On statistical parsing of French with supervised and semi-supervised strategies

Marie Candito, Benoît Crabbé, Djamé Seddah

On statistical parsing of French with supervised and semi-supervised strategies

Marie Candito*, Benoît Crabbé* and Djamé Seddah◇

* Université Paris 7
UFRL et INRIA (Alpage)
30 rue du Château des Rentiers
F-75013 Paris — France

◇ Université Paris 4
LALIC et INRIA (Alpage)
28 rue Serpente
F-75006 Paris — France

Abstract

This paper reports preliminary results on grammatical induction for French. We investigate how to best train a parser on the French Treebank (Abeillé et al., 2003), viewing the task as a trade-off between generalizability and interpretability. We compare on French a supervised lexicalized parsing algorithm with a semi-supervised unlexicalized algorithm (Petrov et al., 2006) along the lines of (Crabbé and Candito, 2008). We report the best results known to us on French statistical parsing with the semi-supervised learning algorithm, and the reported experiments can give insights for the task of grammatical learning for a morphologically-rich language, with a relatively limited amount of training data, annotated with a rather flat structure.

1 Natural language parsing

Despite the availability of annotated data, there have been relatively few works on French statistical parsing. Together with a treebank, the availability of several supervised or semi-supervised grammatical learning algorithms, primarily set up on English data, allows us to figure out how they behave on French.

Before that, it is important to describe the characteristics of the parsing task. In the case of statistical parsing, two different aspects of syntactic structures are to be considered : their capacity to capture regularities and their interpretability for further processing.

Generalizability Learning for statistical parsing requires structures that capture best the underlying regularities of the language, in order to apply these patterns to unseen data.

Since capturing underlying linguistic rules is also an objective for linguists, it makes sense to use supervised learning from linguistically-defined generalizations. One generalization is typically the use of phrases, and phrase-structure rules that govern the way words are grouped together. It has to be stressed that these syntactic rules exist at least in part independently of semantic interpretation.

Interpretability But the main reason to use supervised learning for parsing, is that we want structures that are as *interpretable* as possible, in order to extract some knowledge from the analysis (such as deriving a semantic analysis from a parse). Typically, we need a syntactic analysis to reflect how words *relate* to each other. This is our main motivation to use supervised learning : the learnt parser will output structures as defined by linguists-annotators, and thus interpretable within the linguistic theory underlying the annotation scheme of the treebank. It is important to stress that this is more than capturing syntactic regularities : it has to do with the *meaning* of the words.

It is not certain though that both requirements (generalizability / interpretability) are best met in the same structures. In the case of supervised learning, this leads to investigate different instantiations of the training trees, to help the learning, while keeping the maximum interpretability of the trees. As we will see with some of our experiments, it may be necessary to find a trade-off between generalizability and interpretability.

Further, it is not guaranteed that syntactic rules infered from a manually annotated treebank produce the best language model. This leads to

methods that use semi-supervised techniques on a treebank-infered grammar backbone, such as (Matsuzaki et al., 2005; Petrov et al., 2006).

The plan of the paper is as follows : in the next section, we describe the available treebank for French, and how its structures can be interpreted. In section 3, we describe the typical problems encountered when parsing using a plain probabilistic context-free grammar, and existing algorithmic solutions that try to circumvent these problems. Then we describe experiments and results when training parsers on the French data.

2 Interpreting the French trees

The French Treebank (Abeillé et al., 2003) is a publicly available sample from the newspaper *Le Monde*, syntactically annotated and manually corrected for French.

```
<SENT>
  <NP fct="SUJ">
    <w cat="D" lemma="le" mph="ms" subcat="def">le</w>
    <w cat="N" lemma="bilan" mph="ms" subcat="C">bilan</w>
  </NP>
  <VN>
    <w cat="ADV" lemma="ne" subcat="neg">n'</w>
    <w cat="V" lemma="être" mph="P3s" subcat="">est</w>
  </VN>
  <AdP fct="MOD">
    <w compound="yes" cat="ADV" lemma="peut-être">
       <w catint="V">peut</w>
       <w catint="PONCT">-</w>
       <w catint="V">être</w>
    </w>
    <w cat="ADV" lemma="pas" subcat="neg">pas</w>
  </AdP>
  <AP fct="ATS">
    <w cat="ADV" lemma="aussi">aussi</w>
    <w cat="A" lemma="sombre" mph="ms" subcat="qual">sombre</w>
  </AP>
  <w cat="PONCT" lemma="." subcat="S">.</w>
</SENT>
```

Figure 1: Simplified example of the FTB

To encode syntactic information, it uses a combination of labeled constituents, morphological annotations and functional annotation for verbal dependents as illustrated in Figure 1. This constituent and functional annotation was performed in two successive steps : though the original release (Abeillé et al., 2000) consists of 20,648 sentences (hereafter FTB-V0), the functional annotation was performed later on a subset of 12351 sentences (hereafter FTB). This subset has also been revised, and is known to be more consistently annotated. This is the release we use in our experiments. Its key properties, compared with the Penn Treebank, (hereafter PTB) are the following :

Size : The FTB is made of 385 458 tokens and 12351 sentences, that is the third of the PTB. The average length of a sentence is 31 tokens in the FTB, versus 24 tokens in the PTB.

Inflection : French morphology is richer than English and leads to increased data sparseness for statistical parsing. There are 24098 types in the FTB, entailing an average of 16 tokens occurring for each type (versus 12 for the PTB).

Flat structure : The annotation scheme is flatter in the FTB than in the PTB. For instance, there are no VPs for finite verbs, and only one sentential level for sentences whether introduced by complementizer or not. We can measure the corpus flatness using the ratio between tokens and non terminal symbols, excluding preterminals. We obtain 0.69 NT symbol per token for FTB and 1.01 for the PTB.

Compounds : Compounds are explicitly annotated (see the compound *peut-être* in Figure 1) and very frequent : 14,52% of tokens are part of a compound. They include digital numbers (written with spaces in French *10 000*), very frozen compounds *pomme de terre (potato)* but also named entities or sequences whose meaning is compositional but where insertion is rare or difficult (*garde d'enfant (child care)*).

Now let us focus on what is expressed in the French annotation scheme, and why syntactic information is split between constituency and functional annotation.

Syntactic categories and constituents capture distributional generalizations. A syntactic category groups forms that share distributional properties. Nonterminal symbols that label the constituents are a further generalizations over sequences of categories or constituents. For instance about anywhere it is grammatical to have a given NP, it is implicitly assumed that it will also be grammatical - though maybe nonsensical - to have instead any other NPs. Of course this is known to be false in many cases : for instance NPs with or without determiners have very different distributions in French (that may justify a different label) but they also share a lot. Moreover, if words are taken into account, and not just sequences of categories, then constituent labels are a very coarse generalization. Constituents also encode dependencies : for instance the different PP-attachment for the sentences *I ate a cake with cream / with a fork* reflects that *with cream* depends on *cake*, whereas *with a fork* depends on *ate*. More precisely, a syntagmatic tree can be interpreted as a dependency structure using the following conventions : for each constituent, given the dominating symbol

and the internal sequence of symbols, (i) a head symbol can be isolated and (ii) the siblings of that head can be interpreted as containing dependents of that head. Given these constraints, the syntagmatic structure may exhibit various degree of flatness for internal structures.

Functional annotation Dependencies are encoded in constituents. While X-bar inspired constituents are supposed to contain all the syntactic information, in the FTB the shape of the constituents does not necessarily express unambiguously the *type* of dependency existing between a head and a dependent appearing in the same constituent. Yet this is crucial for example to extract the underlying predicate-argument structures. This has led to a "flat" annotation scheme, completed with functional annotations that inform on the type of dependency existing between a verb and its dependents. This was chosen for French to reflect, for instance, the possibility to mix post-verbal modifiers and complements (Figure 2), or to mix post-verbal subject and post-verbal indirect complements : a post verbal NP in the FTB can correspond to a temporal modifier, (most often) a direct object, or an inverted subject, and in the three cases other subcategorized complements may appear.

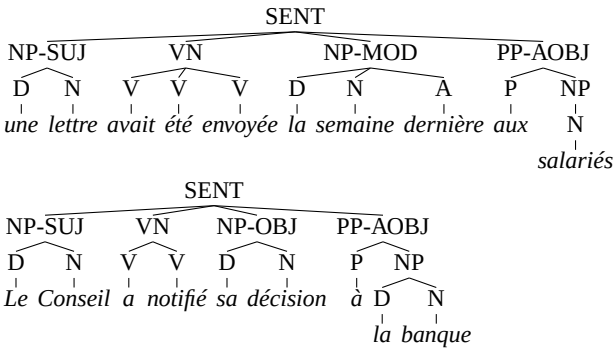

Figure 2: Two examples of post-verbal NPs : a direct object and a temporal modifier

3 Algorithms for probabilistic grammar learning

We propose here to investigate how to apply statistical parsing techniques mainly tested on English, to another language – French –. In this section we briefly introduce the algorithms investigated.

Though Probabilistic Context Free Grammars (PCFG) is a baseline formalism for probabilistic parsing, it suffers a fundamental problem for the purpose of natural language parsing : the independence assumptions made by the model are too strong. In other words all decisions are local to a grammar rule.

However as clearly pointed out by (Johnson, 1998) decisions have to take into account non local grammatical properties: for instance a noun phrase realized in subject position is more likely to be realized by a pronoun than a noun phrase realized in object position. Solving this first methodological issue, has led to solutions dubbed hereafter as *unlexicalized statistical parsing* (Johnson, 1998; Klein and Manning, 2003a; Matsuzaki et al., 2005; Petrov et al., 2006).

A second class of non local decisions to be taken into account while parsing natural languages are related to handling lexical constraints. As shown above the subcategorization properties of a predicative word may have an impact on the decisions concerning the tree structures to be associated to a given sentence. Solving this second methodological issue has led to solutions dubbed hereafter as *lexicalized parsing* (Charniak, 2000; Collins, 1999).

In a supervised setting, a third and practical problem turns out to be critical: that of *data sparseness* since available treebanks are generally too small to get reasonable probability estimates. Three class of solutions are possible to reduce data sparseness: (1) enlarging the data manually or automatically (e.g. (McClosky et al., 2006) uses self-training to perform this step) (2) smoothing, usually this is performed using a markovization procedure (Collins, 1999; Klein and Manning, 2003a) and (3) make the data more coarse (i.e. clustering).

3.1 Lexicalized algorithm

The first algorithm we use is the lexicalized parser of (Collins, 1999). It is called lexicalized, as it annotates non terminal nodes with an additional latent symbol: the head word of the subtree. This additional information attached to the categories aims at capturing bilexical dependencies in order to perform informed attachment choices.

The addition of these numerous latent symbols to non terminals naturally entails an over-specialization of the resulting models. To ensure generalization, it therefore requires to add additional simplifying assumptions formulated as a variant of usual naïve Bayesian-style simplifying assumptions: the probability of emitting a non

head node is assumed to depend on the head and the mother node only, and not on other sibling nodes[1].

Since Collins demonstrated his models to significantly improve parsing accuracy over bare PCFG, lexicalization has been thought as a major feature for probabilistic parsing. However two problems are worth stressing here: (1) the reason why these models improve over bare PCFGs is not guaranteed to be tied to the fact that they capture bilexical dependencies and (2) there is no guarantee that capturing non local lexical constraints yields an optimal language model.

Concerning (1) (Gildea, 2001) showed that full lexicalization has indeed small impact on results : he reimplemented an emulation of Collins' Model 1 and found that removing all references to bilexical dependencies in the statistical model[2], resulted in a very small parsing performance decrease (PARSEVAL recall on WSJ decreased from 86.1 to 85.6). Further studies conducted by (Bikel, 2004a) proved indeed that bilexical information were used by the most probable parses. The idea is that most bilexical parameters are very similar to their back-off distribution and have therefore a minor impact. In the case of French, this fact can only be more true, with one third of training data compared to English, and with a much richer inflection that worsens lexical data sparseness.

Concerning (2) the addition of head word annotations is tied to the use of manually defined heuristics highly dependent on the annotation scheme of the PTB. For instance, Collins' models integrate a treatment of coordination that is not adequate for the FTB-like coordination annotation.

3.2 Unlexicalized algorithms

Another class of algorithms arising from (Johnson, 1998; Klein and Manning, 2003a) attempts to attach additional latent symbols to treebank categories without focusing exclusively on lexical head words. For instance the additional annotations will try to capture non local preferences like the fact that an NP in subject position is more likely realized as a pronoun.

The first unlexicalized algorithms set up in this trend (Johnson, 1998; Klein and Manning, 2003a) also use language dependent and manually defined heuristics to add the latent annotations. The specialization induced by this additional annotation is counterbalanced by further naïve simplifying assumptions : dubbed markovization (Klein and Manning, 2003a).

Using hand-defined heuristics remains problematic since we have no guarantee that the latent annotations added in this way will allow to extract an optimal language model.

A further development has been first introduced by (Matsuzaki et al., 2005) who recasts the problem of adding latent annotations as an unsupervised learning problem: given an observed PCFG induced from the treebank, the latent grammar is generated by combining every non terminal of the observed grammar to a predefined set H of latent symbols. The parameters of the latent grammar are estimated from the *observed trees* using a specific instantiation of EM.

This first procedure however entails a combinatorial explosion in the size of the latent grammar as $|H|$ increases. (Petrov et al., 2006) (hereafter BKY) overcomes this problem by using the following algorithm: given a PCFG G_0 induced from the treebank, iteratively create n grammars $G_1 \dots G_n$ (with $n = 5$ in practice), where each iterative step is as follows :

- SPLIT Create a new grammar G_i from G_{i-1} by splitting every non terminal of G_i in two new symbols. Estimate G_i's parameters on the observed treebank using a variant of inside-outside. This step adds the latent annotation to the grammar.

- MERGE For each pair of symbols obtained by a previous split, try to merge them back. If the likelihood of the treebank does not get significantly lower (fixed threshold) then keep the symbol merged, otherwise keep the split.

- SMOOTH This step consists in smoothing the probabilities of the grammar rules sharing the same left hand side.

This algorithm yields state-of-the-art results on

[1]This short description cannot do justice to (Collins, 1999) proposal which indeed includes more fine grained informations and a backoff model. We only keep here the key aspects of his work relevant for the current discussion.

[2]Let us consider a dependent constituent C with head word Chw and head tag Cht, and let C be governed by a constituent H, with head word Hhw and head tag Hht. Gildea compares Collins model wher the emission of Chw is conditioned on Hhw, and a "mono-lexical" model, where the emission of Chw is not conditioned on Hhw.

English[3]. Its key interest is that it directly aims at finding an optimal language model without (1) making additional assumptions on the annotation scheme and (2) without relying on hand-defined heuristics. This may be viewed as a case of semi-supervised learning algorithm since the initial supervised learning step is augmented with a second step of unsupervised learning dedicated to assign the latent symbols.

4 Experiments and Results

We investigate how some treebank features impact learning. We describe first the experimental protocol, next we compare results of lexicalized and unlexicalized parsers trained on various "instantiations" of the xml source files of the FTB, and the impact of training set size for both algorithms. Then we focus on studying how words impact the results of the BKYalgorithm.

4.1 Protocol

Treebank setting For all experiments, the treebank is divided into 3 sections : training (80%), development (10%) and test (10%), made of respectively 9881, 1235 and 1235 sentences. We systematically report the results with the compounds merged. Namely, we preprocess the treebank in order to turn each compound into a single token both for training and test.

Software and adaptation to French For the Collins algorithm, we use Bikel's implementation (Bikel, 2004b) (hereafter BIKEL), and we report results using Collins model 1 and model 2, with internal tagging. Adapting model 1 to French requires to design French specific head propagation rules. To this end, we adapted those described by (Dybro-Johansen, 2004) for extracting a Stochastic Tree Adjoining Grammar parser on French. And to adapt model 2, we have further designed French specific argument/adjunct identification rules.

For the BKY approach, we use the Berkeley implementation, with an horizontal markovization h=0, and 5 split/merge cycles. All the required knowledge is contained in the treebank used for training, except for the treatment of unknown or rare words. It clusters unknown words using typographical and morphological information. We adapted these clues to French, following (Arun and Keller, 2005).

Finally we use as a baseline a standard PCFG algorithm, coupled with a trigram tagger (we refer to this setup as TNT/LNCKY algorithm[4]).

Metrics For evaluation, we use the standard PARSEVAL metric of labeled precision/recall, along with unlabeled dependency evaluation, which is known as a more annotation-neutral metric. Unlabeled dependencies are computed using the (Lin, 1995) algorithm, and the Dybro-Johansen's head propagation rules cited above[5]. The unlabeled dependency F-score gives the percentage of input words (excluding punctuation) that receive the correct head.
As usual for probabilistic parsing results, the results are given for sentences of the test set of less than 40 words (which is true for 992 sentences of the test set), and punctuation is ignored for F-score computation with both metrics.

4.2 Comparison using minimal tagsets

We first derive from the FTB a minimally-informed treebank, TREEBANKMIN, instantiated from the xml source by using only the major syntactic categories and no other feature. In each experiment (Table 1) we observe that the BKY algorithm significantly outperforms Collins models, for both metrics.

parser / metric	BKY	BIKEL M1	BIKEL M2	TNT/ LNCKY
PARSEVAL LP	85.25	78.86	80.68	68.74
PARSEVAL LR	84.46	78.84	80.58	67.93
PARSEVAL F_1	84.85	78.85	80.63	68.33
Unlab. dep. Prec.	90.23	85.74	87.60	79.50
Unlab. dep. Rec.	89.95	85.72	86.90	79.37
Unlab. dep. F_1	90.09	85.73	87.25	79.44

Table 1: Results for parsers trained on FTB with minimal tagset

[3](Petrov et al., 2006) obtain an F-score=90.1 for sentences of less than 40 words.

[4]The tagger is TNT (Brants, 2000), and the parser is LNCKY, that is distributed by Mark Johnson (`http://www.cog.brown.edu/~mj/Software.htm`). Formally because of the tagger, this is not a strict PCFG setup. Rather, it gives a practical trade-off, in which the tagger includes the lexical smoothing for unknown and rare words.

[5]For this evaluation, the gold constituent trees are converted into pseudo-gold dependency trees (that may contain errors). Then parsed constituent trees are converted into parsed dependency trees, that are matched against the pseudo-gold trees.

4.3 Training data size impact

How do the unlexicalized and lexicalized approaches perform with respect to size? We compare in figure 3 the parsing performance BKY and COLLINSM1, on increasingly large subsets of the FTB, in perfect tagging mode[6] and using a more detailed tagset (CC tagset, described in the next experiment). The same 1235-sentences test set is used for all subsets, and the development set's size varies along with the training set's size. BKY outperforms the lexicalized model even with small amount of data (around 3000 training sentences). Further, the parsing improvement that would result from more training data seems higher for BKY than for Bikel.

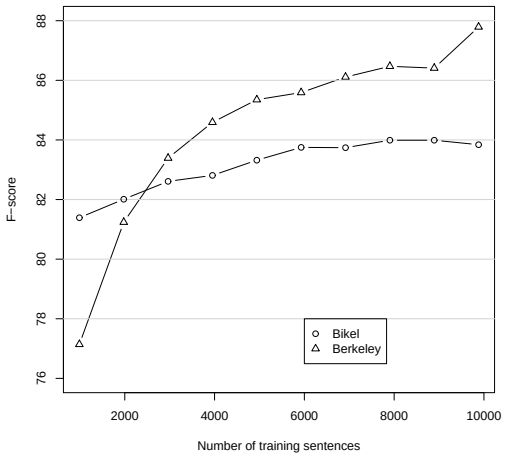

Figure 3: Parsing Learning curve on FTB with CC-tagset, in perfect-tagging

This potential increase for BKY results if we had more French annotated data is somehow confirmed by the higher results reported for BKY training on the Penn Treebank (Petrov et al., 2006) : F_1=90.2. We can show though that the 4 points increase when training on English data is not only due to size : we extracted from the Penn Treebank a subset comparable to the FTB, with respect to number of tokens and average length of sentences. We obtain F_1=88.61 with BKY training.

4.4 Symbol refinements

It is well-known that certain treebank transformations involving symbol refinements improve PCFGs (see for instance parent-transformation of (Johnson, 1998), or various symbol refinements in (Klein and Manning., 2003b)). Lexicalization itself can be seen as symbol refinements (with backoff though). For BKY, though the key point is to automatize symbol splits, it is interesting to study whether manual splits still help.

We have thus experimented BKY training with various tagsets. The FTB contains rich morphological information, that can be used to split preterminal symbols : main coarse category (there are 13), subcategory (`subcat` feature refining the main cat), and inflectional information (`mph` feature).

We report in Table 2 results for the four tagsets, where terminals are made of : MIN: main cat, SUBCAT: main cat + subcat feature, MAX: cat + subcat + all inflectional information, CC: cat + verbal mood + wh feature.

Tagset	Nb of tags	Parseval F_1	Unlab. dep F_1	Tagging Acc
MIN	13	84.85	90.09	97.35
SUBCAT	34	85.74	–	96.63
MAX	250	84.13	–	92.20
CC	28	**86.41**	**90.99**	96.83

Table 2: Tagset impact on learning with BKY (own tagging)

The corpus instantiation with CC tagset is our best trade-off between tagset informativeness and obtained parsing performance[7]. It is also the best result obtained for French probabilistic parsing. This demonstrates though that the BKY learning is not optimal since manual a priori symbol refinements significantly impact the results.

We also tried to learn structures with functional annotation attached to the labels : we obtain PARSEVAL F_1=78.73 with tags from the CC tagset + grammatical function. This degradation, due to data sparseness and/or non local constraints badly captured by the model, currently constrains us to use a language model without functional informations. As stressed in the introduction, this limits the interpretability of the parses and it is a trade-off between generalization and interpretability.

4.5 Lexicon and Inflection impact

French has a rich morphology that allows some degree of word order variation, with respect to

[6]For BKY, we simulate perfect tagging by changing words into word+tag in training, dev and test sets. We obtain around 99.8 tagging accuracy, errors are due to unknown words.

[7]The differences are statistically significant : using a standard t-test, we obtain p-value=0.015 between MIN and SUBCAT, and p-value=0.002 between CC and SUBCAT.

English. For probabilistic parsing, this can have contradictory effects : (i) on the one hand, this induces more data sparseness : the occurrences of a French regular verb are potentially split into more than 60 forms, versus 5 for an English verb; (ii) on the other hand, inflection encodes agreements, that can serve as clues for syntactic attachments.

Experiment In order to measure the impact of inflection, we have tested to cluster word forms on a morphological basis, namely to partly cancel inflection. Using lemmas as word form classes seems too coarse : it would not allow to distinguish for instance between a finite verb and a participle, though they exhibit different distributional properties. Instead we use as word form classes, the couple lemma + syntactic category. For example for verbs, given the CC tagset, this amounts to keeping 6 different forms (for the 6 moods).
To test this grouping, we derive a treebank where words are replaced by the concatenation of lemma + category for training and testing the parser. Since it entails a perfect tagging, it has to be compared to results in perfect tagging mode : more precisely, we simulate perfect tagging by replacing word forms by the concatenation form+tag.
Moreover, it is tempting to study the impact of a more drastic clustering of word forms : that of using the sole syntactic category to group word forms (we replace each word by its tag). This amounts to test a pure unlexicalized learning.

Discussion Results are shown in Figure 4. We make three observations : First, comparing the terminal=tag curves with the other two, it appears that the parser does take advantage of lexical information to rank parses, even for this "unlexicalized" algorithm. Yet the relatively small increase clearly shows that lexical information remains underused, probably because of lexical data sparseness.
Further, comparing terminal=lemma+tag and terminal=form+tag curves, we observe that grouping words into lemmas helps reducing this sparseness. And third, the lexicon impact evolution (i.e. the increment between terminal=tag and terminal=form+tag curves) is stable, once the training size is superior to approx. 3000 sentences[8]. This suggests that only very frequent words matter, otherwise words' impact should be more and more important as training material augments.

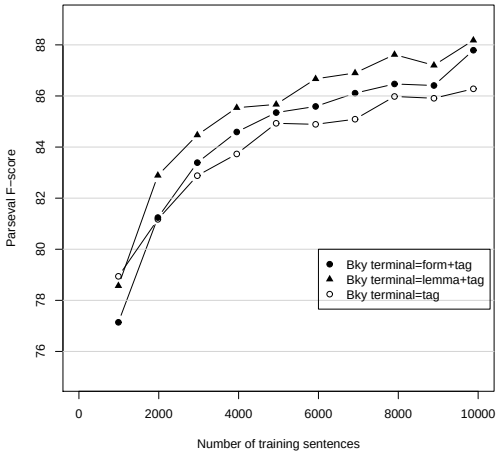

Figure 4: Impact of clustering word forms (training on FTB with CC-tagset, in perfect-tagging)

5 Related Work

Previous works on French probabilistic parsing are those of (Arun and Keller, 2005), (Schluter and van Genabith, 2007), (Schluter and van Genabith, 2008). One major difficulty for comparison is that all three works use a different version of the training corpus. Arun reports results on probabilistic parsing, using an older version of the FTB and using lexicalized models (Collins M1 and M2 models, and the bigram model). It is difficult to compare our results with Arun's work, since the treebank he has used is obsolete (FTB-V0). He obtains for Model 1 : LR=80.35 / LP=79.99, and for the bigram model : LR=81.15 / LP=80.84, with minimal tagset and internal tagging. The results with FTB (revised subset of FTB-V0) with minimal

[8] This is true for all points in the curves, except for the last step, i.e. when full training set is used. We performed a 10-fold cross validation to limit sample effects. For the BKYtraining with CC tagset, and own tagging, we obtain an average F-score of 85.44 (with a rather high standard deviation σ=1.14). For the clustering word forms experiment, using the full training set, we obtain : 86.64 for terminal=form+tag (σ=1.15), 87.33 for terminal=lemma+tag (σ=0.43), and 85.72 for terminal=tag (σ=0.43). Hence our conclusions (words help even with unlexicalized algorithm, and further grouping words into lemmas helps) hold independently of sampling.

tagset (Table 1) are comparable for COLLINSM1, and nearly 5 points higher for BKY.

It is also interesting to review (Arun and Keller, 2005) conclusion, built on a comparison with the German situation : at that time lexicalization was thought ((Dubey and Keller, 2003)) to have no sizable improvement on German parsing, trained on the Negra treebank, that uses a flat structures. So (Arun and Keller, 2005) conclude that since lexicalization helps much more for parsing French, with a flat annotation, then word-order flexibility is the key-factor that makes lexicalization useful (if word order is fixed, cf. French ad English) and useless (if word order is flexible, cf. German). This conclusion does not hold today. First, it can be noted that as far as word order flexibility is concerned, French stands in between English and German. Second, it has been proven that lexicalization helps German probabilistic parsing (Kübler et al., 2006). Finally, these authors show that markovization of the unlexicalized Stanford parser gives almost the same increase in performance than lexicalization, both for the Negra treebank and the Tüba-D/Z treebank. This conclusion is reinforced by the results we have obtained : the unlexicalized, markovized, PCFG-LA algorithm outperforms the Collins' lexicalized model.

(Schluter and van Genabith, 2007) aim at learning LFG structures for French. To do so, and in order to learn first a Collins parser, N. Schluter created a modified treebank, the MFT, in order (i) to fit her underlying theoretical requirements, (ii) to increase the treebank coherence by error mining and (iii) to improve the performance of the learnt parser. The MFT contains 4739 sentences taken from the FTB, with semi-automatic transformations. These include increased rule stratification, symbol refinements (for information propagation), coordination raising with some manual re-annotation, and the addition of functional tags. MFT has also undergone a phase of error mining, using the (Dickinson and Meurers, 2005) software, and following manual correction. She reports a 79.95% F-score on a 400 sentence test set, which compares almost equally with Arun's results on the original 20000 sentence treebank. So she attributes her results to the increased coherence of her smaller treebank. Indeed, we ran the BKY training on the MFT, and we get F-score=84.31. While this is less in absolute than the BKY results obtained with FTB (cf. results in table 2), it is indeed very high if training data size is taken into account. This good result raises the open question of identifying which modifications in the MFT (error mining and correction, tree transformation, symbol refinements) have the major impact.

6 Conclusion

This paper reports results in statistical parsing for French with both unlexicalized (Petrov et al., 2006) and lexicalized parsers. To our knowledge, both results are state of the art on French for each paradigm.

Both algorithms try to overcome PCFG's simplifying assumptions by some specialization of the grammatical labels. For the lexicalized approach, the annotation of symbols with lexical head is known to be rarely used in practice (Gildea, 2001), what is really used being the category of the lexical head. So comparing lexicalized and unlexicalized latent variables approach merely amounts to comparing a manual refinement of the symbols with head category versus an automated learning of this refinement.

We observe that the second approach (BKY) constantly outperforms the lexicalist strategy *à la* (Collins, 1999). We observe however that (Petrov et al., 2006)'s semi-supervised learning procedure is not fully optimal since a manual refinement of the treebank labelling turns out to improve the parsing results.

Finally we observe that the semi-supervised BKY algorithm does take advantage of lexical information : removing words degrades results. The preterminal symbol splits percolates lexical distinctions. Further grouping words into lemmas helps for a morphologically rich language such as French. So an intermediate clustering standing between syntactic category and lemma is thought to yield better results in the future.

7 Acknowledgments

We thank N. Schluter and J. van Genabith for kindly letting us run BKY on the MFT. We also thank the reviewers for valuable comments and references. The work of the second author was partly funded by the "Prix Diderot Innovation 2007", from University Paris 7.

References

Anne Abeillé, Lionel Clément, and Alexandra Kinyon. 2000. Building a treebank for french. In *Proceedings of the 2nd International Conference Language Resources and Evaluation (LREC'00)*.

Anne Abeillé, Lionel Clément, and François Toussenel, 2003. *Building a treebank for French*. Kluwer, Dordrecht.

Abhishek Arun and Frank Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of french. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 306–313, Ann Arbor, MI.

Daniel M. Bikel. 2004a. A distributional analysis of a lexicalized statistical parsing model. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP 2004)*, volume 4, pages 182–189, Barcelona, Spain.

Daniel M. Bikel. 2004b. Intricacies of Collins' Parsing Model. *Computational Linguistics*, 30(4):479–511.

Thorsten Brants. 2000. Tnt – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP)*, Seattle-WA.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the Annual Meeting of the North American Association for Computational Linguistics (NAACL-00)*, pages 132–139, Seattle, Washington.

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

Benoit Crabbé and Marie 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.

Markus Dickinson and W. Detmar Meurers. 2005. Prune diseased branches to get healthy trees! how to find erroneous local trees in treebank and why it matters. In *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for german using sister-head dependencies. In *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103.

Ane Dybro-Johansen. 2004. Extraction automatique de grammaires á partir d'un corpus français. Master's thesis, Université Paris 7.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Dan Klein and Christopher D. Manning. 2003a. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics Morristown, NJ, USA.

Dan Klein and Christopher D. Manning. 2003b. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse german? In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 111–119, Sydney, Australia, July. Association for Computational Linguistics.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *International Joint Conference on Artificial Intelligence*, pages 1420–1425, Montreal.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 75–82.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan 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, July. Association for Computational Linguistics.

Natalie Schluter and Josef van Genabith. 2007. Preparing, restructuring, and augmenting a french treebank: Lexicalised parsers or coherent treebanks? In *Proceedings of PACLING 07*.

Natalie Schluter and Josef van Genabith. 2008. Treebank-based acquisition of lfg parsing resources for french. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may.