

Statistical Parsing of the French Treebank

Abhishek Arun

School of Informatics

University of Edinburgh

Master of Science

Cognitive Science and Natural Language

School of Informatics

University of Edinburgh

September 2004

Abstract

This thesis presents the first ever results of applying probabilistic models to the task of parsing French using the newly available French Le Monde corpus.

We start off by describing the annotation scheme of this brand new corpus, and highlight its differences compared to the Penn Treebank. We then use an unlexicalised PCFG to give us a baseline parsing model. This model is steadily enriched to the level of Collins' Model 2 by adding lexicalisation and subcategorisation information. Word-level orthographical and morphological information is incorporated as feature vectors to help reduce Part of Speech tag assignment ambiguity for unknown words.

This enriched model significantly outperforms the baseline model, achieving labeled precision and recall of up to 80% on sentences with ≤ 40 words, an improvement of almost 15% over the baseline.

We also examine an alternative "bigram" model, where modifying nonterminals are also conditioned on the previously generated nonterminal, in an attempt to take into account the flatness of the French Treebank. This model achieves precision and recall of up to 81 %.

Acknowledgements

Thanks to: Frank Keller, for his advice and support throughout the project; Bjorn Nelson, for much needed help with XSLT; Dave Lambert, for converting me to Linux, but failing with Emacs; Dan Bikel, for allowing me to use his parser and finding time to answer my questions; to Helmut Schmid, for BitPar and VPF; and to my parents and brother for constant encouragement.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

*(Abhishek Arun
School of Informatics
University of Edinburgh)*

Table of Contents

1	Introduction	1
1.1	Probabilistic Parsing	1
1.2	The French Treebank	1
1.3	Aims	2
1.4	Experimentation	3
1.5	Results	4
2	Probabilistic Parsing	7
2.1	Parsing	7
2.2	Probabilistic Parsing	8
2.3	Previous work	9
2.3.1	Work on the Penn Treebank	10
2.3.2	Other languages	12
3	Corpus	16
3.1	Tagging	17
3.2	Parsing	18

3.2.1	Verbal nucleus	21
3.2.2	Coordinated phrase	21
3.3	Format	22
3.4	Corpus discussion	23
3.4.1	Tagging	23
3.4.2	Compounds	26
3.4.3	Parsing	28
3.5	Comparison between English and French	30
4	Experiments	33
4.1	Formatting	33
4.1.1	XML to bracketed expressions	33
4.1.2	Compounds	34
4.1.3	Punctuation	35
4.1.4	Raising coordination	35
4.2	Data Sets	37
4.3	Experiments with unlexicalised PCFG	38
4.3.1	PCFG	38
4.3.2	BitPar	41
4.4	Problems with PCFG	42
4.5	Collins' parser	45
4.5.1	Collins' Model 1	45
4.5.2	Model 2	51

4.5.3	Intricacies of Collins' Parsing Models	57
4.5.4	Smoothing	62
4.5.5	Part-of-speech tagging	63
4.5.6	Parsing	64
4.5.7	Varying the order of the Markov assumption	65
4.5.8	Modification to subcategorization frames	67
5	Results	69
5.1	PARSEVAL measures	69
5.2	Results - I	71
5.3	Results - II	73
5.4	Data sparsity	73
5.5	Comparison with PTB	75
5.6	Error analysis	76
5.6.1	Unlexicalised vs Lexicalised	76
5.6.2	Model 2 vs Bigram model	76
6	Conclusion	89
6.1	Summary	89
6.2	Future work	90
A	Head rules	93
B	Argument identification rules	95
C	Word features	96

D Parameter classes	99
Bibliography	102

List of Figures

2.1	A parse tree	8
3.1	Example of a FTB tree	29
3.2	Example of a coordinated FTB tree	29
3.3	Coordination - French Treebank annotation style	30
3.4	Coordination - Penn Treebank annotation style	30
4.1	Coordination before transformation	37
4.2	Coordination after transformation	37
4.3	PTB coordination annotation	37
4.4	A lexicalised tree	44
4.5	Collins' Model 1 example	47
4.6	Distance measure - Correct parse tree	49
4.7	Distance Measure - Incorrect parse tree	49
4.8	Verb-intervening PTB - Correct parse	51
4.9	Verb-intervening PTB - Incorrect parse	52
4.10	Verb-intervening FTB - Correct parse	53
4.11	Verb-intervening FTB - Incorrect parse	54

4.12	Tree with complement/adjunct distinction	54
4.13	Distance vs Subcategorization	55
4.14	A base NP	58
4.15	Base NP with a post-nominal adjective	59
4.16	A raised tree	60
4.17	Conjunction example	62
5.1	Learning curve	74
5.2	Wrong PP-attachment analysis in BitPar	77
5.3	Correct PP-attachment analysis in Model 1 emulation	77
5.4	Wrong parse analysis in Model 2 emulation	78
5.5	Correct parse analysis in bigram model	79
5.6	Model 2 not sensitive to flat structure - 1	79
5.7	Bigram model sensitive to flat structure - 1	80
5.8	Model 2 not sensitive to flat structure - 2	80
5.9	Bigram model sensitive to flat structure - 2	80
5.10	Model 2 sensitive to flat structure	81
5.11	Bigram model not sensitive to flat structure	81
5.12	Correct VPinf attachment in Model 2 -1	82
5.13	Wrong VPinf attachment in bigram model -1	82
5.14	Correct VPinf attachment in Model 2 -2	83
5.15	Wrong VPinf attachment in bigram model -2	83

List of Tables

2.1	PTB Parsing results for sentences \leq to 40 words	12
2.2	Negra parsing results	15
3.1	French Treebank POS tags	19
3.2	French Treebank syntactic tagset	20
3.3	Invalid POS tags	24
4.1	New FTB punctuation tagset	36
4.2	Average number of child nodes per Treebank per Constituent	65
4.3	Parsing results for Negra - 2	66
5.1	Results for lexicalised and unlexicalised models for sentences \leq 100 words long. Each model performed its own POS tagging. CR refers to the raised coordination transformation. All Collins models were run on the contracted comp	85
5.2	Results for lexicalised and unlexicalised models for sentences \leq 40 words long. Each model performed its own POS tagging. CR refers to the raised coordination transformation.	86

5.3	Results for lexicalised and unlexicalised models for sentences ≤ 100 words long. The correct POS tags were supplied to the models. CR refers to the raised coordination transformation.	87
5.4	Results for lexicalised and unlexicalised models for sentences ≤ 40 words long. The correct POS tags were supplied to the models. CR refers to the raised coordination transformation.	88
5.5	Comparison between PTB and FTB for sentence length ≤ 100 words .	88
5.6	Comparison between PTB and FTB for sentence length ≤ 40 words .	88
A.1	Head rules table. <i>Parent</i> is the non-terminal on the left-hand side of a rule <i>Direction</i> specifies whether search starts from left or right end of the rule. <i>Priority</i> gives a priority ranking, with priority decreasing when moving down the list	94
D.1	Parameter class for head generation	99
D.2	Parameter class for partially lexicalised modifier	100
D.3	Parameter class for lexicalised modifier	100
D.4	Parameter class for Base NP generation	101
D.5	Parameter class for coordination and punctuation generation	101

Chapter 1

Introduction

1.1 Probabilistic Parsing

Given its ability to handle the extreme ambiguity produced by context-free natural language grammars, it is not surprising that a lot of research has been done in the field of treebank-based probabilistic parsing over the past few years, resulting in parsing models that achieve both broad coverage as well as high parsing accuracy (e.g., [Collins, 1997, Charniak, 1999]). However, most of these models have been developed for English and trained on the Penn Treebank (PTB) [Marcus et al., 1994] which raises the question whether these models generalise to other languages, and to annotation schemes that are different to the PTB [Dubey and Keller, 2003].

1.2 The French Treebank

For this thesis, we took advantage of the recently made available Le Monde corpus [A. Abeillé and F.Toussenel, 2003], to address this question for French. This corpus consists of 20648 sentence extracts and 580945 words from the daily newspaper *Le Monde*, ranging from 1989 to 1993, and covering a variety of authors and domains (economy, literature, politics etc.), representative of contemporary written French. This

morphologically and syntactically annotated French Treebank differs from the Penn Treebank in a number of ways, such as format (XML instead of Lisp style bracketing), tagset, syntactic categories and treatment of compounds. This thesis constitutes, to our knowledge, the first attempt of constructing a probabilistic parsing model for French. Moreover, as this thesis is almost certainly the first systematic use of the *Le Monde* corpus by an outside researcher, it is not surprising to note the presence of certain inconsistencies and inaccuracies in the annotation of the corpus, resulting in the discarding of approximately 50% of the sentences in the dataset. Chapter 3 presents in detail the particularities of the French Treebank.

1.3 Aims

A further aim of this project was to assess the impact of **lexicalisation** and **markovization** on parsing performance. While lexicalisation (where head words annotate phrasal nodes) has been shown to dramatically increase parsing accuracy in English [Collins, 1997, Charniak, 1997, Charniak, 1999]; in the case of the German Negra corpus, a lexicalised model following [Collins, 1997]’s Model 1 failed to outperform the unlexicalised baseline model [Dubey and Keller, 2003]. In the case of Chinese [Bikel and Chiang, 2000, Levy and Manning, 2003, Chiang and Bikel, 2002] and Czech [Collins et al., 1999], lexicalised parsing models have been successfully applied but since their performance was not compared to a baseline unlexicalised model, a doubt about the usefulness of lexicalisation in languages other than English remains.

A major drawback of treebank grammars is that many rule types will only be seen once (and therefore have their probabilities overestimated), and many rules which occur in test sentences will never have been seen in training (and therefore have their probabilities underestimated - see [Collins, 1999] for analysis). [Collins, 1997]’s models have shown markovization to be a successful way to combat data sparsity of rules that is inherent in treebank grammars and therefore, we follow his work in the course of this project.

1.4 Experimentation

To begin with, the XML-style formatted French Treebank (FTB) was converted to the Penn Treebank (PTB) bracketed standard. A peculiarity of the FTB is its use of compound words.

E.g:

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

Note: The format above is explained in further details in Chapter 3.

Compounds were treated in 2 different ways (see Section 4.1.2 for more details).

a) The compounds were contracted by concatenating their subwords using an '_':

```
(P d'_entre)
```

b) The compounds were expanded, with the subword retaining their tag and adding 'Cmp' to the tag of the compound, effectively expanding the tagset:

```
(PCmp (P d') (P entre))
```

Moreover, in order to be consistent with the PTB, punctuations (. , ! " etc) which are tagged as 'PONCT' in the FTB, were converted to their respective PTB POS tags.

Due to various inconsistencies in the dataset, described further in Chapter 3, around 50% of the sentences had to be discarded, leaving 10552 sentences in total (222 569 words). The first 8552 sentences constituted the training set leaving 1000 sentences each for the development and test sets. The test set was used only once all parameters had been tuned using the development set.

1.5 Results

For the unlexicalised PCFG model (henceforth **baseline model**), we used BitPar [Schmid, 2004], a parser based on a bit-vector implementation of the well-known CKY algorithm [Kasami, 1965]. A grammar and lexicon were read off the FTB. In BitPar, a POS tag distribution for unknown words has to be specified, which is then used to tag unknown words in the test data. The contracted compound model achieves 66.11% Labelled Recall (LR) and 65.55% Labelled Precision (LP) while the expanded compound model achieves 60.75% LR and 60.57% LP. The poor performance of the expanded compound model can be explained by the higher number of grammar rules that are induced by this model which in turn leads to data sparsity issues. The contracted compound model was retained for further experimentation.

The head-lexicalised model was implemented using Dan Bikel's Multi-lingual, parallel processing statistical parsing engine [Bikel, 2002] which is basically a Java implementation of [Collins, 1997]'s parser. Lexicalisation 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. French-specific word level orthographical and morphological information were incorporated to help reduce POS assignment ambiguity for unknown words.

This first model is essentially Model 1 of [Collins, 1997], and achieves 80.35% LR and 79.99% LP, a substantial improvement over the baseline, confirming that lexicalisation does, in fact, help in parsing French. This model was augmented to the level of Model 2 of [Collins, 1997], which involves complement/adjunct distinction (through use of heuristics) and probabilities over subcategorization frames. This model achieved 80.49% LR and 79.98% LP, an improvement that is not statistically significant.

An alternative model, referred to in [Dubey and Keller, 2003] as capturing **sister-head** relationships, was looked into. In this model, the modifying non-terminal is conditioned on the previously generated modifying non-terminal instead of the head non-terminal. This is a way of taking into account the flatness of tree structures in the

corpus. This model achieves a slightly lower LP (80.47%) but a higher LR (80.56%), again not a statistically significant improvement in performance.

A **bigram** model, which is essentially the Collins Model 2 but where the conditioning context associated with the generation of the modifying non-terminal is extended to include the previously generated modifier, was implemented next. This model performed at 81.15% LR and 80.84% LP (f-score of 80.99%), a significant improvement over the previous models.

Finally, we looked at a mechanism to present a cleaner division of subcategorization. This is done by differentiating sentences where the subject is expressed explicitly through a constituent, from sentences where the subject is encapsulated in the verbal nucleus construct that is prevalent in the French Treebank annotation scheme (see Chapter 4 for further explanation). This model achieved an f-score of 81.03%, a slight improvement of the previously described model.

These are quite promising results, even though they fall quite short of the results obtained using similar parsing models on the PTB. As a comparison, the Bikel parser for the Penn Treebank was trained on a similar sized training set and evaluated on a test set of 1000 sentences from the standard section 23 of the PTB, resulting in 86.43% LR and 86.79% LP. It is hoped that further research on the FTB will yield improvements over my results.

The remainder of this thesis is organised as follows:

The next chapter 2 will look at the theoretical aspect of Probabilistic parsing and its importance in Natural Language processing and will review work done till now, in this field, for both English as well as other languages.

Chapter 3 will address in detail the particularities of the *Le Monde* corpus annotation scheme as well as the major similarities and differences between English and French.

Chapter 4 will describe the models that were implemented and the experiments that were performed.

Chapter 5 will present the results obtained from the experiments. We will also analyse

these results and try to explain the failings of the models presented as well as discuss ways for improvement.

We conclude with a brief summary of our work and possible new directions worth looking at in the future.

Chapter 2

Probabilistic Parsing

2.1 Parsing

Parsing, or more accurately, syntactic parsing, is the task of recognising a sentence and assigning a syntactic structure to it.

The parse tree in Figure 2.1 (from [Collins, 1999]) will help us illustrate the practical motivation for parsing. A parse tree represents several layers of information. Words, annotated with their part-of-speech tags, are grouped together to form a hierarchy of phrases ; for example, noun phrases (**NP**) like "IBM" or "IBM, long-time rival of Microsoft" and verb phrases (**VP**) like "acquired Lotus on Wednesday".

The tree also represents grammatical relations between phrases or words. For example, given a rule (**S** → **NP VP**), the (**NP**) is the subject of the verb within the (**VP**) (in this case, "IBM, long-time rival of Microsoft" is the subject of "acquired"). Similarly, the rule (**VP** → **VBD NP PP**) indicates an object-verb relationship ("Lotus" is the object of "acquired").

The tree allows us to directly read predicate-argument relations off it; e.g. "IBM" being the subject is the acquirer whereas "Lotus" being the object is the acquiree. Thus, parsing is an important intermediate stage for semantic analysis, making it fundamental for Natural Language Processing (NLP) applications such as Information Retrieval and

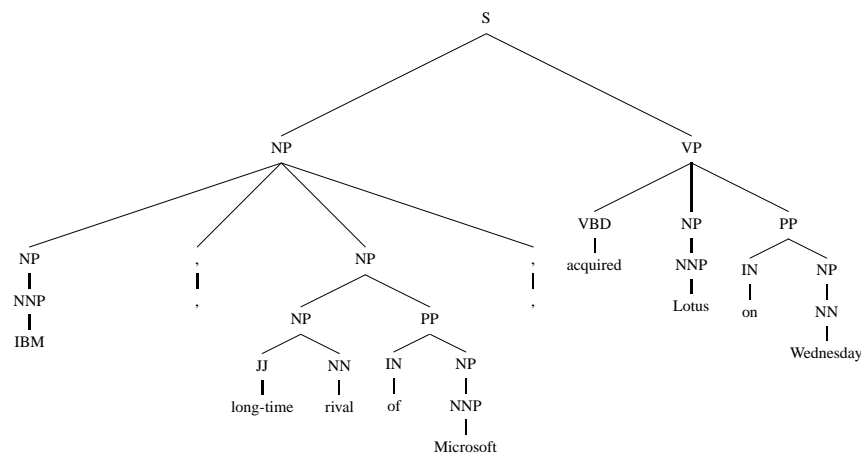


Figure 2.1: A parse tree

Question-Answering.

For example, an Information Retrieval query such as - "Find all companies bought by IBM" could be implemented by retrieving, from a database of parsed sentences, all sentences where "IBM" is the subject of verbs such as "buy", "acquire" or "purchase".

2.2 Probabilistic Parsing

However, due to ambiguity, a sentence can have multiple parses. Examples of ambiguity are :

- Part-of-Speech (POS) ambiguity. For example, the word *book* can be either a verb or a noun.
- Coordination ambiguity. In the following phrase (reproduced from [Collins, 1999]),

a program to promote safety in trucks and vans,

vans can be coordinated with *trucks*, *safety* and *program*.

While semantically, the only plausible interpretation is the coordination between *trucks* and *vans*, the other coordination relationships are also syntactically well-

formed. Therefore, a parser armed with only grammatic rules, will recover multiple analyses for this sentence, since it lacks the linguistic intuition and world knowledge possessed by humans to eliminate the semantically implausible analyses.

- Prepositional-Phrase (PP) attachment ambiguity. The sentence :

I saw a man with a telescope

has at least 2 syntactic trees : one where the PP *with a telescope* modifies *man* and the other where it modifies *saw*.

In contrast to the previous example, in this sentence, both analyses are semantically plausible.

The extreme ambiguity involved in broad coverage parsing leads to an exponentially growing number of parse analyses. Probabilistic parsing offers a way to disambiguate between multiple analyses of a given sentence : choose the most probable parse. This is also referred to in parsing literature as **parse selection**. The parse selection task is therefore to build a parser that selects the most likely parse from all possible parses of a sentence.

2.3 Previous work

The aim of this thesis is to build a system that maximises parsing accuracy for French. We use a supervised learning approach for this task where supervised training involves the use of a treebank of sentence/parse tree pairs as training data - in our case, using the recently released *Le Monde* corpus. Till recently, parsed training corpora was only available for English (Penn Treebank [Marcus et al., 1994]) such that most research in this field was focused on that particular language. However, lately, more and more treebanks have been made available in other languages, driving research in these non-English languages. In this section, we will review a number of treebank-based supervised learning approaches to the parse selection task for English as well as other languages.

2.3.1 Work on the Penn Treebank

2.3.1.1 Unlexicalised PCFGs

In this section, we look at past work involving unlexicalised parsing of the Penn Treebank. Intricacies of parsing with an unlexicalised PCFG are presented in section 4.3.1.

While [Charniak, 1997] describes a lexicalised PCFG model that we will discuss later in section 2.3.1.2, he also gives results for a baseline unlexicalised PCFG model. On sentences of length ≤ 40 words, the baseline model scores 71.7% LR and 75.8% LP. This result is directly comparable to the other results on PTB parsing that we will report in this thesis, since the model was trained and tested on the same datasets as those other models and evaluation was performed using the standard PARSEVAL measures [Black et al., 1991].

[Charniak, 1996] also describes results for a PCFG trained and tested on the PTB. However, these results are less comparable since both the training and testing datasets differ from the standard datasets and evaluation is given on the recovery of unlabelled constituents. Interestingly, the paper includes a strategy to add bias towards right-branching structures prevalent in English which leads to a 2.3% and 1.7% improvement in recall and precision respectively.

2.3.1.2 Lexicalised PCFGs

[Charniak, 1997] presents a probability model for a lexicalised PCFG. The probability of a lexicalised rule is decomposed into the product of 2 terms :

1. A probability that predicts the non-lexicalised part of a rule r , conditioned on the parent non-terminal t , its head-word h and ,interestingly, the non-terminal above the parent, l .

$$\text{i.e. } P(r \mid h, t, l)$$

2. A probability of generating the lexical head, l_w of each modifier in a rule, conditioned on the label of the head L , its parent P , and the head word of the parent,

w .

i.e $P_L(lw | L, P, w)$

This model is refined by adding several features such as smoothing which is performed using deleted interpolation of progressively coarser but less weighted back-off contexts, as well as by conditioning on automatically derived word clusters.

This model achieves 86.7% LR and 86.6% LP on ≤ 40 words.

[Collins, 1997] describes three generative parsing models, each model a refinement on the previous one, and achieving improved performance. The general approach of the model is to start with a parent node and a head and then to model the successive generation of dependents on both the left and right hand side of the head. In the first model, modifiers are generated more or less independent of each other. Non-recursive NPs, also called base NPs, are generated with a probability model different than those for other constituents, to account for the flatness of their annotation in the Penn Treebank. In the next models, Collins introduces more linguistic knowledge in the models. For example, model 2 makes use of argument/adjunct distinction and probabilities over subcategorization frames. Model 3 incorporates wh-movement through the use of traces.

Our work in this thesis is inspired a lot by Model 1 and 2 of [Collins, 1997] and more details about the models are presented in Section 4.5.

[Charniak, 1999] describes a maximum-entropy inspired parser that performs much better than [Collins, 1997], which till then had the best results for the task. This model, like [Collins, 1997] is based on a probabilistic generative model, but uses a maximum-entropy inspired technique for conditioning and smoothing purposes. The advantage of maximum-entropy models is that the features used to encode the different conditional probability events do not have to be statistically independent of each other. Moreover, while in [Collins, 1997], the generation of modifier is a zeroth Markov grammar (the modifiers are generated independently of each other), in this model, the generation of

Model	LR	LP	CBs	0 CB	≤ 2 CBs
Unlexicalised PCFG - Charniak 97	71.7	75.8	2.03	39.5	68.1
Lexicalised PCFG - Charniak 97	87.5	87.4	1.00	62.1	86.1
Model 1 - Collins 97	87.9	88.2	0.95	65.8	86.3
Model 2 - Collins 97	88.5	88.7	0.92	66.7	87.1
Model 3 - Collins 97	88.6	88.7	0.90	67.1	87.4
Charniak 2000	90.1	90.1	0.74	70.1	89.6

Table 2.1: PTB Parsing results for sentences \leq to 40 words

a modifier is conditioned on its previous 2 sisters (second order Markov grammar ¹). Finally, this model also included conditioning on the **grandparent** node i.e the node that dominates the parent node.

Table 2.1 summarises the performance of the models described in this section.

2.3.2 Other languages

2.3.2.1 Czech

[Collins et al., 1999] present a statistical parser for Czech using the Prague Dependency Treebank (PDT) [Hajič, 1998]. Czech differs radically from English in at least two respects : it is a highly inflected language and it has a relatively free word order. The treebank is modeled after the PTB with one major difference : syntactic annotation is based on dependencies rather than phrase structures. Thus, instead of nonterminal symbols at the non-leaves of the tree, the PDT uses so-called *analytical functions* capturing the type of relation between a dependent and its governing node. The PDT also contains a traditional morpho-syntactic annotation (tags) at each word position, together with a lemma, uniquely identifying the underlying lexical unit. Given that Czech is a highly inflected language, the corpus was found to contain over 1000 tags.

¹ Charniak refers to it as a third-order Markov grammar since he also counts the head label. Most other works assume conditioning on the head label as given and do not include it in the Markov order count

The authors decided to build over the work of [Collins, 1997] for parsing purposes. They therefore had to first convert the dependency structures in the training data into lexicalised trees. The mapping from dependency structure to lexicalised tree is one-to-many: the lexicalised tree can be made as flat as possible or various forms of binary branching trees can be chosen. For the baseline model, [Collins et al., 1999] decided to use the simplest possible conversion scheme which was to have the flattest tree possible. Moreover, only the first letter of the POS set representing broad categories such as noun, verb etc was chosen to annotate the training trees.

Parsing accuracy was defined as the ratio of correct dependency links vs the total number of dependency links in a sentence.

The baseline model gave a result of 72.3% accuracy.

The authors then refined the model by successive modifications. The training tree transformation was changed to take into account relative clauses, coordination and punctuation. The probability model for the generation of modifiers was altered by the addition of the previously generated modifier to the conditioning context. This model was called a **bigram** model.

This new model performed at a 80.0% accuracy level; a 7.7% absolute improvement and a 27.8% relative improvement over the baseline model.

2.3.2.2 Chinese

[Bikel and Chiang, 2000] present the first-ever results of applying statistical parsing models to the Chinese Treebank [Ircs, 2002]. They compare two models, both inspired from Model 2 of [Collins, 1999]. [Bikel and Chiang, 2000] implemented Chinese Treebank specific head-finding and argument-adjunct distinction rules for both models. The first model is a bigram model, similar to that employed by [Collins et al., 1999], that achieves performance of 69.0% LR and 74.8% LP. The second model, adapted from [Chiang, 2000] is based on stochastic Tree-Adjoining Grammar (TAG). In this model, a parse tree is built up not out of lexicalised phrase-structure rules but by tree fragments (called elementary trees) which are lexicalised in the sense that each

fragment contains exactly one lexical item. This model achieved LR of 76.8% and LP of 77.8%.

In [Chiang and Bikel, 2002], the authors present a learning method that, given a model with initial hand written tree-augmentation rules, re-estimates the parameters of the model using the Inside-Outside algorithm. This algorithm was used to fine-tune the head-finding rules for the Chinese Treebank, leading to improved results of 78.79% LR and 81.06% LP.

2.3.2.3 German

[Dubey and Keller, 2003] make use of Negra [Skut et al., 1997], a syntactically annotated corpus, to present the first probabilistic, treebank-based parser for German. The latter differs from English in a number of ways; most importantly, it is a **semi-free wordorder** language such that a context-free grammar model has to contain more rules than for a fixed wordorder language like English. Moreover, the annotation scheme of Negra is much flatter than the PTB. For example, there is no **S** \rightarrow **NP VP** rule. Instead, the subject, the verb and its objects are all sisters of each other, dominated by an **S** node. There is also no **PP** \rightarrow **P NP** rule; the preposition and the noun it selects are sisters dominated by a **PP** node.

[Dubey and Keller, 2003] compare the results of a baseline unlexicalised PCFG model with two different head-lexicalised models.

The first head-lexicalised model PCFG is that of Carroll and Rooth (1998) where rule probabilities are defined as :

$$P(RHS | LHS) = P_{rule}(C_1 \dots C_n | P, l(P)) \times \prod_{i=1..n} P_{choice}(l(C_i) | C_i, P, l(P)) \quad (2.1)$$

where

- **P** is the mother category of the rule.
- $C_1 \dots C_n$ are daughters.

Model	LR	LP	CBs	0 CB	≤ 2 CBs
Baseline	70.56	66.69	1.03	58.21	84.46
Carroll and Rooth	68.04	60.07	1.31	52.08	79.54
Collins Model 1	67.91	66.07	0.73	65.67	89.52
Sister-head all	71.32	70.93	0.61	69.53	91.72

Table 2.2: Negra parsing results

- $l(C)$ the lexical head of the constituent C .
- $P_{rule}(C_1 \dots C_n \mid P, l(P))$, the probability that category P with lexical head $l(P)$ is expanded by the rule $P \rightarrow C_1 \dots C_n$
- $P_{choice}(l(C) \mid C, P, l(P))$ is the probability that the modifier category C has the lexical head $l(C)$ given that its mother is P with lexical head $l(P)$.

The authors re-implemented Model 1 of [Collins, 1999] for the second head-lexicalised parsing model. This implementation, however, treated base NPs similar to all other syntactic categories.

Surprisingly, none of the head-lexicalised models managed to outperform the unlexicalised baseline model. This is at odds with what has been found for English. The authors then hypothesise that the lexicalised models perform poorly because they cannot cope with the flatness of Negra trees. They implement an alternative model whereby they extend [Collins, 1999]’s base NP model to all syntactic constituents. They call this model as capturing **sister-head relationships**. This model outperforms the baseline model (2.2) leading to the authors to conclude that the sister-head model is successful because it provides a way of binarizing the flat rules in the Negra corpus.

They also observe that the success of the sister-head model for German is at odds with what is known about parsing for English where this model has been found useful only for base NPs.

Chapter 3

Corpus

Annotated reference corpora (such the Penn Treebank) have helped both the development of English NLP tools and English corpus linguistics. A linguistically (error free) annotated corpus allows one to build (or improve) sizable linguistic resources (such as lexicons or grammars) and also to evaluate (or improve) most computational analyzers. A syntactically annotated corpus (tree bank) provides, on top of the category for each word, some of the following informations: constituent boundaries (sentence, NP etc), grammatical function of words or constituents, dependencies between words or constituents, verb valence (and valence alternation). More recently, however, a wider variety of parsed corpora has become available in other languages, such as Negra [Skut et al., 1997] for German, the Penn Chinese Treebank [Ircs, 2002] and the Prague Dependency Treebank [Hajič, 1998] for Czech, enabling researchers to address similar NLP issues for these languages.

The French *Le Monde* corpus is the latest addition to the list of available treebanks. Work on this project started in 1997 [A. Abeillé and F.Toussenet, 2003] and the corpus was made available for research purposes in May 2004. The corpus provided to us consists of 20648 sentence extracts (totalling 580945 words) from the daily newspaper *Le Monde*, ranging from 1989 to 1993, and covering a variety of authors and domains (economy, literature, politics etc.), representative of contemporary written French. The corpus first went through a tagging phase followed by a parsing phase. Each of these

phases consisted of automatic annotation followed by human correction and validation.

3.1 Tagging

The complete morphosyntactic tagset of the corpus is defined to be as follows:

1. Part of Speech (POS), for example Determiner.
2. Subcategorization, for example possessive or cardinal. ¹
3. Inflection, for example masculine singular.
4. Lemma (canonical form)
5. Parts (with similar morphosyntactic tags) for compounds.

The reasons for which the designers of the corpus chose to annotate more than just parts of speech are various. Some parts of speech are too inclusive (e.g. conjunctions or nouns) and further distinctions (called "subcategories") were made (e.g. proper and common for nouns, subordinating or coordinating for conjunctions). Inflectional morphology was annotated since morphological endings are important for grouping constituents (based on agreement marks) and also because many forms in French are ambiguous with respect to mood, person, number or gender. For example, the determiner *des* can be either masculine or feminine, the verb form *change* can be either indicative or subjunctive, and either first or third person, or even second person imperative. Lemmas are annotated to further aid disambiguation: *suis* is an indicative verb form first person singular which can correspond to the lemma *être* (be) or to the lemma *suivre* (follow).

Compounds are annotated since they may comprise words which do not exist otherwise (e.g. *insu* in the compound preposition *à l'insu de* = unbeknownst to) or exhibit sequences of tags otherwise non-grammatical (e.g. *à la va vite* = Prep + Det + finite verb + adverb, meaning 'in a hurry'), or sequences with different grammatical prop-

¹This should not be confused with the normal use of the term which implies verbal subcategorization frames.

erties than expected from those of the parts: *peut-être* is a compound adverb made of two verb forms.

The designers give the following examples to illustrate cases where some sequences are ambiguous between compound and literal interpretations:

- (1) Paul veut bien que Marie vienne
Paul wants indeed that Mary come
Paul indeed wants Mary to come
- (2) Paul pleure bien que Marie vienne
Paul cries although Mary come
Paul is crying although Marie is coming
- (3) Paul en fait a raison
Paul in fact has reason
Paul in fact is right
- (4) Paul en fait trop
Paul acting too much
Paul is acting too much

The designers explain that in (1), there is no compound: *bien* is an adverb (well) and *que* a subordination conjunction (that); whereas in (2) the same sequence *bien que* is a compound subordination conjunction (although). In (3), the sequence *en fait* is a compound adverb (in fact), whereas in (4) the same sequence must be decomposed into *en* as a clitic and *fait* (does) as a finite verb.

Compounds are annotated with the same tagset as non-compounds, but tags are added for each of their parts. Since the borderline between compounds and free sequences is subject to much linguistic debate, the compound parts as well were annotated, leaving the possibility of ignoring the compounds.

POS	Subcategory	Morphology	Description
A	cardinal,ordinal,poss,qualif,indef,inter	f,m + s,p + 1,2,3	Adjectives
Adv	-, inter,exclam,negative		Adverbs
CL	subj,refl,obj,-	f,m + s,p + 1,2,3	Clitic pronouns
C	subord,coord	-	Conjunctions
D	card,dem,def,indef,exclam,negative, poss,inter,partitive	f,m + s,p + 1,2,3	Determiners
ET	-	-	Foreign words
I	-		Interjections
N	common,proper	f,m + s,p	Nouns
P	-	-	Prepositions
PRO	inter,pers,card,neg,poss,rel,indef	f,m + s,p + 1,2,3	Other pronouns
PONCT	strong,weak	-	Punctuation
PREF	-	-	Prefixes
V	-	f,m + s,p + 1,2,3 + W,G,K,PI,I,J,F,T,C,S,Y	Verbs

Table 3.1: French Treebank POS tags

Phrasal category	Description	Head
<NP>	Noun Phrases	Noun,Pronoun,Adjective,Interjection
<VN>	Verbal nucleus	Verb
<VPinf>	Infinitive clause	Infinitive verb
<VPpart>	Participle clause	Participle verb (present or past)
<PP>	Prepositional Phrases	Preposition
<AdP>	Adverbial Phrases	Adverb
<AP>	Adjectival Phrases	Adjective
<SENT>	Sentences	Verbal Nucleus
<Srel>	Subordinate clause	Verbal Nucleus
<Ssub>	Relative clause	Verbal Nucleus
<Sint>	Other clause	Verbal Nucleus
<COORD>	Coordinated phrases	Coordinating Conjunction

Table 3.2: French Treebank syntactic tagset

3.2 Parsing

The syntactic tagset of the corpus is shown in table 3.2:

As per [A. Abeillé and F.Toussenet, 2003], only major phrases were annotated, with little internal structure (determiners and modifying adjectives at the same level in the noun phrase for example). For the sake of simplicity, parsimonious use of unary phrases are made: there are unary NPs for proper names and pronouns, but not for bare common nouns, there are unary APs for predicative adjectives but not for modifying ones, and there are unary VNs for verbs but no unary AdP for Adverbs. For rigid sequences of categories, such as dates or addresses where it is difficult to determine the head, there is one global NP with no internal constituents.

In order to be as theory neutral as possible, empty categories and functional phrases (such as DP or CP) are not used. Headless phrases (elliptical NP lacking a head Noun as in (10.7)) or sentential clauses lacking a verbal nucleus as in (10.8) are allowed.

- (5) Ce sont <NP> les:D meilleurs:A </NP>
 they are <NP> the:D best:A </NP>
 they are <NP> the:D best:A </NP>
- (6) plus vieille <Ssub> que:CS <NP>toi:PRO</NP></Ssub>
 more old <Ssub> than:CS <NP>you:PRO</NP></Ssub>
 older than you

We will now look at two peculiarities of this corpus:

1. Verbal nucleus (VN)
2. Coordinated phrase (COORD)

3.2.1 Verbal nucleus

For verbal phrases, only the minimal verbal nucleus (clitics, auxiliaries, negation and verb) were annotated, because the traditional VP (with complements) is subject to much linguistic debate and is often discontinuous in French ([A. Abeillé and F. Toussenel, 2003]).

- (7) Les actions qu'a mises IBM sur le marché
 The shares that have put IBM on the market
 The shares that IBM put on the market
- (8) Les actionnaires décideront certainement une augmentation de capital
 The stock holders will decide certainly an increase of capital
 The stock holders will certainly decide on an increase in capital

In (7) the NP subject (IBM) is postverbal and precedes the locative complement (*sur le marché*). In (8), the adverb *certainement* is also postverbal and precedes the NP object (*une augmentation de capital*).

3.2.2 Coordinated phrase

Unlike the Penn Treebank, the French Treebank annotates coordinated phrases with the tag 'COORD'. We could not find an explanation for this design choice.

E.g.

A Londres <COORD> comme:C à New-York</COORD>,la tonne vaut environ 1000 dollars.

In London <COORD> like:C in New-York</COORD>,the ton is worth around 1000 dollars.

3.3 Format

The French Treebank has been formatted in XML following the TEI and XCES guidelines [Ide, 1998]. The corpus provided to us consists of 24 XML files.

Below is an extract from a file.

```
<SENT nb="1000">
  <NP>
    <w lemma="six" ei="PROmp" ee="PRO-card-mp"
      cat="PRO" subcat="card" mph="mp">Six</w>
  <PP>
    <w compound="yes" lemma="d'entre" ei="P" ee="P" cat="P">
      <w catint="P">d'</w>
      <w catint="P">entre</w>
    </w>
  <NP>
    <w lemma="eux" ei="PROmp" ee="PRO-3mp"
      cat="PRO" subcat="3mp">eux</w>
  </NP>
</PP>
```

```

</NP>
<VPpart>
  <w lemma="," ei="PONCTW" ee="PONCT-W" cat="PONCT" subcat="W">,</w>
  <w lemma="seulement" ei="ADV" ee="ADV" cat="ADV">seulement</w>
  <w cat="V" ee="V-Kmp" ei="VKmp" lemma="blesser"
    mph="Kmp" subcat="">blessés</w>
  <w lemma="," ei="PONCTW" ee="PONCT-W" cat="PONCT" subcat="W">,</w>
</VPpart>
<VN>
  <w cat="V" ee="V-P3p" ei="VP3p" lemma="avoir"
    mph="P3p" subcat="">ont</w>
  <w cat="V" ee="V-Kms" ei="VKms" lemma="pouvoir"
    mph="Kms" subcat="">pu</w>
</VN>
:
</SENT>

```

The beginning of a sentence is marked by a <SENT> tag and a sentence number (nb="1000"), and its end marked by a closing tag (</SENT>). However, the sentence numbers are not continuous between files as we were provided with only a subset of the whole corpora (20648 sentences instead of 32000 mentioned in [A. Abeillé and F.Toussenel, 2003]).

3.4 Corpus discussion

In this section, we will go over some inaccuracies and inconsistencies we encountered while working on this corpus, as well as discuss the possible impact of certain design decisions on the parsing implementation.

3.4.1 Tagging

An example of a tagged word in the FTB:

POS	Count
ADVP	1
Dmp	1
S	1
unknown	1

Table 3.3: Invalid POS tags

```
<w cat="V" ee="V-Kmp" ei="VKmp" lemma="blesser"
  mph="Kmp" subcat="" >blessés</w>
```

where:

1. *cat* represents POS tag
2. *lemma* is the lemma of the word
3. *subcat* is the subcategory
4. *ei* is the inflection

The categories *ee* and *mph* are not documented anywhere. While, *mph* can be guessed to be meaning morphology, it's not clear what *ee* represents. Moreover, only *cat*, *lemma* and *ei* were found to be obligatory for every word.

3.4.1.1 Incorrect POS tags

While the POS tag table (3.1) lists 13 such tags, there are actually more.

Since there only 4 of them, these erroneous tags were left in the corpus.

3.4.1.2 No tag information

There are 101 cases where the entire tag information for a word is missing. E.g:

```
<w>des</w>
```

Since it is impossible to fix this problem automatically and doing so manually would involve making linguistically motivated decisions, sentences containing this type of error were discarded.

3.4.1.3 Tag information present but word missing

There are 16490 cases where the information for a word (or a part of a compound) is present but the word (or compound part) is missing. E.g

```
<NP>
  <w lemma="le" ei="Dmp" ee="D-def-mp" cat="D"
    subcat="def" mph="mp" />
  <w lemma="secouriste" ei="NCmp" ee="N-C-mp" cat="N"
    subcat="C" mph="mp">secouristes</w>
</NP>

<w compound="yes" lemma="de les" ei="Dmp" ee="D--mp" cat="D"
  subcat="" mph="mp">
  <w catint="D">des</w>
  <w catint="D" />
</w>
```

It was not easy to decide what to do with these errors given the sheer number of them. Discarding the guilty sentences would reduce the dataset to almost 50% of its original size. One possible solution was to skip the lines without word information. However, we were unsure of the impact it would have on the training algorithms of the different parsing models. We ultimately decided to discard these sentences as it was deemed preferable to have a 'clean' albeit reduced dataset.

3.4.1.4 Typographical errors

There are some random characters ('y', 10460, '>') in the corpus (outside of sentence barriers, it has to be said) that we treated as typographical errors. These characters

were automatically filtered out of the corpus.

3.4.1.5 Comments

Compared to the Penn Treebank, the basic tagset of the French Treebank is smaller. All punctuation marks are represented as the single 'PONCT' tag, there are no modal verbs, no wh-determiners/pronouns/adverbs and no possessives. Adverbs and prepositions are more coarsely defined. On the other hand, the French Treebank introduces the clitic pronoun (CL) for weak pronouns according to the generative tradition [Kayne, 1975].

As for verbs, nouns and adjectives, while they too are coarsely defined (if we just make use of the 'cat' information), we can arrive at a finer level of granularity than that of the PTB by utilising the other morphosyntactic annotation available. However, this has been shown to not always be present and therefore, we decided to use only the basic 13 POS tags listed in 3.1.

French is a morphologically much richer language than English, especially in the case of verbs, which makes POS tagging a harder problem. Keeping the tagset small is a way to counteract the adverse affects of incorrect POS tagging for unknown words.

3.4.2 Compounds

A major feature of the corpus is the presence of compounds. The rationale behind this design decision has been explained previously in this chapter (see 3.1). However, we are unsure about the decision to treat certain words as compounds, particularly the 2 examples below:

a) The word *aujourd'hui* (today) is represented as:

```
<w compound="yes" lemma="aujourd'hui" ei="ADV" ee="ADV" cat="ADV">
  <w catint="P">aujourd'</w>
  <w catint="N">hui</w>
</w>
```

This is undoubtedly a bizarre choice, since on their own, neither *aujourd'* nor *hui* are valid French words.

b) Numbers values with more than 3 digits are represented as compounds as well:

```
<w compound="yes" lemma="1 000" ei="Dmp" ee="D-card-mp"
  cat="D" subcat="card" mph="mp">
  <w catint="">1</w>
  <w catint="">000</w>
</w>
```

As a comparison, in the Penn Treebank, numbers with more than 3 digits are represented with a comma after every 3 digits and are tagged as **CD**.

e.g. (NP (CD 14,789,000) (NNS tons))

3.4.2.1 Missing POS tags for compound parts

While as mentioned in 3.1, all compound parts are supposed to be tagged, we find that this is not the case. e.g

```
<w compound="yes" lemma="1 000" ei="Dmp" ee="D-card-mp"
  cat="D" subcat="card" mph="mp">
  <w catint="">1</w>
  <w catint="">000</w>
</w>

<w compound="yes" lemma="entendre parler" ei="VW" ee="V--W"
  cat="V" subcat="" mph="W">
  <w catint="">entendre</w>
  <w catint="">parler</w>
</w>
```

In the examples, the POS tags denoted by **catint** are missing for the compound parts.

The decision on what to do in such situations is dependent on what the approach towards compounds intends to be. We will come back to this in the next chapter.

3.4.3 Parsing

3.4.3.1 Erroneous annotation

There are 8 sentences where the POS tag 'PONCT' has been used as a syntactic category.

Ex:

```
<PONCT>
  <w cat="PONCT" ee="PONCT-W" ei="PONCTW" lemma="( " subcat="W"></w>
</PONCT>
```

These were manually corrected.

3.4.3.2 Comments

Annotating only the verb nucleus (VN) and not the verbal phrase (VP), results in a flatter treebank.

Instead of having rules of type :

```
SENT -> NP VP
VP -> V NP PP
```

we now have

```
SENT -> NP VN NP PP
```

Moreover, since the arguments of a VN are sisters to it in the tree headed by SENT, it is harder to implement the complement/adjunct distinction that's fundamental to Collins' Model 2, and therefore training the model to learn subcategorization frames for verbs becomes much more difficult too, as we will show in 4.

```
(SENT
  (VN (I put))
  (NP (some very nice little words)))
```

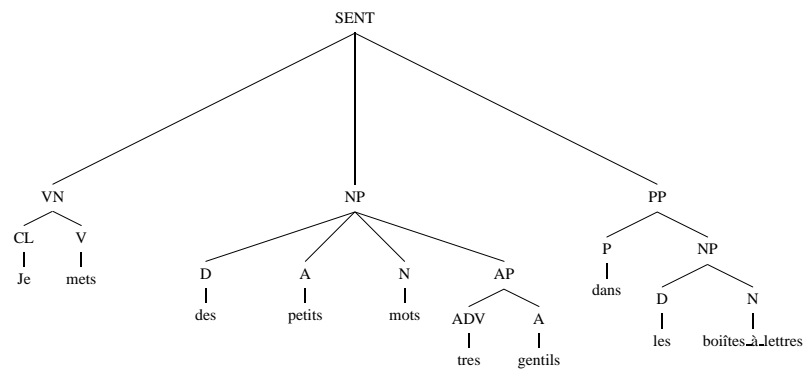


Figure 3.1: Example of a FTB tree

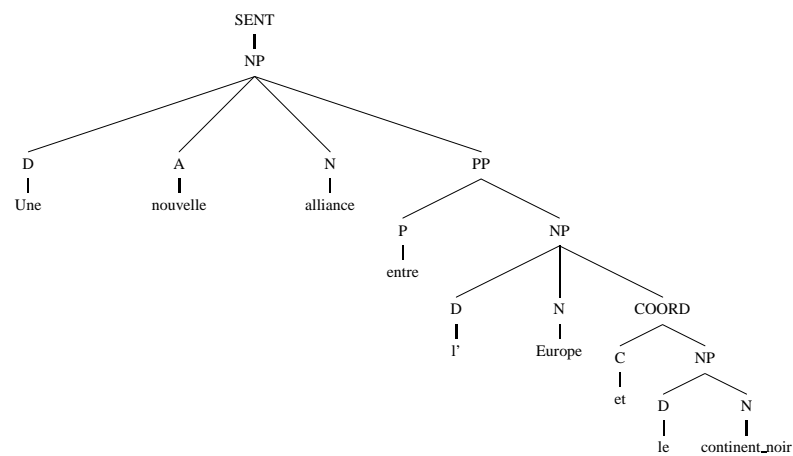


Figure 3.2: Example of a coordinated FTB tree

(PP (in the mailboxes)))

On the other had, having coordination as a syntactic category adds binary branching to trees.

(SENT (NP A new alliance between Europe and the black continent))

It is not exactly clear whether this annotation style will help or harm parsing accuracy. My intuition is that since as mentioned in Chapter 12 of [Manning and Schütze, 1999], the standard PARSEVAL measures [Black et al., 1991] are biased in favour of flat structures (there are less bracketing decisions to make), parsing accuracy would be negatively impacted. On the other hand, with the binary branching structure, presumably the brackets for the COORD constituent are easy to detect, since they are always

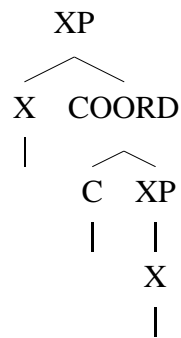


Figure 3.3: Coordination - French Treebank annotation style

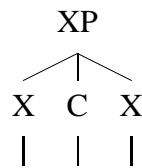


Figure 3.4: Coordination - Penn Treebank annotation style

adjacent to a coordinating conjunction.

In chapter 4, we will present experiments that were performed using both the current annotation scheme and the Penn Treebank scheme, created by getting rid of the 'COORD' syntactic category and 'raising coordination' up a level in the tree.

3.5 Comparison between English and French

Both English and French are right-branching, **SVO** languages with similar verb orders.

	French	Je	mets	des petits mots très gentils	dans les boîtes `a lettre
Declarative sentence		<i>Subj</i>	<i>Verb</i>	<i>Object</i>	
	English	I	put	some very nice little words	in the mailboxes.
		<i>Subj</i>	<i>Verb</i>	<i>Object</i>	

Yes/No question :

French: A-t-il posté la lettre ?

English: Has he posted the letter ?

Imperative sentence :

French: Donne-moi le livre!

English: Give me the book!

While in English, adjectives appear before the noun, in French, they can stand either before or after the noun and in some cases the meaning of the adjective changes slightly according to where it is placed. The spelling of the adjective is affected by the gender and number of the noun it is affecting.

Hence, in the previous example,

French	petits	mots	très	gentils
	<i>ADJ (plural)</i>	<i>Noun</i>	<i>ADV</i>	<i>ADJ (plural)</i>
English	very	nice	little	words
	<i>ADV</i>	<i>ADJ</i>	<i>ADJ</i>	<i>Noun</i>

While English has relatively simple inflectional system: only nouns, verbs, and sometimes adjectives can be inflected, and the number of possible inflectional affixes is quite small, French is highly inflected especially in the case of verbs.

In French, verbs can be grouped into three different categories called groups.

First group all verbs with infinitives finishing with *-er* except *aller*. Particularity: this is the most regular group because the radical does not change during the conjugation: (*aimer*: *aim-e*; *aim-ons*; *aim-ent*).

Second group all verbs with infinitives finishing with *-ir* and past participles with *-issant*. This is a regular group. Those verbs always use a double radical. One for the singular and the second one for plural: (*fin-is*; *finiss-ons*).

Third group All the irregular verbs. They can be further divided into four main sub-categories:

1. verbs in *-ir* (like *mourir*: *mour-ant*, *mour-ons*)
2. verbs in *-oir* (like *recevoir*: *recev-ant*, *recev-ons*)
3. verbs in *-re* (like *rendre*: *rend-ant*, *rend-ons*)
4. *aller* even if it terminates with *-er*.

Verbs of each category inflect depending on the 6 possible subject number and case combination (je, tu, il - elle, nous, vous, ils - elles) and the verb tense (around 20 different tenses and moods exist).

This leads to a very large number of possible word forms, and consequently sparse data problems during lexicalisation are to be expected. On the positive side, inflectional information should provide cues to parse structure, if the parsing model is somehow parametrised to make use of this information [Collins et al., 1999].

Chapter 4

Experiments

In this chapter, we will go over the experiments that we ran and will give detailed description of the probability models that were used.

4.1 Formatting

4.1.1 XML to bracketed expressions

Most existing parsers expect as input Penn Treebank style bracketed expressions. The XML formatted French Treebank was therefore converted to the required format using XSL (eXtensible Stylesheet Language) transformations. We decided to only keep the POS tag and discard the morphological information for each terminal. For example:

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

gets transformed to:

```
(NP (PRO eux))
```

4.1.2 Compounds

With respect to compounds, we decided to try two different approaches.

a) Collapsing the compound.

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

would yield

```
(P d'_entre)
```

i.e concatenate the compound parts using ' _ ' and pick up the "cat" information supplied at the compound level.

b) Expanding the compound.

The previous example yields,

```
(PCmp (P d') (P entre))
```

Here, the compound parts are treated as individual words with their own POS tag (from the *catint* tag), and the tag of the Compound is appended to the suffix 'Cmp', effectively expanding the tagset.

However, things are not always this straight forward. There are cases where the POS tag of the compound part is missing (i.e the value of *catint* is blank), e.g.:

```
<w compound="yes" lemma="la plupart de" ei="Dmp" ee="D-mp"
  cat="D" subcat="mp">
  <w catint="">la</w>
  <w catint="">plupart</w>
</w>
```

In cases like this, we decided to substitute the missing POS tags with the POS tag of the compound.

Therefore the previous example is mapped to,

```
(DCmp (D la) (D plupart))
```

This is not ideal since the POS tags are being "guessed" (e.g. "*plupart*" is tagged as Noun or Adjective in the corpus) but it is a work-around that usually gets it right.

4.1.3 Punctuation

The French Treebank has punctuations tagged as *PONCT*. We decided to follow the Penn Treebank convention and reassigned the punctuations new POS tags in line with the PTB tagset. The reason behind this decision is that Collins's parsers have quite a few rules that are dependent on these tags and our intent was to make as few changes as possible to the parsing algorithm. Moreover, punctuations that can be problematic to some computing systems e.g. ("(", ")") , "[" , "]") were replaced by a more convenient representation (4.1).

4.1.4 Raising coordination

We maintain a dataset whereby a raising coordination transformation is applied. As mentioned previously, coordination structures have their own constituent label (*COORD*) in the FTB annotation scheme. Some of the models that we plan to use have coordination specific rules, where coordination is marked in the Penn Treebank style. Also, we postulate that having a flatter *COORD* representation would lead to a favourable bias with respect to the PARSEVAL measures.

The raising coordination transformation therefore converts a tree such as that depicted in figure 4.1 to one like figure 4.2.

It is important to note that, in the FTB annotation scheme, a coordinating conjunction is **always** followed by a syntactic category. The resulting tree, though flatter, is still not compatible with the PTB treatment of coordinations.

Original Punctuation	New Punctuation	New tag
!	!	.
”	”	”
,	,	,
-	-	:
.	.	.
/	/	:
:	:	:
;	;	:
=	=	:
?	?	.
[-LSB-	-LRB-
]	-RSB-	-RRB-
(-LRB-	-LRB-
)	-RRB-	-RRB-
...	...	:
(...)	(...)	:
(*)	-LRB-*-RRB-	.

Table 4.1: New FTB punctuation tagset

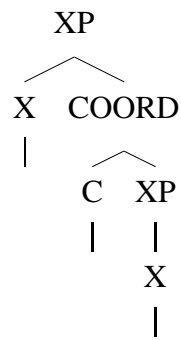


Figure 4.1: Coordination before transformation

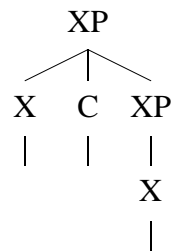


Figure 4.2: Coordination after transformation

4.2 Data Sets

After discarding the 10,096 inconsistently annotated sentences, the remaining data set of 10,552 sentences (222,569 words at an average sentence length of 21.1 words) was split, as per the usual statistical NLP experimental methodology, into a training set, a development set - used to test the parsing models and to tune their parameters, and a test set, unseen during the development phase, and used only once all parameters were fixed.

The training set was constituted from the first 8,552 sentences, with the following

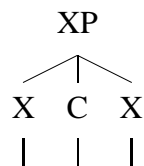


Figure 4.3: PTB coordination annotation

1000 sentences serving as the development set and the final 1000 sentences forming the test set. All results reported in this thesis were obtained on the test set, unless stated otherwise.

4.3 Experiments with unlexicalised PCFG

Lexicalisation has been shown to improve parsing performance for the Penn Treebank (e.g [Collins, 1997, Charniak, 1997, Charniak, 1999]). However, in the case of the German Negra corpus, a lexicalised model following Model 1 of [Collins, 1997] failed to outperform an unlexicalised baseline model [Dubey and Keller, 2003].

This thesis aims to evaluate the impact of lexicalisation over the French Treebank parsing accuracy. We follow the work of [Dubey and Keller, 2003] by starting off with a baseline unlexicalised standard probabilistic context-free grammar (PCFG).

Four different sets of experiments were ran on the baseline model:

1. Corpus with compounds contracted, coordination not raised and unary rules removed.
2. Corpus with compounds contracted, coordination raised and unary rules removed.
3. Corpus with compounds expanded, coordination not raised and unary rules removed.
4. Corpus with compounds expanded, coordination raised and unary rules removed.

4.3.1 PCFG

A context-free grammar G is defined by four parameters (N, Σ, P, S) :

1. a set of non-terminal symbols (or "variables") N
2. a set of terminal symbols Σ (disjoint from N)

3. a set of productions P , each of the form $A \rightarrow \beta$, where A is a non-terminal and β is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$
4. a designated start symbol S

e.g. For the French Treebank, a simple CFG (with start symbol SENT) would be:

SENT \rightarrow NP VN NP

NP \rightarrow D N

VN \rightarrow V

PP \rightarrow P NP

D \rightarrow La | le

N \rightarrow fille | garçon

V \rightarrow aime

which would yield a sentence like:

- (1) La fille aime le garçon.
 The girl likes the boy
 The girl likes the boy

A probabilistic context-free grammar augments each rule in P with a conditional probability:

$$A \rightarrow \beta[p] \tag{4.1}$$

A PCFG is thus a 5-tuple $G = (N, \Sigma, P, S, D)$, where D is a function assigning probabilities to each rule in P . This function expresses the probability p that the given non-terminal A will be expanded to the sequence β ; We will refer to it as $P(A \rightarrow \beta|A)$.

Formally, this is the conditional probability of a given expansion given the left-hand side non-terminal A . Thus if we consider all the possible expansions of a non-terminal, the sum of their probabilities must be 1.

A PCFG assigns a probability to each parse-tree T of a sentence S . During probabilistic parsing, the derivation having the highest probability for a given sentence is selected.

The probability of a particular parse T is the product of the probabilities of all the rules r used to expand each node n in the parse tree.

$$P(T, S) = \prod_{n \in T} p(r(n)) \quad (4.2)$$

This is the joint probability of the parse and the sentence. However,

$$P(T, S) = P(T)P(S|T) \quad (4.3)$$

and since a parse tree includes all words of the sentence, $P(S|T)=1$. Thus:

$$P(T, S) = P(T)P(S|T) = P(T) \quad (4.4)$$

The job of a parser is to find the most probable tree for a sentence S out of the set of parse trees for S (which I'll call $\tau(S)$).

$$\hat{T}(S) = \operatorname{argmax}_{T \in \tau(S)} P(T|S) \quad (4.5)$$

By definition, the probability $P(T|S)$ can be rewritten as $P(T, S)/P(S)$, thus leading to:

$$\hat{T}(S) = \operatorname{argmax}_{T \in \tau(S)} \frac{P(T, S)}{P(S)} \quad (4.6)$$

The term $P(S)$ can be dropped from the equation since we are maximising over all possible parse trees for the same sentence.

$$\hat{T}(S) = \operatorname{argmax}_{T \in \tau(S)} P(T, S) \quad (4.7)$$

and since from 4.4, $P(T, S) = P(T)$, finding the most likely parse simplifies to

$$\hat{T}(S) = \operatorname{argmax}_{T \in \tau(S)} P(T) \quad (4.8)$$

4.3.2 BitPar

This subsection describes the functionality of BitPar ([Schmid, 2004]), a parser based on a bit-vector implementation of the well-known CKY algorithm ([Kasami, 1965]). The most probable parse is computed in four steps. First, a CKY-style recogniser fills the chart with constituents. This is followed by a top-down filtering of the chart, the bottom-up computation of the Viterbi probabilities and finally the top-down extraction of the best parse.

For each of the four experiments performed, a grammar and a lexicon for BitPar were read off the training set. Since the CKY algorithm requires a grammar to be in Chomsky normal form, where the right-hand side of each rule either consists of two non-terminals or a single terminal symbol, BitPar has to split rules with more than 2 non-terminals on the right-hand side into binary rules. In contrast to most beam searching parsing strategies, BitPar is guaranteed to return the most probable analysis.

4.3.2.1 Parameter estimation

The parameters of the PCFG (the probability of each expansion of a non-terminal) are estimated using Maximum Likelihood Estimation (MLE).

$$P(\alpha \rightarrow \beta \mid \alpha) = \frac{\operatorname{Count}(\alpha \rightarrow \beta)}{\sum_{\gamma} \operatorname{Count}(\alpha \rightarrow \gamma)} = \frac{\operatorname{Count}(\alpha \rightarrow \beta)}{\operatorname{Count}(\alpha \rightarrow \beta)} \quad (4.9)$$

where the count of events are obtained from the grammar (when β is one or more nonterminals) and the lexicon (when α is a POS tag and β is a terminal).

4.3.2.2 Smoothing

MLE assigns zero probability to all unobserved events. Smoothing allows the model to assign a positive probability to events which are not observed in the training corpus, as the test data is likely to contain some of them.

BitPar uses a variant of the absolute discounting method [Ney et al., 1994] for this purpose.

4.3.2.3 Unknown words

In BitPar, a POS tag distribution for unknown words has to be specified, which is then used to tag unknown words in the test data.

This is done by supplying to BitPar 2 files, one containing open class frequencies for capitalised words, and the other for uncapitalised words.

BitPar simply assigns all the POS tags specified in the files to an unknown word (depending on its capitalisation). The probability $P(\text{tag}|\text{word})$ is computed simply by MLE from the tag frequencies specified in the appropriate POS tag distribution file.

BitPar can also be run in a "perfect tagging" mode, by passing the parser the POS tags instead of the words.

Ex:

D

N

V

.

At the same time, the lexicon is modified so that it maps all POS tags to themselves.

D D

N N

V V

A A

Experiments were ran in both modes, the "perfect tagging" mode providing an upper bound for parsing performance.

4.4 Problems with PCFG

While PCFGs are a natural extension to context-free grammars, they suffer from a number of shortcomings that limit their effectiveness as probability estimators. One problem with PCFGs comes from their fundamental independence assumptions. By definition, a CFG assumes that the expansion of any one non-terminal is independent of the expansion of any other non-terminal. Similarly, each PCFG rule is assumed to be independent of every other rule, and the rules are multiplied together. However, this is not true in the case of natural language. In English, the choice of how a node expands is dependent on the location of the node in the parse tree, e.g. based on statistics from the Penn Treebank, pronouns, proper names and definite NPs appear more commonly in the subject position while NPs containing post-head modifiers and bare nouns occur more commonly in the object position [Manning and Schütze, 1999]. This reflects the fact that the subject normally expresses the sentence-internal topic. Given the similarities between English and French, this finding is expected to hold true for French as well. The PCFG model does not allow to capture such kind of structural preferences. Similarly, PCFGs fail to address the preference for right-branching structures that are prevalent in both the French and the Penn Treebanks.

PCFGs are also criticised for their lack of lexical sensitivity. Lexical information in a PCFG can only be represented via the probability of pre-terminal nodes (*Verb*, *Noun*, *Determiner*) to be expanded lexically. But there are a number of other kinds of lexical and other dependencies that turn out to be important in modelling syntactic probabilities. It is a well studied fact that lexical information plays an important fact in selecting the correct parsing of an ambiguous prepositional-phrase attachment [Hindle and Rooth, 1991].

Consider the example shown before :

- (2) Je mets des petits mots très gentils dans les boîtes à lettres.
 I put some little words very nice in the box of letters
 I put some nice little words in the mail box

Here, the prepositional phase *dans les boîtes à lettres* can be attached either to the NP *des petits mots très gentils* or to the verbal nucleus headed by *mets*. In the PCFG derived from the FTB, the attachment choice comes down to the choice between two rules:

- $NP \rightarrow NP PP$ (NP-attachment)
- $SENT \rightarrow VN NP PP$ (VN-attachment) Given the absence of VPs from the FTB annotation scheme, the structure above will be referred to, for lack of a better word, as a VN-attachment.

In the example above, the correct attachment is to the verb; because the verb *mettre* subcategorises for a location, which can be expressed with the preposition *dans*. Thus a model which kept separate lexical dependency statistics for different verbs would be able to choose the correct parse in these cases.

The simplest and most common way of adding lexical sensitivity to a CFG is to annotate each nonterminal by its head word. Central to this model of lexicalisation is the idea that the strong lexical dependencies are between heads and their dependents, e.g. between a head noun and a modifying adjective, or between a verb and a noun phrase object. Figure 4.4 shows an example of a lexicalised tree.

Therefore, from the observations above, we should be able to build a much better probabilistic parser than one based on a PCFG by better taking into account lexical and structural context. In the next section, we will look at one such parser, the Collins parser, used to perform the remaining experiments of this thesis.

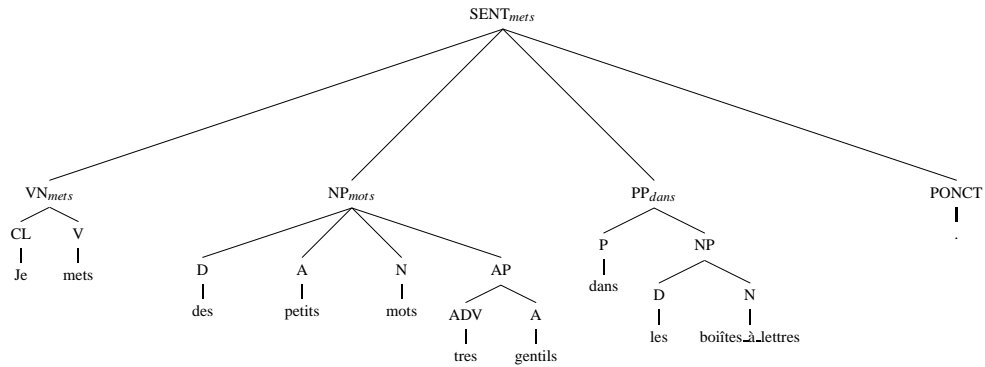


Figure 4.4: A lexicalised tree

4.5 Collins' parser

Lexicalised parsing experiments were run using Dan Bikel's multi-lingual, parallel processing, statistical parsing Java engine [Bikel, 2002] which in addition to replicating the models described in [Collins, 1999], also provides a convenient interface to develop corresponding parsing models for other languages, by extending a Java language package. Such an implementation for the Chinese Treebank resulted in state-of-the-art results for that corpus [Bikel, 2002].

The parsing model implemented by PCFGs defines a conditional probability $P(T|S)$, for each candidate parse tree T for a sentence S . [Collins, 1997] proposes a generative model whereby the decoding task involves maximising $P(T, S)$ instead of $P(T|S)$:

$$\begin{aligned} T_{best} &= \operatorname{argmax}_T P(T|S) = \operatorname{argmax}_T \frac{P(T|S)}{P(S)} \\ &= \operatorname{argmax}_T P(T, S) \quad (\text{since } P(S) \text{ is a constant}) \end{aligned} \quad (4.10)$$

$P(T, S)$ is then estimated by attaching probabilities to a top-down derivation of the tree. In a PCFG, for a tree derived by n applications of context-free re-write rules $LHS_i \rightarrow RHS_i, 1 \leq i \leq n$,

$$P(T, S) = \prod_{i=1..n} P(RHS_i | LHS_i) \quad (4.11)$$

4.5.1 Collins' Model 1

The models implemented by the Collins' parser lexicalise a PCFG by associating a word w and a POS tag t with each non-terminal \mathbf{X} in the tree [Collins, 1997]. Thus, a non-terminal is written as $X(x)$ where $x = \langle w, t \rangle$ and \mathbf{X} a constituent label. Each rule now has the form:

$$P(h) \rightarrow L_n(l_n) \dots L_1(l_1) H(h) R_1(r_1) \dots R_m(r_m) \quad (4.12)$$

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

The head rules were constructed based on the FTB annotation guidelines provided along with the dataset (see 3.2), as well as by eye-balling the grammar rules read off the training date set. They are presented in Appendix A.

The addition of lexical heads leads to an enormous number of potential rules, making direct estimation of $P(RHS|LFS)$ infeasible because of sparse data problems. Therefore, the generation of the RHS of a rule such as 4.12, given the LHS, is decomposed into three steps, namely, first generating the head, then making the independence assumptions that the left and right modifiers are generated by separate zeroth-order Markov processes:

1. Generate the head constituent label of the phrase, with probability $P_H(H | P, h)$.

2. Generate modifiers to the right of the head with probability

$$\prod_{i=1 \dots m+1} P_R(R_i(r_i) | P, h, H).$$

$R_{m+1}(r_{m+1})$ is defined as *STOP* - the *STOP* symbol is added to the vocabulary of non-terminals, and the model stops generating right modifiers when it is generated.

3. Generate modifiers to the left of the head with probability

$$\prod_{i=1 \dots n+1} P_L(L_i(l_i) | P, h, H),$$

where $L_{n+1}(l_{n+1}) = \text{STOP}$.

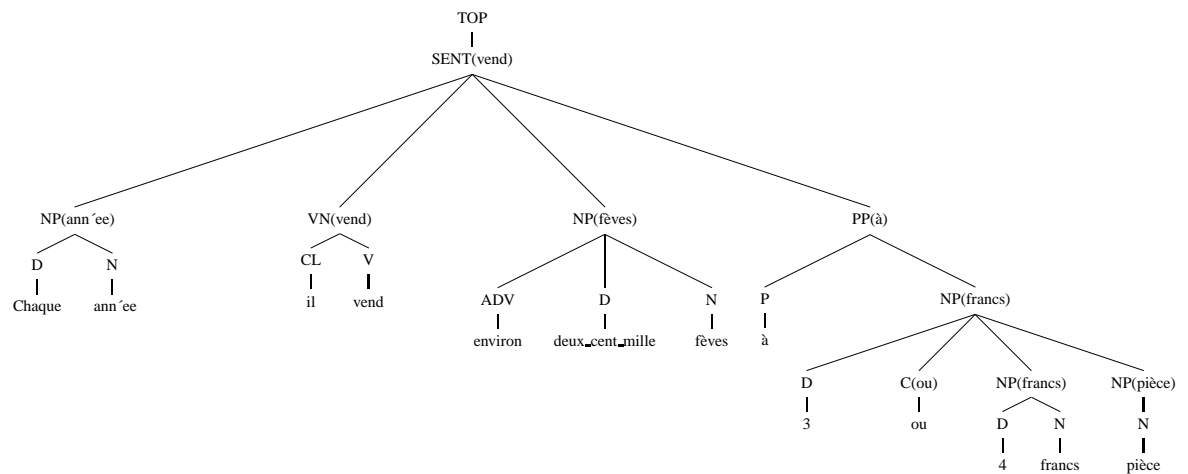


Figure 4.5: Collins' Model 1 example

Let's illustrate all this with an example:

- (3) Chaque année il vend environ deux cent mille fèves à 3 ou 4 francs
 Every year he sells around two hundred thousand beans at 3 or 4 francs
 pièce
 each
 Every year he sells around two hundred thousand beans beans for 3 or 4 francs
 each.

TOP → SENT(vend)
 SENT(vend) → NP(année) VN(vend) NP(fèves) PP(à)
 NP(année) → D(chaque) N(année)
 VN(vend) → CL(il) V(vend)
 NP(fèves) → ADV(environ) D(deux_cent_mille) N(fèves)
 PP(à) → P(à) NP(francs)
 NP(francs) → D(3) C(ou) NP(francs) NP(pièce)
 NP(francs) → D(4) N(francs)
 NP(pièce) → N(pièce)

For example, the probability of the rule

SENT(vend) → NP(année) VN(vend) NP(fèves) PP(à)

would be estimated as:

$$\begin{aligned}
& P_h(VN \mid SENT, vend) \times P_l(NP(année) \mid SENT, VN, vend) \times \\
& P_l(STOP \mid SENT, VN, vend) \times P_r(NP(fèves) \mid SENT, VN, vend) \times \\
& P_r(PP(\grave{a}) \mid SENT, VN, vend) \times P_r(STOP \mid SENT, VN, vend)
\end{aligned} \tag{4.13}$$

A zeroth order Markov assumption has been made:

$$\begin{aligned}
P_l(L_i(l_i) \mid H, P, h, L_1(l_1) \dots L_{i-1}(l_{i-1})) &= P_l(L_i(l_i) \mid H, P, h) \\
P_r(R_i(r_i) \mid H, P, h, R_1(r_1) \dots R_{i-1}(r_{i-1})) &= P_r(R_i(r_i) \mid H, P, h)
\end{aligned} \tag{4.14}$$

In subsequent experiments, we will tinker with the order of the Markov assumption in an attempt to take into account the flatter structure of the French Treebank.

4.5.1.1 The distance metric

Thus far the model has assumed that the modifiers are generated independently of each other. The dependency between the modifiers is increased by adding a **distance metric** in the conditioning during generation of the modifiers.

i.e.

$$\begin{aligned}
P_l(L_i(l_i) \mid H, P, h, L_1(l_1) \dots L_{i-1}(l_{i-1})) &= P_l(L_i(l_i) \mid H, P, h, distance_l(i-1)) \\
P_r(R_i(r_i) \mid H, P, h, R_1(r_1) \dots R_{i-1}(r_{i-1})) &= P_r(R_i(r_i) \mid H, P, h, distance_r(i-1))
\end{aligned} \tag{4.15}$$

$distance_l$ and $distance_r$ are functions of the surface string between the head and the previously generated modifier.

The distance measure is a vector composed of 2 elements:

1. Is the string of zero length ? (1 if true, 0 otherwise) [Collins, 1999] states that this parameter enables the model to learn a preference for right-branching structures. Since French, like English, is considered to be a right-branching language, we decided to include this parameter in our model.

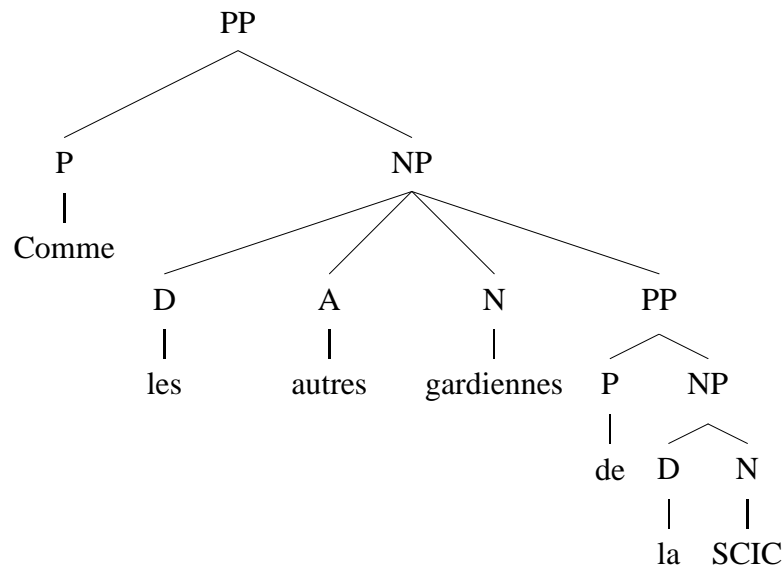


Figure 4.6: Distance measure - Correct parse tree

2. Does the string contain a verb ? (1 if true, 0 otherwise) This allows the model to learn a preference for modification of the most recent verb.
- (4) Comme les autres gardiennes de la SCIC
 Like the other female guards of the SCIC
 Like the other female guards of the SCIC

The example in Figures 4.6 and 4.7 shows how useful the distance measure can be. The tree in figure 4.6 is the correct parse tree for this sentence. It has a right-branching structure at the PP level. In the incorrect parse (figure 4.7) the second NP (headed by *SCIC*) attaching to the preposition *comme* gives a dependency ($SCIC \rightarrow comme, R, \langle NP, PP, P \rangle, 00$) whereas the first NP attachment gives ($gardiennes \rightarrow comme, R, \langle NP, PP, P \rangle, 10$) where *R* indicates that the modifier is to the right of the head and the terms in the angled-brackets represent the constituent label of the modifier, the head constituent label and the tag of the head word. The distance variable differentiates between a dependency where the NP is/isn't adjacent to the preposition: the result is that ($SCIC \rightarrow comme, R, \langle NP, PP, P \rangle, 00$) will get very low probability (a PP will almost never be seen with an NP modifier at distance 00) and the parse tree will get a low score.

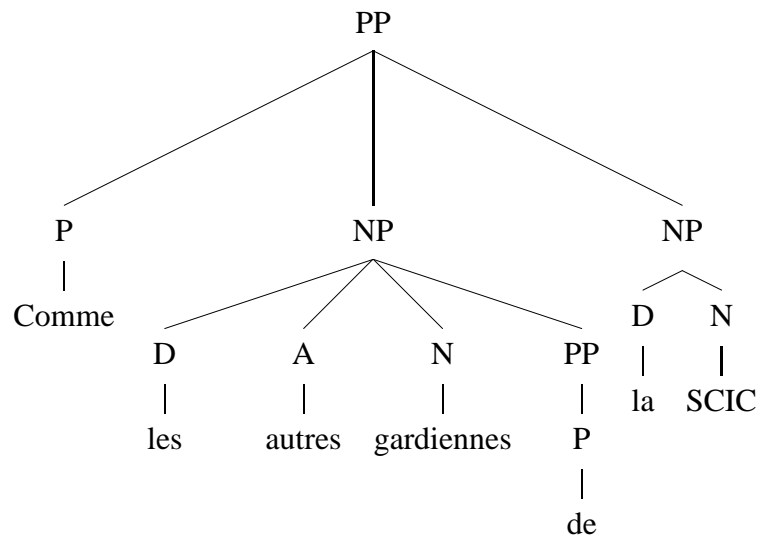


Figure 4.7: Distance Measure - Incorrect parse tree

However, the verb intervening flag is less helpful in the French Treebank compared to the PTB, because of the absence of a VP label.

Collins gives the example of Figures 4.8 and 4.9 to illustrate the usefulness of the verb intervening flag.

He argues that the 2 trees only differ by a single dependency (*by* → *shot*) vs (*by* → *believed*), and that since both dependencies are conditioned on $\langle PP, VP, VBN \rangle$, the distance variable allows the model to discriminate between the first attachment as an attachment that does not cross the verb (distance 00) and the second attachment as one that does cross a verb (distance 01).

A corresponding example is shown from the FTB (Figures 4.10 and 4.10).

- (5) Il faudrait discuter avec les habitants
 He will have to discuss with the inhabitants
 One will have to discuss with the inhabitants

Here, the two competing dependencies are (*avec* → *discuter*, $R, \langle PP, VPinf, VN \rangle, 00$) for the first tree vs (*avec* → *faudrait*, $R, \langle PP, SENT, VN \rangle, 01$) for the second tree. The head nonterminal label is already differing because of the annotation scheme, making

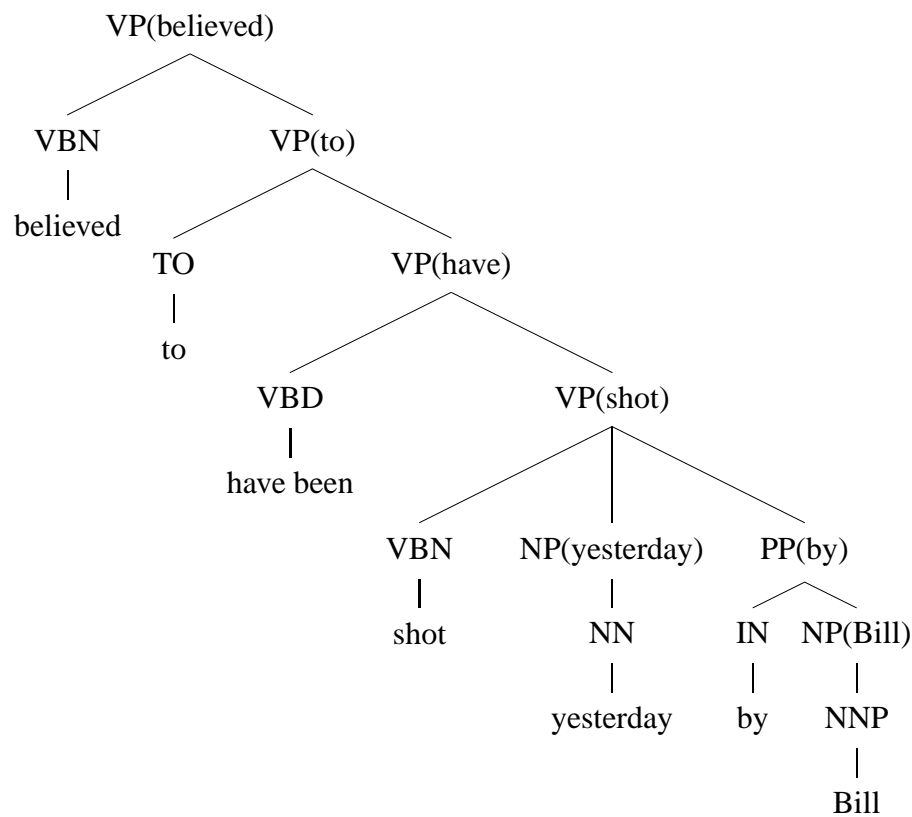


Figure 4.8: Verb-intervening PTB - Correct parse

the verb intervening feature less important from a discriminatory perspective. However, this feature was maintained for the purpose of this thesis.

4.5.2 Model 2

Model 2 incorporates a complement/adjunct distinction and probabilities over sub-categorization frames. These added features model a crucial bit of linguistic reality, which is that words often have well-defined sets of complements and adjuncts, occurring with some well-defined distribution in the right hand sides of a (context-free) rewriting scheme [Bikel, 2004]. Collins showed that this enriched model improved parsing accuracy (88.1% LR and 88.6% LP compared to 87.4% LR and 88.1% LP on Model 1) for the Penn Treebank.

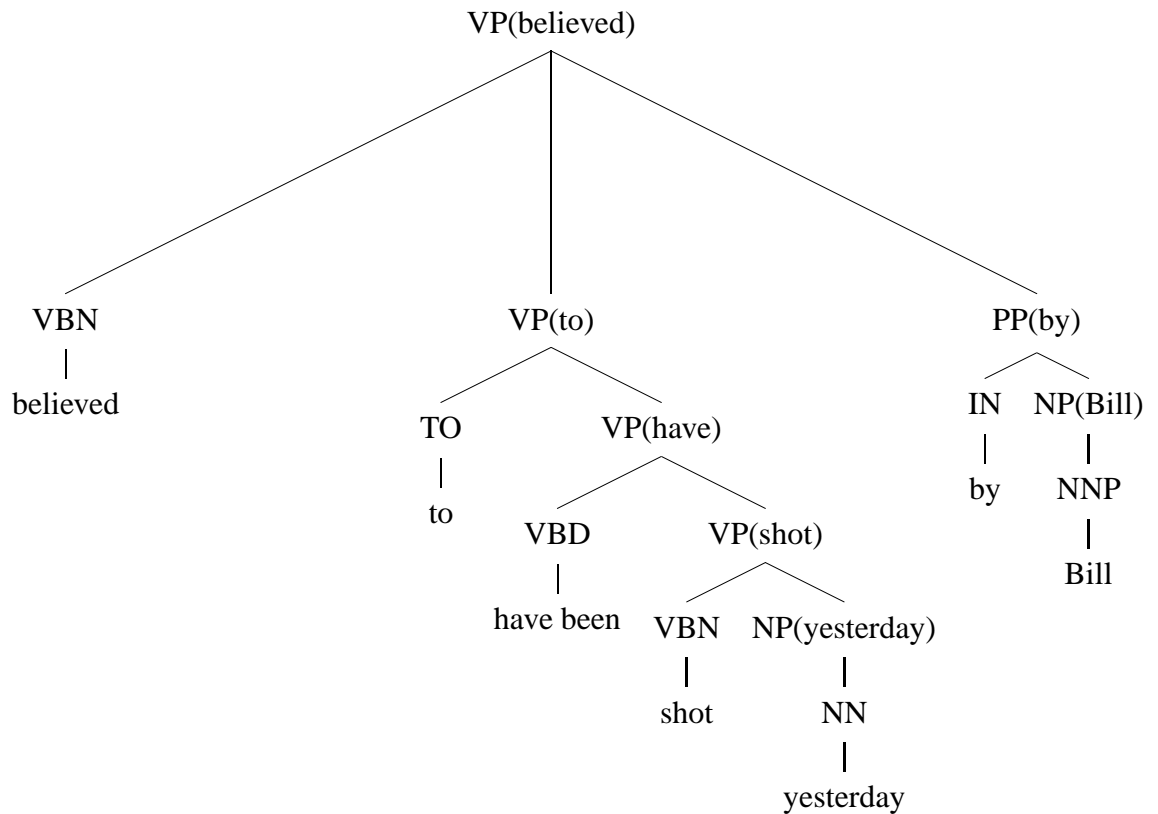


Figure 4.9: Verb-intervening PTB - Incorrect parse

Model 1 is then retrained on the training data with the enhanced set of non-terminals, and it might learn the lexical properties which distinguish complements and adjuncts. The generative process is enhanced to include a probabilistic choice of left and right subcategorization frames [Collins, 1999]:

1. Choose a head H with probability $P_H(H | P, h)$.
2. Choose left and right subcat frames, LC and RC , with probabilities $P_{lc}(LC | P, H, h)$ and $P_{rc}(RC | P, H, h)$.

Each subcat frame is a multiset specifying the complements that the head requires in its left or right modifiers.

3. Generate the left and right modifiers with probabilities $P_l(L_i, l_i | H, P, h, distance_l(i-1), LC)$ and $P_r(R_i, r_i | H, P, h, distance_r(i-1), RC)$ respectively. Thus, the subcat require-

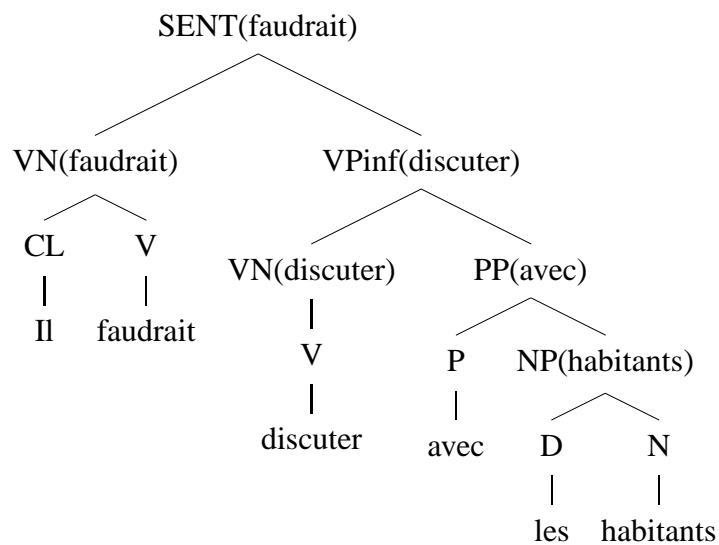


Figure 4.10: Verb-intervening FTB - Correct parse

ments are added to the conditioning context. As complements are generated, they are removed from the appropriate subcat multiset. Most importantly, the probability of generating a STOP symbol will be 0 when the subcat frame is *non-empty*, and the probability of generation of a complement will be 0 when it is not in the subcat frame; thus only the required complements will be generated.

- (6) Ce jour l`a un certain parfum de krach a même r´egn´e dans les salles de march´e
 That day a certain perfume of crash was even ruling in the rooms of market
 That day a certain smell of crash was even present in the market rooms

For example, in the sentence in Figure 4.12, we mark the complements with a marker (‘-C’ in this case) in the training phase based on argument identification rules. The argument identification rules are listed in the appendix B. The two used here are: (a) if the parent node is SENT, mark the last NP before the head as an argument, and (b) mark the first child following the head of a prepositional phrase as a complement.

The probability of the rule $\text{SENT}(\text{régner}) \rightarrow \text{NP}(\text{jour}) \text{NP-C}(\text{parfum}) \text{VN}(\text{régner}) \text{PP}(\text{dans})$

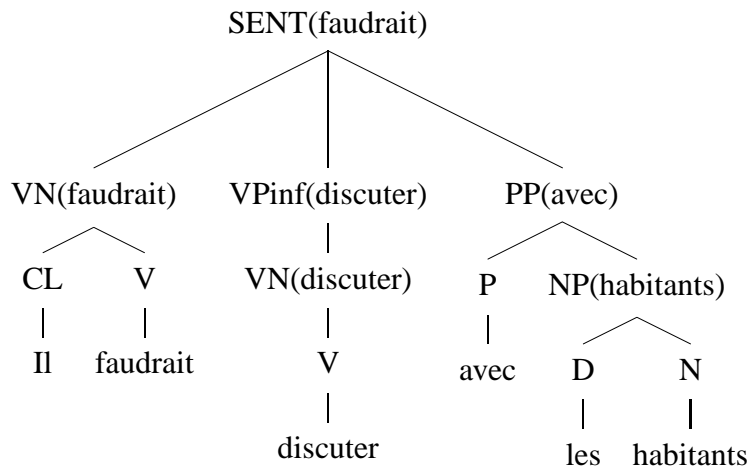


Figure 4.11: Verb-intervening FTB - Incorrect parse

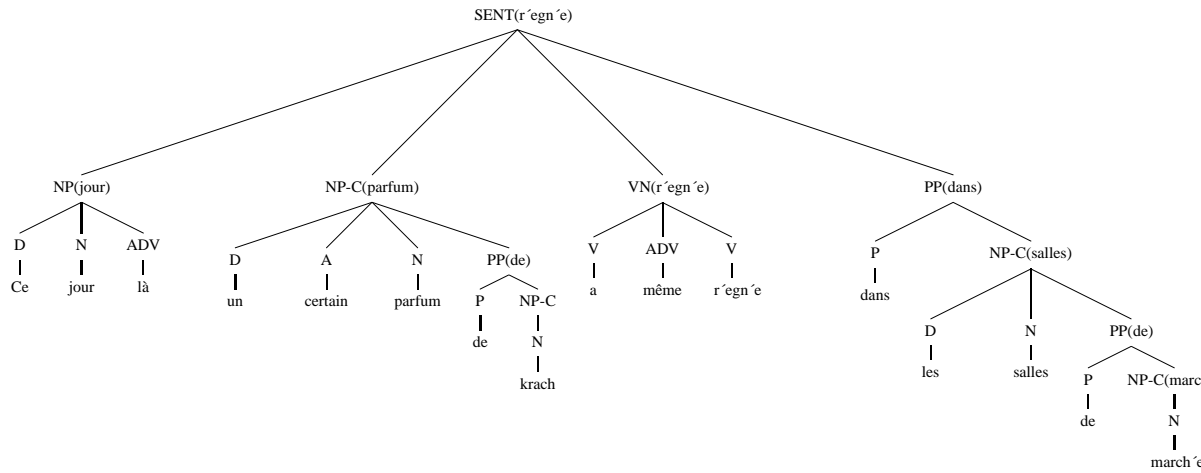


Figure 4.12: Tree with complement/adjunct distinction

is:

$$\begin{aligned}
 &P_h(VN \mid SENT, regné) \times P_{lc}(\{(NP - C)\} \mid SENT, VN, regné) \times \\
 &P_{rc}(\{\} \mid SENT, VN, regné) \times P_l(NP - C(parfum) \mid SENT, VN, regné, \{(NP - C)\}) \times \\
 &P_l(NP(jour) \mid SENT, VN, regné, \{\}) \times \\
 &P_l(STOP \mid SENT, VN, regné, \{\}) \times P_r(PP(dans) \mid SENT, VN, regné, \{\}) \times \\
 &P_r(STOP \mid SENT, VN, regné, \{\})
 \end{aligned}
 \tag{4.16}$$

The head initially decides to take a single **NP-C**(subject) to its left, and no complements to its right. **NP-C(parfum)** is immediately generated as the required subject, and **NP-C** is removed from *LC*, leaving it empty when the next modifier **NP(jour)** is generated.

It is hard to say, in the abstract, whether Model 2 would be of real benefit to parsing the French Treebank. First of all, as noted in [Collins, 1999], there is a real overlap between the distance measure and the subcategorization probabilities.

e.g

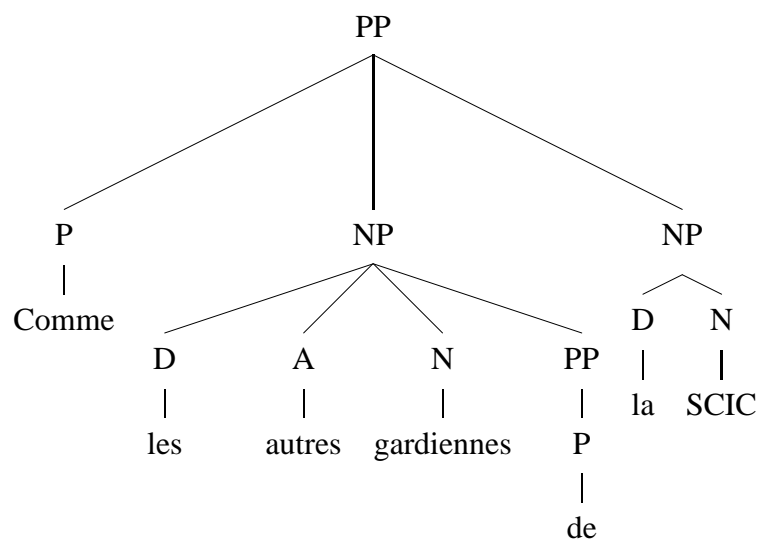


Figure 4.13: Distance vs Subcategorization

For the example in Fig 4.13, the probability frame for the preposition will be $\langle PP, P, de, \{\} \rangle$ and will have a low probability since PPs in training data will almost always be seen with one complement, and very rarely with 0 or 2 complements. Collins goes on to mention that a model with distance features but without subcategorization frames performs as well as a model that includes both.

Moreover, given the annotation scheme of the FTB, argument identification is more problematic. One of the rules that Collins implements, makes use of the semantic tags that are available in the PTB to block nonterminals marked with certain of these tags to be identified as arguments.

e.g:

Last week IBM bought Lotus.

In this sentence, while the NP "Last week" is marked with the TMP (temporal) tag and would therefore not be identified as a complement, the NP "IBM", which is not marked with any semantic tag, would be identified as one (the subject of the VP "bought Lotus").

As mentioned before, we have a rule that marks the closest NP to the head of a SENT as a complement. Hence the NP *Un certain parfum de krach* is marked as a complement in the sentence:

Ce jour là, un certain parfum de krach a même régné dans les salles de marché

If the positions of the first 2 NPs are swapped, to create the perfectly valid sentence:

- (7) Un certain parfum de krach, ce jour là, a même régné dans les salles de marché
 A certain perfume of krash, that day, was even ruling in the rooms of market
 A certain smell of krach, that day, was even present in the market rooms

we would get the marking wrong.

Very frequently, VNs include a clitic pronoun which is the subject of the verb it is attached to.

e.g the subject NP *un certain parfum* can be moved to the end of the sentence and replaced by the clitic pronoun *Il* as left-adjacent to the head verb to give the sentence:

- (8) Ce jour là, il a même régné dans les salles de marché un certain parfum de krach
 That day, he was even ruling in the rooms of market a certain smell of crash
 That day, there was even present in the market rooms a certain smell of crash

In this construction, the modifier NP (*Ce jour là*) would be marked as the subject and the VN would be *il a même régné*.

The presence of the clitic in the VN makes subject identification that much harder.

Additionally, since there are no VP → V NP or VP → V NP PP rules, a systematic

object identification becomes unfeasible. The constituent following a VN can be an object (the PP *dans les boîtes à lettres* in *Je mets des petits mots très gentils dans les boîtes à lettres*) or could just be an adjunct (the PP *dans les salles de marche* in the sentence above).

If we want the model to learn that the verb *mettre* can be ditransitive and subcategorize for a NP and a PP, we will have to identify the correct arguments in the training data. This will not be always possible as shown in the previous paragraph.

4.5.3 Intricacies of Collins' Parsing Models

The previous subsections described the basic framework of Collins' parsing models. However, his models make provision for a few special cases. We believe that it is crucial to look into these special cases closely since this is where Collins brings English/Penn Treebank specific linguistic knowledge into play. A thorough understanding of these provisions would help us incorporate French Treebank motivated changes in our models.

4.5.3.1 Base NPs

Collins implements a different probability model for Base NPs. These are defined as NPs that do not directly dominate an NP themselves, unless that NP is a possessive NP [Collins, 1999]. The base NPs are labelled as NPB in the model.

The main reasons for special treatment of NPs are:

1. The boundaries of base NPs are often strongly marked: particularly the start points, which are often marked with a determiner.
2. The annotation standard in the Penn Treebank leaves the internal structures of base NPs very flat and underspecified. E.g., both *pet food volume* (where *pet* modifies *food* and *food* modifies *volume*) and *vanilla ice cream* (where both *vanilla* and *ice* modify *cream*) have the structure **NPB** → **NN NN NN**. Because

of this, there's no reason to believe that modifiers within NPBs are dependent on the head rather than the previous modifier.

Hence, the independence assumptions are different when the parent non-terminal is an **NPB**.

Specifically

$$\begin{aligned} P_l(L_i(l_i) \mid H, P, h, L_1(l_1) \dots L_{i-1}(l_{i-1})) &= P_l(L_i(l_i) \mid P, L_{i-1}(l_{i-1})) \\ P_r(R_i(r_i) \mid H, P, h, R_1(r_1) \dots R_{i-1}(r_{i-1})) &= P_r(R_i(r_i) \mid P, R_{i-1}(r_{i-1})) \end{aligned} \quad (4.17)$$

The distance variable is dropped since the modifier and the previous-modifier non-terminals are always adjacent. In this model, $L_0(l_0)$ and $R_0(r_0)$ are defined to be $H(h)$.

With this conditioning, the model is able to learn that the STOP symbol is very likely to follow a determiner, generating leftwards from the head noun.

NPs are annotated quite similarly in the French Treebank too (apart from the fact that there's no possessive tag) (see Figure 4.14).

- (9) un curieux sentiment
 a strange feeling
 a strange feeling

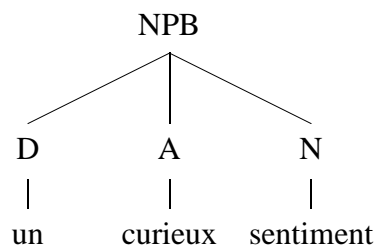


Figure 4.14: A base NP

Moreover, while in English, adjectival modifiers are pre-nominal, in French they can either be pre-nominal or post-nominal. When in a post-nominal position, the adjective is a unary projection of an adjectival phrase.

e.g.,

- (10) la gestion locale
 the management local
 the local management

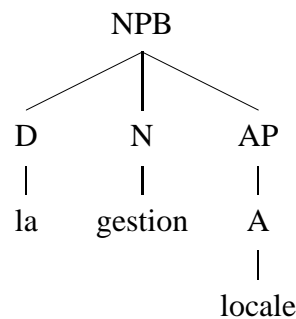


Figure 4.15: Base NP with a post-nominal adjective

In this case as well (see Figure 4.15), the model will be able to learn that a STOP symbol is very likely to follow an adjectival phrase, generating rightwards from the head noun.

4.5.3.2 Coordination and punctuation

Coordination and punctuation are two other examples where the independence assumptions in the basic model.

Given a typical 'raised' tree like in Figure 4.16,

- (11) sportifs et culturels
 sports and cultural
 sports and cultural

which has the rule: $AP(\text{sportifs}) \rightarrow A(\text{sportifs}) C(\text{et}) AP(\text{culturels})$,

the probability is

$$\begin{aligned}
 &P_h(A|AP, \text{sportifs}) \times P_l(STOP | A, AP, \text{sportifs}) \times P_r(C(\text{et}) | A, AP, \text{sportifs}) \times \\
 &P_r(AP(\text{culturels}) | A, AP, \text{sportifs}) \times P_r(STOP | A, AP, \text{sportifs})
 \end{aligned}
 \tag{4.18}$$

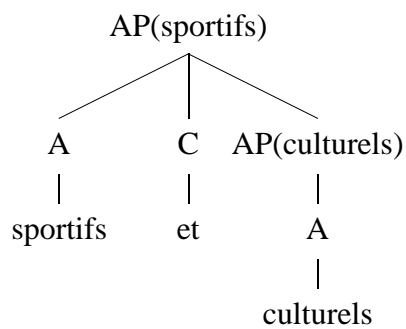


Figure 4.16: A raised tree

The independence assumptions mean that the model fails to learn the rules that there is always exactly one phrase following the coordinator (**C**). The basic probability models will give much too high probability to unlikely phrases such as

$AP \rightarrow A C$ or

$AP \rightarrow A C AP AP$

As per [Collins, 1999], punctuation at the beginning and end of sentences is removed altogether from the training/test data. All punctuation items apart from those tagged as comma or colon are also removed. Any remaining punctuations are raised as high as possible in the parse tree so that punctuation can only appear between two non-terminals, as opposed to appearing at the end of a phrase. This means, that in some sense, punctuation acts very much like a coordinating conjunction, in that it "conjoins" the two siblings between which it sits. Observing that it might be helpful for conjunctions to be generated conditioning on both their conjuncts, Collins introduced two new parameter classes, P_{punc} and P_{CC} .

However, as Dan Bikel mentions in [Bikel, 2004], Collins' implementation of these parameters causes the model to be inconsistent (see [Bikel, 2004] for more details). Bikel, therefore, re-implements these parameter classes using a mechanism that does not yield an inconsistent model.

The main change he makes is to treat punctuation and coordination pre-terminals as first-class objects that are generated just like any other modifying nonterminals, so that there is no longer need for the 2 parameter classes introduced by Collins.

In Bikel's model, the generation of a modifying non-terminal is described by:

$$P_M(M(t)_i | P, H, w_h, t_h, subcat_{side}, vi(M_i), \delta M_{i-1}, side) \quad (4.19)$$

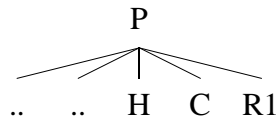
where M_i is some modifier L_i or R_i , w_h is the head word, t_h is the head POS tag, $side$ is a boolean that indicates whether the modifier is on the left or right side of the head and $subcat_{side}$ is the subcategorization frame on that side.

The distance metric is altered to consist solely of the verb-intervening (vi) predicate. A mapped version of the previously generated modifier is added to the conditioning context, according to the following mapping function:

$$\delta(M_i) = \left\{ \begin{array}{lll} +START+ & \text{if} & i = 0 \\ CC & \text{if} & M_i = C \\ +PUNC+ & \text{if} & M_i = , \text{ or } : \\ +OTHER+ & \text{otherwise} & \end{array} \right\} \quad (4.20)$$

Note that the $i = 0$ condition incorporates the "no intervening" component of Collins' distance metric.

So, given a tree like



the conjunction of some node R_1 with a head H and a parent P is:

$$\hat{p}_H(H | P) \times \hat{p}_R(C | P, H, +START+) \times \hat{p}_{R1}(H | P, H, C) \quad (4.21)$$

Let's illustrate it fully using a real example from the corpus:

- (12) Il y a une façon de les prendre, et ça marche
 He there is a way to them take, and it works
 There is a way to take them, and it works

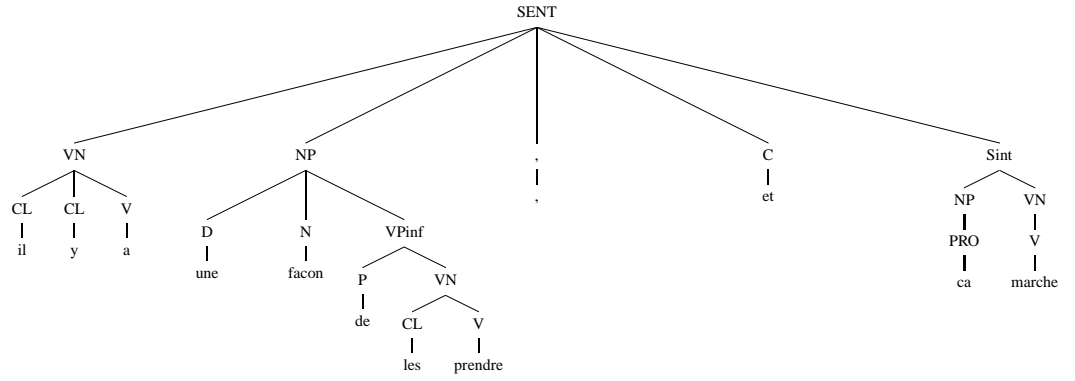


Figure 4.17: Conjunction example

The probability of generating the five children is now:

$$\begin{aligned}
 & \hat{p}_H(VN \mid SENT, a) \times \\
 & \hat{p}_M(NP(facon, N) \mid SENT, VN, a, V, , false, +START+, right) \times \\
 & \hat{p}_M(, (,) \mid SENT, VN, a, V, , true, +OTHER+, right) \times \\
 & \hat{p}_M(C(et, C) \mid SENT, VN, a, V, , true, +PUNC+, right) \times \\
 & \hat{p}_M(Sint(marche, V) \mid SENT, VN, a, V, , true, C, right) \times
 \end{aligned} \tag{4.22}$$

4.5.4 Smoothing

The parameter classes in Collins' models define conditional probabilities with very large conditioning contexts. This causes huge data sparsity issues. Collins uses the technique of *deleted interpolation*, which smooths the distributions based on full contexts with coarser models that use less of the context, by successively deleting elements from the context at each back-off level.

For example, the head parameter class smooths $P_{H_0}(H \mid P, w_h, t_h)$ with $P_{H_1}(H \mid P, t_h)$ and $P_{H_2}(H \mid P)$

For some conditional probability $p(A \mid B)$, we denote the reduced context at the i th back-off level $\phi_i(B)$ where typically $\phi_0(B) = B$. Each estimate in the back-off chain is computed using maximum likelihood estimation, and the overall smoothed estimate is computed using $n - 1$ smoothing weights for n back-off levels, denoted $\lambda_0, \dots, \lambda_{n-2}$.

These weights are used in a recursive fashion; the smoothed version $\tilde{e}_i = \tilde{p}_i(A|\phi_i(B))$ of an unsmoothed ML estimate $e_i = \hat{p}_i(A|\phi_i(B))$ at back-off level is i computed via the formula

$$\tilde{e}_i = \lambda_i e_i + (1 - \lambda_i) \tilde{e}_{i+1}, 0 \leq i \leq n-1, \tilde{e}_{n-1} = e_{n-1} \quad (4.23)$$

So, for example, with three levels of back-off, the overall smoothed estimate would be

$$\tilde{e}_0 = \lambda_0 e_0 + (1 - \lambda_0) [\lambda_1 e_1 + (1 - \lambda_1) e_2] \quad (4.24)$$

The formula for the smoothing weight is:

$$\lambda_i = \frac{c_i}{c_i + 5u_i} \quad (4.25)$$

where c_i is the count of the history context $\phi_i(B)$ and u_i is the *diversity* of that context, which is equal to the number of unique futures observed in training for that history context. The multiplicative constant 5, called the *smoothing factor* was optimised by looking at overall parsing performance on the development test set, Section 00 of the Penn Treebank. Unfortunately, we could not do likewise on our dev set, and had to use the same PTB optimised value.

A complete list of all parameter classes, along with their back-off structures, used in the implementation of Bikel's parsing model is presented in Appendix D.

4.5.5 Part-of-speech tagging

Part of speech tags are generated along with the words in the models, so tagging is fully integrated. In effect, all possible tag sequences are considered. The parser can be run in two modes:

1. POS tags supplied.

The supplied tags are only used for words that have not previously been seen in training (truly unknown words). For seen words, the parser uses all all possible tags that have been seen in training with that word.

2. POS tags not supplied.

Instead of using an outside POS tagger, we forced the parser to do its own POS tagging. To do so, Bikel's parser requires the implementation of a `WordFeatures` Java class. This is inspired by the unknown-word model of [Weischedel et al., 1993].

The `WordFeatures` Java class provides a mapping of lexical items to orthographic/morphological word feature-vectors to reduce ambiguity during part-of-speech tagging of unknown words, where unknown words refer to words occurring 5 or fewer times in the training data. This threshold was optimised on the development set.

The features implemented (capitalisation, hyphenation, inflection, derivation and compound) were also optimised on the development set. They are explained in further details in Appendix C.

During training, feature vectors are computed for every word appearing in the training data and every *word - feature vector - POS tag* triple is stored in a look-up table.

During decoding, if the number of occurrences in the training data set of the word being parsed is more than 5, POS tag look-up is done using the word itself as the key. Otherwise, the feature vector is used to search the table.

4.5.6 Parsing

This is performed via a probabilistic version of the CKY chart-parsing algorithm. As with normal CKY, even though the model is defined in a top-down, generative manner, decoding proceeds bottom-up.

Syntactic category	FTB	PTB	Negra
VP	-	2.32	2.59
Sint	3.44	-	-
Ssub	4.41	-	-
AdP	2.24	-	-
VPinf	3.07	-	-
VPpart	2.51	-	-
AP	1.34	-	-
PP	2.10	2.03	3.08
Srel	3.92	-	-
VN	1.76	-	-
NP	2.45	2.20	3.08
SENT	5.84	2.22	4.22

Table 4.2: Average number of child nodes per Treebank per Constituent

To speed up decoding, the algorithm implements beam pruning: the chart memorises the highest-scoring theory in each span, and if a proposed chart item for that span is not within a certain factor of the top-scoring item, it is not added to the chart. 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.

4.5.7 Varying the order of the Markov assumption

We have mentioned a few times now that the French Treebank trees are quite flat. Table 4.2 shows the average number of children for each syntactic category in the FTB and compares them to the figures available for Negra and PTB ([Dubey and Keller, 2003]).

The degree of flatness of French Treebank, in the case of NPs and PPs is in between that of the Penn Treebank and that of Negra, while the absence of VPs explains the very high level of flatness at the **SENT**, **Sint**, **Ssub**, **Srel**, **VPinf** and **VPpart** node level. [Collins, 1999] acknowledges that:

Model	LR	LP	CBs	0 CB	≤ 2 CBs
Collins - NoBaseNP	67.91	66.07	0.73	65.67	89.52
Sister-head NP	67.84	65.96	0.75	65.85	88.97
Sister-head PP	70.27	68.45	0.69	66.27	90.33
Sister-head all	71.32	70.93	0.61	69.53	91.72

Table 4.3: Parsing results for Negra - 2

The models developed in chapter 7 have tacitly assumed the Penn-treebank style of annotation, and will perform badly given other representations...

1. To counteract the flatness of Negra, [Dubey and Keller, 2003] implement an alternative parsing model whereby Collins' baseNP model is extended to the generation of nonterminal modifiers of **all syntactic categories**.

$$\begin{aligned}
 P_l(L_i(l_i) \mid H, P, h, L_1(l_1) \dots L_{i-1}(l_{i-1})) &= P_l(L_i(l_i) \mid P, L_{i-1}(l_{i-1})) \\
 P_r(R_i(r_i) \mid H, P, h, R_1(r_1) \dots R_{i-1}(r_{i-1})) &= P_r(R_i(r_i) \mid P, R_{i-1}(r_{i-1}))
 \end{aligned}
 \tag{4.26}$$

They call this model as capturing **sister-head relationships**, and argue that it implicitly adds binary branching to the grammar.

Results of their work on Negra is shown in Table 4.3:

2. While we followed [Dubey and Keller, 2003]'s work by assessing the impact of a sister-head relationship model on the FTB, we also investigated other ways of modifying our probability model.

Recall that in Bikel's implementation of the Collins parser, the maximal context for the modifying non-terminal parameter class is defined as:

$$P_M(M(t)_i \mid P, H, w_h, t_h, subcat_{side}, vi(M_i), \delta M_{i-1}, side)
 \tag{4.27}$$

where

$$\delta(M_i) = \left\{ \begin{array}{lll} +START+ & \textit{if} & i = 0 \\ CC & \textit{if} & M_i = C \\ +PUNC+ & \textit{if} & M_i = , \textit{or} : \\ +OTHER+ & \textit{otherwise} & \end{array} \right\} \quad (4.28)$$

We decided to include the previous modifier as well in the conditioning context. This is similar to the BBN model presented in [Bikel and Chiang, 2000] with the difference being that in our model we also include subcat frames and the distance feature. [Bikel and Chiang, 2000] also call this model as a "bigram of nonterminals" model, since it looks a lot like a bigram language model. We will refer to this model as the **bigram model**.

The model is implemented simply by altering the definition of the δ function, which now looks like:

$$\delta(M_i) = \left\{ \begin{array}{lll} +START+ & \textit{if} & i = 0 \\ M_i & \textit{otherwise} & \end{array} \right\} \quad (4.29)$$

This model, in effect, implements a 1st order Markov assumption and has been used by [Collins et al., 1999] for parsing the Prague Dependency Treebank. The intuition behind this approach is that we hope e.g., that the model will learn that the STOP symbol is more likely to follow 'heavy' phrases such as Srel, Ssub, Sinf.

3. The third model we looked at, implements the "bigram of nonterminals" model only for categories with high degrees of flatness ("SENT", "Srel", "Ssub", "Sint", "VPinf", "VPpart").

4.5.8 Modification to subcategorization frames

We mentioned earlier that given the FTB annotation scheme, subject identification is hard. More specifically, given the phrase below

(NP Ce jour là), (VN (il a même régné)) dans les salles de marché un certain parfum de krach

the training algorithm will mark the temporal NP modifier *Ce jour là* as the subject of the sentence, whereas the actual subject, the pronominal clitic *Il* is inside the VN construct.

This is a problem for the subcategorization probabilities of model 2 - as the probability of having the construct whereby the clitic inside the VN is the subject of the sentence, $P_{lc}(\{\} | SENT, VN, verb)$ will be low.

We attempt to solve this problem by marking VNs containing CLs with a new nonterminal tag - VNG. NPs appearing before VNGs in a sentence are no longer marked as arguments.

The resulting model will have a cleaner division of subcategorization:

$$\begin{aligned}
 P_{lc}(\{\} | SENT, VNG, verb) &\approx 1 \\
 P_{lc}(\{NP\} | SENT, VNG, verb) &\approx 0 \\
 P_{lc}(\{\} | SENT, VN, verb) &\approx 0 \\
 P_{lc}(\{NP\} | SENT, VN, verb) &\approx 1
 \end{aligned}$$

Chapter 5

Results

5.1 PARSEVAL measures

All experiments were run using the first 8552 sentences of the FTB as training set, the following 1000 sentences as development set and the final 1000 sentences as the test set. The average sentence length of the test set is 21 words. The development set was used to tune the parameters of the model and the test set was only used once all the parameters had been set. All results reported in this chapter were obtained on the test set, unless stated otherwise.

As common practice in parsing literature, we use the standard PARSEVAL measures [Black et al., 1991] to compare parsing performance.

We report on 9 main categories :

1. **Labelled Precision (LP)** = $\frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in proposed parse}}$
2. **Labelled Recall (LR)** = $\frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in treebank parse}}$
3. **F-Score** = $\frac{2 \times LP \times LR}{LP + LR}$ This is the harmonic mean of LP and LR.
4. **Average Crossed Brackets (CBs)** where **Crossed Brackets** = number of constituents which violate constituent boundaries with a constituent in the treebank parse

Token	Punctuation tag	Other tags	Count
-	:	ADV	33
-	:	P	12
/	:	C	1
/	:	P	6

5. **0 CBs** = percentage of sentences with 0 crossing brackets.
6. **≤ 2 CBs** = percentage of sentences with less or equal to 2 crossing brackets.
7. **Complete match** = percentage of sentences where precision and recall are both 100
8. **Coverage** = percentage of sentences that parse.
9. **Tagging accuracy** = percentage of correct POS tags.

For a constituent to be correct, it must span the same set of words and have the same label as a constituent on the treebank parse. In the case of the Clitical Verbal Nucleus marked model, the labels **VN** and **VNG** are considered equivalent.

All punctuation items i.e words tagged as comma, colon, double quotes, period, left round bracket and right rounded bracket are ignored for evaluation purposes. This is consistent with [Collins, 1999]. However, there exists 2 punctuation tokens in the dataset that sometimes appear tagged as a non-punctuation item (see Table 5.1).

Since punctuation tags are ignored during the PARSEVAL measures calculation, a situation where a token is tagged as a punctuation in one file and as a non-punctuation item in the other file, will lead to a length mismatch. In such cases, all the PARSEVAL scores of that sentence are set to 0.

Parsing results are shown in the tables 5.1 and 5.2. Note that for models where coordination has not been raised (BitPar - Comp Expanded, BitPar - Comp Contracted and Collins Model 1 - see tables below), both the parsed outputs and the gold standard files are first converted by applying the raising coordination transformation and evaluation is subsequently performed. This ensures that the different results across experiments

are comparable.

However, in the case of the expanded compound vs the contracted compound models, we do not have comparable results, since the expanded compound model has many more brackets than the contracted model. We tried collapsing the compounds for evaluation purposes, e.g.

converting,

(PCmp (P d') (P entre))

to

(P d'_entre)

This approach can only work, if we are certain that the model is tagging the right words as compounds. Unfortunately, this is very rarely the case.

e.g

the model has the following output :

(NCmp (N jours) (N commerçants))

while in the gold standard file, "jours" and "commerçants" are two distinct NPs. Collapsing the compounds will therefore lead to a huge number of sentences with length mismatches, such that evaluation becomes pointless.

All tests were performed on the contracted compound format of the dataset unless stated otherwise. Results are reported both for the standard ≤ 40 and ≤ 100 sentence length thresholds.

5.2 Results - I

The expanded compound models fare very badly. This is due to the explosion in grammar rules (11704 rules in the expanded model compared to 10299 rules in the contracted model) and tagset (24 labels in expanded vs 11 in contracted model) associated

with that transformation. Additionally, as we saw earlier with ((*NCmp (N jours) (N commerçants)*)), compound identification becomes a very arduous task without proper lexical cues.

Given that the baseline model is a vanilla PCFG that derives its parameters by simple MLE, sparsity of data is a major issue. The unlexicalised model with contracted compounds boost performance by about 5%. Raising coordination improves accuracy by about 1% across the models.

Based on our results, we can assert that lexicalisation definitely helps parsing performance for French. The increase in accuracy between the unlexicalised model and the head lexicalised model is approximately 14.5%. This is consistent with what has been reported for English on the PTB by [Charniak, 1997].

While adding complement/adjunct distinction and probability over subcategorization frames to the model improves parsing performance, the increase was not found to be statistically significant using a Chi-square test. This confirms our suspicions that the annotation scheme of the FTB does not lend itself particularly well to the demands of Model 2. Moreover, as mentioned in [Collins, 1999], some of the benefits of Model 2 are already captured by inclusion of the distance measure.

The improvement using the sister-head model as well proved not to be statistically significant. Applying the bigram model to those labels that have been found to have a high degree of flatness, actually decreases accuracy. However, extending the model to all constituents gives a statistically significant improvement over the Collins Model 1 inspired model. A chi-square test confirmed that the bigram-all model performed significantly better than model 1 (**recall** chi-square = 3.91, DoF = 1, $p \leq 0.048$, and **precision** chi-square = 3.97, DoF = 1, $p \leq 0.046$). Adding the Clitic Verbal Nucleus identification, pushes the F-Score to 81.03 but here again, the improvement is not statistically significant.

The improved performance with the bigram-all model is consistent with the findings of [Collins et al., 1999] for Czech, where the model upped accuracy by about 0.9%, as well as for English where [Charniak, 1999] reports an increase in f-score of approxi-

mately 0.3%.

On the other hand, [Bikel and Chiang, 2000] report that for Chinese, the bigram model performs significantly lower than a model based on stochastic Tree-Adjoining Grammars (TAG) [Chiang, 2000], but that the poor performance could be due to poor POS tagging.

5.3 Results - II

The models in 5.2 implemented their own POS tagging. Tagging accuracy was in the 91 to 93% range for BitPar and around 96% for the word-feature enhanced tagging model of the Bikel parser. POS tags are an important cue for parsing as they provide the model with a level of generalisation as to how classes of words tend to behave, what roles they play in sentences, and what other classes they combine with [Collins et al., 1999]. 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 uses the tags provided, the Bikel parser only uses them for words that appear fewer times than the unknown word threshold (6).

Perfect tagging increases parsing performance in the BitPar models by around 3% (while tagging accuracy goes up by 7-9%) (see tables 5.3 and 5.4). This is quite considerable and shows that poor POS tagging is one of the reasons of low performance. Its impact is less drastic on the Bikel parser (around 1% increase in F-Score) given that tagging accuracy is increasing by only around 2%.

5.4 Data sparsity

While the parsing performance of our best model hovers around a very promising 80% level, it still falls well short of the best performances obtained for English on the PTB (around 90%, [Collins, 1997, Charniak, 1997, Charniak, 1999]). This could be due to lack of sufficient training data; our FTB training data set contains 8,552 sentences,

much less than the 40,000 sentences available in the Penn Treebank, which could lead to data sparsity issues.

To verify this hypothesis, we computed learning curves for the unlexicalised model with raised coordination and contracted compounds and the model emulating Collins Model 2 (both with their own POS tagging mechanism).

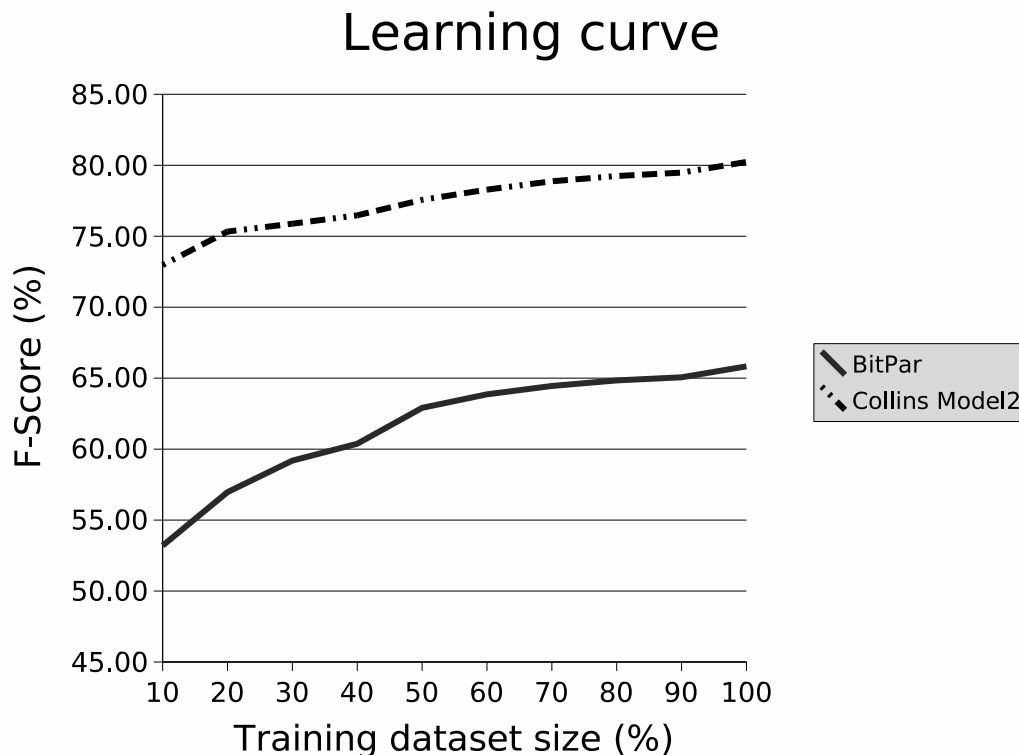


Figure 5.1: Learning curve

The result (5.1) shows that there is no evidence for sparse data. For both models, a fairly high f-score is obtained using only 10% of the training data. Increasing the training dataset size leads to a slow increase in performance. While the Collins model learning curve shows distinct signs of flattening out as we reach the 100% mark, this is less pronounced in the BitPar model. These findings are consistent with what [Dubey and Keller, 2003] report.

The Collins model appears more impervious to training set size. This is probably due

to the way in which its rule probabilities are computed which is by decomposing larger rules through Markov chains.

5.5 Comparison with PTB

We take advantage of Bikel's emulation of Collins' Model 2 for English on the PTB to verify parsing performance on the latter on training and testing sets of the same size as our dataset. We select the sentences from section 02,03,04,05 and 08 (total of 8345 sentences) to form the training set and the first 1000 sentences of section 23 to form our test set.

In this emulation, for both models, words appearing less than 6 times (the unknown word threshold) are mapped to a single token +UNKNOWN+. During parsing, for all 'unknown' words, all the POS tags associated with the +UNKNOWN+ token are used to seed the chart.

This gives us a uniform tagging method to make the models comparable.

The results are shown in tables 5.5 and 5.6.

Even with comparable dataset size, parsing performance on the PTB is superior to that on the FTB - almost 8% higher on f-score and 1.5% on exact match. This is to be expected since the Bikel's parser has been optimised for the Penn Treebank annotation scheme. [Collins et al., 1999] mentions that the Wall Street Journal may be an easy domain since a reasonable proportion of sentences come from a sub-domain, financial news, which is relatively restricted. The *Le Monde* corpus on the other hand consists of a wide variety of domains (economy, literature, politics etc.).

It is interesting to note that as far as crossing brackets measures (Average, $0 \leq 2$) are concerned, the FTB fares considerably better than the PTB. This is consistent with the fact that trees in the FTB are flatter than their counterpart in the PTB, so that there are fewer brackets to be predicted and thus fewer crossing brackets. To emphasise this point, we note that [Dubey and Keller, 2003]'s best parsing model for Negra, an even

flatter treebank, has a best average crossing bracket figure of 0.61, much lower than the FTB (0.78) and the PTB (0.90) [Collins, 1999].

5.6 Error analysis

Error analysis is an especially arduous task in the realm of parsing. Unlike classification tasks where the output domain is restricted; in parsing, the domain of investigation is exponentially big so that finding patterns of behaviour is hard. In this section, we will try to illustrate how the failings of the models we implemented and how we tried correcting these failings.

5.6.1 Unlexicalised vs Lexicalised

As discussed in 4.4, vanilla PCFGs have always been criticised for their lack of lexical sensitivity. This especially comes to the fore when dealing with PP-attachments, where it is a well studied fact that lexical information plays an important part in selecting the correct parsing of an ambiguous prepositional-phrase attachment ([Hindle and Rooth, 1991]).

The sentence

- (1) L'Aérospatiale met 7_000 salariés en chômage partiel.
 The Aérospatiale puts 7_000 salaried in unemployment partial
 The Aérospatiale puts 7_000 salaried workers on part time employment

is an example where BitPar gets it wrong but Model 1 gets it right (see figures 5.2 and 5.3).

The PCFG attaches the PP headed by *en* to the NP. However, the correct attachment is at the sentence level, because the verb *mettre* subcategorizes for a location, which can be expressed with the preposition *en*. This lexical dependency fact is only captured by the lexicalised model.

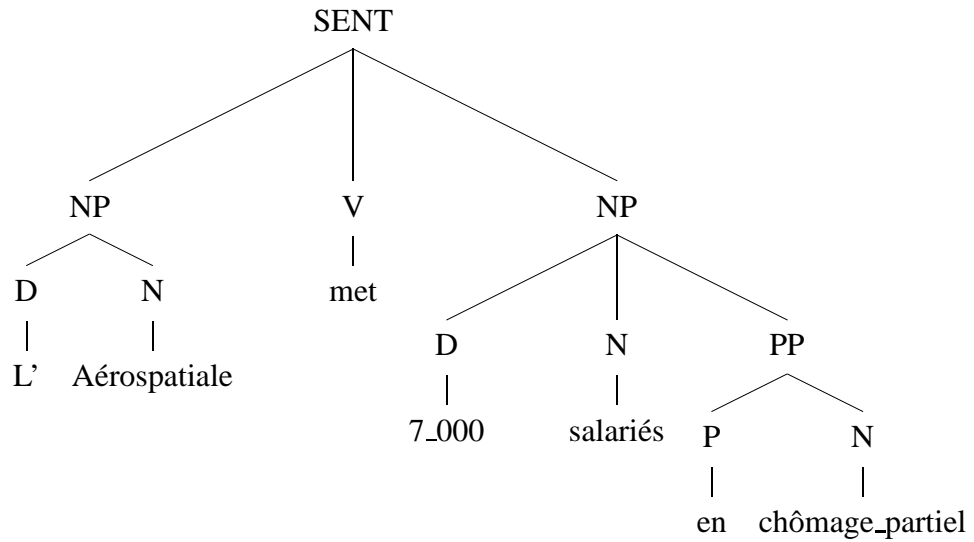


Figure 5.2: Wrong PP-attachment analysis in BitPar

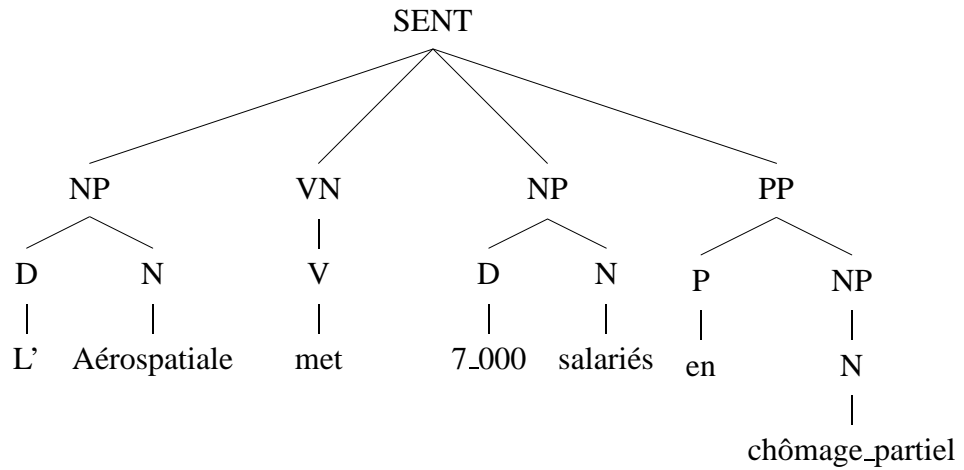


Figure 5.3: Correct PP-attachment analysis in Model 1 emulation

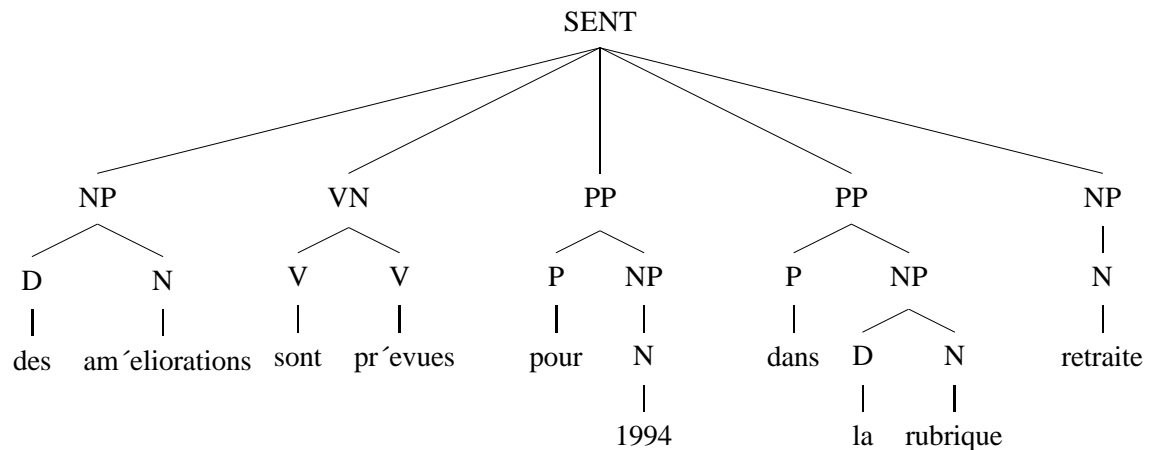


Figure 5.4: Wrong parse analysis in Model 2 emulation

5.6.2 Model 2 vs Bigram model

We skip the analysis of Model 1 vs Model 2 given that they are not statistically significant. Instead we compare Model 2 to the bigram-all model. The conditioning of the latter would suggest it to be more sensitive towards flatter structures.

The sentence,

- (2) des améliorations sont prévues pour 1994 dans la rubrique retraite
 some improvements are expected for 1994 in the sector retirement
 some improvements are expected in the retirement sector in 1994

is an example where the bigram model gets it right (see figures 5.4 and 5.5)

While in Model 2, *retraite* is the unary child of an NP and is a sister of the preceding PP, in the correct parse of the bigram model, it gets attached to the NP child of the PP.

We suspect that the bigram model learns that the probability of an NP following a PP in a sentence headed by a VN,

$P(NP(N, retraite) \mid SENT, VN, prévues, V, PP)$ is lower than

$P(+STOP+ \mid SENT, VN, prévues, V, PP)$ the probability of a +STOP+ following a PP.

The two sentences below,

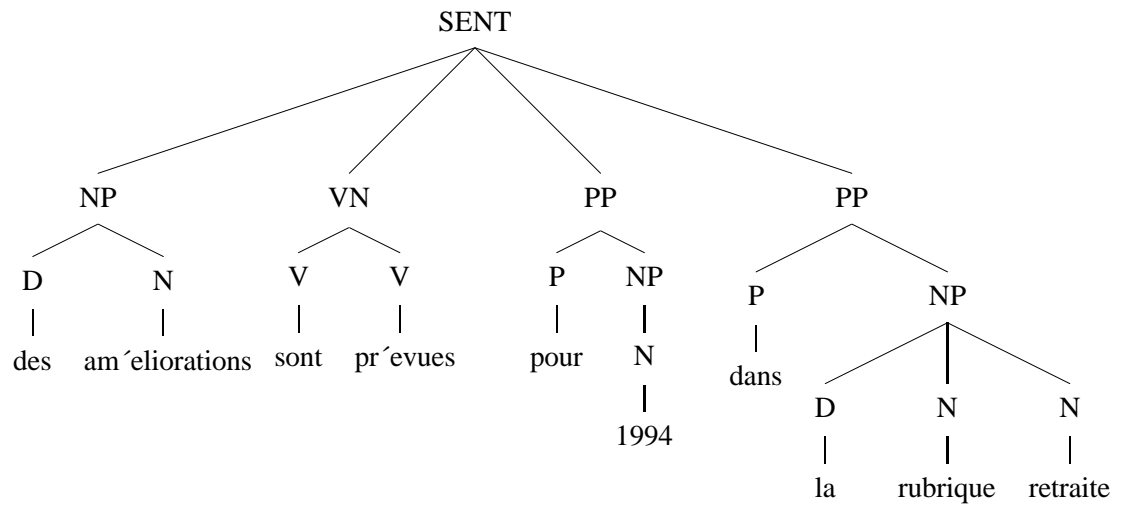


Figure 5.5: Correct parse analysis in bigram model

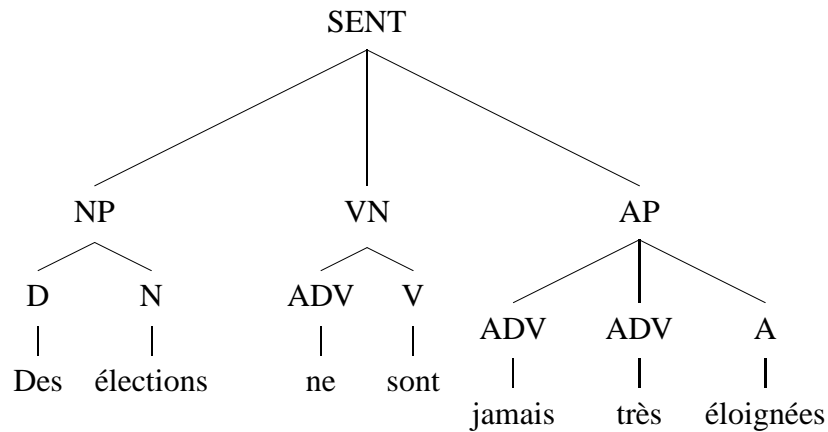


Figure 5.6: Model 2 not sensitive to flat structure - 1

(3) Des élections ne sont jamais très éloignées
 Some elections not are never very far
 Elections are never far too away

(4) Les conséquences sur l'embauche sont alors nulles
 The consequences on the employment are therefore nil
 The consequences on employment are therefore nil

are examples where the bigram model is more sensitive to a flatter structure (see figures 5.6,5.7, 5.8 and 5.9)

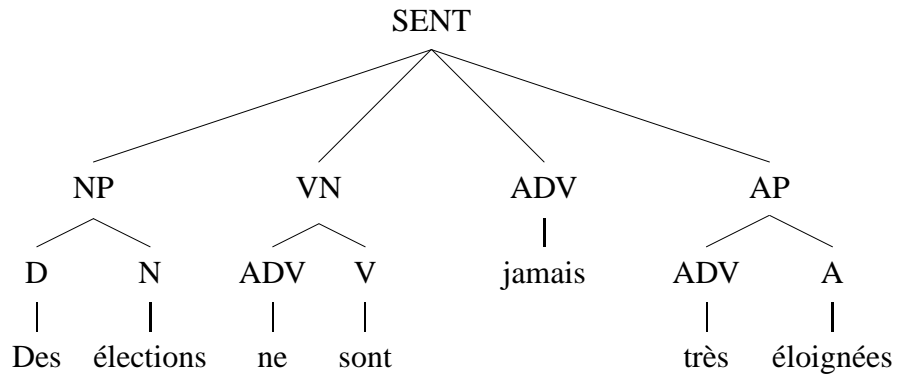


Figure 5.7: Bigram model sensitive to flat structure - 1

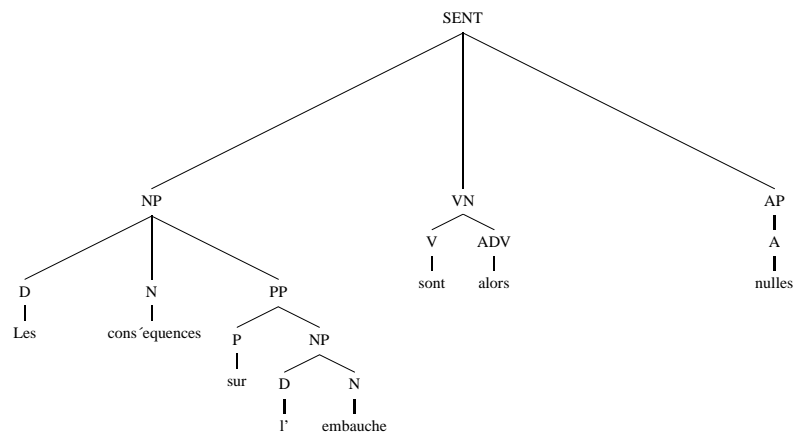


Figure 5.8: Model 2 not sensitive to flat structure - 2

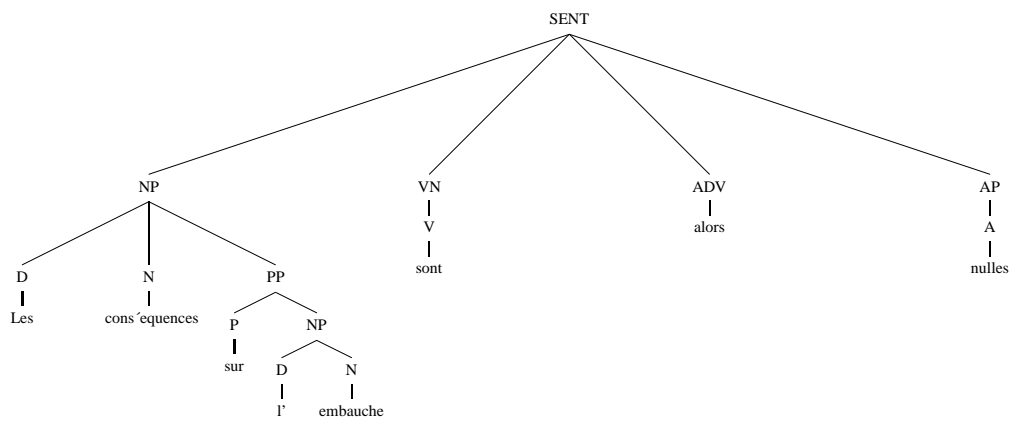


Figure 5.9: Bigram model sensitive to flat structure - 2

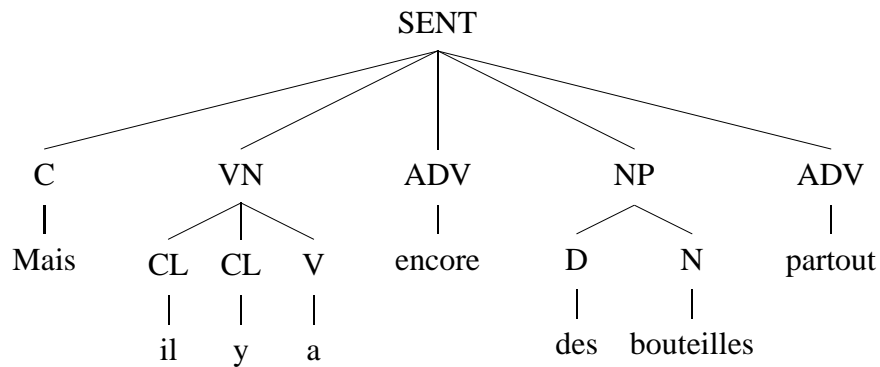


Figure 5.10: Model 2 sensitive to flat structure

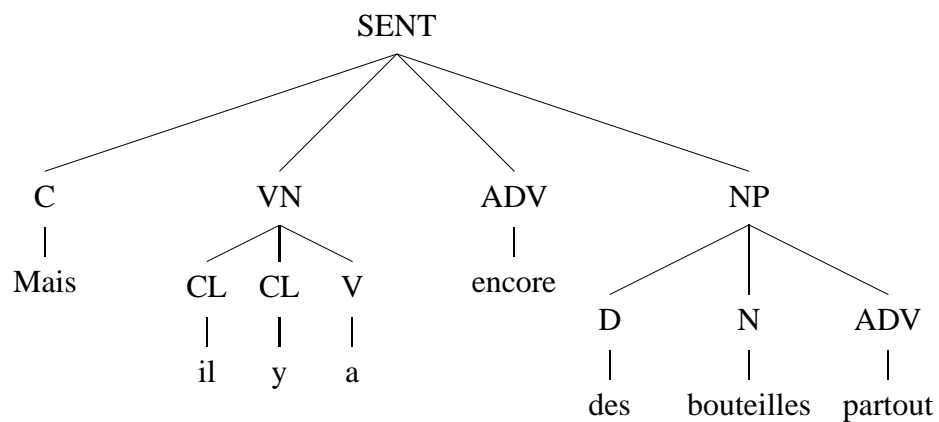


Figure 5.11: Bigram model not sensitive to flat structure

However, there are cases where the bigram model performs worse than Model 2:

- (5) Mais il y a encore des bouteilles partout
 But he there is still some bottles everywhere
 But there are still bottles everywhere
- (6) Il n'est pas dans mes habitudes de me chercher des excuses
 He not is not in my habits to me find some excuses
 It is not in my nature to make excuses for myself
- (7) est-on en état de dire ce qu'il convient de faire
 are-we in state to say what is suits to do
 Are we able to say what should be done

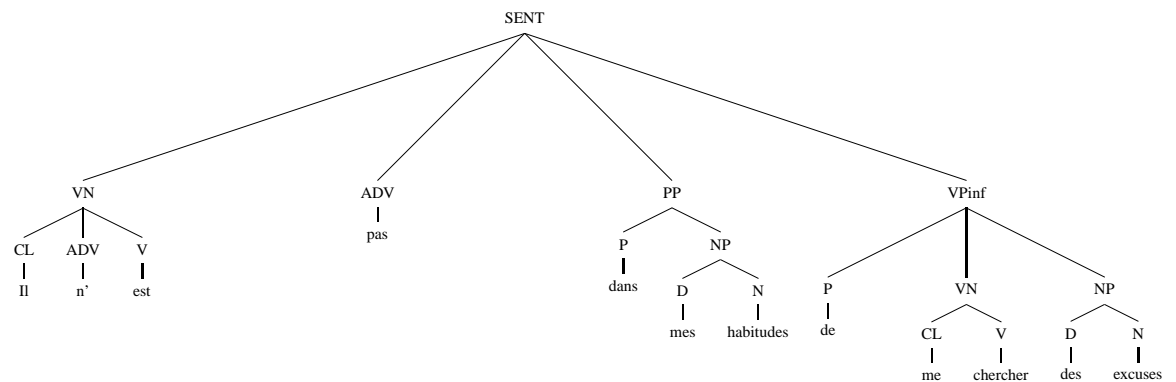


Figure 5.12: Correct VPinf attachment in Model 2 -1

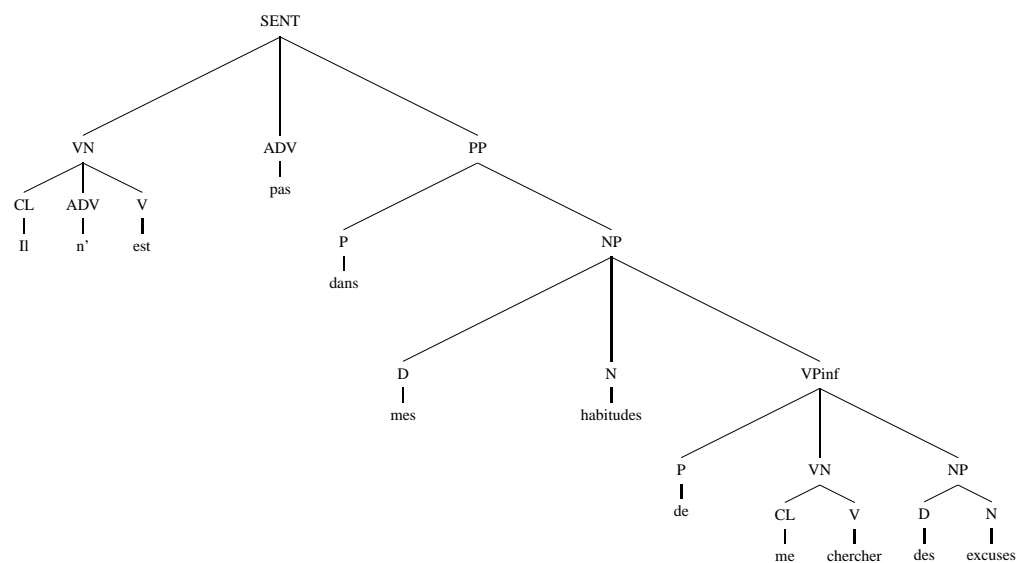


Figure 5.13: Wrong VPinf attachment in bigram model -1

In this case, it seems like the model is trained to expect the likelihood of a +STOP+ being generated after a NP more likely than after an ADV; thus, getting the tree structure wrong (figures 5.10 and 5.11).

The parse trees in figures 5.12, 5.13, 5.14 and 5.15 highlight the tendency of the bigram model of getting VPInf attachments wrong. While Model 2 correctly places the VPInfs as the last child of the sentence, the bigram model attaches them to the NP which is the child of the last PP.

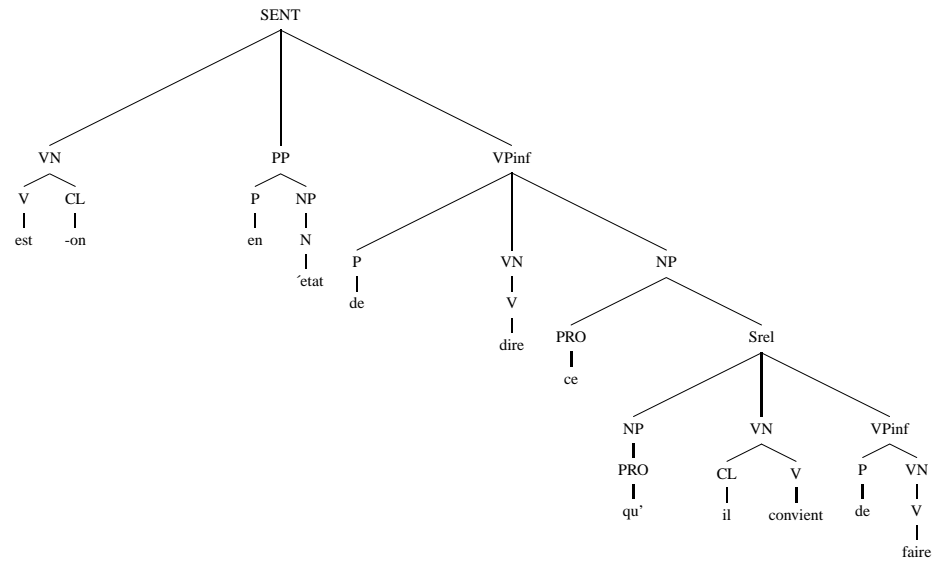


Figure 5.14: Correct VPinf attachment in Model 2 -2

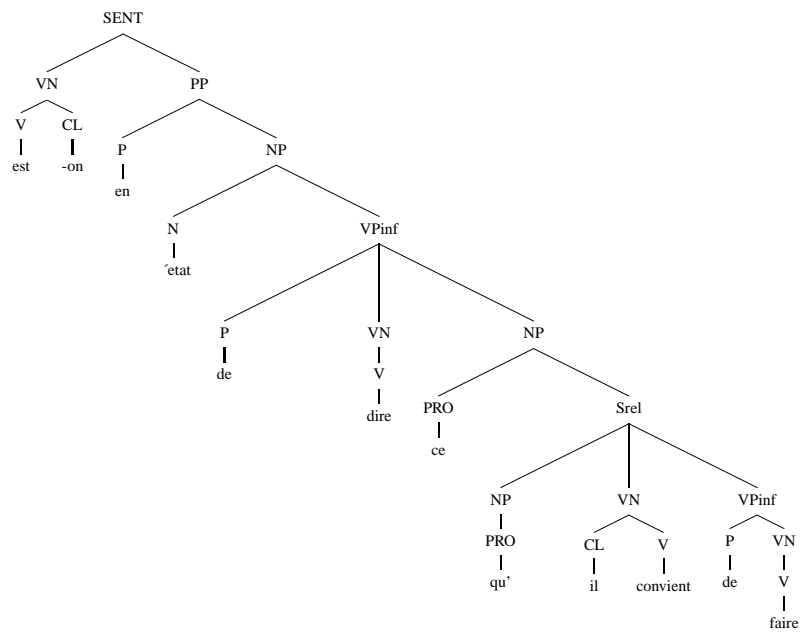


Figure 5.15: Wrong VPinf attachment in bigram model -2

Model (≤ 100 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
BitPar - Comp Expanded	57.09	56.34	56.71	2.42	34.38	65.62	9.94	99.3	90.9
BitPar - Comp Expanded (CR)	57.84	57.88	57.86	2.16	35.98	67.48	10.37	99.2	90.9
BitPar - Comp Contracted	62.01	62.21	62.11	1.90	43.21	72.11	15.83	98.7	93.2
BitPar - Comp Contracted (CR)	63.72	63.01	63.36	1.76	43.56	73.74	16.48	98.2	93.1
Collins Model 1	76.98	77.81	77.39	1.08	60.06	83.8	25.25	99.7	96.8
Collins Model 1 (CR)	78.52	78.14	78.33	1.05	61.11	85.33	27.34	99.8	96.8
Collins Model 2 (CR)	78.87	78.30	78.58	1.02	60.62	86.1	27.29	99.7	96.8
Collins Model 2 (CR) + Sister-head	78.80	78.81	78.80	1.02	60.87	85.31	27.16	99.6	96.9
Collins Model 2 (CR) + Bigram-f _{hat}	78.55	78.27	78.41	1.04	60.50	84.82	27.64	99.6	96.7
Collins Model 2 (CR) + Bigram-all	79.54	79.17	79.35	0.98	61.25	86.18	28.76	99.5	96.8
Collins Model 2 (CR) + Bigram-all + VNG	79.58	79.26	79.42	0.98	61.17	85.95	28.61	99.4	96.9

Table 5.1: Results for lexicalised and unlexicalised models for sentences ≤ 100 words long. Each model performed its own POS tagging. CR refers to the raised coordination transformation. All Collins models were run on the contracted comp

Model (≤ 40 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
BitPar	59.97	58.64	59.30	1.74	39.05	73.23	11.36	99.2	91.0
Comp Expanded									
BitPar	60.75	60.57	60.66	1.57	40.77	75.03	11.85	99.1	91.1
Comp Expanded (CR)									
BitPar	64.19	64.61	64.40	1.50	46.74	76.80	17.13	98.5	93.3
Comp Contracted									
BitPar	66.11	65.55	65.83	1.39	46.99	78.95	17.82	97.9	93.2
Comp Contracted (CR)									
Collins Model 1	78.64	79.52	79.08	0.82	63.98	88.25	27.31	99.6	96.9
Collins Model 1 (CR)	80.35	79.99	80.17	0.78	65.22	89.46	29.57	99.7	96.9
Collins Model 2 (CR)	80.49	79.98	80.23	0.77	64.85	90.1	29.49	99.7	96.8
Collins Model 2 (CR)	80.47	80.56	80.51	0.78	64.96	89.34	29.27	99.6	96.9
+ Sister-Head									
Collins Model 2 (CR)	80.30	80.05	80.17	0.77	64.78	89.13	29.89	99.6	96.7
+ Bigram-flat									
Collins Model 2 (CR)	81.15	80.84	80.99	0.74	65.21	90.51	30.97	99.5	96.8
+ Bigram-all									
Collins Model 2 (CR)	81.16	80.90	81.03	0.74	65.07	90.07	30.90	99.4	96.9
+ Bigram-all + VNG									

Table 5.2: Results for lexicalised and unlexicalised models for sentences ≤ 40 words long. Each model performed its own POS tagging. CR refers to the raised coordination transformation.

Model (≤ 100 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
BitPar	61.76	55.58	58.51	2.18	37.04	67.41	11.23	98.8	100.0
Comp Expanded									
BitPar	63.07	62.25	62.66	1.98	37.64	70.67	12.92	97.5	100.0
Comp Expanded (CR)									
BitPar	64.95	64.53	64.74	1.86	43.67	71.84	17.96	98.0	100.0
Comp Contracted									
BitPar	67.04	65.55	66.29	1.69	43.96	75.44	18.68	96.9	100.0
Comp Contracted (CR)									
Collins Model 1	77.71	78.66	78.18	1.10	59.34	83.63	27.61	99.9	98.5
Collins Model 1 (CR)	79.7	79.51	79.60	1.03	60.64	85.54	29.92	99.9	98.5
Collins Model 2 (CR)	79.86	79.65	79.75	1.05	59.74	85.74	30.32	99.9	98.5
Collins Model 2 (CR)	79.63	79.95	79.79	1.02	60.4	85.93	29.45	99.8	98.5
+ Sister-head									
Collins Model 2 (CR)	79.50	79.50	79.50	1.05	59.44	84.54	30.42	99.8	98.8
+ Bigram-flat									
Collins Model 2 (CR)	80.21	80.84	80.52	1.01	60.83	85.10	30.61	99.7	98.7
+ Bigram-all									
Collins Model 2 (CR)	80.32	80.25	80.28	1.02	60.73	85.2	31.32	99.8	98.7
+ Bigram-all + VNG									

Table 5.3: Results for lexicalised and unlexicalised models for sentences ≤ 100 words long. The correct POS tags were supplied to the models. CR refers to the raised coordination transformation.

Model (≤ 40 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
BitPar	64.26	58.21	61.09	1.61	42.2	74.45	12.83	98.7	100.0
Comp Expanded									
BitPar	65.5	64.76	65.13	1.49	42.36	77.48	14.70	97.8	100.0
Comp Expanded (CR)									
BitPar	67.06	66.97	67.01	1.45	47.29	76.57	19.45	97.9	100.0
Comp Contracted									
BitPar	69.35	67.93	68.63	1.34	47.43	80.25	20.20	97.0	100.0
Comp Contracted (CR)									
Collins Model 1	79.38	80.44	79.91	0.84	63.30	87.95	29.86	99.8	98.5
Collins Model 1 - CR	81.51	81.43	81.47	0.78	64.6	89.25	32.36	99.8	98.5
Collins Model 2 (CR)	81.69	81.59	81.64	0.78	63.84	89.69	32.79	99.8	98.6
Collins Model 2 (CR)	81.08	81.56	81.32	0.79	64.35	89.57	31.74	99.7	98.5
+ Sister-Head									
Collins Model 2 (CR)	81.14	81.19	81.16	0.81	63.37	88.80	32.72	99.7	98.8
+ Bigram- <i>fst</i>									
Collins Model 2 (CR)	81.78	81.91	81.84	0.78	64.96	89.12	32.97	99.6	98.8
+ Bigram- <i>all</i>									
Collins Model 2 (CR)	81.89	81.95	81.92	0.79	64.67	89.12	33.70	99.6	98.8
+ Bigram- <i>all</i> + VNG									

Table 5.4: Results for lexicalised and unlexicalised models for sentences ≤ 40 words long. The correct POS tags were supplied to the models. CR refers to the raised coordination transformation.

Model (≤ 100 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
PTB	85.73	86.14	85.93	1.37	55.44	80.34	27.12	99.3	94.9
FTB	77.39	76.74	77.06	1.11	59.01	85.30	25.48	99.7	95.7

Table 5.5: Comparison between PTB and FTB for sentence length ≤ 100 words

Model (≤ 40 words)	LR	LP	F-Score	CBs	0CB	≤ 2CBs	Complete Match	Coverage (%)	Tagging Accuracy
PTB	86.43	86.79	86.61	1.17	57.80	82.44	29.12	99.7	95.0
FTB	79.20	78.58	78.89	0.83	63.33	89.23	27.53	99.7	95.8

Table 5.6: Comparison between PTB and FTB for sentence length ≤ 40 words

Chapter 6

Conclusion

6.1 Summary

This thesis presented the first ever attempt at statistically parsing French by making use of the recently available *Corpus Le Monde* treebank project. We started off by implementing a baseline unlexicalised vanilla PCFG model which achieved an f-score of 65.83%. Applying Collins' head-lexicalised Model 1 help improved parsing accuracy by a massive 14.5% to reach an f-score of 80.17%, demonstrating the usefulness of lexicalisation in parsing French. In this model, all nonterminal nodes of a parse tree are augmented by their head word and its associated tag.

We then proceeded to enhance this model by adding complement/adjunct distinction as well as probabilities over subcategorization frames, but this did not improve parsing accuracy. There are two explanations for this; firstly, the annotation scheme of the French Treebank (where verbal phrases are replaced by a minimal verbal nucleus and complements are sisters of the verbal nucleus), makes it very hard to properly implement the argument/adjunct distinction during the training phase. Secondly, as noted in [Collins, 1999], there is a real overlap between the distance measure already implemented in Model 1 and the subcategorization probabilities.

Given that a simple corpus study showed that the tree structure of the corpus is flatter

than that of the Penn Treebank, we looked at models that would take that fact into account. We first looked at a sister-head model [Dubey and Keller, 2003] which basically entails extending Collins' base NP model to all constituents. However, this model did not significantly improve parsing accuracy. On the other hand, a bigram-model in which the conditioning context is extended to include the previous generated modifier proved to be a very good parsing model and achieved an f-score of 80.99%.

In our final experiment, we extended the bigram model by implementing a mechanism to differentiate between head verbal nuclei that have a clitic pronoun child with those that do not. When a clitic pronoun appears as the child of a verbal nucleus, it usually plays the role of the subject of the sentence. In cases where the verbal nucleus does not have a pronoun child, the subject role is usually played by a preceding constituent. We hoped that this distinction would allow us to have a cleaner implementation of probabilities over subcategorization frames. As it turns out, this model did not make a significant improvement over the bigram model, but still managed to increase the f-score to 81.03%.

This thesis, although the first work of its kind for the new French Treebank, reports a highly encouraging accuracy figure. However, it falls short of state of the art results for the Penn Treebank (around 90% f-score), which however is achieved with a much larger training corpus (approximately. 40,000 sentences). To have a more comparable benchmark, we trained and tested Bikel's emulation of Collins' Model 2 on training and test sets of similar size to our corpus. This model achieved an f-score of 8% higher than the FTB model.

6.2 Future work

We believe that although the results obtained in this thesis are highly satisfactory and compare favourably with similar work for other languages, there is a lot of room for further improvement.

The parsing models we have explored are data-driven. Hence, our main wish is to have

a larger dataset. While we realise that the first release of any work of the magnitude of a treebank project is bound to contain inconsistencies, we expect that as more and more people make use of this corpus and give their feedback to the corpus designers, future releases will be more accurate and hopefully larger.

A major part of emulating Collins' models is to come up with head and argument identification rules. We used guidelines provided with the dataset, our own linguistic intuition and empirical study of the data to come up with these rules. We cannot possibly be certain that these are the optimal rules for the dataset, where optimal refers to rules that maximise the likelihood of the training data. [Chiang and Bikel, 2002] present a learning method that, given a model with initial hand written tree-augmentation rules such as the ones mentioned above, re-estimates the parameters of the model using the Inside-Outside algorithm. We believe that this is an approach worth exploring.

We saw that parsing performance is directly proportional to tagging accuracy. Our best performing bigram-all model has an upper f-score limit of 81.84% given perfect tagging, an improvement of 0.85% over our current integrated tagging method. Here again, future work will involve fine-tuning the existing feature set to improve unknown word POS assignment disambiguation.

We would also like to do a more thorough corpus study especially as regards to punctuation and coordination. Collins treatment of the latter two is very Penn Treebank specific. We feel that we did not devote sufficient time in trying to understand how these 2 phenomena occur in the French Treebank and how best to integrate them in the parsing model. It would be interesting to see how the model performs were coordination represented exactly as in the Penn Treebank i.e. by raising the children of the syntactic category which is a sister of the coordination conjunction to the level of the conjunct. Moreover, to improve a model, it is essential to understand its failings, and therefore a rigorous error analysis is called for. As mentioned by [Collins, 1999], evaluating the parser's performance in recovering dependencies between words is a more informative strategy than accuracy by constituent type. This, combined with a deeper study of the intricacies of the probability model, will allow us to address cases where the model is systematically failing e.g., the incorrect VPinf attachment that the bigram

model, unlike the original Collins Model 2, is guilty of.

Finally, it would be interesting to extend the conditioning context by adding information about the grandparent node as well. [Klein and Manning, 2003, Charniak, 1999] show that this form of vertical markovization can be very beneficial. Obviously, increasing the context in such a way would make n-gram estimation methods infeasible; a more sophisticated framework such as maximum-entropy (as in [Charniak, 1999]) would be worth looking at. Another option would be to do n-gram estimation using word lemmas instead of the words. French is morphologically quite rich and therefore we expect to have less sparse data issues using word lemmas which, crucially, are already annotated in the corpus.

Appendix A

Head rules

This appendix describes the rules used to find heads of constituents in the treebank; i.e., for a context-free rule $\langle X \rightarrow Y_1 \dots Y_n \rangle$, these rules decide which of $Y_1 \dots Y_n$ is the head of the rule.

Parent Nonterminal	Direction	Priority list
AP	Right	A N V
AdP	Right	ADV
AdP	Left	P D C
COORD	Left	C
COORD	Right	
NP	Right	N PRO A ADV
NP	Left	NP
NP	Right	
PP	Right	P CL A ADV V N
SENT	Left	VNG VN V NP Srel Ssub Sint
Sint	Left	VNG VN V
Sint	Right	
Srel	Left	VNG VN V
Ssub	Left	VNG VN V
Ssub	Right	
VN	Right	V
VNG	Right	V
VPinf	Left	VNG VN V
VPinf	Right	
VPpart	Left	VNG VN V
VPpart	Right	
PONCT	Right	

Table A.1: Head rules table. *Parent* is the non-terminal on the left-hand side of a rule. *Direction* specifies whether search starts from left or right end of the rule. *Priority* gives a priority ranking, with priority decreasing when moving down the list

Appendix B

Argument identification rules

This appendix describes the rules for identifying complements in the treebank

1. If the parent is a **SENT**, then perform a left-to-right search on the right side of the head, where the first child found whose label is one of { **NP**, **Srel**, **Ssub**, **Sint**, **VPinf**, **VPpart** } is relabelled as an argument. If the parent is a **SENT**, and the head nonterminal is not **VNG**, relabel the last **NP** before the head as an argument.
2. Relabel the first non-punctuation child following the head of a **PP** as argument.
3. Relabel the first non-punctuation child following the head of a **Ssub** as argument. Relabel the first **C** or **NP** child before the head of a **Ssub** as argument.
4. Relabel the first non-punctuation child following the head of a **Srel** as argument. Relabel the last **NP** child before the head of a **Srel** as argument.
5. Relabel the last **NP**, **Ssub** or **VPinf** child following the head of an **Sint** as argument.

Appendix C

Word features

The word features class provides a mapping of lexical items to orthographical/morphological word feature vectors, to help disambiguation during part-of-speech tagging of unknown words.

The pseudo code for assigning the distinct values of each of the six features of the word vector are presented below:

1. Is the word (after removal of ”-”, ”_”, ”,” characters) numerical ?
2. Capitalisation features.

```
If first letter of word is upperCase
    if word is sentence initial
        return C1
    elsif whole word is upperCase and contains a numerical
        return C2
    elsif whole word is upperCase and does not contain a numerical
        return C3
    else
        return C4
else
    return C0
```

3. Hyphenation features.

```

if word contains "-"
    return H1
elsif word contains "'"
    return H2
elsif word contains "$"
    return H3
else
    return H4

```

4. Compound features

A word is considered a compound if it contains a "_" character. Prepositions are ("À", "à", "de", "du", "des", "d'", "au", "aux", "Au", "Aux", "A", "De", "Des", "D'", "Du")

```

If word is compound
    if first part of compound is a preposition and last part is a preposition
        return P2
    elif first part of compound is a preposition
        return P1
    elif last part of compound is a preposition
        return P3
    else
        return P4
else
    return P0

```

5. Derivational features

Possible derivation are : { "âtre", "aphe", "aphie", "ment", "aire", "if", "ien", "age", "al", "ale", "er", "ère", "ique", "tion", "able", "aux", "enne", "ive", "eur", "ois", "oise", "eux" }

If word ends with a derivation

```

    return D + position in derivation array
else
    return D0

```

6. Inflectional features

If the word length is greater than 3 and its derivational feature is D0, its inflectional features are computed.

Inflections are :

```

{ "issons", "issez", "issent", "isse", "isses", "issions", "issiez", "issant", "is-
sais", "issait", "issaient", "îmes", "îtes", "irent", "irai", "iras", "irons", "iront",
"irez", "irais", "irait", "irions", "iriez", "iraient", "erai", "eras", "erons", "erez",
"eront", "erais", "erait", "erions", "eriez", "eraient", "ions", "iez", "ant", "ais",
"ait", "aient", "as", "âmes", "âtes", "èrent", "ons", "ez", "ent", "es", "ées" }

```

If word ends with an inflection

```

    return I + position in inflection array
else
    return I0

```

Appendix D

Parameter classes

This appendix describes the parameter classes with their back-off structures implemented in Bikel’s emulation of the Collins’ parser

The head-generation parameter class, P_H and subcat generation parameter classes, P_{subcat_L} and P_{subcat_R} , have back-off structures as follows.

Note that α refers to the argument removal operation.

Hence $\alpha(NP - C) = NP$

Back-off level	$P_H(H \mid \dots)$	$P_{subcat_L}(subcat_L \mid \dots) P_{subcat_R}(subcat_R \mid \dots)$
0	P, w_h, t_h	$\alpha(P), \alpha(H), w_h, t_h$
1	P, t_h	$\alpha(P), \alpha(H), t_h$
2	P, t_h	$\alpha(P), \alpha(H)$

Table D.1: Parameter class for head generation

A fully lexicalised nonterminal has 3 components: the non-terminal label, the head word and its part of speech. The generation of fully-lexicalised modifying nonterminals is done in two steps, to allow for the parameters to be independently smoothed, which, in turn, is done to avoid sparse data problems. These two steps estimate the joint event of all three components using the chain rule.

The two parameter classes for generating modifying nonterminals that are not dominated by a baseNP, P_M (for the partially lexicalised version consisting of the unlexicalised label and the part of speech of the head word) and P_{M_w} (for the generation of the head word) have the following back-off structures:

For notational brevity, $M(w, t)_i$ is used to refer to the three items M_i, t_{M_i} and w_{M_i} that constitute some fully lexicalised modifying nonterminal, and similarly $M(t)_i$ refers to the two items M_i, t_{M_i} that constitute some partially-lexicalised modifying nonterminal.

The "bigram of non-terminals" model has the same parameter classes except for the redefinition of the δ function.

Back-off level	$P_M(M(t)_i, coord, punc \mid \dots)$
0	$\alpha(P), H, w_h, t_h, subcat_{side}, side, vi(M_i), \delta(M_{i-1})$
1	$\alpha(P), H, t_h, subcat_{side}, side, vi(M_i), \delta(M_{i-1})$
2	$\alpha(P), H, subcat_{side}, side, vi(M_i), \delta(M_{i-1})$

Table D.2: Parameter class for partially lexicalised modifier

Back-off level	$P_{M_w}(w_{M_i} \mid \dots)$
0	$M(t)_i, coord, punc, alpha(P), H, w_h, t_h, subcat_{side}, side, vi(M_i), \delta(M_{i-1})$
1	$M(t)_i, coord, punc, alpha(P), H, t_h, subcat_{side}, side, vi(M_i), \delta(M_{i-1})$
2	t_{M_i}

Table D.3: Parameter class for lexicalised modifier

The two parameter classes for generating modifying nonterminals that are children of base NPs have the following back-off structures. Note that there's no coord flag as coordinating conjunctions are generated like regular modifying nonterminals when they are dominated by NPB. M_0 is defined as H , the head nonterminal label of the base NP that was generated using a P_H parameter.

These parameter classes are used for the sister-head model as well.

The two parameter classes for generating punctuation and coordinating conjunctions, P_{punc} and P_{coord} , have the following back-off structures, where

Back-off level	$P_{M,NPB}(M(t)_i, punc \dots)$	$P_{M_w,NPB}(w_{M_i} \dots)$
0	$P, M(w, t)_{i-1}, side$	$M_i, t_{M_i}, punc, P, M(w, t)_{i-1}, side$
1	$P, M(t)_{i-1}, side$	$M_i, t_{M_i}, punc, P, M(t)_{i-1}, side$
2	t_{M_i}	

Table D.4: Parameter class for Base NP generation

- $type$ is a flag that obtains the value p in the history contexts of P_{punc} parameters and c in the history contexts of P_{coord} parameters,
- $M(w, t)_i$ is the modifying preterminal that is being conjoined to the head child,
- t_p/t_c is the particular POS tag that is conjoining the modifier to the head child (such as "C" or ":",) and
- w_p/w_c is the particular word that is conjoining the modifier to the head child (such as "et" or ":",)

Back-off level	$P_{coord}(t_c \dots), P_{punc}(t_p \dots)$	$P_{coord_w}(w_c \dots), P_{punc_w}(w_p \dots)$
0	$P, H, M(w, t)_i, type, w_h, t_h$	$P, H, M(w, t)_i, type, w_h, t_h, t_{type}$
1	$P, H, M(t)_i, type, t_h$	$P, H, M(t)_i, type, t_h, t_{type}$
2	$type$	t_{type}

Table D.5: Parameter class for coordination and punctuation generation

Bibliography

- [A. Abeillé and F.Toussenel, 2003] A. Abeillé, L. C. and F.Toussenel (2003). Building Treebank For French.
- [Bikel, 2002] Bikel, D. (2002). Design of a Multi-lingual, Parallel-processing.
- [Bikel, 2004] Bikel, D. (2004). Intricacies of Collins' Parsing Model.
- [Bikel and Chiang, 2000] Bikel, D. and Chiang, D. (2000). Two statistical parsing models applied to the Chinese Treebank.
- [Black et al., 1991] Black, E., Abney, S. P., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M. P., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings DARPA Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA. Morgan Kaufmann.
- [Charniak, 1996] Charniak, E. (1996). Tree-Bank Grammars. In *AAAI/IAAI, Vol. 2*, pages 1031–1036.
- [Charniak, 1997] Charniak, E. (1997). Statistical Parsing with a Context-Free Grammar and Word Statistics. In *AAAI/IAAI*, pages 598–603.
- [Charniak, 1999] Charniak, E. (1999). A Maximum-Entropy-Inspired Parser. Technical Report CS-99-12.
- [Chiang, 2000] Chiang, D. (2000). Statistical parsing with an automatically-extracted tree adjoining grammar.
- [Chiang and Bikel, 2002] Chiang, D. and Bikel, D. M. (2002). Recovering Latent Information in Treebanks.
- [Collins, 1997] Collins, M. (1997). Three Generative, Lexicalized Models for Statistical Parsing. In Philip R. Cohen and Wolfgang Wahlster, editor, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational*

- Linguistics*, pages 16–23, Somerset, New Jersey. Association for Computational Linguistics.
- [Collins et al., 1999] Collins, M., Ramshaw, L., Hajic;, J., and Tillmann, C. (1999). A statistical parser for Czech. In *Proceedings of the 37th conference on Association for Computational Linguistics*, pages 505–512. Association for Computational Linguistics.
- [Collins, 1999] Collins, M. J. (1999). *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Philadelphia.
- [Dubey and Keller, 2003] Dubey, A. and Keller, F. (2003). Probabilistic Parsing for German using Sister-Head Dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103, Sapporo.
- [Hajič, 1998] Hajič, J. (1998). Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In Hajičová, E., editor, *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová*, pages 12–19. Prague Karolinum, Charles University Press.
- [Hindle and Rooth, 1991] Hindle, D. and Rooth, M. (1991). Structural Ambiguity and Lexical Relations. In *Meeting of the Association for Computational Linguistics*, pages 229–236.
- [Ide, 1998] Ide, N. (1998). *Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora*.
- [Ircs, 2002] Ircs, N. X. (2002). Building a Large-Scale Annotated Chinese Corpus.
- [Kasami, 1965] Kasami (1965). An efficient recognition and syntax analysis algorithm for context-free languages.
- [Kayne, 1975] Kayne, R. (1975). *French Syntax : the transformational cycle*.
- [Klein and Manning, 2003] Klein, D. and Manning, C. D. (2003). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.
- [Levy and Manning, 2003] Levy, R. and Manning, C. (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.
- [Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Lanugage Processing*. MIT.
- [Marcus et al., 1994] Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

- [Ney et al., 1994] Ney, H., U.Essen, and R.Knesser (1994). On structuring probabilistic dependencies in stochastic language modelling.
- [Schmid, 2004] Schmid, H. (2004). Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors.
- [Skut et al., 1997] Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. (1997). An annotation scheme for free word order languages.
- [Weischedel et al., 1993] Weischedel, R., Schwartz, R., Palmucci, J., Meter, M., and Ramshaw, L. (1993). Coping with ambiguity and unknown words through probabilistic models. *Comput. Linguist.*, 19(2):361–382.