

# Chunking + Island-Driven Parsing = Full Parsing

Alicia Ageno and Horacio Rodríguez

Talp Research Center, UPC

Jordi Girona, 1-3

Barcelona 08034, Spain

{ageno, horacio}@lsi.upc.es

## Abstract

We present a novel method for improving parsing performance, using a stochastic island-driven chart parser preceded by a chunking process for identifying initial islands. Two different stochastic models have been developed for the island-driven parsing. Some experiments with nominal chunking using broad-coverage grammars derived from the Penn Treebank have been performed with remarkable results.

## 1 Introduction

Recently there has been increasing interest in facilitating the parsing process in full parsing (guided by broad-coverage Context-Free Grammars) in order to improve performance. Mainly three directions have been followed: 1) deriving a regular approximation of the initial grammar (Nederhof 2000) and generating a language that could be either a subset or a superset of the language generated by the original grammar, 2) splitting the parsing process into a sequence of simpler steps, each one governed by a simple, usually regular, grammar (Ciravegna & Lavelli 99; Abney 96), and 3) guiding the parsing process by certain type of heuristics, usually informed by stochastic models. These models are normally Stochastic Context-Free Grammars, SCFG (Charniak et al. 98; Blaheta & Charniak 99) or extensions (Collins 97; Geman & Johnson 2000).

The three directions present advantages and limitations. In the first case, only an approximation of the original language is obtained. This may be insufficient for some applications, depending on the distance between the language generated by the original grammar and the approximation. In the second case, the grammar must be structured as a cascade of simpler grammars, which prevents us from using a general-purpose pre-existing grammar. Our work is in the line of the third approach, although our method is completely independent of the knowledge source and the language, in contrast with approaches as (Collins 97). We will be using sort of FOMs as (Charniak et al. 98) or (Blaheta & Charniak 99), but based on the concept of islands and applying these FOMs to their extension.

What we propose here is a way of splitting the parsing process into two steps, allowing the use of any full-coverage SCFG as an input. Firstly, a chunking step (using a grammar of chunks automatically extracted from the initial grammar), in which a partial parsing of the

input is performed. Secondly, an island-driven parsing step, where a probabilistic bidirectional parsing is performed starting from islands which are the previously detected chunks.

In the remainder of this paper, we start by briefly describing our island-driven parser, along with the stochastic models it uses, in section 2. In section 3, we describe the global methodology. In sections 4 and 5 we discuss the experiments (and their results) and the evaluation of the quality of the results respectively. Finally, in section 6 we state the conclusions.

## 2 The Probabilistic Island-Driven Approach

Although most methods for CFG parsing are based on a uniform way of guiding the parsing process (e.g. top-down, bottom-up, left-corner), there have been several attempts to introduce more flexibility, allowing bidirectionality, in order to make parsers more sensitive to linguistic phenomena (Satta & Stock 94; Sikkil & op den Akker 96; Ritchie 99). Particularly in island-driven parsing, the conventional left-to-right approach of chart parsing is enhanced with two features: bidirectionality (parsing can take place either left-to-right or right-to-left) and the islands themselves (dynamically determined positions of the sentence from which the process starts). Island-driven flexibility permits the use of optimal heuristics, in order to deliver a single best-first analysis, that could not be applied to unidirectional strategies. These heuristics are based on two stochastic models, local and *neighbouring*, which allow to select the most probable island, to be extended in the most probable side. Our island-driven chart parser performs a combination of bottom-up expansion and top-down prediction, guided by the stochastic parameters. The algorithm has been described in (Ageno & Rodríguez 2000), therefore we will focus on the description of the stochastic models.

### 2.1 The Local Model

The local approach is based on regarding the probability of an edge to be extended (and the same applies to the prediction) as the probability of the next symbol to be expanded having the terminal(s) symbol(s) in the corresponding position of the sentence as either left or right corner. Being  $G$  a SCFG,  $T$  the set of terminal symbols of  $G$ ,  $N$  the set of nonterminal symbols of  $G$ ,  $R_i$  the  $i$ -th production of  $G$  and  $P(R_i)$  its attached probability;  $[A, i, j]$  is an island of category  $A$  spanning positions  $i$  to  $j$ ,

and  $\{left|right\}_corner$  are functions from  $N \times T$  to  $[0,1]$ , being  $\{left|right\}_corner(A, a)$  the probability that a derivation tree rooted  $A$  could have symbol  $a$  as a left or right corner:

$$\forall A \in N, a \in T : right\_corner(A, a) = P(A \gg a/G)^1$$

Similarly,  $\{left|right\}_corner^*$  are functions from  $N \times T^*$  to  $[0,1]$ , so that, for any list of symbols  $la$ :

$$right\_corner^*(A, la) = \sum_{a \in la} right\_corner(A, a)$$

Left\_corner probabilities are symmetrically defined. All these probabilities are pre-computed, so that:

- For expansion to the left of an island (inactive edge) labelled  $A$ :

$$P_{island}^{left}([A, i, j]/G, w) = \sum_{R_i: X \rightarrow \alpha A} P(R_i)$$

- For expansion to the left of (or prediction to the left from) an active edge<sup>2</sup>:

$$P_{arc}^{left}([A \rightarrow \alpha B. \beta. \gamma, i, j]/G, w) = right\_corner^*(B, lt)$$

Special cases where either  $\alpha$  or  $\beta$  are empty are also considered. Expansions and predictions to the right are symmetrically defined.

## 2.2 The Neighbouring Model

In this approach, to take the decision of extending an island we'll consider both the information provided by the neighbours and the distances to them (the lengths of the *gaps*, the segments of the input sentence between adjacent islands). Roughly speaking, we intend to model the distances (in terms of number of terminals) between nodes in the parse tree, and guide the decisions accordingly. Hence, the probabilities of length distributions for each rule of the grammar must be learnt from a training corpus.

Given two islands  $[A, i, j]$  and  $[B, j+d, l]$ , separated by a distance  $d$ , three types of relationship have been considered<sup>3</sup>:

$$R^1 = \{r : X \rightarrow \alpha A \beta B \gamma, d = |\beta|\}$$

$$R^2 = \{r : X \rightarrow \alpha A \beta H \gamma, H \xrightarrow{*} \delta B \mu, d = |\beta| + |\delta|\}$$

$$R^3 = \{r : X \rightarrow \alpha H \beta B \gamma, H \xrightarrow{*} \delta A \mu, d = |\mu| + |\beta|\}$$

And we'll denote each probability, for  $i=1..3$ :

$$P^i(d/r, A, B) \quad \text{and} \quad P_{acc}^i(d/A, B) = \sum_{r \in R^i} P^i(d/r, A, B)$$

These probabilities are pre-computed for each possible pair of islands and distance  $d=0..limit$  (being all cases of  $d > limit$  treated as a whole). The *limit* is a parameter that in our experiments has been set to 3<sup>4</sup>. The application of

<sup>1</sup>  $P(A \gg a/G)$  denotes the probability that, starting with the nonterminal  $A$ , successive application of rules in grammar  $G$  produces a sequence starting with terminal  $a$ .

<sup>2</sup>  $lt$  being the list of terminal categories of word  $w_{i-1}$

<sup>3</sup> These cases only account for those situations in which there is one rule that includes directly at least one of the islands considered, according to our notion of *neighbourhood*. Therefore, to get a full coverage, a back-off to other method is needed.

<sup>4</sup> Considering average distances between islands.

this model to the expansions and predictions to the right follows (left direction symmetrical):

- For expansion of an island  $[A, i, j]$ , being  $[B, j+d, l]$  the closest island to the right:

$$P_{island}^{right}([A, i, j]/G, w, [B, j+d, l]) = P_{acc}^1(d/A, B) + \sum_{Z \in N} \sum_{l=0}^{\min(3, d)} P_{acc}^1(d-l/A, Z) \times P_{acc}^2(l/Z, B) + P_{acc}^1(d-l/A, Z) \times P_{acc}^3(l/Z, A)$$

The first addend accounts for cases of  $A$  and  $B$  being in the same rule righthand side while the second one considers all possibilities of  $B$  being derived in one or more steps from a nonterminal  $Z$  which is in the same rule righthand side as  $A$ , plus all possibilities of  $A$  being derived from a  $Z$  in the same rule righthand side as  $B$ .

- For expansion of (or prediction from) an active edge  $[A \rightarrow \beta. A_l \alpha A_r. \gamma, i, j]$ , being  $[B, j+d, l]$  the closest island to the right:

$$P_{arc}^{right}([A \rightarrow \beta. A_l \alpha A_r. \gamma, i, j]/G, w, [B, j+d, l]) = P^1(d/r, A_r, B) + \sum_{\gamma_i \in N, i \leq d, 0 \leq d-l \leq 3} prob(|\gamma_i - \gamma_{i-1}| = l) \times P_{acc}^2(d-l/\gamma_i, B)$$

The idea is the same, albeit particularising to the rule of the active edge. To the left the formula is symmetrical, using  $P^3$  instead of  $P^2$  and  $A_l$  instead of  $A_r$ . *Prob* provides for the distribution of probabilities of the lengths of any subsequence of terminal and nonterminal symbols<sup>5</sup>.

## 3 Overall methodology

Our approach takes advantage of both a syntactically motivated way of detecting islands (instead of essentially lexical as in previous approaches) and a stochastic way of guiding the parsing process (that is, which islands should be extended and in which direction).

So far, non-ambiguous words have been considered our initial islands, saving at the same time the necessity of a tagger. The results of previous experiments testing the performance of both island-driven stochastic models on the Penn Treebank are described in (Ageno & Rodríguez 2001). However, we intend to introduce a more informed method to choose the islands, such as using base-NPs<sup>6</sup> as candidates. Nevertheless, the complexity introduced by this additional step should be minimised so that it was worth the improvements obtained.

Although the task of finding base-NPs is relatively simple, its difficulty stems from the attempt to obtain a high level of accuracy. However, as we will show in section 5, this accuracy is not that critical anymore when base-NPs are just going to be used as the input for our island-driven parser. In other words, our approach does not rely on being able to detect all possible base-NPs, and conversely, the base-NPs detected will be acceptable only if they drive to an island-driven analysis. Thus, we have implemented a straightforward algorithm to find base-NPs which, despite its simplicity, accomplishes an outstanding

<sup>5</sup> Several heuristics, described in (Ageno & Rodríguez 2001) have been adopted in this strategy.

<sup>6</sup> Base-NPs are just one kind of chunks. Other classes can be considered (Abney 1996). Our results with base-NPs may be easily extended to other types.

improvement in the results regarding both efficiency and quality. Several variations of this algorithm have been tested.

We take our base-NPs' definition from (Cardie & Pierce 98), that is, we define base-NPs to be simple, nonrecursive noun phrases (i.e., not containing other noun phrase descendants). However, we will not apply their method for selection of the chunks. Our algorithm is composed by three steps:

1. A base-NP grammar, a subset of our initial grammar containing only those rules whose lefthand side is a nonrecursive NP, is extracted.

2. A partial parse to find all possible base-NPs for each sentence in the input corpus is performed by a chunker. For the sake of comparison, PoS for words in the test set are also ambiguous, so eventually many of these base-NPs may not be correct.

3. A process of selection of the obtained chunks, according to their types, is carried out.

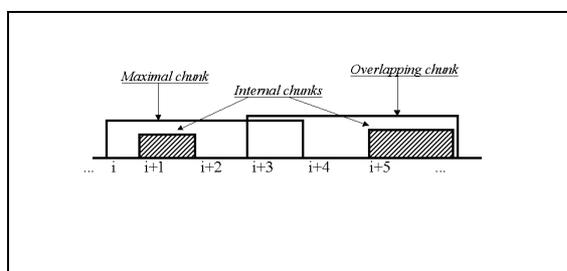


Figure 2: Example of kinds of base-NPs

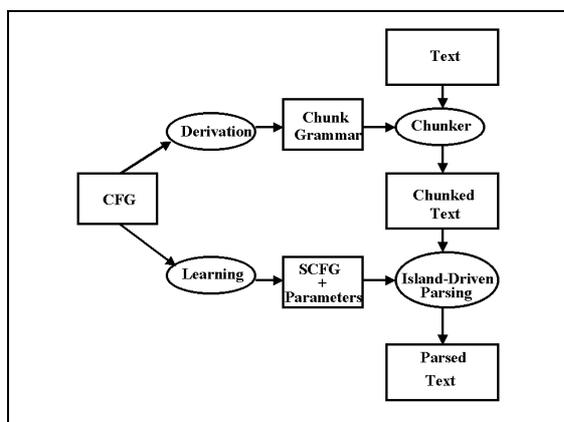


Figure 3: A two-steps approach to parsing

Three types of base-NPs have been distinguished (see Figure 2), namely the *maximal* NPs (the longest ones out of those starting at a certain position of the sentence), the *overlapping* NPs (those overlapping with previous *maximal* NPs), and the *internal* NPs (the rest of the NPs extracted). Selection of chunks has been performed considering these three classes, so that we have evaluated, for both stochastic models, the following alternatives:

1. Taking every base-NP extracted, hereafter *local+chunks1* and *neighbouring+chunks1* respectively.
2. Selecting only *maximal* and *overlapping* NPs, hereafter *local+chunks2* and *neighbouring+chunks2*.
3. Culling only those maximal NPs, hereafter *local+chunks3* and *neighbouring+chunks3*.

The overall process is depicted in Figure 3.

## 4 Evaluation

### 4.1 Setting

We hasten to emphasise that our experiments have been aimed at comparing both methods for selection of islands, as well as at comparing, in the same environment, the performance of our 'chunks+island-driven' approach with the classical bottom-up<sup>7</sup>. By classical bottom-up (henceforth BU) we mean a chart parser which operates combining the edges of the chart bottom-up and left-to-right. We consider that the parse returned by this method is the first analysis found, so that the process will stop as soon as this happens, possibly leaving items in the agenda.

Our methodology does not supply a specific knowledge source (as in Collins 97), but it can be applied to any existent SCFG. Besides, for the sake of comparison we might be starting from a corpus that is not morphologically disambiguated. Our approach has been tested using several artificial grammars, and even a limited-coverage grammar for Spanish (see Ageno & Rodríguez 2000). However, we wished to compare our strategies using a grammar as close as possible to a real one, so we chose corpus Penn Treebank II (Marcus et al. 93), 1,25Mw. The grammar underlying the bracketing was extracted, but its size (17534 rules) was simply too big to contemplate for our parser. Therefore, following (Gaizauskas 95), we applied a thresholding mechanism to prune rules from the grammar, obtaining a grammar with 941 rules, 26 nonterminals and 45 terminals<sup>8</sup>.

In order to estimate the parameters of both models, a training corpus of 49208 sentences was used (previously, probabilities attached to the grammar rules were learnt). While local parameters can be considered accurately learnt, *neighbouring* parameters are far more complex, which implies the sparseness problems that will be described below. A corpus of 1000 sentences extracted randomly from sections 13 and 23 (from those sentences covered by our grammar) was used for testing. The base-NP derived grammar is composed by 33 rules.

Efficiency has been measured in terms of the number of inactive and active edges created during the parsing process, that is, the ones required to find the first parse.

### 4.2 Results

We dare compare our chunking approach with a plain method as BU, which does not take advantage of this pre-process. This is because an experiment on the performance of BU using as an input the test sentences previously chunked (treating these chunks as terminal categories) resulted in a 56% coverage, due to the lack of

<sup>7</sup> As expected, top-down approach produced far worse results, so it was not considered.

<sup>8</sup> We have not worried about the subsequent reduction of coverage, inasmuch as our goal is to compare our approach with our baseline in the same environment.

accuracy of the base-NPs extracted. Anyway, we compared the average results for this subset of 560 sentences, being 12517 edges for the “BU+chunks”, in front of 5833 of *local+chunks2*. We conclude that, if we want to keep simple and completely automatic this pre-process, it definitely makes no sense to apply it to such unidirectional strategies.

PTB-II	Inactive edges	Active edges
BU	6679	53164
Local-noamb.	2569	13777
Local+chunks1	1180	9631
Local+chunks2	<b>634</b>	<b>6524</b>
Local+chunks3	674	6593
Neighb.-noamb.	1488	14402
Neighb.+chunks1	1457	12610
Neighb.+chunks2	982	8849
Neighb.+chunks3	1045	9434

Table 1: Comparative results for corpus PTB-II

Overall figures are shown in Table 1. Both local and *neighbouring* strategies dramatically outperform BU. In general, the use of SCFGs has proven to be successful if an appropriate grammar for a given language is available, together with a large enough labelled corpus of written sentences so that the model parameters can be estimated with acceptable precision. Certainly the *neighbouring* model suffers from data sparseness. This drawback has been partially overcome by using hybrid techniques, as described in (Ageno & Rodríguez 2001).

Focusing on the differences due to the two methods of island selection, we find that the base-NPs approach outperforms the *nonambiguous* one (hereafter, *local-noamb* and *neighbouring-noamb*). Moreover, we observe a more significant improvement when being more selective with the base-NPs (56% for *local+chunks2*). Surprisingly we find that the longest-match strategy (*local+chunks3*), the one most often used in application systems, performs slightly worse than *local+chunks2*. The fact that our stochastic model permits to select the most appropriate base-NPs to be dealt with could explain that it is worth it to try and compensate the lack of accuracy of the base-NPs selection process by adding more alternative base-NPs to the initial set.

## 5 Assessing the Quality of the Parses

So far, the evaluation of the parses returned by each method has been performed on the basis of the number of edges created in order to complete the analysis. Two kinds of measures to account for the quality of the results will be considered next. Once more, let us remark that our aim is to compare our approaches against each other as well as against our baseline BU.

### 5.1 Probabilities

The probability of a parse is usually regarded as the product of the probabilities of the rules involved. Average probabilities were computed for each basic method (see results in Table 2).

	Probability
PTB	0.932
BU	0.636
Local-noamb.	0.774
Local+chunks1	0.658
Local+chunks2	<b>0.832</b>
Local+chunks3	0.822
Neighb.-noamb.	0.389
Neighb.+chunks1	0.513
Neighb.+chunks2	0.655
Neighb.+chunks3	0.625

Table 2: Average probabilities for each method

As expected, the maximum average probability corresponds to the PTB parses. The following method is the local approach. Although the introduction of base-NPs implies a reduction for *chunks1*, again *chunks2* and *chunks3* represent a significant improvement. As to the *neighbouring* model, the change from the *nonambiguous* approach to the base-NPs improves the probability by more than 40% for *chunks2*. Thus, by choosing the appropriate chunking method, *neighbouring* approach also outperforms BU.

### 5.2 Other evaluation metrics

Additionally, we have tried to compute the accuracy of the parses returned by our methods, by means of the metrics described in (Goodman 96) plus two precision rates, namely: Labelled and Bracketed Recall Rates (LR and BR), Consistent Brackets Recall Rate (CBR), and Labelled and Bracketed Precision Rates (LP and BP).

	LR	BR	CBR	LP	BP
Viterbi	0.577	0.633	0.746	0.541	0.592
BU	0.412	<b>0.514</b>	<b>0.705</b>	0.299	0.369
Local-noamb.	0.423	0.497	0.640	0.344	0.403
Local-chunks1	0.402	0.478	0.643	0.298	0.352
Local-chunks2	<b>0.433</b>	0.506	0.634	<b>0.398</b>	<b>0.462</b>
Local-chunks3	0.427	0.500	0.636	0.384	0.449
Neighb.-noamb.	0.373	0.460	0.675	0.230	0.282
Neighb+chunks1	0.402	0.487	0.666	0.275	0.332
Neighb+chunks2	0.401	0.481	0.646	0.309	0.368
Neighb+chunks3	0.395	0.479	0.649	0.301	0.362
“Worse”	0.347	0.445	0.696	0.175	0.223

Table 3: Evaluation metrics for untagged corpus

LR and LP compute recall and precision by considering both the spanning of each constituent of the tree as well as its label. BR and BP are less strict, and account only for constituent matching, ignoring the nonterminal label. CBR is even less strict and regards only the constituents whose intervals cross, that is, that could never be in the same parse tree.

Table 3 shows the obtained results for the 1000 sentences in the test set. “Viterbi” and “worse” parses (the ones maximising and minimising the probability), our upper and lower bounds, are also included.

As expected, the best results correspond to the “Viterbi” parses, and the “worse” ones obtain the worst ranks for all but one measure. Local model, which outperformed BU (in three out of the five measures) using the *nonambiguous* approach, improves results even more when using *chunks3* and (specially) *chunks2*.

*Neighbouring* model, which did not get to beat BU using the *nonambiguous* approach, gets quite comparable values with the *chunks2* approach. Somewhat surprisingly we find that, for the CBR measure, better results are obtained by methods that do not stand out for the other measures, such as even the “worse” parses. The main reason seems that these parse trees are basically composed by unary and binary rules (average length of 1.6 for the rules used by “worse” against 2.1 for the rules by *local-chunks2*), which makes more difficult a crossing bracket to happen.

It is important to compare the different approaches between the upper and lower bounds, as all the results are rather low due to the fact that sentences were not tagged. A new set of experiments was conducted in order to evaluate the effects of tagging the corpus in the accuracy of the results. The test set was previously tagged and then parsed by means of all the *chunks* approaches plus the BU method (see Table 3). Obviously it made no sense to test the *nonambiguous* approaches, as all words in each sentence would have been islands.

	LR	BR	CBR	LP	BP
BU	0.674	0.707	0.814	0.496	0.519
Local-chunks1	0.683	0.705	0.798	0.530	0.547
Local-chunks2	<b>0.750</b>	<b>0.770</b>	<b>0.827</b>	<b>0.668</b>	<b>0.686</b>
Local-chunks3	0.746	0.767	0.826	0.661	0.679
Neighb+chunks1	0.660	0.685	0.789	0.482	0.500
Neighb+chunks2	0.717	0.743	0.810	0.583	0.603
Neighb+chunks3	0.720	0.744	0.810	0.586	0.605

Table 3: Evaluation metrics for tagged corpus

It can be observed that local approaches systematically obtain better measures than BU, as well as all the *neighbouring* approaches but the first one. Increases of around 30% in recall and 25% in precision are accomplished by adding the previous tagging process.

## 6 Conclusions

A parsing method using a stochastic island-driven chart parser preceded by a chunking process for identifying initial islands has been presented. The method has been proved useful for improving parsing performance without lost of coverage. It uses a SCFG from which a grammar of chunks can be automatically extracted. The chunking process can be carried out quite straightforwardly in an efficient way. The island-driven chart parsing process is performed based on a stochastic model which provides the probability of extension of the islands. Two stochastic models, a local model, considering only the SCFG, and a *neighbouring* model, regarding also the adjacent islands, have been developed and compared. Both the probabilities attached to the SCFG and the parameters of the models can be learnt from annotated corpora.

The system has been tested on PTB-II corpus with remarkable results. For instance, the local method using a conventional island-selection mechanism (*local-noamb*) reduces the BU average number of (active + inactive) edges by a factor of 4, whereas the chunking criterion reduces it by 8. We find that, although both methods clearly outperform the baseline BU, the use of a more informed strategy, the base-NPs approach, provides a

significant improvement, specially when only *maximal* and *overlapping* NPs are selected. As to the quality measures considered (the probabilities of the different parses and their accuracy), mostly our parses outperform BU results, obtaining quite comparable figures in all cases. The change of island-selection strategy also improves these evaluation metrics. We are currently investigating extensions such as the ideas of ‘work’ and ‘competitorship’ described in (Blaheta & Charniak 99) to improve both our performance and accuracy.

## Acknowledgements

This work has been partially funded by the EU (IST-1999-12392), and by the Spanish (TIC2000-0335-C03-02, TIC2000-1735-C02-02) and Catalan (1997-SGR-00051) Governments.

## References

- (Abney 96) S. Abney *Partial Parsing via Finite-State Cascades*. In Proceedings of the ESSLI-96 Robust Parsing Workshop, 1996.
- (Ageno & Rodríguez 2000) A. Ageno and H. Rodríguez *Extending Bidirectional Parsing with a Stochastic Model*. In Proceedings of the 3<sup>rd</sup> International Workshop on Text, Speech and Dialogue, 2000.
- (Ageno & Rodríguez 2001) A. Ageno and H. Rodríguez *Probabilistic Modeling of Island-Driven Parsing*. In Proceedings of the 7<sup>th</sup> International Workshop on Parsing Technologies. Beijing, 2001.
- (Blaheta & Charniak 99) D. Blaheta and E. Charniak *Automatic Compensation for Parser Figure-of-Merit Flaws*. In Proceedings of the 37<sup>th</sup> Annual Meeting of the Association for Computational Linguistics. Maryland, 1999.
- (Cardie & Pierce 98) C. Cardie and D. Pierce *Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification*. In Proceedings of the 36<sup>th</sup> Annual Meeting of the ACL and COLING. Montreal, 1998.
- (Charniak et al. 98) E. Charniak, S. Goldwater, and M. Johnson *Edge-Based Best-First Chart Parsing*. In 6<sup>th</sup> Annual Workshop for Very Large Corpora. Montreal, 1998.
- (Ciravegna & Lavelli 99) F. Ciravegna and A. Lavelli *Full Text Parsing using Cascades of Rules: an Information Extraction Perspective*. In Proceedings of the 9<sup>th</sup> Conference of the EACL. Bergen, 1999.
- (Collins 97) M. Collins *Three Lexicalised Models for Statistical Parsing*. In Proceedings of the 35<sup>th</sup> Annual Meeting of the ACL and the 8<sup>th</sup> EACL. Madrid, 1997.
- (Gaizauskas 95) R. Gaizauskas *Investigations into the Grammar Underlying the Penn Treebank II*. Research Report CS-95-25, Univ. of Sheffield, 1995.
- (Geman & Johnson 2000) S. Geman and M. Johnson *Probability and Statistics in Computational Linguistics, a brief review*. Internal Publication, Division of Applied Mathematics, Brown Univ., 2000.
- (Goodman 96) J. Goodman *Parsing Algorithms and Metrics*. In Proceedings of the 34<sup>th</sup> Annual Meeting of the Association for Computational Linguistics, 1996.
- (Marcus et al. 93) M. Marcus, M.A. Marcinkiewicz, and B. Santorini *Building a large annotated corpus of English: The Penn Treebank*. Computational Linguistics, 19(2), pp. 313-330, 1993.
- (Nederhof 2000) M. Nederhof *Practical Experiments with Regular Approximation of Context-free Languages*. Computational Linguistics, 26(1), pp. 17-44, 2000.
- (Ritchie 99) G. Ritchie *Completeness Conditions for Mixed Strategy Bidirectional Parsing*. Computational Linguistics, 25(4), 1999.
- (Satta & Stock 94) G. Satta and O. Stock *Bidirectional Context-Free Grammar Parsing for Natural Language Processing*. Artificial Intelligence, 69, pp. 123-164, 1994.
- (Sikkel & op den Akker 96) K. Sikkel and R. op den Akker *Predictive Head-Corner Chart Parsing*. In “Recent Advances in Parsing Technology”, H. Bunt & M. Tomita, ed., Kluwer Academic, Netherlands, chapter 9, 1996.