Lexicalization in Crosslinguistic Probabilistic Parsing: The Case of French

Abhishek Arun and **Frank Keller**
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
a.arun@sms.ed.ac.uk, keller@inf.ed.ac.uk

Abstract

This paper presents the first probabilistic parsing results for French, using the recently released French Treebank. We start with an unlexicalized PCFG as a baseline model, which is enriched to the level of Collins' Model 2 by adding lexicalization and subcategorization. The lexicalized sister-head model and a bigram model are also tested, to deal with the flatness of the French Treebank. The bigram model achieves the best performance: 81% constituency F-score and 84% dependency accuracy. All lexicalized models outperform the unlexicalized baseline, consistent with probabilistic parsing results for English, but contrary to results for German, where lexicalization has only a limited effect on parsing performance.

1 Introduction

This paper brings together two strands of research that have recently emerged in the field of probabilistic parsing: crosslinguistic parsing and lexicalized parsing. Interest in parsing models for languages other than English has been growing, starting with work on Czech (Collins et al., 1999) and Chinese (Bikel and Chiang, 2000; Levy and Manning, 2003). Probabilistic parsing for German has also been explored by a range of authors (Dubey and Keller, 2003; Schiehlen, 2004). In general, these authors have found that existing lexicalized parsing models for English (e.g., Collins 1997) do not straightforwardly generalize to new languages; this typically manifests itself in a severe reduction in parsing performance compared to the results for English.

A second recent strand in parsing research has dealt with the role of lexicalization. The conventional wisdom since Magerman (1995) has been that lexicalization substantially improves performance compared to an unlexicalized baseline model (e.g., a probabilistic context-free grammar, PCFG). However, this has been challenged by Klein and Manning (2003), who demonstrate that an unlexicalized model can achieve a performance close to the state of the art for lexicalized models. Furthermore, Bikel (2004) provides evidence that lexical information (in the form of bi-lexical dependencies) only makes a small contribution to the performance of parsing models such as Collins's (1997).

The only previous authors that have directly addressed the role of lexicalization in crosslinguistic parsing are Dubey and Keller (2003). They show that standard lexicalized models fail to outperform an unlexicalized baseline (a vanilla PCFG) on Negra, a German treebank (Skut et al., 1997). They attribute this result to two facts: (a) The Negra annotation assumes very flat trees, which means that Collins-style head-lexicalization fails to pick up the relevant information from non-head nodes. (b) German allows flexible word order, which means that standard parsing models based on context free grammars perform poorly, as they fail to generalize over different positions of the same constituent.

As it stands, Dubey and Keller's (2003) work does not tell us whether treebank flatness or word order flexibility is responsible for their results: for English, the annotation scheme is non-flat, and the word order is non-flexible; lexicalization improves performance. For German, the annotation scheme is flat and the word order is flexible; lexicalization fails to improve performance. The present paper provides the missing piece of evidence by applying probabilistic parsing models to French, a language with non-flexible word order (like English), but with a treebank with a flat annotation scheme (like German). Our results show that French patterns with English: a large increase of parsing performance can be obtained by using a lexicalized model. We conclude that the failure to find a sizable effect of lexicalization in German can be attributed to the word order flexibility of that language, rather than to the flatness of the annotation in the German treebank.

The paper is organized as follows: In Section 2, we give an overview of the French Treebank we use for our experiments. Section 3 discusses its annotation scheme and introduces a set of tree transformations that we apply. Section 4 describes the pars-

Proceedings of the 43rd Annual Meeting of the ACL, pages 306–313,
Ann Arbor, June 2005.

```
<NP>
<w lemma="eux" ei="PROmp"
   ee="PRO-3mp" cat="PRO"
   subcat="3mp">eux</w>
</NP>
```

Figure 1: Word-level annotation in the French Treebank: *eux* 'they' (`cat`: POS tag, `subcat`: subcategory, `ei`, `ee`: inflection)

```
<w compound="yes" lemma="d'entre"
   ei="P" ee="P" cat="P">
   <w catint="P">d'</w>
   <w catint="P">entre</w>
</w>
```

Figure 2: Annotation of compounds in the French Treebank: *d'entre* 'between' (`catint`: compound-internal POS tag)

ing models, followed by the results for the unlexicalized baseline model in Section 6 and for a range of lexicalized models in Section 5. Finally, Section 7 provides a crosslinguistic comparison involving data sets of the same size extracted from the French, English, and German treebanks.

2 The French Treebank

2.1 Annotation Scheme

The French Treebank (FTB; Abeillé et al. 2000) consists of 20,648 sentences extracted from the daily newspaper *Le Monde*, covering a variety of authors and domains (economy, literature, politics, etc.).[1] The corpus is formatted in XML and has a rich morphosyntactic tagset that includes part-of-speech tag, 'subcategorization' (e.g., possessive or cardinal), inflection (e.g., masculine singular), and lemma information. Compared to the Penn Treebank (PTB; Marcus et al. 1993), the POS tagset of the French Treebank is smaller (13 tags vs. 36 tags): all punctuation marks are represented as the single PONCT tag, there are no separate tags for modal verbs, *wh*-words, and possessives. Also verbs, adverbs and prepositions are more coarsely defined. On the other hand, a separate clitic tag (CL) for weak pronouns is introduced. An example for the word-level annotation in the FTB is given in Figure 1

The phrasal annotation of the FTB differs from that for the Penn Treebank in several aspects. There is no verb phrase: only the *verbal nucleus* (VN) is annotated. A VN comprises the verb and any clitics, auxiliaries, adverbs, and negation associated with it. This results in a flat syntactic structure, as in (1).

(1) (VN (V sont) (ADV systématiquement) (V arrêtés)) 'are systematically arrested'

The noun phrases (NPs) in the FTB are also flat; a noun is grouped together with any associated determiners and prenominal adjectives, as in example (2). Note that postnominal adjectives, however, are adjoined to the NP in an adjectival phrase (AP).

(2) (NP (D des) (A petits) (N mots) (AP (ADV très) (A gentils))) 'small, very gentle words'

Unlike the PTB, the FTB annotates *coordinated phrases* with the syntactic tag COORD (see the left panel of Figure 3 for an example).

The treatment of *compounds* is also different in the FTB. Compounds in French can comprise words which do not exist otherwise (e.g., *insu* in the compound preposition *à l'insu de* 'unbeknownst to') or can exhibit sequences of tags otherwise ungrammatical (e.g., *à la va vite* 'in a hurry': Prep + Det + finite verb + adverb). To account for these properties, compounds receive a two-level annotation in the FTB: a subordinate level is added for the constituent parts of the compound (both levels use the same POS tagset). An example is given in Figure 2.

Finally, the FTB differs from the PTB in that it does not use any empty categories.

2.2 Data Sets

The version of the FTB made available to us (version 1.4, May 2004) contains numerous errors. Two main classes of inaccuracies were found in the data: (a) The word is present but morphosyntactic tags are missing; 101 such cases exist. (b) The tag information for a word (or a part of a compound) is present but the word (or compound part) itself is missing. There were 16,490 instances of this error in the dataset.

Initially we attempted to correct the errors, but this proved too time consuming, and we often found that the errors cannot be corrected without access to the raw corpus, which we did not have. We therefore decided to remove all sentences with errors, which lead to a reduced dataset of 10,552 sentences.

The remaining data set (222,569 words at an average sentence length of 21.1 words) was split into a training set, a development set (used to test the parsing models and to tune their parameters), and a test set, unseen during development. The training set consisted of the first 8,552 sentences in the corpus, with the following 1000 sentences serving as the development set and the final 1000 sentences forming the test set. All results reported in this paper were obtained on the test set, unless stated otherwise.

[1] The French Treebank was developed at Université Paris 7. A license can be obtained by emailing Anne Abeillé (`abeille@linguist.jussieu.fr`).

3 Tree Transformations

We created a number of different datasets from the FTB, applying various tree transformation to deal with the peculiarities of the FTB annotation scheme. As a first step, the XML formatted FTB data was converted to PTB-style bracketed expressions. Only the POS tag was kept and the rest of the morphological information for each terminal was discarded. For example, the NP in Figure 1 was transformed to:

(3) (NP (PRO eux))

In order to make our results comparable to results from the literature, we also transformed the annotation of punctuation. In the FTB, all punctuations is tagged uniformly as PONCT. We reassigned the POS for punctuation using the PTB tagset, which differentiates between commas, periods, brackets, etc.

Compounds have internal structure in the FTB (see Section 2.1). We created two separate data sets by applying two alternative tree transformation to make FTB compounds more similar to compounds in other annotation schemes. The first was *collapsing the compound* by concatenating the compound parts using an underscore and picking up the `cat` information supplied at the compound level. For example, the compound in Figure 2 results in:

(4) (P d'_entre)

This approach is similar to the treatment of compounds in the German Negra treebank (used by Dubey and Keller 2003), where compounds are not given any internal structure (compounds are mostly spelled without spaces or apostrophes in German).

The second approach is *expanding the compound*. Here, the compound parts are treated as individual words with their own POS (from the `catint` tag), and the suffix `Cmp` is appended the POS of the compound, effectively expanding the tagset.[2] Now Figure 2 yields:

(5) (PCmp (P d') (P entre)).

This approach is similar to the treatment of compounds in the PTB (except hat the PTB does not use a separate tag for the mother category). We found that in the FTB the POS tag of the compound part is sometimes missing (i.e., the value of `catint` is blank). In cases like this, the missing `catint` was substituted with the `cat` tag of the compound. This heuristic produces the correct POS for the subparts of the compound most of the time.

[2] An alternative would be to retain the `cat` tag of the compound. The effect of this decision needs to be investigated in future work.

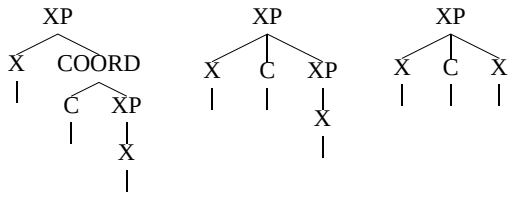

Figure 3: Coordination in the FTB: before (left) and after transformation (middle); coordination in the PTB (right)

As mentioned previously, coordinate structures have their own constituent label COORD in the FTB annotation. Existing parsing models (e.g., the Collins models) have coordination-specific rules, presupposing that coordination is marked up in PTB format. We therefore created additional datasets where a transformation is applied that *raises coordination*. This is illustrated in Figure 3. Note that in the FTB annotation scheme, a coordinating conjunction is always followed by a syntactic category. Hence the resulting tree, though flatter, is still not fully compatible with the PTB treatment of coordination.

4 Probabilistic Parsing Models

4.1 Probabilistic Context-Free Grammars

The aim of this paper is to further explore the crosslinguistic role of lexicalization by applying lexicalized parsing models to the French Treebank parsing accuracy. Following Dubey and Keller (2003), we use a standard unlexicalized PCFG as our baseline. In such a model, each context-free rule $RHS \rightarrow LHS$ is annotated with an expansion probability $P(RHS|LHS)$. The probabilities for all the rules with the same left-hand side have to sum up to one and the probability of a parse tree T is defined as the product of the probabilities of each rule applied in the generation of T.

4.2 Collins' Head-Lexicalized Models

A number of lexicalized models can then be applied to the FTB, comparing their performance to the unlexicalized baseline. We start with Collins' Model 1, which lexicalizes a PCFG by associating a word w and a POS tag t with each non-terminal X in the tree. Thus, a non-terminal is written as $X(x)$ where $x = \langle w,t \rangle$ and X is constituent label. Each rule now has the form:

(1) $P(h) \rightarrow L_n(l_n) \ldots L_1(l_1) H(h) R_1(r_1) \ldots R_m(r_m)$

Here, H is the head-daughter of the phrase, which inherits the head-word h from its parent P. $L_1 \ldots L_n$ and $R_1 \ldots R_n$ are the left and right sisters of H. Either n or m may be zero, and $n = m$ for unary rules.

The addition of lexical heads leads to an enormous number of potential rules, making direct estimation of $P(RHS|LHS)$ infeasible because of sparse data. Therefore, the generation of the RHS of a rule given the LHS is decomposed into three steps: first the head is generated, then the left and right sisters are generated by independent 0th-order Markov processes. The probability of a rule is thus defined as:

(2) $P(RHS|LHS) =$
$P(L_n(l_n)\ldots L_1(l_1)H(h),R_1(r_1)\ldots R_m(r_m)|P(h))$
$= P_h(H|P,h) \times \prod_{i=1}^{m+1} P_r(R_i(r_i)|P,h,H,d(i))$
$\times \prod_{i=1}^{n+1} P_l(L_i(l_i)|P,h,H,d(i))$

Here, P_h is the probability of generating the head, P_l and P_r are the probabilities of generating the left and right sister respectively. $L_{m+1}(l_{m+1})$ and $R_{m+1}(r_{m+1})$ are defined as stop categories which indicate when to stop generating sisters. $d(i)$ is a distance measure, a function of the length of the surface string between the head and the previously generated sister.

Collins' Model 2 further refines the initial model by incorporating the complement/adjunct distinction and subcategorization frames. The generative process is enhanced to include a probabilistic choice of left and right subcategorization frames. The probability of a rule is now:

(3) $P_h(H|P,h) \times P_{lc}(LC|P,H,h) \times P_{rc}(RC|P,H,h)$
$\times \prod_{i=1}^{m+1} P_r(R_i(r_i)|P,h,H,d(i),RC)$
$\times \prod_{i=1}^{n+1} P_l(L_i(l_i)|P,h,H,d(i),LC)$

Here, *LC* and *RC* are left and right subcat frames, multisets specifying the complements that the head requires in its left or right sister. The subcat requirements are added to the conditioning context. As complements are generated, they are removed from the appropriate subcat multiset.

5 Experiment 1: Unlexicalized Model

5.1 Method

This experiment was designed to compare the performance of the unlexicalized baseline model on four different datasets, created by the tree transformations described in Section 3: compounds expanded (Exp), compounds contracted (Cont), compounds expanded with coordination raised (Exp+CR), and compounds contracted with coordination raised (Cont+CR).

We used BitPar (Schmid, 2004) for our unlexicalized experiments. BitPar is a parser based on a bit-vector implementation of the CKY algorithm. A grammar and lexicon were read off our training set, along with rule frequencies and frequencies for lexical items, based on which BitPar computes the rule probabilities using maximum likelihood estimation. A frequency distribution for POS tags was also read off the training set; this distribution is used by BitPar to tag unknown words in the test data.

Model	LR	LP	CBs	0CB	≤2CB	Tag	Cov
Exp	59.97	58.64	1.74	39.05	73.23	91.00	99.20
Exp+CR	60.75	60.57	1.57	40.77	75.03	91.08	99.09
Cont	64.19	64.61	1.50	46.74	76.80	93.30	98.48
Cont+CR	**66.11**	**65.55**	1.39	46.99	78.95	93.22	97.94

Table 1: Results for unlexicalized models (sentences ≤40 words); each model performed its own POS tagging.

All models were evaluated using standard Parseval measures of labeled recall (LR), labeled precision (LP), average crossing brackets (CBs), zero crossling brackets (0CB), and two or less crossing brackets (≤2CB). We also report tagging accuracy (Tag), and coverage (Cov).

5.2 Results

The results for the unlexicalized model are shown in Table 1 for sentences of length ≤40 words. We find that contracting compounds increases parsing performance substantially compared to expanding compounds, raising labeled recall from around 60% to around 64% and labeled precision from around 59% to around 65%. The results show that raising coordination is also beneficial; it increases precision and recall by 1–2%, both for expanded and for non-expanded compounds.

Note that these results were obtained by uniformly applying coordination raising during evaluation, so as to make all models comparable. For the Exp and Cont models, the parsed output and the gold standard files were first converted by raising coordination and then the evaluation was performed.

5.3 Discussion

The disappointing performance obtained for the expanded compound models can be partly attributed to the increase in the number of grammar rules (11,704 expanded vs. 10,299 contracted) and POS tags (24 expanded vs. 11 contracted) associated with that transformation.

However, a more important point observation is that the two compound models do not yield comparable results, since an expanded compound has more brackets than a contracted one. We attempted to address this problem by collapsing the compounds for evaluation purposes (as described in Section 3). For example, (5) would be contracted to (4). However, this approach only works if we are certain that the model is tagging the right words as compounds. Un-

fortunately, this is rarely the case. For example, the model outputs:

(6) (NCmp (N jours) (N commerçants))

But in the gold standard file, *jours* and *commerçants* are two distinct NPs. Collapsing the compounds therefore leads to length mismatches in the test data. This problem occurs frequently in the test set, so that such an evaluation becomes pointless.

6 Experiment 2: Lexicalized Models

6.1 Method

Parsing We now compare a series of lexicalized parsing models against the unlexicalized baseline established in the previous experiment. Our is was to test if French behaves like English in that lexicalization improves parsing performance, or like German, in that lexicalization has only a small effect on parsing performance.

The lexicalized parsing experiments were run using Dan Bikel's probabilistic parsing engine (Bikel, 2002) which in addition to replicating the models described by Collins (1997) also provides a convenient interface to develop corresponding parsing models for other languages.

Lexicalization requires that each rule in a grammar has one of the categories on its right hand side annotated as the head. These head rules were constructed based on the FTB annotation guidelines (provided along with the dataset), as well as by using heuristics, and were optimized on the development set. Collins' Model 2 incorporates a complement/adjunct distinction and probabilities over subcategorization frames. Complements were marked in the training phase based on argument identification rules, tuned on the development set.

Part of speech tags are generated along with the words in the models; parsing and tagging are fully integrated. To achieve this, Bikel's parser requires a mapping of lexical items to orthographic/morphological word feature vectors. The features implemented (capitalization, hyphenation, inflection, derivation, and compound) were again optimized on the development set.

Like BitPar, Bikel's parser implements a probabilistic version of the CKY algorithm. As with normal CKY, even though the model is defined in a top-down, generative manner, decoding proceeds bottom-up. To speed up decoding, the algorithm implements beam search. Collins uses a beam width of 10^4, while we found that a width of 10^5 gave us the best coverage vs. parsing speed trade-off.

Label	FTB	PTB	Negra	Label	FTB	PTB	Negra
SENT	5.84	2.22	4.55	VPpart	2.51	–	–
Ssub	4.41	–	–	VN	1.76	–	–
Sint	3.44	–	–	PP	2.10	2.03	3.08
Srel	3.92	–	–	NP	2.45	2.20	3.08
VP	–	2.32	2.59	AdvP	2.24	–	2.08
VPinf	3.07	–	–	AP	1.34	–	2.22

Table 2: Average number of daughter nodes per constituents in three treebanks

Flatness As already pointed out in Section 2.1, the FTB uses a flat annotation scheme. This can be quantified by computing the average number of daughters for each syntactic category in the FTB, and comparing them with the figures available for PTB and Negra (Dubey and Keller, 2003). This is done in Table 2. The absence of sentence-internal VPs explains the very high level of flatness for the sentential category SENT (5.84 daughters), compared to the PTB (2.44), and even to Negra, which is also very flat (4.55 daughters). The other sentential categories Ssub (subordinate clauses), Srel (relative clause), and Sint (interrogative clause) are also very flat. Note that the FTB uses VP nodes only for non-finite subordinate clauses: VPinf (infinitival clause) and VPpart (participle clause); these categories are roughly comparable in flatness to the VP category in the PTB and Negra. For NP, PPs, APs, and AdvPs the FTB is roughly as flat as the PTB, and somewhat less flat than Negra.

Sister-Head Model To cope with the flatness of the FTB, we implemented three additional parsing models. First, we implemented Dubey and Keller's (2003) sister-head model, which extends Collins' base NP model to all syntactic categories. This means that the probability function P_r in equation (2) is no longer conditioned on the head but instead on its previous sister, yielding the following definition for P_r (and by analogy P_l):

(4) $P_r(R_i(r_i)|P,R_{i-1}(r_{i-1}),d(i))$

Dubey and Keller (2003) argue that this implicitly adds binary branching to the grammar, and therefore provides a way of dealing with flat annotation (in Negra and in the FTB, see Table 2).

Bigram Model This model, inspired by the approach of Collins et al. (1999) for parsing the Prague Dependency Treebank, builds on Collins' Model 2 by implementing a 1st order Markov assumption for the generation of sister non-terminals. The latter are now conditioned, not only on their head, but also on the previous sister. The probability function for P_r (and by analogy P_l) is now:

(5) $P_r(R_i(r_i)|P,h,H,d(i),R_{i-1},RC)$

Model	LR	LP	CBs	0CB	≤2CB	Tag	Cov
Model 1	80.35	79.99	0.78	65.22	89.46	96.86	99.68
Model 2	80.49	79.98	0.77	64.85	90.10	96.83	99.68
SisterHead	80.47	80.56	0.78	64.96	89.34	96.85	99.57
Bigram	**81.15**	**80.84**	0.74	65.21	90.51	96.82	99.46
BigramFlat	80.30	80.05	0.77	64.78	89.13	96.71	99.57

Table 3: Results for lexicalized models (sentences ≤40 words); each model performed its own POS tagging; all lexicalized models used the Cont+CR data set

The intuition behind this approach is that the model will learn that the stop symbol is more likely to follow phrases with many sisters. Finally, we also experimented with a third model (BigramFlat) that applies the bigram model only for categories with high degrees of flatness (SENT, Srel, Ssub, Sint, VPinf, and VPpart).

6.2 Results

Constituency Evaluation The lexicalized models were tested on the Cont+CR data set, i.e., compounds were contracted and coordination was raised (this is the configuration that gave the best performance in Experiment 1).

Table 3 shows that all lexicalized models achieve a performance of around 80% recall and precision, i.e., they outperform the best unlexicalized model by at least 14% (see Table 1). This is consistent with what has been reported for English on the PTB.

Collins' Model 2, which adds the complement/adjunct distinction and subcategorization frames achieved only a very small improvement over Collins' Model 1, which was not statistically significant using a χ^2 test. It might well be that the annotation scheme of the FTB does not lend itself particularly well to the demands of Model 2. Moreover, as Collins (1997) mentions, some of the benefits of Model 2 are already captured by inclusion of the distance measure.

A further small improvement was achieved using Dubey and Keller's (2003) sister-head model; however, again the difference did not reach statistical significance. The bigram model, however, yielded a statistically significant improvement over Collins' Model 1 (recall $\chi^2 = 3.91$, $df = 1$, $p \leq .048$; precision $\chi^2 = 3.97$, $df = 1$, $p \leq .046$). This is consistent with the findings of Collins et al. (1999) for Czech, where the bigram model upped dependency accuracy by about 0.9%, as well as for English where Charniak (2000) reports an increase in F-score of approximately 0.3%. The BigramFlat model, which applies the bigram model to only those labels which have a high degree of flatness, performs at roughly the same level as Model 1.

Model	LR	LP	CBs	0CB	≤2CB	Tag	Cov
Exp+CR	65.50	64.76	1.49	42.36	77.48	100.0	97.83
Cont+CR	69.35	67.93	1.34	47.43	80.25	100.0	96.97
Model1	81.51	81.43	0.78	64.60	89.25	98.54	99.78
Model2	81.69	81.59	0.78	63.84	89.69	98.55	99.78
SisterHead	81.08	81.56	0.79	64.35	89.57	98.51	99.57
Bigram	**81.78**	**81.91**	0.78	64.96	89.12	98.81	99.67
BigramFlat	81.14	81.19	0.81	63.37	88.80	98.80	99.67

Table 4: Results for lexicalized and unlexicalized models (sentences ≤40 words) with correct POS tags supplied; all lexicalized models used the Cont+CR data set

The models in Tables 1 and 3 implemented their own POS tagging. Tagging accuracy was 91–93% for BitPar (unlexicalized models) and around 96% for the word-feature enhanced tagging model of the Bikel parser (lexicalized models). POS tags are an important cue for parsing. To gain an upper bound on the performance of the parsing models, we reran the experiments by providing the correct POS tag for the words in the test set. While BitPar always uses the tags provided, the Bikel parser only uses them for words whose frequency is less than the unknown word threshold. As Table 4 shows, perfect tagging increased parsing performance in the lexicalized models by around 3%. This shows that the poor POS tagging performed by BitPar is one of the reasons of the poor performance of the lexicalized models. The impact of perfect tagging is less drastic on the lexicalized models (around 1% increase). However, our main finding, viz., that lexicalized models outperform unlexicalized models considerable on the FTB, remains valid, even with perfect tagging.[3]

Dependency Evaluation We also evaluated our models using dependency measures, which have been argued to be more annotation-neutral than Parseval. Lin (1995) notes that labeled bracketing scores are more susceptible to cascading errors, where one incorrect attachment decision causes the scoring algorithm to count more than one error.

The gold standard and parsed trees were converted into dependency trees using the algorithm described by Lin (1995). Dependency accuracy is defined as the ratio of correct dependencies over the total number of dependencies in a sentence. (Note that this is an unlabeled dependency measure.) Dependency accuracy and constituency F-score are shown

[3]It is important to note that the Collins model has a range of other features that set it apart from a standard unlexicalized PCFG (notably Markovization), as discussed in Section 4.2. It is therefore likely that the gain in performance is not attributable to lexicalization alone.

Model	Dependency	F-score
Cont+CR	73.09	65.83
Model 2	83.96	80.23
SisterHead	84.00	80.51
Bigram	**84.20**	**80.99**

Table 5: Dependency vs. constituency scores for lexicalized and unlexicalized models

in Table 5 for the most relevant FTB models. (F-score is computed as the geometric mean of labeled recall and precision.)

Numerically, dependency accuracies are higher than constituency F-scores across the board. However, the effect of lexicalization is the same on both measures: for the FTB, a gain of 11% in dependency accuracy is observed for the lexicalized model.

7 Experiment 3: Crosslinguistic Comparison

The results reported in Experiments 1 and 2 shed some light on the role of lexicalization for parsing French, but they are not strictly comparable to the results that have been reported for other languages. This is because the treebanks available for different languages typically vary considerably in size: our FTB training set was about 8,500 sentences large, while the standard training set for the PTB is about 40,000 sentences in size, and the Negra training set used by Dubey and Keller (2003) comprises about 18,600 sentences. This means that the differences in the effect of lexicalization that we observe could be simply due to the size of the training set: lexicalized models are more susceptible to data sparseness than unlexicalized ones.

We therefore conducted another experiment in which we applied Collins' Model 2 to subsets of the PTB that were comparable in size to our FTB data sets. We combined sections 02–05 and 08 of the PTB (8,345 sentences in total) to form the training set, and the first 1,000 sentences of section 23 to form our test set. As a baseline model, we also run an unlexicalized PCFG on the same data sets. For comparison with Negra, we also include the results of Dubey and Keller (2003): they report the performance of Collins' Model 1 on a data set of 9,301 sentences and a test set of 1,000 sentences, which are comparable in size to our FTB data sets.[4]

The results of the crosslinguistic comparison are shown in Table 6.[5] We conclude that the effect of lexicalization is stable even if the size of the training set is held constant across languages: For the FTB we find that lexicalization increases F-score by around 13%. Also for the PTB, we find an effect of lexicalization of about 14%. For the German Negra treebank, however, the performance of the lexicalized and the unlexicalized model are almost indistinguishable. (This is true for Collins' Model 1; note that Dubey and Keller (2003) do report a small improvement for the lexicalized sister-head model.)

Corpus	Model	LR	LP	CBs	0CB	$\leq$2CB
FTB	Cont+CR	66.11	65.55	1.39	46.99	78.95
	Model 2	79.20	78.58	0.83	63.33	89.23
PTB	Unlex	72.79	75.23	2.54	31.56	58.98
	Model 2	86.43	86.79	1.17	57.80	82.44
Negra	Unlex	69.64	67.27	1.12	54.21	82.84
	Model 1	68.33	67.32	0.83	60.43	88.78

Table 6: The effect of lexicalization on different corpora for training sets of comparable size (sentences $\leq$40 words)

8 Related Work

We are not aware of any previous attempts to build a probabilistic, treebank-trained parser for French. However, there is work on chunking for French. The group who built the French Treebank (Abeillé et al., 2000) used a rule-based chunker to automatically annotate the corpus with syntactic structures, which were then manually corrected. They report an unlabeled recall/precision of 94.3/94.2% for opening brackets and 92.2/91.4% for closing brackets, and a label accuracy of 95.6%. This result is not comparable to our results for full parsing.

Giguet and Vergne (1997) present use a memory-based learner to predict chunks and dependencies between chunks. The system is evaluated on texts from *Le Monde* (different from the FTB texts). Results are only reported for verb-object dependencies, for which recall/precision is 94.04/96.39%. Again, these results are not comparable to ours, which were obtained using a different corpus, a different dependency scheme, and for a full set of dependencies.

9 Conclusions

In this paper, we provided the first probabilistic, treebank-trained parser for French. In Experiment 1, we established an unlexicalized baseline model, which yielded a labeled precision and recall of about 66%. We experimented with a number of tree transformation that take account of the peculiarities of the annotation of the French Tree-

[4] Dubey and Keller (2003) report only F-scores for the reduced data set (see their Figure 1); the other scores were provided by Amit Dubey. No results for Model 2 are available.

[5] For this experiments, the same POS tagging model was applied to the PTB and the FTB data, which is why the FTB figures are slightly lower than in Table 3.

bank; the best performance was obtained by raising coordination and contracting compounds (which have internal structure in the FTB). In Experiment 2, we explored a range of lexicalized parsing models, and found that lexicalization improved parsing performance by up to 15%: Collins' Models 1 and 2 performed at around 80% LR and LP. No significant improvement could be achieved by switching to Dubey and Keller's (2003) sister-head model, which has been claimed to be particularly suitable for treebanks with flat annotation, such as the FTB. A small but significant improvement (to 81% LR and LP) was obtained by a bigram model that combines features of the sister-head model and Collins' model.

These results have important implications for crosslinguistic parsing research, as they allow us to tease apart language-specific and annotation-specific effects. Previous work for English (e.g., Magerman, 1995; Collins, 1997) has shown that lexicalization leads to a sizable improvement in parsing performance. English is a language with non-flexible word order and with a treebank with a non-flat annotation scheme (see Table 2). Research on German (Dubey and Keller, 2003) showed that lexicalization leads to no sizable improvement in parsing performance for this language. German has a flexible word order and a flat treebank annotation, both of which could be responsible for this counter-intuitive effect. The results for French presented in this paper provide the missing piece of evidence: they show that French behaves like English in that it shows a large effect of lexicalization. Like English, French is a language with non-flexible word order, but like the German Treebank, the French Treebank has a flat annotation. We conclude that Dubey and Keller's (2003) results for German can be attributed to a language-specific factor (viz., flexible word order) rather than to an annotation-specific factor (viz., flat annotation). We confirmed this claim in Experiment 3 by showing that the effects of lexicalization observed for English, French, and German are preserved if the size of the training set is kept constant across languages.

An interesting prediction follows from the claim that word order flexibility, rather than flatness of annotation, is crucial for lexicalization. A language which has a flexible word order (like German), but a non-flat treebank (like English) should show no effect of lexicalization, i.e., lexicalized models are predicted not to outperform unlexicalized ones. In future work, we plan to test this prediction for Korean, a flexible word order language whose treebank (Penn Korean Treebank) has a non-flat annotation.

References

Abeillé, Anne, Lionel Clement, and Alexandra Kinyon. 2000. Building a treebank for French. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. Athens.

Bikel, Daniel M. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of the 2nd International Conference on Human Language Technology Research*. Morgan Kaufmann, San Francisco.

Bikel, Daniel M. 2004. A distributional analysis of a lexicalized statistical parsing model. In Dekang Lin and Dekai Wu, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, pages 182–189.

Bikel, Daniel M. and David Chiang. 2000. Two statistical parsing models applied to the Chinese treebank. In *Proceedings of the 2nd ACL Workshop on Chinese Language Processing*. Hong Kong.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*. Seattle, WA, pages 132–139.

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, pages 16–23.

Collins, Michael, Jan Hajič, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. University of Maryland, College Park.

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

Giguet, Emmanuel and Jacques Vergne. 1997. From part-of-speech tagging to memory-based deep syntactic analysis. In *Proceedings of the International Workshop on Parsing Technologies*. Boston, pages 77–88.

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

Levy, Roger and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo.

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

Magerman, David. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, MA, pages 276–283.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

Schiehlen, Michael. 2004. Annotation strategies for probabilistic parsing in German. In *Proceedings of the 20th International Conference on Computational Linguistics*. Geneva.

Schmid, Helmut. 2004. Effi cient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics*. Geneva.

Skut, Wojciech, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Washington, DC.