Cross Parser Evaluation and Tagset Variation : a French Treebank Study

Djamé Seddah[†], Marie Candito[‡] and Benoît Crabbé[‡]

[†] Université Paris-Sorbonne
LALIC & INRIA (ALPAGE)
28 rue Serpente
F-75006 Paris — France

[‡] Université Paris 7
INRIA (ALPAGE)
30 rue du Château des Rentiers
F-75013 Paris — France

Abstract

This paper presents preliminary investigations on the statistical parsing of French by bringing a complete evaluation on French data of the main probabilistic lexicalized and unlexicalized parsers first designed on the Penn Treebank. We adapted the parsers on the two existing treebanks of French (Abeillé et al., 2003; Schluter and van Genabith, 2007). To our knowledge, mostly all of the results reported here are state-of-the-art for the constituent parsing of French on every available treebank. Regarding the algorithms, the comparisons show that lexicalized parsing models are outperformed by the unlexicalized Berkeley parser. Regarding the treebanks, we observe that, depending on the parsing model, a tag set with specific features has direct influence over evaluation results. We show that the adapted lexicalized parsers do not share the same sensitivity towards the amount of lexical material used for training, thus questioning the relevance of using only one lexicalized model to study the usefulness of lexicalization for the parsing of French.

1 Introduction

The development of large scale symbolic grammars has long been a lively topic in the French NLP community. Surprisingly, the acquisition of probabilistic grammars aiming at stochastic parsing, using either supervised or unsupervised methods, has not attracted much attention despite the availability of large manually syntactic annotated data for French. Nevertheless, the availability of the Paris 7 French Treebank (Abeillé et al., 2003), allowed (Dybro-Johansen, 2004) to carry out the extraction of a Tree Adjoining Grammar (Joshi, 1987) and led (Arun and Keller, 2005) to induce the first effective lexicalized parser for French. Yet, as noted by (Schluter and van Genabith, 2007), the use of the treebank was "challenging". Indeed, before carrying out successfully any experiment, the authors had to perform a deep restructuring of the data to remove errors and inconsistencies. For the purpose of building a statistical LFG parser, (Schluter and van Genabith, 2007; Schluter and van Genabith, 2008) have reannotated a significant subset of the treebank with two underlying goals: (1) designing an annotation scheme that matches as closely as possible the LFG theory (Kaplan and Bresnan, 1982) and (2) ensuring a more consistent annotation. On the other hand, (Crabbé and Candito, 2008) showed that with a new released and corrected version of the treebank[1] it was possible to train statistical parsers from the original set of trees. This path has the advantage of an easier reproducibility and eases verification of reported results.

With the problem of the usability of the data source being solved, the question of finding one or many accurate language models for parsing French raises. Thus, to answer this question, this paper reports a set of experiments where five algorithms, first designed for the purpose of parsing English, have been adapted to French: a PCFG parser with latent annotation (Petrov et al., 2006), a Stochastic Tree Adjoining Grammar parser (Chiang, 2003), the Charniak's lexicalized parser (Charniak, 2000) and the Bikel's implementation of Collins' Model 1 and 2 (Collins, 1999) described in (Bikel, 2002). To ease further comparisons, we report results on two versions of the treebank: (1) the last version made available in December 2007, hereafter FTB , and described in (Abeillé and Barrier, 2004) and the (2) LFG inspired version of (Schluter and van Genabith, 2007).

The paper is structured as follows : After a brief presentation of the treebanks, we discuss the use-

[1] This has been made available in December 2007.

fulness of testing different parsing frameworks over two parsing paradigms before introducing our experimental protocol and presenting our results. Finally, we discuss and compare with related works on cross-language parser adaptation, then we conclude.

2 Treebanks for French

This section provides a brief overview to the corpora on which we report results: the French Treebank (FTB) and the Modified French Treebank (MFT).

2.1 The French Treebank

THE FRENCH TREEBANK is the first treebank annotated and manually corrected for French. It is the result of a supervised annotation project of newspaper articles from *Le Monde* (Abeillé and Barrier, 2004). The corpus is annotated with labelled constituent trees augmented with morphological annotations and functional annotations of verbal dependents as shown below :

```
<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: "Le bilan n'est peut-être pas aussi sombre." *(i.e. The result is perhaps not as bleak)*

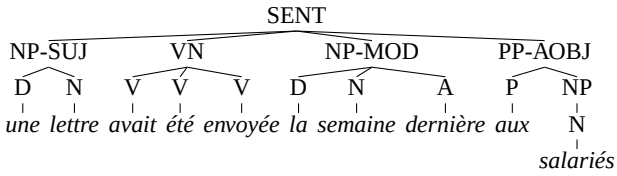
Though the original release (*in 2000*) consists of 20,648 sentences, the subset of 12351 functionally annotated sentences is known to be more consistently annotated and therefore is the one used in this work. Its key properties, compared with the Penn Treebank (hereafter PTB, (Marcus et al., 1994)), are the following :

Size: The FTB consists of 385,458 tokens and 12,351 sentences, that is the third of the PTB. It also entails that the average length of a sentence is 27.48 tokens. By contrast the average sentence length in the PTB is 24 tokens.

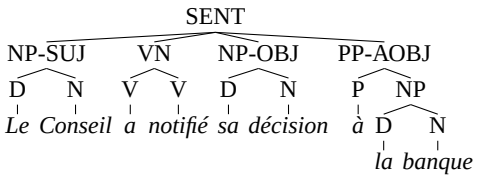
Inflection: French morphology is richer than English and leads to increased data sparseness issues for the purpose of statistical parsing. There are 24,098 types in the FTB, entailing an average of 16 tokens occurring for each type.

A Flat Annotation Scheme: Both the FTB and the PTB are annotated with constituent trees. However, the annotation scheme is flatter in the FTB. For instance, there are no VPs for finite verbs and only one sentential level for clauses or sentences whether or not introduced by a complementizer. Only *verbal nucleus* (VN) is annotated and comprises the verb, its clitics, auxiliaries, adverbs and surrounding negation.

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 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 all cases, other subcategorized complements may appear.



(a) A letter had been sent last week to the employees



(b) The Council has notified his decision to the bank

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

Compounds: Compounds are explicitly annotated and very frequent in the treebank: 14.52% of tokens are part of a compound (see the compound *peut-être 'perhaps'* in Figure 1). They include

digit numbers (written with spaces in French) (e.g. *10 000*), frozen compounds (eg. *pomme de terre 'potato'*) but also named entities or sequences whose meaning is compositional but where insertion is rare or difficult (e.g. *garde d'enfant 'child care'*). As noted by (Arun and Keller, 2005), compounds in French may exhibit ungrammatical sequences of tags as in *à la va vite* 'in a hurry' : Prep+ Det+ finite verb + adverb or can include "words" which do not exist outside a compound (e.g *hui* in *aujourd'hui* 'today'). Therefore, compounds receive a two-level annotation : constituent parts are described in a subordinate level using the same POS tagset as the genuine compound POS. This makes it more difficult to extract a proper grammar from the FTB without merged compounds[2]. This is why, following (Arun and Keller, 2005) and (Schluter and van Genabith, 2007), all the treebanks used in this work contain compounds.

2.2 The Modified French Treebank

THE MODIFIED FRENCH TREEBANK (MFT) has been derived from the FTB by (Schluter and van Genabith, 2008) as a basis for a PCFG-based Lexical Functional Grammar induction process (Cahill et al., 2004) for French. The corpus is a subset of 4739 sentences extracted from the original FTB. The MFT further introduces formal differences of two kinds with respect to the original FTB: structural and labeling modifications.
Regarding structural changes, the main transformations include increased rule stratification (Fig. 3), coordination raising (Fig. 5).

Moreover, the MFT's authors introduced new treatments of linguistic phenomena that were not covered by their initial source treebank. Those include, for example, analysis for 'It'-cleft constructions.[3] Since the MFT was designed for the purpose of improving the task of grammar induction, the MFT's authors also refined its tag set by propagating information (such as mood features added to VN node labels), and added functional paths[4] to the original function labels. The modifications introduced in the MFT meet better the formal requirements of the LFG architecture set up

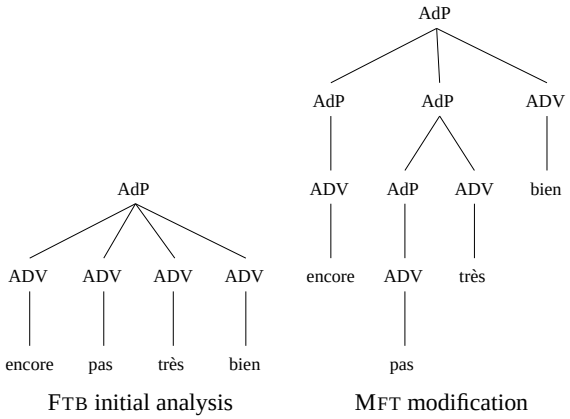


Figure 3: Increased stratification in the MFT : *"encore pas très bien"* ('still not very well')

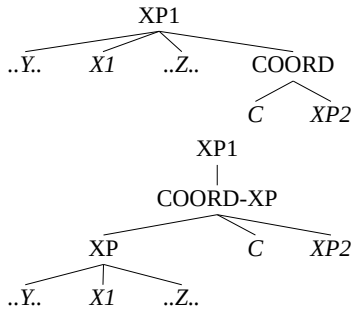


Figure 5: Coordinated structures in the general case, for FTB (up) and MFT (down)

by (Cahill et al., 2004) and reduce the size of the grammars extracted from the treebank. MFT has also undergone a phase of error mining and an extensive manual correction.

2.3 Coordination in French Treebanks

One of the key differences between the two French treebanks is the way they treat coordinate structures. Whereas the FTB represents them with an adjunction of a COORD phrase as a sister or a daughter of the coordinated element, the MFT introduces a treatment closer to the one used in the PTB to describe such structures. As opposed to (Arun and Keller, 2005) who decided to transform the FTB's coordinations to match the PTB's analysis, the COORD label is not removed but extended to include the coordinated label (Fig. 5).

In Figure 5, we show the general coordination structure in the FTB, and the corresponding modified structure in the MFT. A more complicated modification concerns the case of *VP coordinations*. (Abeillé et al., 2003) argue for a flat representation with no VP-node for French, and this is

[2]Consider the case of the compound *peut-être 'perhaps'* whose POS is ADV, its internal structure (Fig. 1) would lead to a CFG rule of the form ADV $\longrightarrow$ V V.

[3]See pages 2-3 of (Schluter and van Genabith, 2007) for details.

[4]Inspired by the LFG framework (Dalrymple, 2001).

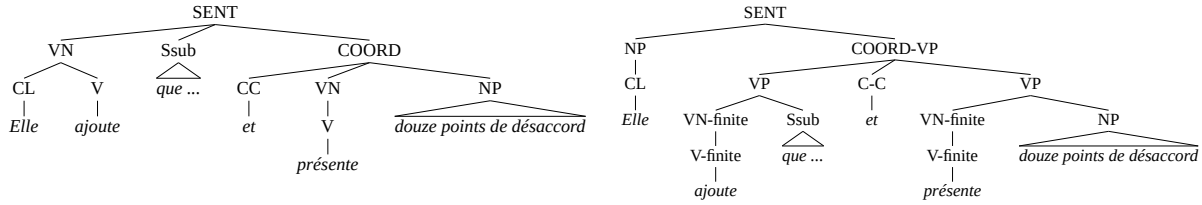


Figure 4: Two representations of "VP coordinations" for the sentence *She adds that ... and presents twelve sticking points*: in the FTB (left) and in the MFT (right)

particularly justified in some cases of subject-verb inversion. Nevertheless, VP phrases are used in the FTB for non-finite VPs only (nodes VPinf and VPpart). In the MFT, finite VPs were introduced to handle *VP coordinations*. In those cases, the FTB annotation scheme keeps a flat structure (Figure 4, left), where the COORD phrase has to be interpreted as a coordinate of the VN node; whereas finite VP nodes are inserted in the MFT (Figure 4, right).

2.4 Summary

In Table 2, we describe the annotation schemes of the treebanks and we provide in Table 1 a numeric summary of some relevant different features between these two treebanks. The reported numbers take into account the base syntactic category labels without functions, part-of-speech tags without any morpho-syntactic information (ie. no 'gender' or number').

properties	FTB	MFT
# of sentences	12351	4739
Average sent. length	27.48	28.38
Average node branching	2.60	2.11
PCFG size (without term. prod.)	14874	6944
# of NT symbols	13	39
# of POS tags	15	27

Table 1: Treebanks Properties

3 Parsing Algorithms

Although Probabilistic Context Free Grammars (PCFG) are a baseline formalism for probabilistic parsing, it is well known that they suffer from two problems: (a) The independence assumptions made by the model are too strong, and (b) For Natural Language Parsing, they do not take into account lexical probabilities. To date, most of the results on statistical parsing have been reported for English. Here we propose to investigate how to apply these techniques to another language – French – by testing two distinct enhancements

	FTB	MFT
POS tags	A ADV C CL D ET I N P P+D P+PRO PONCT PREF PRO V	A A_card ADV ADV_int ADVne A_int CC CL C_S D D_card ET I N N_card P P+D PONCT P+PRO_rel PREF PRO PRO_card PRO_int PRO_rel V_finite V_inf V_part
NT labels	AP AdP COORD NP PP SENT Sint Srel Ssub VN VPinf VPpart	AdP AdP_int AP AP_int COORD_XP COORD_UC COORD_unary NC NP NP_int NP_rel PP PP_int PP_rel SENT Sint Srel Ssub VN_finite VN_inf VN_part VP VPinf VPpart VPpart_rel

Table 2: FTB's and MFT's annotation schemes

over the bare PCFG model carried out by two class of parser models: an unlexicalized model attempting to overcome problem (a) and 3 different lexicalized models attempting to overcome PCFG's problems (a) and (b)[5].

3.1 Lexicalized algorithms

The first class of algorithms used are lexicalized parsers of (Collins, 1999; Charniak, 2000; Chiang, 2003). The insight underlying the lexicalized algorithms is to model lexical dependencies between a governor and its dependants in order to improve attachment choices.

Even though it has been proven numerous times that lexicalization was useful for parsing the *Wall Street Journal* corpus (Collins, 1999; Charniak, 2000), the question of its relevance for other languages has been raised for German (Dubey and Keller, 2003; Kübler et al., 2006) and for French

[5]Except (Chiang, 2003) which is indeed a TREE INSERTION GRAMMAR (Schabes and Waters, 1995) parser but which must extract a lexicalized grammar from the set of context free rules underlying a treebank.

(Arun and Keller, 2005) where the authors argue that French parsing benefits from lexicalization but the treebank flatness reduces its impact whereas (Schluter and van Genabith, 2007) argue that an improved annotation scheme and an improved treebank consistency should help to reach a reasonable state of the art. As only Collins' models 1 & 2 have been used for French as instances of lexicalised parsers, we also report results from the history-based generative parser of (Charniak, 2000) and the Stochastic Tree Insertion Grammar parser of (Chiang, 2003) as well as (Bikel, 2002)'s implementation of the Collins' models 1 & 2 (Collins, 1999). Most of the lexicalized parsers we use in this work are well known and since their releases, almost ten years ago, their core parsing models still provide state-of-the-art performance on the standard test set for English.[6] We insist on the fact that one of the goals of this work was to evaluate raw performance of well known parsing models on French annotated data. Thus, we have not considered using more complex parsing architectures that makes use of reranking (Charniak and Johnson, 2005) or self-training (McClosky et al., 2006) in order to improve the performance of a raw parsing model. Furthermore, studying and designing a set of features for a reranking parser was beyond the scope of this work. However, we did use some of these models in a non classical way, leading us to explore a Collins' model 2 variation, named model X, and a Stochastic Tree Adjoining Grammar (Schabes, 1992; Resnik, 1992) variant[7] , named Spinal Stochastic Tree Insertion Grammars (hereafter SPINAL STIG), which was first used to validate the heuristics used by our adaptation of the Bikel's parser to French. The next two subsections introduce these variations.

Collins' Model 2 variation During the exploratory phase of this work, we found out that a specific instance of the Collins' model 2 leads to significantly better performance than the canonical model when applied to any of the French Treebanks. The difference between those two models relies on the way probabilities associated to so-called "modifier non terminals" nodes are handled by the generative model.

To explain the difference, let us recall that a lexicalized PCFG can roughly be described as a set of stochastic rules of the form: $\boxed{P \rightarrow L_n\ L_{n-1}\ ..L_1\ H\ R_1\ ..\ R_{m-1}\ R_m}$ where L_i, H, R_i and P are all lexicalized non terminals; P inherits its head from H (Bikel, 2004). The Collins' model 2 deterministically labels some nodes of a rule to be arguments of a given Head and the remaining nodes are considered to be modifier non terminals (hereafter MNT).

In this model, given a left-hand side symbol, the head and its arguments are first generated and then the MNT are generated from the head outward. In Bikel's implementation of Collins's model 2 (Bikel, 2004), the MNT parameter class is the following (for clarity, we omit the *verb intervening*, *subcat* and *side* features which are the same in both classes) :

- model 2 (canonical) :
  $\boxed{p(M(t)_i|P, H, w_h, t_h, map(M_{i-1}))}$
  Where $M(t)_i$ is the POS tag of the i^{th} MNT, P the parent node label, H the head node label, w_h the head word and t_h its POS tag. $map(M_{i-1})$ is a mapped version of the previously-generated modifier added to the conditioning context (see below for its definition).

$$map(M_i) = \left\{ \begin{array}{ll} +START+ & \text{if } i = 0 \\ CC & \text{if } M_i = CC \\ +PUNC+ & \text{if } M_i =, \\ & \text{or } M_i =: \\ +OTHER+ & \text{otherwise} \end{array} \right\}$$

Whereas in the model we call X [8], the mapping version of the previously generated non terminal is replaced by a complete list of all previously generated non terminals.

- Model X :
  $\boxed{p(M(t)_i|P, H, w_h, t_h, (M_{i-1}, ..., M_{i-k}))}$

The FTB being flatter than the PTB, one can conjecture that giving more context to generate MNT will improve parsing accuracy, whereas clustering MNT in a X-bar scheme must help to reduce data sparseness. Note that the Model X, to the best of our knowledge, is not documented but included in Bikel's parser.

[6] Section 23 of the Wall Street Journal section of the PTB.

[7] The formalism actually used in this parser is a context free variant of Tree Adjoining Grammar, Tree Insertion Grammars (TIG), first introduced in (Schabes and Waters, 1995).

[8] See file NonTerminalModelStructure1.java in Bikel's parser source code at `http://www.cis.upenn.edu/~dbikel/download/dbparser/1.2/install.sh`.

The spinal STIG model In the case of the STIG parser implementation, having no access to an argument adjunct table leads it to extract a grammar where almost all elementary trees consist of a suite of unary productions from a lexical anchor to its maximal projection (i.e. spine[9]). Therefore extracted trees have no substitution node.
Moreover, the probability model, being split between lexical anchors and tree templates, allows a very coarse grammar that contains, for example, only 83 tree templates for one treebank instantiation, namely the FTB-CC (cf. section 5).
This behavior, although not documented[10], is close to Collins' model 1, which does not use any argument adjunct distinction information, and led to results interesting enough to be integrated as the "Chiang Spinal" model in our parser set. It should be noted that, recently, the use of similar models has been independently proposed in (Carreras et al., 2008) with the purpose of getting a richer parsing model that can use non local features and in (Sangati and Zuidema, 2009) as a mean of extracting a Lexicalized Tree Substitution Grammar. In their process, the first extracted grammar is actually a spinal STIG.

3.2 Unlexicalized Parser

As an instance of an unlexicalized parser, the last algorithm we use is the Berkeley unlexicalized parser (BKY) of (Petrov et al., 2006). This algorithm is an evolution of treebank transformation principles aimed at reducing PCFG independence assumptions (Johnson, 1998; Klein and Manning, 2003).

Treebank transformations may be of two kinds (1) structure transformation and (2) labelling transformations. The Berkeley parser concentrates on (2) by recasting the problem of acquiring an optimal set of non terminal symbols as an semi-supervised learning problem by learning a PCFG with Latent annotations (PCFG-LA): 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 actual treebank trees (or *observed trees*) using a specific instanciation of EM.

4 Experimental protocol

In this section, we specify the settings of the parsers for French, the evaluation protocol and the different instantiations of the treebanks we used for conducting the experiments.

4.1 Parsers settings

Head Propagation table All lexicalized parsers reported in this paper use head propagation tables. Adapting them to the French language requires to design French specific head propagation rules. To this end, we used those described by (Dybro-Johansen, 2004) for training a Stochastic Tree Adjoining Grammar parser on French. From this set, we built a set of meta-rules that were automatically derived to match each treebank annotation scheme.
As the Collins Model 2 and the STIG model need to distinguish between argument and adjunct nodes to acquire subcategorization frames probabilities, we implemented an argument-adjunct distinction table that takes advantage of the function labels annotated in the treebank. This is one of the main differences with the experiments described in (Arun and Keller, 2005) and (Dybro-Johansen, 2004) where the authors had to rely only on the very flat treebank structure without function labels, to annotate the arguments of a head.

Morphology and typography adaptation Following (Arun and Keller, 2005), we adapted the morphological treatment of unknown words proposed for French when needed (BKY's and BIKEL's parser). This process clusters unknown words using typographical and morphological information. Since all lexicalized parsers contain specific treatments for the PTB typographical convention, we automatically converted the original punctuation parts of speech to the PTB's punctuation tag set.

4.2 Experimental details

For the BKY parser, we use the Berkeley implementation, with an initial horizontal markovization h=0, and 5 split/merge cycles. For the COLLINS' MODEL, we use the standard parameters set for the model 2, without any argu-

[9] Not to be confused with the "spine" in the Tree Adjunct Grammar (Joshi, 1987) framework which is the path from a foot node to the root node.

[10] We mistakenly "discovered" this obvious property during the preliminary porting phase.

ment adjunct distinction table, as a rough emulation of the COLLINS MODEL 1. The same set of parameters used for COLLINS' MODEL 2 is used for the MODEL X except for the parameters "Mod{Nonterminal,Word}ModelStructureNumber" set to 1 instead of 2.

4.3 Protocol

For all parsers, we report parsing results with the following experimental protocol: a treebank is divided in 3 sections : test (first 10%), development (second 10%) and training (remaining 80%). The MFT partition set is the canonical one (3800 sentences for training, 509 for the dev set and the last 430 for the test set). We systematically report the results with compounds merged. Namely, we preprocess the treebank in order to turn each compound into a single token both for training and test.

4.4 Evaluation metrics

Constituency Evaluation: we use the standard labeled bracketed PARSEVAL metric for evaluation (Black et al., 1991), along with unlabeled dependency evaluation, which is described as a more annotation-neutral metric in (Rehbein and van Genabith, 2007). In the remainder of this paper, we use PARSEVAL as a shortcut for Labeled Brackets results on sentence of length 40 or less.
Dependency Evaluation: unlabeled dependencies are computed using the (Lin, 1995) algorithm, and the Dybro Johansens's head propagation rules cited above[11]. The unlabeled dependency accuracy gives the percentage of input words (excluding punctuation) that receive the correct head. All reported evaluations in this paper are calculated on sentences of length less than 40 words.

4.5 Baseline : Comparison using minimal tagsets

We compared all parsers on three different instances, but still comparable versions, of both the FTB and the MFT. In order to establish a baseline, the treebanks are converted to a minimal tag set (only the major syntactic categories.) without any other information (no mode propagation as in the MFT) except for the BIKEL's parser in Collins' model 2 (resp. model X) and the STIG parser (i.e. STIG-pure) whose models needs function labels to perform.
Note that by stripping all information from the node labels in the treebanks, we do not mean to compare the shape of the treebanks or their *parsability* but rather to present an overview of parser performance on each treebank regardless of tagset optimizations. However, in each experiment we observe that the BKY parser significantly outperforms the other parsers in all metrics.
As the STIG parser presents non statistically significant PARSEVAL results differences between its two modes (PURE & SPINAL) with a f-score p-value of 0.32, for the remaining of the paper we will only present results for the STIG's parser in "spinal" mode.

		FTB-min	MFT-min
COLLINS MX	PARSEVAL	81.65	79.19
	UNLAB. DEP	88.48	84.96
COLLINS M2	PARSEVAL	80.1	78.38
	UNLAB. DEP	87.45	84.57
COLLINS M1	PARSEVAL	77.98	76.09
	UNLAB. DEP	85.67	82.83
CHARNIAK	PARSEVAL	82,44	81.34
	UNLAB. DEP	88.42	84.90
CHIANG-SPINAL	PARSEVAL	80.66	80.74
	UNLAB. DEP	87.92	85,14
BKY	PARSEVAL	84,93	83.16
	UNLAB. DEP	90.06	87.29
CHIANG-PURE	PARSEVAL	80.52	79.56
	UNLAB. DEP	87,95	85.02

Table 3: Labeled F_1 scores for unlexicalised and lexicalised parsers on treebanks with minimal tagsets

5 Cross parser evaluation of tagset variation

In (Crabbé and Candito, 2008), the authors showed that it was possible to accurately train the Petrov's parser (Petrov et al., 2006) on the FTB using a more fine grained tag set. This tagset, named CC[12] annotates the basic non-terminal labels with verbal mood information, and wh-features. Results were shown to be state of the art with a F_1 parseval score of 86.42% on less than 40 words sentences.
To summarize, the authors tested the impact of tagset variations over the FTB using constituency measures as performance indicators.
Knowing that the MFT has been built with PCFG-based LFG parsing performance in mind (Schluter

[11] 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.

[12] TREEBANKS+ in (Crabbé and Candito, 2008).

and van Genabith, 2008) but suffers from a small training size and yet allows surprisingly high parsing results (PARSEVAL F-score (<=40) of 79.95 % on the MFT gold standard), one would have wished to verify its

performance with more annotated data.

However, some semi-automatic modifications brought to the global structure of this treebank cannot be applied, in an automatic and reversible way, to the FTB. Anyway, even if we cannot evaluate the influence of a treebank structure to another, we can evaluate the influence of one tagset to another treebank using handwritten conversion tools. In order to evaluate the relations between tagsets and parsing accuracy on a given treebank, we extract the optimal tagsets[13] from the FTB, the CC tagset and we convert the MFT POS tags to this tagset. We then do the same for the FTB on which we apply the MFT's optimal tagset (ie. SCHLU). Before introducing the results of our experiments, we briefly describe these tagsets.

1. **min** : Preterminals are simply the main categories, and non terminals are the plain labels
2. **cc** : (Crabbé and Candito, 2008) best tagset. Preterminals are the main categories, concatenated with a wh- boolean for A, ADV, PRO, and with the mood for verbs (there are 6 moods). No information is propagated to non terminal symbols. This tagset is shown in Table 4, and described in (Crabbé and Candito, 2008).

   ADJ ADJWH ADV ADVWH CC CLO CLR CLS CS DET DETWH ET I NC NPP P P+D P+PRO PONCT PREF PRO PROREL PROWH V VIMP VINF VPP VPR VS

   Table 4: CC tagset

3. **schlu** : N. Schluter's tagset (Table 2. Preterminals are the main categories, plus an inf/finite/part verbal distinction, and int/card/rel distinction on N, PRO, ADV, A. These distinctions propagate to non terminal nodes projected by the lexical head. Non terminals for coordinating structures are split according to the type of the coordinated phrases.

Results of these experiments, presented in Table 5, show that BKY displays higher performances in every aspects (constituency and dependency, except for the MFT-SCHLU). Regardless of the parser type, we note that unlabeled dependency scores are higher with the SCHLU tagset than with the CC tagset. That can be explained by the finest granularity of the SCHLU based rule set compared to the other tagset's rules. As these rules have all been generated from meta description (a general COORD label rewrites into COORD_vfinite, COORD_Sint, etc..) their coverage and global accuracy is higher. For example the FTB-CC contains 18 head rules whereas the FTB-SCHLU contains 43 rules.

Interestingly, the ranking of lexicalized parsers w.r.t PARSEVAL metrics shows that CHARNIAK has the highest performance over both treebank tagsets variation even though the MFT's table (table 5) exhibits a non statistically significant variation between CHARNIAK and STIG-spinal on PARSEVAL evaluation of the MFT-CC.[14]

One the other hand, unlabeled dependency evaluations over lexicalized parsers are different among treebanks. In the case of the FTB, CHARNIAK exhibits the highest F-score (FTB-CC: 89.7, FTB-SCHLU: 89.67) whereas SPINAL STIG performs slightly better on the MFT-SCHLU (MFT-CC: 86,7, MFT-SCHLU: 87.16). Note that both tested variations of the Collins' model 2 display very high unlabeled dependency scores with the SCHLU tagset.

6 Related Works

As we said in the introduction, the initial work on the FTB has been carried by (Dybro-Johansen, 2004) in order to extract Tree Adjunct Grammars from the treebank. Although parsing results were not reported, she experienced the same argument adjunct distinction problem than (Arun and Keller, 2005) due to the treebank flatness and the lack of functional labels in this version. This led Arun to modify some node annotations (VNG to distinguish nodes dominating subcategorized subject clitics and so on) and to add bigrams probabilities to the language model in order to enhance the overall COLLINS' MODEL' performance. Although our treebanks cannot be compared (20.000 sentences for Arun's one vs 12351 for the FTB), we report his best PARSEVAL results (<=40): 80.65 LP, 80.25 LR, 80.45 F1.

However, our results are directly comparable with

[13] W.r.t constituent parsing accuracy

[14] Precision P-value = 0.1272 and Recall = 0.06.

Parser	Parseval		Dependency		Parseval		Dependency	
	MFTCC	MFTSCH.	MFTCC	MFTSCH.	FTBCC	FTBSCH.	FTBCC	FTBSCH.
Collins (MX)	80.2	80.96	85.97	**87.98**	82.52	82.65	88.96	89.12
Collins (M2)	78.56	79.91	84.84	87.43	80.8	79.56	87.94	87.87
Collins (M1)	74	78.49	81.31	85.94	79.16	78.51	86.66	86.93
Charniak	82.5	82.66	86.45	86.94	84.27	83.27	89.7	89.67
Chiang (Sp)	82.6	81.97	86.7	87.16	81.73	81.54	88.85	89.02
Bky	**83.96**	**82.86**	**87.41**	86.87	**86.02**	**84.95**	**90.48**	**90.73**

Table 5: Evaluation Results: MFT-CC vs MFT-SCHLU and FTB-CC vs FTB-SCHLU

(Schluter and van Genabith, 2007) whose best PARSEVAL F-score on raw text is 79.95 and our best 82.86 on the MFT-SCHLU.

PARSER	FTBARUN	MFTSCHLU
Arun (acl05)	**80.45**	-
Arun (this paper)	**81.08**	-
Schluter (pacling07)	-	**79.95**
Collins (Mx)	81.5	80,96
Collins (M2)	79.36	79,91
Collins (M1)	77.82	-
Charniak	82.35	82,66
Chiang (Sp)	80.94	81,86
Bky	84.03	82.86

Table 6: Labeled bracket scores on Arun's FTB version and on the MFT

In order to favour a "fair" comparison between our work and (Arun and Keller, 2005), we also ran their best adaptation of the COLLINS MODEL 2 on their treebank version using our own head rules set[15] and obtained 81.08% of F_1 score (Table 6). This shows the important influence of a fine grained head rules set and argues in favor of data driven induction of this kind of heuristics. Even though it was established, in (Chiang and Bikel, 2002), that unsupervised induction of head rules did not lead to improvement over an extremely hand crafted head rules set, we believe that for resource poor languages, such methods could lead toward significant improvements over parsing accuracy. Thus, the new unsupervised head rules induction method presented in (Sangati and Zuidema, 2009) seems very promising for this topic.

However, it would be of interest to see how the Arun's model would perform using the MODEL X parameter variations.

7 Discussion

Regarding the apparent lack of success of a genuine COLLINS' MODEL 2 (in most cases, its performance is worse than the other parsers w.r.t to constituent parsing accuracy) when trained on a treebank with annotated function labels, we suspect that this is caused by the increased data sparseness added by these annotations. The same can be said about the pure STIG model, whose results are only presented on the FTB-MIN because the differences between the spinal model and itself were too small and most of the time not statistically significant. In our opinion, there might be simply not enough data to accurately train a pure COLLINS' MODEL 2 on the FTB with function labels used for clues to discriminate between argument and adjuncts. Nevertheless, we do not share the commonly accepted opinion about the potential lack of success of lexicalized parsers.

To the best of our knowledge, most adaptations of a lexicalized model to a western language have been made with Dan Bikel's implementation of COLLINS' MODELS.[16]

In fact, the adaptations of the CHARNIAK and BKY's models exhibit similar magnitudes of performances for French as for English. Evidence of lexicalization usefulness is shown through a learning curve (Figure 6) obtained by running some of our parsers in perfect tagging mode. This experiment was done in the early stages of this work, the goal was to see how well the parsers would behave with the same head rules and the same set of parameters. We only compared the parsers that could be used without argument adjunct distinction table (ie. COLLIN'S MODEL 1, SPINAL STIG, CHARNIAK and BKY).

For this earlier experiment, our implementation of the COLLINS MODEL 1 actually corresponds to the MODEL X without an argument adjunct distinction table. More precisely, the absence of argument nodes, used for the acquisition of subcategorization frames features, makes the MODEL X parsing model consider all the nodes of a rule, ex-

[15] Due to the lack of function annotation labels in this treebank, (Arun and Keller, 2005)'s argument distinction table was used for this experiment.

[16] Note that the CHARNIAK's parser has been adapted for Danish (Zeman and Resnik, 2008) ; the authors report a 80.20 F_1 score for a specific instance of the Danish Treebank.

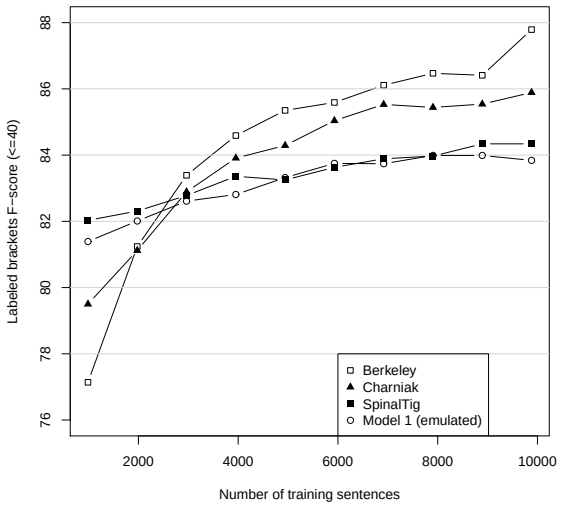


Figure 6: Learning Curve experiment results for parsers in perfect tagging mode

cept the head, as Modifier Non Terminal nodes (MNTs). Hence, because of the impossibility to extract subcategorization frames, the generation of a MNT depends mainly on the parent head word and on the whole list of previously generated MNTs. One can suppose that training on small treebanks would lead this distribution to be sparse, therefore most of the discriminant information would come from less specific distributions. Namely the ones conditioned on the head pos tag and on the last previously generated MNT as shown in this model back-off structure (Table 7).

Back-off level	$p(M(t)_i \vert \cdots)$
0	$P, H, w_h, t_h, \langle M_{i-1}, ..., M_{i-k} \rangle$
1	P, H, t_h, M_{i-1}
2	P, H, f

Table 7: MODEL X simplified parameter class for MNTs

$M(t)_i$ is the POS tag of the i^{th} MNT, P the parent node label, H the head node label, w_h the head word, t_h its POS tag, $\langle M_{i-1}, ..., M_{i-k} \rangle$ the list of previously generated MNTs and f a flag stating if the current node is the first MNT to be generated.

Interestingly, in the SPINAL STIG model, almost all the extracted trees are spinal and consequently are handled by an operation called *Sister Adjunction* whose probability model for a given root node of an elementary tree, also conditions its generation upon the label of the previously generated tree (Chiang, 2003). Furthermore, the second component of the *Sister Adjunction*'s back-off structure (Table 8) is made coarser by the removing of the lexical anchor of the tree where a sister-adjunction is to occur.

Studying in depth the respective impact of these features on the performance of both models is outside the scope of this paper, nevertheless we note that their back-off structures are based on similar principles: a deletion of the main lexical information and a context limited to the root label of the previously generated tree (resp. MNT node label for the MODEL X). This can explain why these formally different parsers display almost the same learning curves (Fig. 6) and more over why they surprisingly exhibit few sensitivity to the amount of lexical material used for training.

Back-off level	$P_{sa}(\gamma \vert \cdots)$
0	$\tau_\eta, \omega_\eta, \eta_\eta, i, X$
1	$\tau_\eta, \eta_\eta, i, X$
2	$\overline{\tau}_\eta, \eta_\eta, i$

Table 8: SPINAL STIG parameter class for Sister-adjoining tree templates (Chiang, 2003)

γ is the tree to be generated on the sister adjunction site (η_η, i) of the tree template τ_η, ω_η is the lexical anchor of τ_η, $\overline{\tau}_\eta$ is τ_η stripped from its anchor POS tag and X is the root label of the previous tree to sister-adjoin at the site (η_η, i).

However, the learning curve also shows that the CHARNIAK's[17] and BKY's parsers have almost parallel curves whereas this specific COLLIN'S MODEL 1 parser and the SPINAL STIG model have very similar shape and seem to reach an upper limit very quickly.[18] The last two parsers having also very similar back-off models (Chiang, 2003), we wonder (1) if we are not actually comparing them because of data sparseness issues and (2) if the small size of commonly used treebanks does not lead the community to consider lexicalized models, via the lone COLLINS' MODELS, as inappropriate to parse other languages than *Wall Street Journal* English.

[17] As opposed to the other parsers, the Charniak's parser tagging accuracy did not reach the 100% limit, 98.32% for the last split. So the comparison is not really fair but we believe that the visible tendency still stands.

[18] We are of course aware that the curve's values are also function of the amount of new productions brought by the increased treebank size. That should be of course taken into account.

Regarding the remarkable performance of the BKY algorithm, it remains unclear why exactly it systematically outperforms the other lexicalized algorithms. We can only make a few remarks about that. First, the algorithm is totally disjoint from the linguistic knowledge, that is entirely taken from the treebank, except for the suffixes used for handling unknown words. This is not true of the Collins' or Charniak's models, that were set up with the PTB annotation scheme in mind. Another point concerns the amount of data necessary for an accurate learning. We had the intuition that lexicalized algorithms would have benefited more than BKY from the training data size increase. Yet the BKY's learning curve displays a somewhat faster progression than lexicalized algorithms such as the SPINAL STIG and our specific instance of the COLLINS' MODEL 1.
In our future work, we plan to conduct self-training experiments using discriminative rerankers on very large French corpora to study the exact impact of the lexicon on this unlexicalized algorithm.

8 Conclusion

By adapting those parsers to French and carrying out extensive evaluation over the main characteristics of the treebank at our disposal, we prove indeed that probabilistic parsing was efficient enough to provide accurate parsing results for French. We showed that the BKY model establishes a high performance level on parsing results. Maybe more importantly we emphasized the importance of tag set model to get distinct state of the art evaluation metrics for FTB parsing, namely the SCHLU tagset to get more accurate unlabeled dependencies and the CC tagset to get better constituency parses. Finally, we showed that the lexicalization debate could benefit from the inclusion of more lexicalized parsing models.

9 Acknowledgments

This work was supported by the ANR Sequoia (ANR-08-EMER-013). We heartily thank A. Arun, J. van Genabith an N. Schluter for kindly letting us use our parsers on their treebanks. Thanks to the anonymous reviewers for their comments. All remaining errors are ours. We thank J. Wagner for his help and we would like to acknowledge the Centre for Next Generation Localization (www.cngl.ie) for providing access to one of its high-memory nodes.

References

Anne Abeillé and Nicolas Barrier. 2004. Enriching a french treebank. In *Proceedings of Language Ressources and Evaluation Conference (*LREC*)*, Lisbon.

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. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of the second international conference on Human Language Technology Research*, pages 178–182. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

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

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 Natural Language Workshop*, pages 306–311, San Mateo (CA). Morgan Kaufman.

Aoife Cahill, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 320–327, Barcelona, Spain.

Xavier Carreras, Mickael Collins, and Terry Koo. 2008. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL)*, pages 9–16.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, Ann Arbor (MI).

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL)*, Seattle.

David Chiang and Daniel M. Bikel. 2002. Recovering latent information in treebanks. In *Proceedings of COLING'02, 19th International Conference on Computational Linguistics*, Taipei, Taiwan, August.

David Chiang, 2003. *Statistical Parsing with an Automatically Extracted Tree Adjoining Grammar*, chapter 16, pages 299–316. CSLI Publications.

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, France.

Mary Dalrymple. 2001. *Lexical-Functional Grammar*, volume 34 of *Syntax and Semantics*. San Diego, CA; London. Academic Press.

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 à partir d'un corpus français. Master's thesis, Université Paris 7.

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

Aravind K. Joshi. 1987. Introduction to tree adjoining grammar. In A. Manaster-Ramer, editor, *The Mathematics of Language*. J. Benjamins.

R. Kaplan and J. Bresnan. 1982. Lexical-functional grammar: A formal system for grammarical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. Mass.: MIT Press.

Dan Klein and Christopher D. Manning. 2003. 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.

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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

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.

Ines Rehbein and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for german. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague.

Philip Resnik. 1992. Probabilistic tree-adjoining grammars as a framework for statistic natural language processing. *COLING'92, Nantes, France*.

F. Sangati and W. Zuidema. 2009. Unsupervised methods for head assignments. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 701–709, Athens, Greece. Association for Computational Linguistics.

Y. Schabes and R.C. Waters. 1995. Tree Insertion Grammar: Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced. *Computational Linguistics*, 21(4):479–513.

Yves Schabes. 1992. Stochastic Lexicalized Tree Adjoining Grammars. In *Proceedings of the 14th conference on Computational linguistics*, pages 425–432, Nantes, France. 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.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, Hajdarábádu, India.