Island Parsing and Bidirectional Charts

Oliviero STOCK (+*)

Rino FALCONE (*%)

Patrizia INSINNAMO (%)

(+) Istituto per la Ricerca Scientifica e Tecnologica, 38050 Povo, Trento, Italy

(*) Istituto di Psicologia-Consiglio Nazionale delle Ricerche, Rome, Italy

(%) Fondazione Ugo Bordoni, Rome, Italy

Abstract

Chart parsing is directional in the sense that it works from the starting point (usually the beginning of the sentence) extending its activity usually in a rightward manner. We shall introduce the concept of a chart that works outward from islands and makes sense of as much of the sentence as it is actually possible, and after that will lead to predictions of missing fragments. So, for any place where the easily identifiable fragments occur in the sentence, the process will extend to both the left and the right of the islands, until possibly completely missing fragments are reached. At that point, by virtue of the fact that both a left and a right context were found, heuristics can be introduced that predict the nature of the missing fragments.

1. Introduction

The goal of using "high level" knowledge sources in recognizing continuous speech is to reduce the hypotheses space generated by acoustic-phonetic analysis (and possibly to implement an interpretation of the utterance) (see for instance Walker 1976, Stringa 1988). Decodifying the vocal signal is a process that must take into account phenomena such as the coarticulatory processes typical of continuous speech and the presence of many sources of variability of the signal (anatomic characteristics of the speaker, emission speed, prosody and so on). These phenomena have as a consequence the fact that, at the level of acoustic-phonetic analysis, it is extremely uncertain how to segment the signal and what labels to give to the segments. Therefore acoustic-phonetic analysis generates a space of possible interpretative hypotheses of the signal; in general, the likelihood of each lexical hypothesis is given a score. A matrix of lexical hypotheses is provided by the lower level processes. Each hypothesis is characterized by: a) the hypothesized string that was recognized; b) the score of this hypothesis; c) the time interval that this hypothesis spans. We consider two different thresholds for the likelihoods: the word hypotheses with score above the higher threshold are to be considered "very reliable", and their role will be to drive the process. The word hypotheses with score between the two thresholds will be included in the analysis without a driving role, while the word hypotheses below the lower threshold are not to be considered, at least in the first pass. This work is about parsing with the above constraints. In this connection it seems advantageous to anchor the recognizing process to those hypotheses that were given a high score. As we shall see, there is also a predictive aspect in our approach: this means that the parser will tell the lower level component to "do its best" to find in the given place an instance of what was predicted. In the simplest case we can think of a direct recovering of a word hypothesis with score below the lower threshold. Our starting point will be a very well founded technique, that has also been experimentally proved as valid, namely chart parsing.

Chart parsing works very well with well formed input, but the technique was not conceived for working with an uncertain input, and even worse, with a fragmentary input. Chart parsing is directional in the sense that it works from the starting point (usually the beginning of the sentence) extending its activity usually in a rightward manner. We shall introduce a different concept, that nonetheless will work with the same linguistic data. The concept is that of a chart that works outward from islands and makes sense of as much of the sentence as it is actually possible. Furthermore, where the signal was just not detected, predictions can be made on the basis of the configuration and of a set of heuristics. After the application of these heuristics, and the introduction of new low level hypotheses, the algorithm works on in the same way and if the situation was not unrecoverable concludes with one (or more) complete analysis of the sentence.

It is worth noting that the proposed solution helps also in dealing with ill-formed written input. There is something more to it: in the general treatment we are giving in this paper we shall refer only to a grammar coded in the traditional form of rewriting rules, but the mechanism can work with a large number of formalisms. Some contemporary linguistic theories emphasize the role of particular words that play the role of head of a constituent (e.g. a noun in a noun phrase). As a matter of general parsing strategy it seems very interesting to couple the localization of the pivot with an island mechanism that guarantees local control of the process in all directions.

2. Chart parsing

Chart parsing is a very powerful idea for parsing natural language. It was introduced by Martin Kay [1973, 1980] and Ronald Kaplan [1973] and historically was inspired by Earley's algorithm [1970]. The most basic goal in introducing the chart was to reduce the complexity of a nondeterministic parsing algorithm.

An advantage of chart parsing is that the mechanism is perfectly suited for both bottom-up and top-down parsing.

A further advantage is that the chart can be complemented with an agenda. In this way, instead of introducing new edges following the rigid application of the algorithm, tasks can be added to the agenda and at every moment a scheduling function can decide the order in which tasks should be executed, in a multiprogramming fashion. Very easily the scheduling function can implement depth-first control and breadth-first control, but any kind of control can in principle be inserted [see for instance Stock 1987].

Also, a particular point is that the input relation with other levels of analysis is very coherent: lexical ambiguity results in the very simple fact that more than one inactive edge are introduced for an ambiguous word.

3. Bidirectionality

Chart parsing has a positive aspect and some evident problems in facing speech recognition. Typical of continuous speech recognition are the following aspects:

1) The separation between words is not univocally given; one of the tasks of the sentence parser is exactly to yield suggestions for word separations. In the chart this can be very well accomplished introducing more vertices, one for every hypothetical separation point. Vertices must be ordered and ordering here is provided by the time order relation. Therefore we can introduce a vertices structure

$V_{t_0}...V_{t_1}...\ \ ...V_{t_i}...\ \ ...V_{t_n},$

with for $i=0, n\text{-}1\ \ t_i < t_{i+1}$

where for every vertex arrives or leaves at least one lexical edge. It just does not matter if the final analysis will not "make use" of all the vertices in the chart.

2) Some words in the input matrix are anchored as "surely" recognized while others are only very tentative interpretations. It makes sense that the analysis privileges elements of the first type as starting points. This is the concept of island parsing, for which the parser tries to make sense of portions of a sentence starting from fixed points (islands), that can occur in any position. The traditional chart mechanism cannot deal with this task.

3) Island parsing is required to get to the extreme borders of the recognizable fragments, and from that situation help in making suggestions for the unrecognized fragments based on both the left and the right contexts. Here again the traditional chart mechanism cannot deal with this task.

We are now going to introduce a new concept: **bidirectional charts.**

Data structures must be rearranged in this connection and the whole parsing process will be different: things get complex if one wants to preserve the good qualities of charts and be reasonably efficient.

We begin with redifining active edges.

An active edge here is a data structure that includes two positions in the rule involved : an initial position and a final position, such that a fragment is covered by the given edge in reference to a fragment of the right handside of the rule.

Therefore an active edge is characterized by *from*, the left vertex, *to*, the right vertex, *rule*, the referred rule, *fromposition*, the first of the two positions in the rule, *toposition*, the second of the positions, and *sub-inactives*, the list of the immediately spanned inactive edges that were included.

Inactive edges are characterized as usual, by *from,to* and *cat*, the category.

Let us now say that an active edge E is locally rightward largest iff there is no other active edge E' with from(E') = from(E), rule(E') = rule(E), fromposition(E') = fromposition(E) and sub-inactives(E') including as an initial substring sub-inactives(E).

Analogously we can define a locally leftward largest edge.

We then define four different rules for introducing a new edge in the chart:

The first rule says, roughly, that if you are trying to build the same thing from the left and from the right you should unify your efforts.

A-A Rule:

If we have two active edges A_1 and A_2,

with to(A_1) = from(A_2)

rule(A_1) = rule(A_2)

toposition(A_1) = fromposition(A_2)

and A_1 is locally leftward largest and A_2 is locally rightward largest, then we can introduce a new active edge A_3 into the chart with

from(A_3) = from(A_1), to(A_3) = to(A_2), rule(A_3) = rule(A_1), fromposition(A_3) = fromposition(A_1), toposition(A_3) = toposition(A_2),sub-inactives(A_3) = concat(sub-inactives(A_1), sub-inactives(A_2)), where concat is the usual string concatenation operator.

If fromposition(A_1) = 0 and toposition(A_2) = n, number of symbols in the right hand side of rule(A_1), an inactive edge I is introduced instead, with from(I) = from(A_1), to(I) = to(A_2) and cat(I) equal to the left hand side of rule(A_1).

We also maintain the usual edge combination rule, with the extension to the two directions.

A-I Rule:

Given an active edge A and an inactive edge I with from(I) = to(A), and, having named *i* toposition(A), with i ≠ n (the number of symbols in the right hand side of the rule), cat(I) = C_{i+1} , i + 1-th symbol of the right hand side of rule(A), then a new edge E can be added to the chart, with from(E) = from(A), to(E) = to(I), and, if i + 1 = n was the last symbol in rule(A) and fromposition(A) = 0, E will be an inactive edge with cat(E) equal to the left hand side of rule(A), if not it will be an active edge with rule(E) = rule(A) and fromposition(E) = fromposition(A), toposition(E) = i + 1.

Similarly, if to(I) = from(A), and having named i fromposition(A), i ≠ 0, cat(I) = C_{i-1} , i-1-th symbol of the right handside of rule(A), then a new edge E can be added to the chart, with from(E) = from(I), to(E) = to(A), and, if i-1 = 0 and toposition(A) is equal to the length of the right handside of rule(A), E will be an inactive edge with cat(E) equal to the left handside of rule(A), if not, it will be an active edge with rule(E) = rule(A), fromposition(E) = i-1, toposition(E) = toposition(A).

Let us now recall our classification of word hypotheses into three classes, say a, b, c, in relation to their scores.

As stated earlier, we consider word hypotheses of class a the islands for our process. The algorithm will proceed outward from the islands and bottom-up when a constituent including an island (however far inside the structure) is completed. Let us say that an edge has another feature, called **withisland**, a boolean that is originally true for lexical edges of class a and false for the others, and during the process is propagated to any new edge that "includes" an edge with withisland = true.

We can now state the

I/bu Rule:

When an inactive edge I, with withisland(I) = true, is introduced in the chart, a new active edge is introduced for every rule R in the grammar that includes on its right hand side the symbol cat(I) and in relation to R for every position i such that cat(I) is the i + 1-th symbol on the right hand side of R. Let us denote such a generic active edge as A; its characteristics will be from(A) = from(I), to(A) = to(I), rule(A) = R, fromposition(A) = i, toposition(A) = i + 1, sub-inactives = list(I).

We have also the usual top-down rule, rivisited consistently with our approach:

A/td Rule:

When an active edge A is added to the chart, if from the vertex to(A) only edges with withisland = false leave rightward, then introduce a cycling active edge on to(A) for every rule that has on the left handside the symbol that comes after the position toposition(A) for rule rule(A), unless there is already an active edge with that rule or an inactive edge with that category. Do likewise on the other vertex.

The meaning of the presence of both the I/bu and the A/td rules is that the process will be a bottom-up one, starting from the islands. When a point is met where only class b words are found, hypotheses of the presence of certain constituents, according to the "island" constraints, are introduced in the form of cycling active edges. This top-down operation will ensure that the parser is led by the most consolidated fragments.

Every time we introduce a new active edge A we must perform a redundancy check to ensure that we do not build, not only now, but also in the forseeable future, anything that has already been built.

r/Check:

A new active edge A can be inserted in the chart unless from the vertex from(A) there is an active edge A' leaving rightward with rule(A') = rule(A), fromposition(A') = fromposition(A) and sub-inactives(A') including as an initial substring sub-inactives(A).

Similarly, A can be inserted in the chart unless from the vertex to(A) there is an active edge A' leaving leftward with rule(A')=rule(A), toposition(A')=toposition(A) and sub-inactives(A') including as a final substring sub-inactives (A).

It is convenient that the above rules be applied in the given order so as to minimize the effort.

As regards the question of control, it seems reasonable that all edge building tasks originated by an island should be carried on in the first place, and the actions resulting from predictions over class b hypotheses be carried out later, in order to avoid an explosion of fuzzy edges in the chart. Still, it is clear that, because of the nature of the algorithm, after the introduction of an edge of the second type, an edge building action originated by an island can take place again.

With this in mind we introduce two agendas, **a-agenda**, where tasks of building edges with withisland=true are added and **b-agenda** where the other tasks are added. Task execution is constrained only by the discipline that a task in b-agenda can be executed only if a-agenda is empty. At the beginning of the process a-agenda is filled with all the tasks originated by the class a word hypotheses.

4. An example

We shall present here an example of parsing with the concepts introduced in this paper. The sentence is :THE BOSS WANTS AN IMMEDIATE CALL TO MILAN.

For clarity's sake, we shall consider vertices univocally detected and lexical interpretations unambiguous. Of course we shall consider words of class *a* (islands) and of class *b*.

class *a* words: MILAN, BOSS

class *b* words: THE WANTS AN IMMEDIATE CALL TO

rules: 1) S -> NP V NP PP

2) S -> NP VP

3) NP -> ProperN

4) NP -> DET N

5) NP -> DET ADJ N

6) PP -> PREP NP

7) VP -> V NP

We shall insert inactive edges in the lower side of the sentence and active edges in the upper side of the sentence. The edge being processed is drawn with a dotted line, the possible other edge considered in the rule that is currently applied is drawn with a dashed line, the resulting edge is drawn with a bold line.

The process starts bottom-up from the islands (class *a* words) MILAN and BOSS, introducing active, inactive and cycling adges into the cahrt, following the *composition* rules introduced before.

Starting from MILAN we get to produce an inactive edge with cat=PP, between vertices 7 and 9, and an active edge with cat=S, relative to rule 1, once more between vertices 7 and 9 , with fromposition=3.

When the word BOSS is analyzed we get to produce an active edge with cat=S, relative to rule 1 , and with toposition=2.

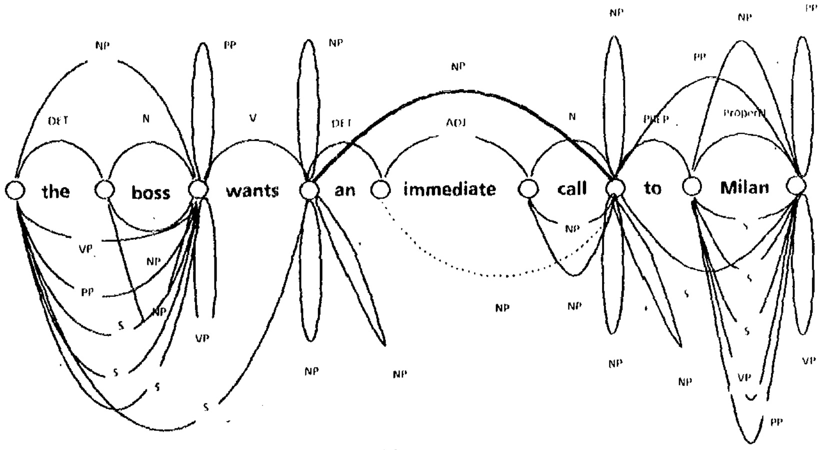

Figure1

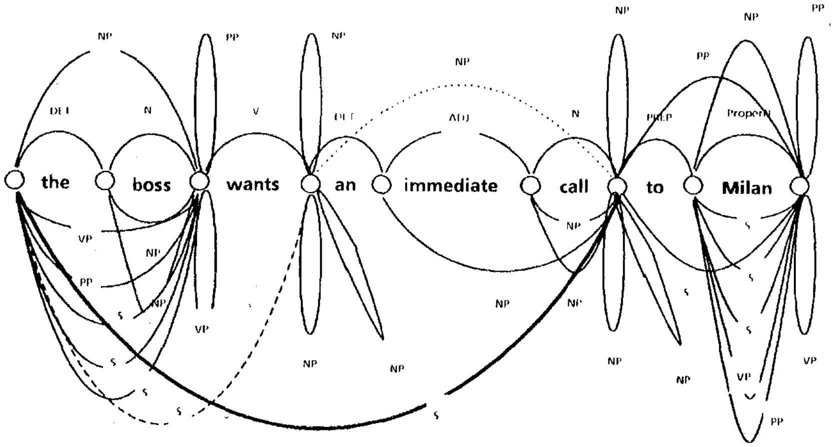

Figure 2

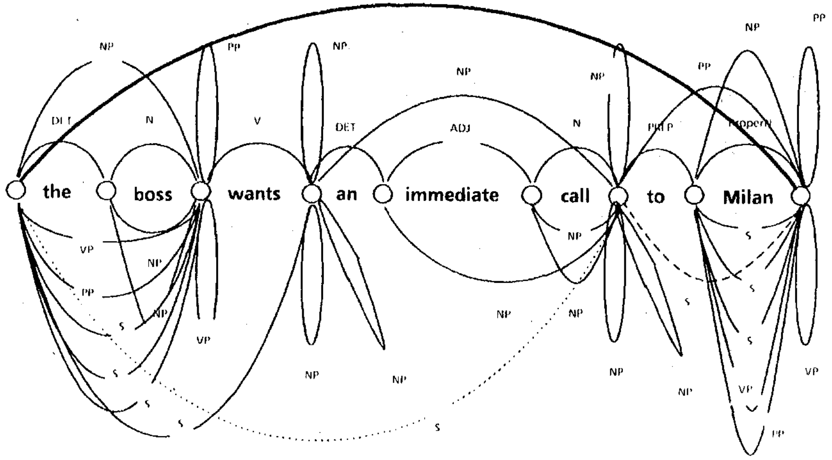

Figure 3

A top-down process is needed after that because a phrase occurs without any islands in it. Through that process a noun phrase is recognized between vertices 4 and 7, so that we have the situation shown in Figure 1.

At this point, by virtue of the A-I composition rule an active edge with cat=S can be inserted into the chart between vertices 1 and 7 as shown in Figure 2.

The last step consists in introducing an inactive edge with cat=S into the chart between vertices 1 and 9, by virtue of the A-A composition rule, as shown in Figure 3. This yields a succesful recognition of the sentence.

Conclusions

A mechanism that extends the chart algorithm with bidirectionality has been introduced. This step is a major one, since a monodirectional chart would not be able to base its processing selectively on easily identifiable fragments (the so called islands) and derive hypotheses about the other parts of the input string. Instead, with the mechanism proposed here, for any place where the easily identifiable fragments occur in the sentence, the process will extend to both the left and the right of the islands, until possibly completely missing fragments are reached. At that point, by virtue of the fact that both a left and a right context were found, heuristics can be introduced that predict the nature of the missing fragments.

The described mechanism is particularly advantageous when dealing with complex sentences, because it is an inherently nondeterministic mechanism, capable of dealing with the complex local ambiguity typical of natural language. An important aspect is that the mechanism is completely independent of the particular linguistic theory adopted. In technical terms, the linguistic representation is reflected only in the particular functional description, and in its particular operations, which are added to the edges of the chart and will provide the necessary information for constraining the process and allowing better predictions.

The use of bidirectional charts seems to be particularly suitable for speech recognition, but also for processing other forms of ill-formed input; lastly, it seems particularly suited even for processing well formed strings when combined with a head-driven linguistic theory, i.e. a theory that privileges particular elements inside constituents [see for instance Stock 1986].

References

Barton, E.; Berwick, R., and Ristad, E. *Computational Complexity and Natural Language.* MIT Press Cambridge, Mass. (1987)

Earley, J. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery.* 13(2): 94-102 (1970)

Kaplan, R. A general syntactic processor. In Rustin, R. (Ed.), *Natural Language Processing.* Englewood Cliffs, N.J.: Prentice-Hall (1973)

Kay, M. *Algorithm Schemata and Data Structures in Syntactic Processing.* Xerox, Palo Alto Research Center (October 1980)

Kay, M. The Mind System. In Rustin, R. (Ed.), *Natural Language Processing.* Englewood Cliffs, N.J.: Prentice-Hall (1973)

Stock O. 'Dynamic Unification in Lexically Based Parsing' *Proceedings of the 7th European Conference on Artificial Intelligence*, Brighton, (1986). Also in Advances in Artificial Intelligence II, B. Du Boulay, D. Hogg & L. Steels Eds., North Holland, Amsterdam,(1987)

Stock O. 'Coping with dynamic syntactic strategies: an experimental environment for an experimental parser' *Proceedings of the Third Conference of the Association for Computational Linguistics*, European Chapter, Copenhagen (1987)

Stringa, L. 'An Artificial Intelligence Approach to Speech Recognition and Understanding' to appear in *Pattern Recognition Letters* (1988).

Thompson, H.S. Chart parsing and rule schemata in GPSG. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics.* Alexandria, Va. (1981)

Walker, D.E. Speech Understanding through syntactic and semantic analysis. In *IEEE Transactions on Computers*, Vol. C-25, no. 4, 1976 (1976)

Wiren, M. A comparison of rule invocation strategies in context-free chart parsing. In Proceedings of the *Third Conference of the Europen Chapter of the Association for Computational Linguistics.* Copenhagen, (1987)