PROBABILISTIC MODELLING OF ISLAND-DRIVEN PARSING

Alicia Ageno and **Horacio Rodríguez**
TALP Research Center
Universitat Politècnica de Catalunya (UPC)
Jordi Girona, 1-3. E-08034 Barcelona, Spain
{ageno,horacio}@lsi.upc.es

Abstract

Two methods for stochastically modelling bidirectionality in chart parsing are presented. A probabilistic island-driven parser which uses such models (either isolated or in combination) has been built and tested on wide-coverage corpora. The best results are accomplished by the hybrid approaches that combine both methods.

1 Introduction

Although most methods for context-free grammar parsing are based on a uniform way of guiding the parsing process (e.g. top-down, bottom-up, left-corner), there have recently been several attempts to introduce more flexibility, allowing bidirectionality, in order to make parsers more sensitive to linguistic phenomena ([1],[2],[3]).

We can roughly classify such approaches into head-driven and island-driven parsing. They respectively assume the existence of a distinguished symbol in each rule, the *head*, and certain distinguished words in the sentence to be parsed, the *islands*, playing a central role on the respective parsing approach. While assigning heads to rules is a heavy knowledge intensive task, selecting islands can be carried out straightforwardly: unambiguous words, base NPs (in the case of textual input), accurately recognised fragments (in the case of speech), might be considered islands.

The problem is, however, that simply starting with islands or heads does not assure improvements over the basic parsing schemata. Only with appropriate heuristics for deciding where and in which direction to proceed can we restrict the syntactic search space and obtain better results, counteracting the overhead that these more complex algorithms suppose.

What we present here are two methods for modelling bidirectionality in parsing, as well as a bidirectional island-driven chart parser that uses such stochastic models. In the remainder of this paper we describe our parser and stochastic models in section 2. We discuss the planning of the experiments and their results in section 3, and the evaluation of the quality of the results in section 4. Finally, we present our conclusions in section 5.

2 The Stochastic Island-Driven Methodology

In island-driven parsing, the conventional left-to-right approach of chart parsing is enhanced with two features: the bidirectionality (parsing can take place either left-to-right or right-to-left) and the islands

(dynamically determined positions of the sentence from which the process starts). Island-driven flexibility permits the use of optimal heuristics that cannot be applied to unidirectional strategies. These heuristics are based on two stochastic models, which allow to select the most probable island, to be extended to the most probable side. Our models provide sort of Figures of Merit (FOMs) as [4] or [5], in order to deliver a single best-first analysis, but based on the concept of islands and applying these FOMs to their extension.

Our island-driven probabilistic chart parser performs a combination of bottom-up expansion and top-down prediction (the latter to be sure that no constituent is lost whenever no island has been selected within a portion of the input) by managing two agendas, guided by the stochastic parameters. The algorithm has been already described in [6], hence we will focus on the description of the stochastic models.

Given a Stochastic Context-Free Grammar (SCFG), what we try to model is the likelihood of extending (either to right or left) an (either inactive or active) edge, or partial analysis, growing islands of 'certainty'. Two basic models have been studied. The local model is static, as it just takes into account grammatical information. The *neighbouring* model considers also the immediate environment around each island, that is, the islands and *gaps* (the segments of the input sentence spanning between adjacent islands) surrounding it.

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, $[A, i, j]$[1] is an island labelled *A* spanning positions *i* to *j*, and *{left|right}_corner* are functions from N x 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|right} corner:

$$\forall A \in N, a \in T : right\_corner\ (A,a) = P(A >> a / G)$$[2]

Similarly, *{left|right}_corner** are functions from N x 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 and stored in two structures (the *Lreachability* and the *Rreachability* tables), so that:

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

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

- For expansion to the left of (or prediction to the left from) an active edge (*lt* being the list of terminal categories of word w_{i-1}):

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

Special cases where either α or β are empty are also considered. Expansions and predictions to the right are symmetrically defined.

Computing the *reachability* tables is far from being a trivial problem. We have using an extension of [6]'s approach for massively recursive grammars, extending it to deal with bidirectionality. The interdependencies between nonterminals are represented as a linear equations system. The problem has been that we

---

[1] We will employ the usual double dotted rule notation for the edges.

[2] *P(A>>a/G)* denotes the probability that, starting with the nonterminal *A*, successive application of rules in grammar G produces a sequence beginning with terminal *a*.

encountered an unfeasible dimension for our grammars. Therefore, the process has been divided into three steps:

1. Computation of the strongly connected components.
2. Solution of a linear system for each component.
3. Development of an algorithm for the combination of the obtained results.

2.2 The *Neighbouring* Model

In this approach, in order to take the decision of extending an island we will consider the information provided by the *neighbours*, that is, the islands and *gaps* immediately surrounding such island, as well as distances to them (the lengths of the *gaps*). 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 previously 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[3] (see Figures 1 to 3) :

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

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

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

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

$$P^i(d / r, A, B)$$

$$P^i_{acc}(d/\mathrm{A}, \mathrm{B}) = \sum_{r \in R^i} P^i(d/r, A, B)$$

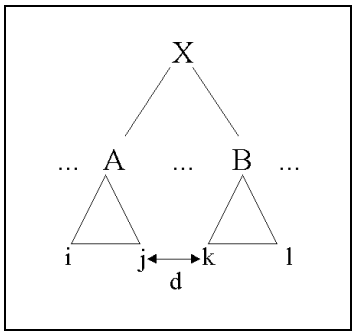

Figure 1: Relationship R^1

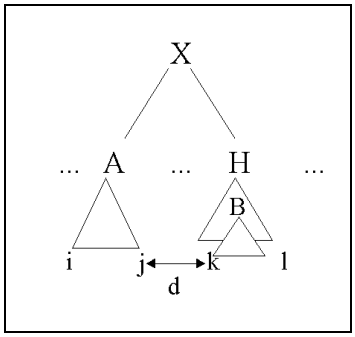

Figure 2: Relationship R^2

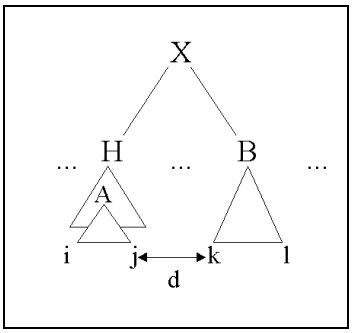

Figure 3: Relationship R^3

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, considering average distances between islands. The application of this model to the expansions and predictions to the right is as follows (left direction symmetrical):

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

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

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

The first addend accounts for cases of *A* and *B* being in the same rule right-hand side (RHS), while the second one considers all possibilities of *B* being derived in 1 or more steps from a nonterminal *H* which is in the same rule RHS as *A*, plus all possibilities of *A* being derived from a nonterminal *H* which is in the same rule RHS as *B*.

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

$$P_{edge}^{right}([A \rightarrow \beta.A_l \alpha A_r.\gamma, i, j]/G, w, [B, j+d, k]) = P^1(d / r, A_r, B) + \sum_{\gamma_i \in N, i \leq d, 0 \leq d-l \leq 3} prob(|\gamma_1..\gamma_{i-1}| = l) \times P_{acc}^2(d - l / \gamma_i, B)$$

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* is a recursive function that, given the 'trained' G, provides for the distribution of probabilities of the lengths of any subsequence of terminal and nonterminal symbols.

Several heuristics have been adopted as regards the *neighbouring* strategy. First of all, *neighbouring* probabilities applied to top-down prediction have empirically shown to generate a significant edge overhead (see comments in section 3.2). Therefore, some limitations have been imposed to their application:

1. For the initial determination of the edges to be used for prediction, local probability acts as a filter (that is, only when local is non-zero will the *neighbouring* probability be used to determine if and when the edge will be used for prediction); for distances $d>2$, local approach is directly used.
2. Subsequent recursive predictions will be guided only by local probabilities, limited in turn by a threshold. This threshold has been empirically set.

In order to avoid the maximum number of computations at run time, the probabilities mentioned above are pre-computed, using the frequencies of distributions of lengths learnt from the training corpus. These data are stored in two tables containing the probabilities of each pair of categories to be at distances from 0 to *limit*, as well as a single case for distances greater than *limit*. Simpler tables, to account for the cases of extensions/predictions to the left of the first island of the sentence (as well as to the right of the last one), are also calculated.

3 The Experiments

3.1 Setting

We hasten to emphasise that our experiments have been aimed at comparing, in the same environment, the performance of the local versus the *neighbouring* approach (including hybrid versions) as well as the performance of our island-driven approach with the classical bottom-up[4], our baseline. By 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 [8] or others do, but it can be applied to any existent SCFG. It has been tested using several artificial grammars, and even a limited-coverage

[4] As expected, the top-down approach led to far worse results, so it was discarded.

grammar for Spanish corpus Lexesp [6]. However, we wished to compare our strategies using a grammar as close as possible to a real one, so we chose Penn Treebank II [9], 1,25Mw. The grammar underlying the bracketing has been extracted, but its size (17534 rules) was simply too big to contemplate for our parser. Therefore, we have applied a simple thresholding mechanism to prune rules from the grammar [10], consisting of removing all rules that account for fewer than n% of rule occurrences of rules in each category. We have used n=22, obtaining a grammar with 941 rules, 26 nonterminals and 45 terminals[5].

In order to estimate the parameters of both models, a training corpus of 49000 sentences has been 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 sparseness problems. A corpus of 1000 sentences extracted randomly from sections 13 and 23 (from those sentences covered by our grammar) was used for testing. Average sentence length of the test set was 21.5 words.

The criterion for the selection of the islands has been to consider as initial islands those non-ambiguous words. Therefore, the analysis of these sentences must be performed without previous PoS-tagging, i.e., words have been *ambiguated*, they own all their possible tags and not only those contained in the PTB. However, section 4 describes some results for a tagged corpus, obtained using base-NPs as initial islands. Additionally, other criteria based on the syntactic ambiguity of the categories according to the grammar are currently being tested.

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.

3.2 Results

Overall figures are shown in Table 1. In general, the use of SCFG has proven to be relatively successful if an *appropriate* grammar for a given language is available, together with a large enough labeled corpus of written sentences so that productions probabilities can be estimated with acceptable precision. A problem with inducing grammars from the PTB is that, because the trees are very flat, there are lots of rare kinds of flat trees with many children. In our case, the flatness itself is a benefit for our methodology, as well as the larger length of the right-hand sides of its rules (3,59 in average), as it allows the expansion of several islands at the same level. However, the variety provokes that the *neighbouring* method suffers from data sparseness. As mentioned in section 2.2, by strict application of *neighbouring* probabilities we do not get a full coverage. Local model is empirically demonstrated to be the best method as a back-off (versus BU). Therefore, henceforth by *neighbouring* we will mean the *neighbouring* model plus a back-off to local method when no analysis is found.

PTB-II	Local	*Neighbouring*	BU
Inactive edges	2569	1488	6679
Active edges	13777	14402	53164

Table 1: Comparative results for corpus PTB-II[6]

[5] 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 framework.

[6] *Neighbouring's* prediction threshold is 0.1.

Detailed Results

We have tried to test the behaviour of each method depending on the kind of sentences being parsed. The idea is to be able to figure out in which cases a more informed model should be applied, using then a sort of hybrid method which chooses the approach on the way. Therefore, the corpus has been divided into groups according to several criteria, and the average number of edges needed to parse the sentences of each group has been computed for each method. The performance of our approaches is quantitatively more appealing than BU´s for all cases, though differences vary and may indicate in turn different behaviours of the models. The examined criteria have been:

1) Length of the sentence (L = #words), starting from group 0 (L<10) to 9 (L>38).
2) Ambiguity rate, A = #tags / #words. Ambiguity groups go from 0 (A<2) to group 9 (A>3.5).
3) Density of islands, D = #islands / #words. Densities span from group 0 (D<0.25) to group 9 (D>0.70).
4) Maximum Island Distance, MID = length of the longest *gap*. We consider MID values from MID<2 (group 0) to MID>11 (group 9).
5) Island Dispersion, DI = Σ length_of_*gaps* / #*gaps*. Dispersions span from group 0 (DI<1.5) until group 9 (DI>7).

Figures 4 and 5 depict the results for criteria 1 and 3.

As regards sentence lengths, notice that both local and *neighbouring* always outperform BU, the performance gap increasing with the sentence length. Local's performance keeps above *neighbouring's* in all groups except for the longest sentences, which is encouraging if we mean to deal with real corpus. As sentences get more ambiguous, BU's performance degrades notoriously, whereas our methods´ is smoother and more nearly monotonic. Regarding the island density, as the number of islands gets close to the total number of words, performance of basic BU is more comparable to local and *neighbouring* (though the latter is always better). MID's graphic presents a suspicious similarity with length's one (though the increment of number of edges is more gradual). By computing the crossover between both measures, we have seen that it may happen because the cases of largest *gaps* often overlap with those of the largest sentence lengths. Once more local and *neighbouring* dispersions are quite comparable, and smoother and more nearly monotonic than BU's.

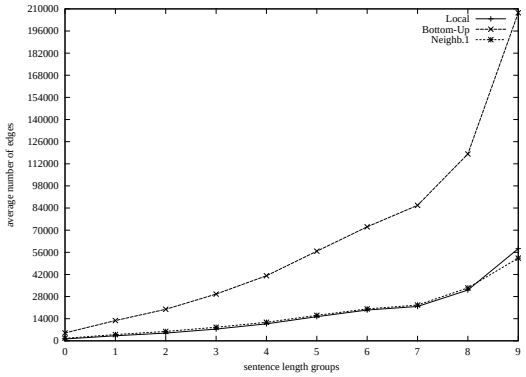

Figure 4: Average #edges/ length

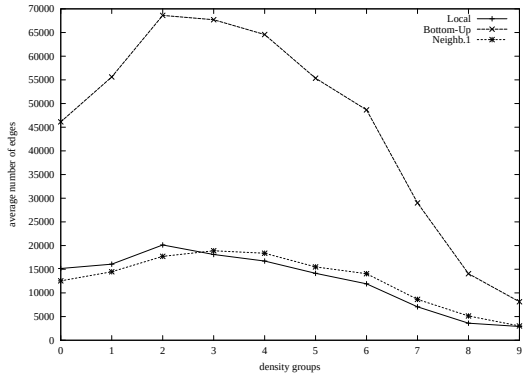

Figure 5: Average #edges/ island density

Hybrid Results

So far, in order to reach a complete coverage of the corpus for the *neighbouring* model, a back-off is performed whenever no parse is found. Using this strategy, *neighbouring's* performance does not improve local's. Hence, why not try the back-off before? Let's present two new heuristic strategies, where the

difference between both will lie in the criteria employed to change to the local approach. In the first one, we will change when a percentage of the sentence has been covered by the islands that are being extended. In the second one, whenever a certain number of extension-prediction loops have been performed. Needless to say both the optimum percentage and number of cycles are computed empirically according to a test set. In Figure 6, we have represented the average number of edges for a coverage percentage from 0% (purely local approach) to 100% (purely *neighbouring* approach). There is a clear minimum for 40% of coverage (hereinafter *neighb-40%*), and it can be seen that performance degrades for both pure approaches. As to the second method, we find a clear minimum for 4 cycles (*neighb-4cycles*), and again performance degrades for non-mixed approaches.

A more thorough study reveals that, one main advantage of the *neighbouring* approach in front of the local one is the extension at lexical level. That is why simply starting the parsing process by introducing terminal and pre-terminal edges into the extension heap according to *neighbouring* probabilities, and then backing off to the local model, represents an improvement in most cases. *Neighbouring* probabilities guide the analysis at a preliminary stage of the extension of the islands, backing off to the local model whenever the former approach would have to start a much more blind process of prediction. The guiding potential of *neighbouring* approach during the extension is higher but, due to the data sparseness, lots of potentially possible cases are assigned probability zero and must be left behind for prediction, which introduces far more overhead than the extension.

Besides, whenever a back-off to the local model must be performed, all lexical edges (and not only the islands) have to be re-introduced in the extension heap in order to be sure of getting a full coverage. The fact that in some cases this mode of operation gets a better performance than a pure local or *neighbouring* approach might indicate that in these cases, the original islands have not been correctly chosen, and point at a new direction of research in other methods of selection [12].

The criteria described above have been applied to the complete test set for both optima obtained. Results can be seen for the *cycles* approach and the *length* criterion in Figure 7. Except for the single case of MID strictly smaller than 2, the *neighb-4cycles* approach clearly outperforms both purely local and *neighbouring* approaches. Similar results are obtained for the *neighb-40%* approach.

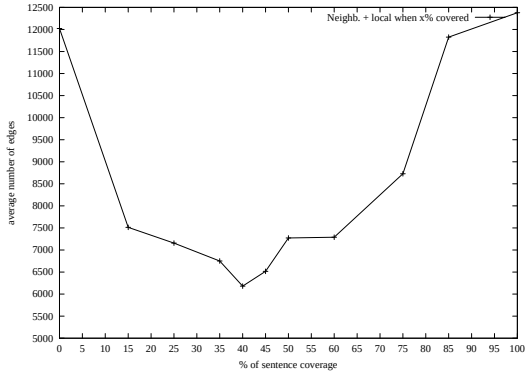

Figure 6: Average #edges/sentence for each percentage of coverage of the sentence

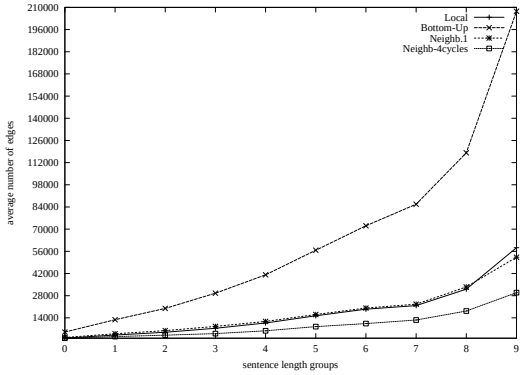

Figure 7: Average #edges/sentence for each group of sentence lengths and for each method

Thresholding

It has been mentioned in section 3.1 that the *neighbouring* statistical parameters learnt by our training process might not be correct, due to the sparseness of the input data. Several experiments have been carried

out in order to evaluate the most adequate threshold from which to consider that a probability is not informative enough (as it is learnt by a not sufficient number of occurrences), making necessary a back-off to the local model. For each matrix of pre-computed probabilities, the distribution of values has been studied, and according to it, a threshold has been defined. For a subset of sentences, a battery of experiments has been performed, each one applying the threshold gradually to the following probabilities:

1. Extension probabilites.
2. Prediction probabilities.
3. Lexical extension probabilities.
4. A special treatment is devoted to certain prediction probabilities. When distances between adjacent islands are larger than the parameter *limit* defined by the user, the lack of occurrences in the training set is particularly critical. This leads to a typical situation: lots of prediction edges entering the prediction heap with high probabilities, learnt by means of a ridiculous number of occurrences. Prediction explodes locally, not allowing the use of other more suitable pending edges situated in other areas of the sentence. As a result, the *neighbouring* probabilities are not informative anymore as a guide to the process. In order to restrict the effects of this situation, another type of threshold (*Tp*) is introduced.

Figure 8 shows the comparison of the average number of edges for the different thresholds. Method 0 is local[7], 1 corresponds to conventional *neighbouring*, and methods 2 to 17 are more and more restrictive applications of thresholds. A particularly steep fall is found from method 7, which is when the application of *Tp* starts. The following methods correspond to different values of *Tp*. An improvement of around 50% is obtained with respect to local and *neighbouring* performance.

Once the different values and combinations of thresholds tested, the optima have been applied to the whole test set. The results are shown in Figure 9. Method 0 corresponds to BU, 1 is local and method 2, *neighbouring*. Methods 3 and 4 are applications of the first three thresholds (the only difference being respectively the application of the first and second threshold for *neighbouring* lexical probabilities). Thus, we can see the difference with respect to methods 5 and 6, which correspond respectively to applications of previous thresholds plus the most optimal threshold *Tp* (hereinafter respectively *neighb-thresh1* and *neighb-thresh2*). The number of edges significantly decreases with respect to the other methods (around 45% for local and *neighbouring*, not to mention BU!).

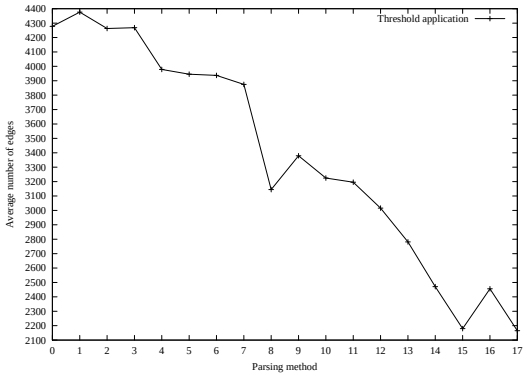

Figure 8: Average #edges/sentence for each threshold

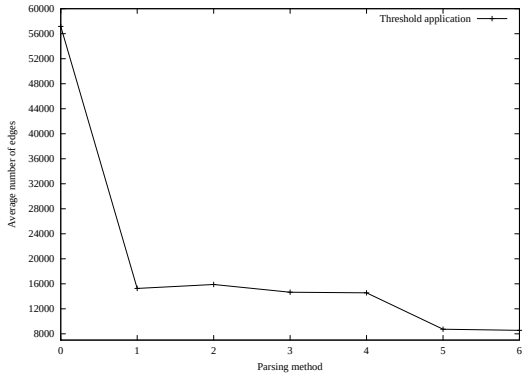

Figure 9: Average #edges/sentence for each method

[7] BU has been avoided in purpose as its quantity of edges is significantly larger, which would have prevented the rest of the data from being seen in detail.

4 Assessing the Quality of the Parses

Up to now, the evaluation of the parse trees returned by each method has been performed on the basis of the number of edges created in order to complete the analysis. Nothing has been done as to the *quality* of the result. In this line, two measures have been considered: probability and accuracy.

4.1 Probabilities

The probability of a parse tree is usually regarded as the product of the probabilities of the rules involved. Average probabilities were computed for each basic method, as well as for the most optimum hybrid ones. Results can be seen in Table 2: as expected, the maximum average probability corresponds to the PTB parses. The following method is the local approach, being the third rank occupied by *neighb-40%*. BU is ranked fourth.

	Probability
PTB	0.932
BU	0.636
Local	**0.774**
Neighbouring	0.389
Neighb-40%	0.641
Neighb-4cycles	0.590
Neighb-thresh1	0.609

Table 2: Average probabilities for each method

4.2 Accuracy Metrics

We have tried to compute the similarity of the PTB parse to the parses returned by our methods, both the homogeneous and the hybrid ones. The metrics computed are those described in [11] 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).

Table 3 shows the obtained results for the 1000 sentences in the test set. As to the hybrid methods, we have included the ones giving optimum results. It is important to emphasize that 'Viterbi' parses (the ones which maximize the probability) and 'worse' parses (the ones that minimize the probability) are going to be our upper and lower bounds, since the specific features of our framework (partial grammar, non-tagged sentences) do not allow to compare our results with other systems.

As to the comparison between our methods and basic BU, local model presents considerably better results, followed by *neighb-40%*. Specially striking are the CBR figures: better results are obtained by methods that do not stand out for the other measures, such as even the "worse" parse trees. Seemingly the reason is that these parses are basically composed by unary and binary rules (average length of 1.6 for the rules used by 'worse' versus 2.0 for those in local), which makes more difficult a crossing bracket to happen.

Although only the hybrid methods giving better average number of edges have been included in Table 3, we have also studied the effects of the different back-off strategies on the accuracy. In general, it can be seen that, with the exception of the CBR metric, accuracy starts improving for the first stages of the hybrid approaches (until number of cycles equals 3, until coverage equals 35%), and then gradually degrades as back-off to local is postponed.

	LR	BR	CBR	LP	BP
'Viterbi'	0.577	0.633	0.746	0.541	0.592
BU	0.412	**0.514**	**0.705**	0.299	0.369
Local	**0.423**	0.497	0.640	**0.344**	**0.403**
Neighbouring	0.373	0.460	0.675	0.230	0.282
Neighb-40%	0.412	0.483	0.641	0.318	0.370
Neighb-4cycles	0.394	0.469	0.634	0.294	0.348
Neighb-thresh2	0.405	0.483	0.641	0.306	0.364
'Worse'	0.347	0.445	0.696	0.175	0.223

Table 3: Evaluation metrics for untagged corpus

Additionally, we show the results of a new set of experiments, in which we have considered previously extracted base NPs as initial islands, thus allowing to start from a tagged corpus (see Table 4). This approach is described in detail in [12], however, we just wanted to show how the fact of both working with a disambiguated corpus and selecting the subset of the test set for which parses in the PTB contain only rules belonging to our reduced grammar can make the accuracy increase. Obviously our grammar is restricted, and we have started from a correctly disambiguated corpus, which is unrealistic, for any tagged corpus would imply the existence of a certain error rate. To what extent would this error affect the accuracy of the parses, the same way that our starting from disambiguated corpus has been affected, remains unexplored.

	LR	BR	CBR	LP	BP
BU	0.824	0.837	0.897	0.676	0.687
Local	**0.906**	**0.914**	**0.934**	**0.888**	**0.896**
Neighbouring	0.860	0.875	0.906	0.778	0.790

Table 4: Evaluation metrics for tagged corpus

5 Conclusions and Future Work

Two stochastic models for dealing with bidirectionality in island-driven chart parsing have been presented. The models provide for the probability of extension of each island given either the stochastic grammar (local model) or both the grammar and the immediately adjacent islands (*neighbouring* model). A chart parser has been built that uses such models, either independently or in combination. Several experiments with a broad coverage (though not complete) grammar of English have been carried out. Parsing performance has been analysed according to several criteria, our approaches dramatically outperforming the baseline BU strategy. Several hybrid methods which combine local and *neighbouring* approaches have also been defined, improving the performance of the single ones. Other evaluation metrics have been considered, including the probabilities of the different parses and its similarity to the PTB ones. Local and *neighb-40%* present the best results.

Performance already improved, the accuracy remains to be increased. In fact, the idea of our hybrid approaches has the same motivation as that of the ideas of 'work' and 'competitorship' of [5], thus pointing out a possible extension for improving both our performance and accuracy; [5] also provides interesting ideas to deal with the data-sparseness which may be applied to our *neighbouring* model.

It has also been mentioned that another source of improvement could be the method of selection of the islands. Several refinements are currently being evaluated, such as considering criteria based on both the degree of ambiguity of the lexical categories of each word, and the degree of ambiguity of the categories according to the grammar.

Acknowledgements

Thanks to the anonymous reviewers for their valuable comments.

References

[1] Giorgio Satta and Oliviero Stock. *Bidirectional CFG Parsing for Natural Language Processing*. Artificial Intelligence, 69:123-164, 1994.

[2] Graeme Ritchie. *Completeness Conditions for Mixed Strategy Bidirectional Parsing*. Computational Linguistics, 25(4), 1999.

[3] Klass Sikkel and Rieks op den Akker. *Predictive Head-Corner Chart Parsing*. In Recent Advances in Parsing Technology, chapter 9, pages 169-182. Kluwer Academic, Netherlands, 1996.

[4] E. Charniak, S. Goldwater, and M. Johnson. *Edge-based Best-first Chart Parsing*. In Proceedings of the 6th Annual Workshop for Very Large Corpora. Montreal, 1998.

[5] D. Blaheta and E. Charniak. *Automatic Compensation for Parser Figure-of-Merit Flaws*. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. Maryland, 1999.

[6] A. Ageno and H. Rodríguez. *Extending Bidirectional Parsing with a Stochastic Model*. In Proceedings of the 3rd International Workshop on Text, Speech and Dialogue. Brno, 2000.

[7] See-Kiong Ng and Masaru Tomita. *Probabilistic LR Parsing for General Context-Free Grammars*. In Proceedings of the 2nd International Worshop on Parsing Technologies. Cancun, 1991.

[8] M. Collins. *Three Lexicalised Models for Statistical Parsing*. In Proceedings of the 35th Annual Meeting of the ACL and the 8th EACL. Madrid, 1997.

[9] M. Marcus, M.A. Marcinkiewicz, and B. Santorini. *Building a large annotated corpus of English: The Penn Treebank.* Computational Linguistics, 19(2):313-330, 1993.

[10] R. Gaizauskas. *Investigations into the Grammar Underlying the Penn Treebank II*. Research Report CS-95-25, University of Sheffield, 1995.

[11] Joshua Goodman. *Parsing Algorithms and Metrics*. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, 1996.

[12] A. Ageno and H. Rodríguez. *Chunking + Island-Driven Parsing = Full Parsing*. In Proceedings of the 3rd Conference on Recent Advances in Natural Language Processing. Tzigov Chark, Bulgaria, 2001.