Extending Bidirectional Chart Parsing with a Stochastic Model

Alicia Ageno and Horacio Rodríguez

LSI Department. Universidad Politècnica de Catalunya (UPC)
Jordi Girona, 1-3. E-08034 Barcelona, Spain
{ageno, horacio}@lsi.upc.es

Abstract. A method for stochastically modeling bidirectionality in chart parsing is presented. A bidirectional parser, which starts analysis from certain dynamically determined positions of the sentence (the *islands*), has been built. This island-driven parser uses the stochastic model to guide the recognition process. The system has been trained and tested over two wide-coverage corpus: Spanish Lexesp and English Penn Treebank. Results regarding comparison of our approach with the basic Bottom-Up are encouraging.

1 Introduction

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 recently been several attempts to introduce more flexibility, for instance allowing bidirectionality, in order to make parsers more sensitive to linguistic phenomena (see [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 quite straightforwardly: unambiguous words, base NPs (in the case of textual input), accurately recognized 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 therefore obtain better results, coming through the obvious overhead that these more complex algorithms suppose.

What we present here is a method for modeling bidirectionality in parsing, as well as a bidirectional island-driven chart parser that uses such a stochastic model. Our framework accounts for bidirectional expansion of partial analysis, which improves the predictive capabilities of the system.

In the remainder of this paper we describe the parsing algorithm we have built for testing the stochastic model in section 2, present this model in section 3, discuss the planning of the experiments and their results in section 4, and give conclusions in section 5.

2 The Parsing Algorithm

The conventional left-to-right approach of chart parsing is enriched with bidirectionality: parsing is started from several dynamically determined positions of the sentence (the *islands*), and then proceed outward in both directions. Island-driven flexibility allows for the use of optimal heuristics that could not be applied to unidirectional strategies. These heuristics are based on a stochastic model, which will be described below.

In island-driven parsing one must deal with cases in which no island at all has been selected within the portion of the input where a constituent is required by the surrounding analyses. Hence the parser must employ *top-down prediction* to be sure that no constituent is lost. Obviously, this *prediction* may take place either at the constituent's left or right boundary. Therefore, we'll talk about *prediction* to the right or to the left.

The parsing algorithm works by following an agenda-based approach. A priority queue, implemented as a heap, is used to deal with the idea of choosing the most probable *island*, according to the stochastic model, to be extended in the most probable side. In fact, the heap's sorting criterion will always be a real number representing a probability attached in a way or another to each chart edge. Two different instances of heap are currently used by the algorithm (though with identical type of contents): the *extension heap* and the *prediction heap*. An element of any of both heaps consists of a bidirectional chart edge (either active or inactive at the *extension heap*, always active at the *prediction* one), a direction attribute indicating whether the edge must be extended/used for prediction to the left or to the right, and a probability attribute stating the probability of extension/fruitful prediction of the edge in question to the indicated direction. Null probabilities are not dealt with at all. The algorithm consists of a loop composed by two stages:

1. A purely bottom-up phase which operates with the *extension heap* as an agenda. It extends the bidirectional chart edges contained in the heap and in turn might add new elements to it, always according to the attached probability. At the very first step of this phase, only those inactive edges representing *islands* are taken into account. The order in which the *extension* (if any) of the existing *islands* to the possible sides will be carried out is therefore determined by the computed probabilities (though once the process started up, new elements with higher probabilities may be added that would delay the *extension* of certain *islands*).
2. Whenever the first phase does not lead to a complete analysis, a top-down *prediction* phase is started. It uses a *prediction heap* which will be updated at the beginning of every step of this type, only with those active edges adjacent to a *gap*[1] (and not used in a previous *prediction* phase yet), always according to a computed probability for each edge and direction. Therefore, a *coverage* structure must be maintained, storing which elements of the sentence form part of an *island*. This second phase lasts until *coverage* is incremented (i.e. one of the *islands* grows a word to one side), which is when we will go back to the first stage, presumably with a non empty *extension heap*. The key idea is to limit *prediction* as much as

[1] *Gaps* are segments of the input sentence spanning between adjacent *islands*.

possible, going back to the *extension* phase as soon as an increment of *coverage* is detected.

3 The Stochastic Model

Given a stochastic CFG, what we try to model is the likelihood of extending (either to the right or to the left) an (either inactive or active) arc, or partial analysis. Two basic models have been studied. The first one, the *local* model, is static, as it just takes into account grammatical information. The second one considers as well the immediate environment around the *island* being dealt with, that is, the *islands* and *gaps* immediately surrounding each *island* (in fact, bidirectional strategies can be used in restricting the syntactic search space for gaps surrounded by two partial analyses). The results obtained for this latter method, the *neighbouring* model, have not entailed relevant improvement over the *local* one yet. Therefore, we'll concentrate on the first one.

The *local* approach is based on regarding the probability of an arc 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 (according to the *expansion*/*prediction* direction)[2]. We'll employ the usual (two-)dotted rule notation for the arcs. Being G an stochastic Context Free Grammar, T the set of terminal symbols of G, N the set of non terminal 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 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. {left|right}_corner* are functions from N x T* to [0,1], being {left|right}_corner* (A, la) the probability that a derivation tree rooted A could have any of the symbols of list la as a {left|right} corner.

$$\forall A \in N, a \in T : right\_corner(A,a) = P(A >> a / G) \tag{1}$$

$$right\_corner*(A,la) = \sum_{a \in la} right\_corner(A,a) \tag{2}$$

Left-corner probabilities are symmetrically defined.

These probabilities are pre-computed and stored in two structures (the *Lreachability* and the *Rreachability* tables), which can be efficiently accessed:

- For *expansion* to the left of an *island* (inactive arc) labeled A:

$$P^{left}_{island}([A, i, j]/\mathrm{G}, \mathrm{w}) = \sum_{R_i : X \rightarrow \alpha A} P(R_i) \tag{3}$$

[2] Following [4]'s notation.

- For *expansion* to the left of (or *prediction* to the left from) an active arc[3]:

$$P_{arc}^{left}([A \rightarrow \alpha B.\beta.\gamma, i, j]/\mathrm{G}, \mathrm{w}) = right\_corner*(B, lt) \qquad \textbf{(4)}$$

Expansions and *predictions* to the right are symmetrically defined.

4 The Experiments

In order to compare the performance of the *local* approach with the classical left-to-right *bottom-up*[4], we have carried out a series of experiments. On one hand, we have used the Lexesp Spanish corpus (5,5Mw), and a grammar for Spanish including 704 rules, 123 non terminal symbols and 310 terminal (Parole compliant) ones [5]. The corpus was simply morphologically analyzed. Hence it had to be syntactically analyzed with the bottom-up chart parser in order to produce a training corpus. Probabilities attached to the grammar rules were learnt by means of this corpus of 10000 sentences, while a corpus of 1000 sentences was reserved for testing.

On the other hand, we have used English Penn Treebank II [6], 1,25Mw. The grammar underlying the bracketing has been extracted, but its size (17534 rules) is simply to big to contemplate for our parser. Therefore, and given that many of the rules occur so infrequently, we have applied a simple thresholding mechanism to prune rules from the grammar [7]. This mechanism consists simply of removing all rules that account for fewer than n% of rule occurrences of rules in each category. In our case we have used n=22, obtaining a grammar with 941 rules, 26 non terminal symbols and 45 terminal ones. This reduction of the grammar has shown to keep a coverage of 60% over the test corpus. Probabilities attached to the grammar rules have also been learnt using for the training process the complete corpus (49208 sentences). A corpus of 1000 sentences extracted randomly from directories 13 and 23 was used for testing.

The criterion chosen for the selection of the *islands* has been considering all non-ambiguous words, taking into account that we are dealing with a categorized but non tagged corpus. Efficiency has been measured in terms of the number of inactive and active arcs created during the parsing process.

Global results corresponding to both corpus are shown in Table 1. In general, the use of SCFG has proven to be successful if a “good” grammar for a given language is available, together with a large enough labeled corpus of written sentences so that productions can be estimated with acceptable precision. In the case of the *Spanish experiments*, the quality of the grammar, as well as the fact that the training process had to be performed from analyses of the bottom-up parser, has shown to be relevant for the global results.

[3] *lt* being the list of terminal categories attached to w_{i-1}

[4] Trying Top-Down approach led, as expected, to far worse results.

Table 1. Comparative results for corpus PTBII and Lexesp

	Local	Bottom-Up
PTBII		
Inactive arcs	2569	6679
Active arcs	13777	53164
Lexesp		
Inactive arcs	116	143
Active arcs	648	645

The corpus has been divided into groups according to the length of the sentences starting from group 0 (length < 10) to group 9 (length > 38). The tendencies for both corpus and languages have been similar. Part of the relevant results obtained for PTBII are shown in Fig. 1. It is obvious that, while *bottom-up's* performance degrades as the length increases, *local's* remains rather constant.

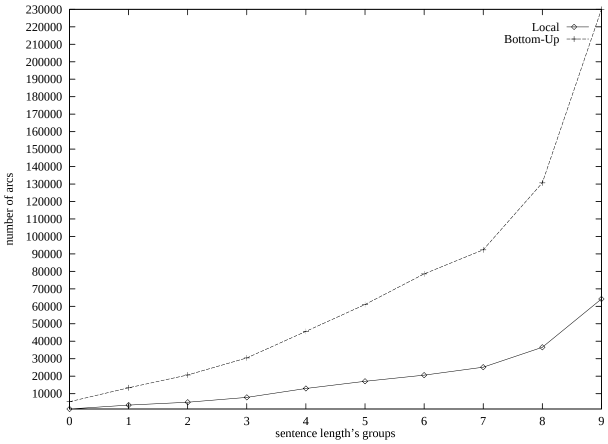

Fig. 1. Average number of arcs for each method and group of sentences of a certain length

Other criteria of classification of sentences have been tested, including *ambiguity rate*, *MID* (Maximum Island Distance) and *island density*. The results corresponding to PTBII and the latter measure are shown in Fig. 2.

5 Conclusions

A stochastic model for dealing with bidirectionality in island-driven chart parsing has been presented. The model, a static one, provides for the probability of extension of each *island* given the stochastic grammar. A chart parser has been built that uses such a model. Several experiments with broad coverage grammars of English and Spanish have been carried out. Parsing performance has been analised according to several metrics (sentence length, ambiguity rate, MID and island density). Our approach clearly outperforms the baseline bottom-up strategy.

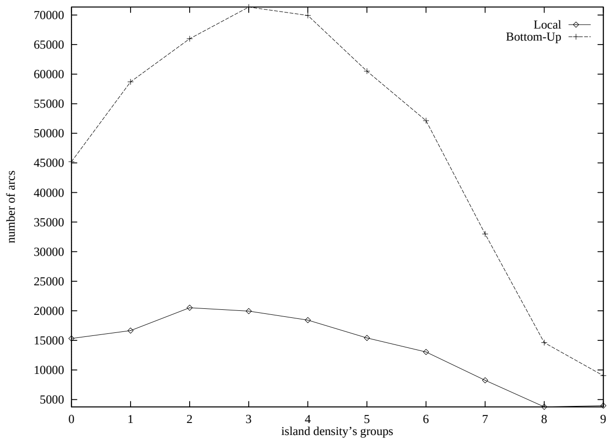

Fig. 2. Average number of arcs for each method and group of sentences of a certain island density

References

1. Satta, G., Stock, O.: Bidirectional Context-Free Grammar Parsing for Natural Language Processing. Artificial Intelligence, 69 (1994) 123-164
2. Sikkel, K., op den Akker, R.: Predictive Head-Corner Chart Parsing. Recent Advances in Parsing Technology. Harry Bunt and Masaru Tomita (eds), Kluwer Academic, Netherlands, chapter 9 (1996) 169-182
3. Ritchie, G.: Completeness Conditions for Mixed Strategy Bidirectional Parsing. Computational Linguistics, Vol. 25, Number 4 (1999) 457-486
4. Jelinek, F., Lafferty, J.D.: Computation of the Probability of Initial Substring Generation by Stochastic Context-Free Grammars. Computational Linguistics, Vol. 17, Number 3 (1991) 315-323
5. Castellón, I., Civit, M., Atserías, J.: Syntactic Parsing of Unrestricted Spanish Text. Proceedings of the 1st International Conference on Language Resources and Evaluation, Granada (1998) 603-609
6. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: Annotating Predicate Argument Structure. Distributed on The Penn Treebank Release 2 CD-ROM by the Linguistic Data Consortium (1995)
7. Gaizauskas, R.: Investigations into the Grammar Underlying the Penn Treebank II. Research Report CS-95-25, University of Sheffield (1995)