this experiment *TableClust* thereafter. For instance, the verb *chérir* (to cherish) belongs to the table *12*. Therefore, the induced word is *#tag_12*, where *#tag* is the POS tag associated with the verb. For an ambiguous verb like *sanctionner* (to punish), belonging to two tables *6* and *12*, the induced word is *#tag_6_12*.

Then, we have done variants of the previous experiment by taking the hierarchy of verb tables into account. This hierarchy is used to obtain clusters of verbs increasingly coarse as the hierarchy level increases, and at the same time, the size of the corpus lexicon is also increasingly reduced. Identifiers combined to the tag depend on the verb and the specific level in the hierarchy. For example, the verb *sanctionner*, belonging to tables *6* and *12*, is replaced by *#tag_QTD2* at level 1. In the case of ambiguous verbs, for a given level in the hierarchy, identifiers are all classes the verb belongs to. This experiment will be called *LexClust* thereafter.

As for clustering method *CatLemma*, we need a Part-Of-Speech tagger in order to assign a tag and a lemma to each verb of a text (table ids can be determined from the lemma). We made the choice to use *MElt* (Denis and Sagot, 2009) which is one of the best taggers for French. Lemmatization process is done with a French dictionary, the Dela (Courtois and Silberztein, 1990), and some heuristics in the case of ambiguities.

7 Experiments and results

7.1 Evaluation metrics

As the FTB-UC is a small corpus, we used a *cross-validation* procedure for evaluations. This method consists in splitting the corpus into p equal parts, then we compute training on *p-1* parts and evaluations on the remaining part. We can iterate this process p times. This allows us to calculate an average score for a sample as large as the initial corpus. In our case, we set the parameter p to 10. Results on evaluation parts are reported using the standard protocol called PARSEVAL (Black et al., 1991) for all sentences. The labeled F-Measure [F1] takes into account the bracketing and labeling of nodes. We also use the unlabeled and labeled attachement scores [UAS, LAS] which evaluate the quality of unlabeled and labeled dependencies between words of the sentence[11]. Punctuation tokens are ignored in all metrics.

7.2 Berkeley parser settings

We used a modified version of BKY enhanced for tagging unknown and rare French words (Crabbé and Candito, 2008)[12]. We can notice that BKY uses two sets of sentences at training, a learning set and a validation set for optimizing the grammar parameters. As in (Candito et al., 2010), we used 2% of each training part as a validation set and the remaining 98% as a learning set. The number of split and merge cycles was set to 5.

7.3 Clustering methods

We have evaluated the impact of clustering methods *TableClust* and *LexClust* on the FTB-UC. For both methods, verbal forms of each training part are replaced by the corresponding cluster and, in order to do it on the evaluation part, we use Melt and some heuristics. So as to compare our results with previous work on word clustering, we have reported results of two clustering methods described in section 4, *DFL* and *DFL+Clust* ($Clust$ is applied on a text that contains $desinflected$ words).

[10] Verbs that are not in the LG remain unchanged.

[11] These scores are computed by automatically converting constituent trees into dependency trees. The conversion procedure is made with the *Bonsaï* software, available at http ://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html.

[12] Available in the *Bonsaï* package.

7.4 Evaluations

The experimental results are shown in the Table 2[13]. The column *#lexicon* represents the size of the FTB-UC lexicon according to word clustering methods. In the case of the method *LexClust*, we varied the level of the verbs hierarchy used. The method *TableClust* slightly improves performances compared with the baseline. Nevertheless, using levels of the hierarchy of verb tables through *LexClust* increases results while considerably reducing the size of the corpus lexicon. We obtain the best results with the level 2 of the hierarchy. These performances are almost identical to those of *DFL*, despite the fact that we only modify verbal forms while *DFL* alters all inflected forms regardless of grammatical categories. However, *DFL+Clust* has high scores and is significantly better than *LexClust*. As of this writing, we tried some combination of methods *LexClust* and *Clust* but we observed that both methods are not easily mergeable.

Method	#lexicon	F1	UAS	LAS	F1<40
Baseline	27,143	83.82	89.43	85.85	86.12
DFL	20,127	84.57	89.91	86.36	86.80
DFL+Clust	1,987	85.22	90.26	86.70	87.39
TableClust	24,743	84.11	89.67	86.10	86.53
LexClust 1	22,318	84.33	89.77	86.22	86.62
LexClust 2	21,833	84.44	89.87	86.32	86.76
LexClust 3	20,556	84.26	89.64	86.10	86.57
Tag	20478	84.11	89.58	86.00	86.40
TagLemma	24722	83.87	89.51	85.91	86.26

TAB. 2: Results from cross-validation evaluation according to clustering methods.

The impact of *TableClust* and *LexClust* on a new text is strongly influenced by the quality of the tagging produced by *Melt*. For evaluating this impact, we computed *Gold* experiments for both clustering method. Each verb of evaluation parts, present in the LG tables, is replaced by correct tag and table ids. We observed a gap of almost 0.5% for both tagging and F1. For instance, on the first evaluation part, *Melt* has high but not perfect scores, with a precision of 98.2% and a recall of 97.2%, for a total of 165 errors[14]. About lemmatization, we have a perfect score of 100%.

Our approach is based on the combination of tags and table ids contained in the syntactic lexicon. In order to validate this approach, we have done two other experiments. A first one, called *Tag*, consists in replacing each verbal lemma by its verbal tag only. The second one, called *TagLemma*, consists in the combination of the tag and the lemma. Results are reported in the Table 2. As for *TableClust* and *LexClust*, we replace only verbal forms that are present in the LG tables. We can see that *Tag* has equal performances to *TableClust*. Therefore, original table ids combined with tags are useless. Maybe, the number of clusters is too high and consequently, the size of the corpus lexicon is still too large. However, *LexClust* is better than *Tag*. About *TagLemma*, results are almost identical to the baseline. According to these observations, we can say that verbal clusters created with our method *LexClust* are relevant and useful for a parser like BKY.

We have indicated in Table 3, the top most F1 absolute gains according to phrase labels, for our best clustering method *LexClust* with level 2 of the hierarchy. For each phrase, the column called *Gain* indicates the average F1 absolute gain in comparison to the baseline F1 for this phrase, and *prop.* is the proportion of the phrase in the whole corpus. We can see that three of the five best corrected phrases relate to verbal phrases (plus one if we consider that AdP is linked to a verbal phrase). Therefore, the integration of syntactic data into a clustering algorithm of verbs improves the recognition of verbal phrases.

Phrase label	Meaning	Gain (prop.)
VPpart	participial phrase	4,4% (2%)
Srel	relative clause	1,6% (1%)
VN	verbal nucleus	1.1% (11%)
VPinf	infinitive phrase	0.9% (0.4%)
AdP	adverbial phrase	0.9% (3%)

TAB. 3: Top most F1 absolute gains according to phrases.

8 Conclusion and future work

In this article, we have shown that by using information on verbs from a syntactic lexicon, like the Lexicon-Grammar, we are able to improve performances of a statistical parser based on a PCFG grammar. In the near future, we plan to reproduce experiments with other grammatical categories.

[13] All experiments have a tagging accuracy of about 97%.

[14] We can compute precision and recall scores because sometimes *Melt* wrongly identifies a word as a verb or miss a verb.

References

A. Abeillé, L. Clément, and F. Toussenel. 2003. Building a treebank for French. In Anne Abeillé, editor, *Treebanks : building and using parsed corpora*, Kluwer, Dordrecht.

E. Black, S.Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the DARPA Speech and Naturale Language Workshop*, pages 306–311.

T. Briscoe and J. Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Fifth Conference on AppliedNatural Language Processing*, pages 356–363, Washington DC, USA.

P. F. Brown, V. J. Della, P. V. Desouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. In *Computational linguistics, 18(4)*, pages 467–479.

M. Candito and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proceedings of the 11th International Conference on Parsing Technology (IWPT'09)*, pages 138–141.

M. Candito and D. Seddah. 2010. Parsing word clusters. In *Proceedings of the first NAACL HLT Workshop on Morphologically-Rich Languages (SPRML2010)*, pages 76–84, Los Angeles, California.

M. Candito, B. Crabbé, and P. Denis. 2010. Statistical French dependency parsing : treebank conversion and first results. In *Proceedings of LREC10*.

J. Carroll and A. C. Fang. 2004. The automatic acquisition of verb subcategorisations and their impact on the performance of an HPSG parser. In *Proceedings of the 1st International Conference onNatural Language Processing*, Sanya City, China.

G. Chrupala, G. Dinu, and J. van Genabith. 2008. Learning morphology with Morfette. In *Proceedings of LREC 2008*.

M. Constant and E. Tolone. 2008. A generic tool to generate a lexicon for nlp from lexicon-grammar tables. In *Actes du 27ème Colloque Lexique et Grammaire*, L'Aquila, Italie.

B. Courtois and M. Silberztein. 1990. Dictionnaires électroniques du français. Présentation. In *Larousse, editor, Langue Française*.

B. Crabbé and M. Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, pages 45–54, Avignon, France.

P. Denis and B. Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *PACLIC 2009*, Hong Kong.

T. Deoskar. 2008. Re-estimation of lexical parameters for treebank PCFGs. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 193–200, Manchester, Great Britain.

J. Dubois and F. Dubois-Charlier. 1997. *Les verbes français*. Larousse-Bordas.

K. Eynde and M. Piet. 2003. La valence : l'approche pronominale et son application au lexique verbal. *Journal of French Language studies*, pages 63–104.

M. Gross. 1994. Constructing Lexicon-grammars. In Atkins and Zampolli, editors, *Computational Approaches to the Lexicon*, pages 213–263.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08*.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of ACL-05*, pages 75–82, Ann Arbor, USA.

R. O'Donovan, A. Cahill, A. Way, M. Burke, and J. van Genabith. 2005. Large-Scale induction and evaluation of lexical resources from the Penn-II and Penn-III Treebanks. In *Computational Linguistics, 31*, pages 329–366.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

B. Sagot. 2010. The Lefff, a freely available, accurate and large-coverage lexicon for French. In *Proceedings of LREC 2010*, La Valette, Malte.

N. Schluter and J. Van Genabith. 2008. Treebank-Based Acquisition of LFG Parsing Resources for French. In *Proceedings of LREC08*, Marrakech, Morocco.

D. Seddah, M. Candito, and B. Crabbé. 2009. Adaptation de parsers statistiques lexicalisés pour le français : Une évaluation complète sur corpus arborés. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'09)*, Senlis, France.

D. Seddah, G. Chrupala, O. Cetinoglu, J. van Genabith, and M. Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *Proceedings of the first NAACL HLT Workshop on Morphologically-Rich Languages (SPRML2010)*.

E. Tolone. 2011. *Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français*. Ph.D. thesis, Université Paris-Est Marne-la-Vallée.