Improving Search Strategies
An Experiment in Best-First Parsing

Hans HAUGENEDER
Manfred GEHRKE

Siemens AG
ZT ZTI INF 23
Otto-Hahn-Ring 6
8000 München 83
W. Germany

Abstract

Viewing the syntactic analysis of natural language as a search problem, the right choice of parsing strategy plays an important role in the performance of natural language parsers. After a motivation of the use of various heuristic criteria, a framework for defining and testing parsing strategies is presented. On this basis systematic tests on different parsing strategies have been performed, the results of which are dicussed. Generally these tests show that a "guided" depth-oriented strategy gives a considerable reduction of search effort compared to the classical depth-first strategy.

1. Introduction

Parsing natural language utterances can be considered a search problem, which is characterized by the application of a set of operators (i.e. the grammar rules) onto the input data (phrase to be processed) in order to yield a final state (derivation tree). In practical applications which are characterized by grammars with a large coverage and a non-trivial complexity of the input (measured e.g. in sentence length and lexical ambiguity) one is confronted with difficulties that seem quite common to various search problems, namely

the size of the search space and
the selection among multiple solutions.

Two quite opposite approaches to these problems have been proposed. In the one approach, the brute force of exhaustive search has been used, possibly augmented with some ranking scheme for the set of parses. In the other approach, the parsing of natural language utterances is considered a deterministic process [Mar80], where a "wait and see" strategy makes the flavour of searching through the alternative application of different grammar rules disappear, at least for grammars with limited coverage.

The approach we are taking to this problem lies between these two extremes: Conceptually, it takes the first view, considering natural language parsing a nondeterministic process; from a performance point of view, it is directed towards the approximation of deterministic behaviour. Thus our aim is to develop a best-first parsing strategy which enables the parser by means of heuristic criteria and information to

- limit the overall search space as much as possible to arrive at the first parse at low costs
- achieve the most plausible analysis as the first one.

With these aims in mind - at present mainly concentrating on the first one - we still want to maintain the ability of our mechanism to find further solutions, since we do not assume the order of the analyses to be correct all the time. Thus "heuristics" is understood as improving the problem solving performance without affecting the competence [Min63].

What we propose is a practically oriented approach to these problems; it is practical in the sense that our primary focus is not to model the human sentence processing mechanism or specify the human parsing strategy. We are rather aiming towards the development of parsing strategies, that are based on heuristic information, enabling the parser to choose the right paths in the search space most of the time.

Although psychological results on human sentence processing strategies may be incorporated in the heuristics to be developed - at least as far as they fit in our framework and do not assume special properties of the underlying processing scheme - we do not understand our work as contributing to the characterization of inherent structures of the human sentence processor.

Thus our goal is not of an "all or nothing" character; we do not expect our parser to make the right choice all the time. What we do want, however, is to develop a more pragmatic strategy, which, when applied to major samples of sentences, is able to give us the first reading with a minimal overall search effort.

After testing some strategies that give the parser more guidance by increasing the information available at the choice points, some promising results have emerged. Work in a similar direction on the MCC Lingo project [Wit86] also seems to give some indication for this.

2. The Use of Heuristic Information in Parsing

In a number of natural language parsers - especially in those with practical orientation and grammars with comprehensive coverage - the problem of dealing with alternative parses has been handled by some sort of scoring measures for sets of alternative parses already produced by breadth-first enumeration.

This is the case in the DIAGRAM parser, where arbitrary sub-procedures (so-called factors) assign likelihood scores to syntactic analyses [Rob82]. In the EPISTLE system, a numerical metric is used for ranking multiple parses which is defined on the form of the phrase structure being built up [Hei82]. And as a last example for that type, the METAL parser performs a scoring of the analyses found, which is based on both grammatical and lexical phenomena [Slo83]. In all these examples, the criteria on which the scoring is based do not influence the parser's behaviour but act as some sort of filter on the parser's results. The major challenge in our approach however is the application of such and similar scoring criteria on the fly during the parsing process instead of applying them after the parser has performed a blind all-paths analysis.

If one thinks of more search intensive applications, like speech understanding with the high degree of ambiguity in the input in the form of numerous word hypotheses, the application of such heuristic criteria during the parsing process seems to have an even larger advantage over the filter approach.

3. A Testbed for Modelling Parsing Strategies

In order to be able to model heuristic parsing strategies, one needs a suitable parsing mechanism which has enough flexibility for such a task. The most obvious choice for doing this is active chart parsing [Kap73], [Kay80] which is a highly general framework for constructing parsers. It combines the concept of an active chart as an extensive bookkeeping mechanism preventing the parser from performing two identical processing steps twice, with an agenda-driven control mechanism which enables a very elegant and highly modularized simulation of different control structures. And it is exactly this second feature that is central for our strategy modelling task (for details see [Hau87]).

Since we view the development of a best-first parsing strategy as an empirical task, i.e. as the result of going through a number of define-test-modify cycles to build up the "final" heuristics, it is necessary (or at least useful from a practical point of view) to have available an environment that enables the user to define and modify the heuristic function easily and supports him in seeing and checking immediately without much effort the effects of a modification.

The APE system, in which this work is embedded, is an ATN grammar development environment which (among other things) offers the functionality needed. By means of a highly interactive, graphically-oriented user interface it offers operational facilities that give the user a number of possibilities for inspecting and debugging the parser's behaviour under a given strategy, as for example an agenda editor, the possibility to specify strategies and change them during parsing, and a chart-based fully graphical parser stepper. An heuristics editor is integrated into APE's user interface in a straightforward way: in addition to the possibility of choosing between several predefined uniform and heuristic strategies, the user can define his own strategies. The specification of the intended heuristic function is performed by giving appropriate weighting factors wf_i to the various heuristic dimensions in a template-based manner.

After the specification of the values for the various weighting factors, each expressing the relevance of the the corresponding criterion, the user is presented with the arithmetic expression associated with the corresponding heuristic function (in standard infix notation), which he can modify further if he finds the system defined composition of the weighted criteria unsatisfactory. This obviously can lead to modifications of the heuristic function's range definition, the consequences of which the user must be aware of when using this option (cf. 4.2). Details of the heuristics specification and manipulation facility are described elsewhere ([Hau87], [Geh88]).

Although the APE system is based on an ATN framework, the characteristics concerning heuristic information for scheduling are independent of the underlying ATN approach; the only critical point is the assumption of an active chart parsing processing scheme. Thus these considerations can be applied to a number of other grammar formalisms as well, especially to those belonging to the paradigms of (procedurally and descriptively) augmented phrase structure grammars.

The implementation of the APE system and the work described here has been performed in Interlisp-D on a Siemens EMS 5822 workstation.

4. Defining Heuristic Strategies

4.1 Factors Influencing Heuristic Strategies

The criteria that can be employed in the specification of an heuristic function though being of a widely differing nature can be divided into two classes. Firstly there are a number of "external" criteria, which are characterized statically in an a priori way. These include:

(1) Characterization of the plausibilities of grammatical rules
(This gives the possibility to scale the grammar with regard to the "strength" of the constructions; thus one can divide the grammar into a core and peripheral part. A quite similar criterion ("syntactic preference") is used in [For82].)

(2) Different values assigned to the various homographic readings of words in the input
(For a number of systematic but not equally distributed homographic ambiguities, as for examples the noun reading of certain verb forms, this offers an elegant way of supressing the "exotic" reading. A similar focussing mechanism also seems to be used during human sentence processing, as indicated in [Car81].)

(3) Complexity of the structure of complete (sub)constituents, measured in terms of number of nodes depth and mode of embedding
(Thus grammatical but hardly acceptable structures, like deep center embeddings for example, can be "postponed".)

(4) Scoring of word hypotheses
(Information of this type becomes relevant with spoken input.)

Besides these criteria there are others which reflect certain aspects of the parsers internal state, as, e.g.:

(5) The weight of the partial analyses, which is the value of the heuristic function associated with the active edge characterizing this partial analysis
(The overall plausibility of a certain partial parse is characterized by this weight.)

(6) The span of an inactive edge as ratio of the edges' span and the total length of the input
(With other factors being equal an inactive edge with a wider span, i.e. a larger constituent will be preferable, since this leads to a wider overall span. At first sight the criterion of length of a constituent sounds a bit awkward, but e.g. in [Fra78] its impact on phrasal attachment is shown.)

(7) The span of an active edge as ratio of the edges' span and the total length of the input
(With other factors being equal an active edge with a wider span, i.e. a larger partial analysis will be preferable, since this leads to a wider overall span)

(8) The number of input items left for processing, expressed as

$$hv_{max} \quad \frac{whyp_{left}}{whyp_{total}}$$

with hv_{max} being the maximal heuristic value (i.e. 1), $whyp_{left}$ being the number of word hypotheses left in the remaining input and $whyp_{total}$ being the total number of word hypotheses. The applicability of this criterion furthermore is coupled to some global threshhold that defines the point in the input from which this factor will be taken into account.
(This sort of information can be used to force the parser to behave in a resource-oriented manner if there are only a few items left to process.)

All this information is fairly inexpensive to compute and making it available to the heuristic function during the parse can be accomplished in a straightforward way by attaching this information to the corresponding components of a task (i.e. the inactive edge, the active edge and the grammar rule), tasks being the fundamental unit in the processing cycle.

Besides these more or less syntactic factors one can also think of integrating semantic criteria (as the possibility of referential interpretation of (noun) phrases or appropriate word sense disambiguation for example) directly as part of an heuristic strategy in our framework. Since the application of the strategy takes place at a very fine-grained level, where one reasonably may not expect semantic "feedback" in the form of a corresponding heuristic value hv_{sem} all the time (i.e. at each choice point), one has to cope with the problem of how to deal with an hv that is not defined. If one adopts the convention that the effect of an hv which no value has been supplied for is totally excluded from the overall heuristic function, one achieves a plausible and attractive style of syntax-semantics interaction. This offers a good deal of flexibility, with the possibility of interaction at the word level as well as at the phrase level without committing to either.

4.2. The Heuristic Function

Assuming that the values for the various heuristic criteria are in the interval [0,1], resulting in an overall heuristic measure in the same interval, i.e. the heuristic function hf has the form specified in (I), there is still the question of how these criteria interact, i.e. how the values accumulate.

hf: $[0,1]^n$	$\Rightarrow$ $[0,1]$	with n being the number of heuristic criteria

(I)

The interaction of these different heuristic values hv_j is handled by a weighting factor wf_j that is associated with each heuristic dimension (such as e.g. complexity of the structure). The weighting factor is intended to express the importance of the corresponding dimension and has a range from 0 to 5, with 0 meaning that the dimension does not play a role at all and 5 giving it maximal relevance.

Obviously, this weighting factor has no real qualitative interpretation; the only fact it expresses is the relevance of a heuristic dimension relative to the other ones. Thus, for each heuristic criterion the actual value is computed by the product of the value of the heuristic dimension and the corresponding weighting factor, i.e. wfj * hvj. For the accumulation of the values of the heuristic criteria we have chosen the arithmetic mean, thus having the overall heuristic value defined by formula (II).

$\frac{\Sigma\ wf_j * hv_j}{\Sigma\ wf_j}$	for all the hv_js and corresponding wf_js, such that the value of hv_j is defined

(II)

5. Results

5.1 Scenario of the Test

For the interpretation of the presented results it seems necessary to specify the experimental conditions under which the tests have been performed. The grammar we have been using covers the following subset of English: declarative sentences, imperative sentences, questions (direct and indirect y/n-questions, direct and indirect wh-questions for NPs, PPs, APs), sentential complements for verbs and nouns, complete and reduced relative clauses, infinitive complements, clausal conjunction, and subordinate clauses.

The test sample consisted of a set of 40 sentences and phrases that range from very simple phrases like "the man" to more complex constructions like "John gives the girl Bill admires a book which he does not expect her to read". The medium sentence length of the sample is 6.5. The homographic ambiguity factor is 1.3, i.e. each word processed is 1.3 times ambiguous on the average.

5.2 Discussion of the Results

When processing the test sample under the various strategies, it turned out that there were many strategies that showed approximately the same overall behaviour, i.e. demanded almost the identical search effort. Especially the variation of the weight for the single factors in general only shows effects when one contrasts the extreme values for the weighting factor (i.e. 0 and 5).The quantitive measures of some selected heuristic functions that have been used in a one-path analysis mode is shown in figure (IV).

The strategies we used are defined as explicated in (III), where AE means overall weight of the incoming active edge, GR weight of the grammar arc, SIE span of the inactive edge, SP span of the active edge to be continued with the inactive edge, IL items left. Besides the heuristic criteria, another important impact on the parsing strategy is the method of insertion of tasks into the agenda. It can take place in a local or in a global mode. While in the latter case (SortAll insertion mode) a general and costly reordering of all tasks in the agenda is performed, in the first case (SortNewToFront insertion mode) only the ordered set of newly generated tasks is put onto the agenda in a stack-like fashion.

strategy	insertion mode	heuristic function
strat1	SortNewToFront	5 * AE, 3 * SIE, 2 * GR, 5 * SP, 1 * IL
strat2	SortNewToFront	5 * GR, 5 * AE
strat3	SortAll	5 * GR, 5 * AE
strat4	SortAll	5 * GR, 4 * SP, 4 * IL

(III)

The strategies we discuss here represent the two best locally operating ones (strat1, strat2), the best global one (strat3) and the worst global one (strat4). The results show among other things that the most promising strategy takes 59% of the search effort depth-first strategy uses. Furthermore it can be seen that with respect to the two best strategies there is an decrease of the search effort on longer, more complex sentences of the sample down to 56% for each.

strategy	one-path: #tasks	search effort (%)
depth-first	2765 (1218)	100 (100)
strat1	1628 (690)	59 (56)
strat2	1702 (685)	62 (56)
strat3	2313 (1138)	84 (97)
strat4	2830 (1363)	113 (112)

The numbers enclosed in brackets are for the subset of "long" sentences in the sample (with length > = 8)

(IV)

strategy	search effort (%)
depth-first	77
strat1	45
strat2	47
strat3	64
strat4	79
all-paths: #tasks	3581 corresponding 100%

(V)

How this relates to the overall search space is documented in (V). These numbers correlate the overall size of the search space to the part of it that has been traversed by the different strategies.

Though we are being far from beleiving that the best strategy we have worked with is *the* strategy, there are some general guidelines. Thus if we try to reflect on the results presented here and the material which has been analyzed during our test, the following picture emerges:

(1) Static weights on the grammar rules are useful.

(2) Span-orientation (i.e. the tendency to further follow the parse that yields the biggest overall span) has shown rather drastic positive effects.

(3) Resource-orientation (i.e. the tendency to continue tasks that have almost reached the end of the input with additional emphasis) gives some minor additional improvements.

(4) Local application of the heuristics (i.e. a heuristically guided depth-first strategy, which corresponds to the insertion mode SortNewToFront) instead of a global reordering of all the pending tasks in the agenda is much more effective with the additional advantage of being much less costly.

(5) Finally our experiment has shown that the simultaneous use of several criteria together leads to a reduction of the search effort as compared to each single's criterion effect. Although the various criteria can locally conflict with each other in certain configurations, their cumulative overall effect is stronger than local "disturbances".

The basically stack-oriented way of updating the agenda - which leads to an "informed" depth-first strategy - makes our approach also compatible with models that determine the scheduling of the parser 's operation with respect to certain phenomena on the basis of purely linguistically oriented principles, as the treatment of syntactic closure in [For82]. As long as such models still retain a certain amount of nondeterminism with depth-first as a default scheduling principle, a guided depth-first strategy of the type discussed here may be favourable to an "uninformed" depth-first strategy.

6. Conclusion and Outlook

The results that have come out of our experiments seem to indicate - though only a subset of the potential criteria has been taken into account systematically - that heuristics of the type presented can be applied fruitfully. We see the extension of the coverage of the grammar as well as the enlargement of the test sample as a logical continuation to confirm our results .

Beyond that we will apply a heuristic approach similar to the one presented here to spoken input, where the complexity is far beyond typed input due to the existence of a large number of word hypotheses (about 5000 for a 6 word sentence on average); thereby the data for the latter work are provided by the SPICOS continuous speech understanding project [Dre87].

References

[Car87] Carpenter, P.A., and Daneman, M., "Lexical Retrieval and Error Recovery in Reading: A Model Based on Eye Fixations". Journal of Verbal Learning and Verbal Behaviour Vol. 20 No. 2 (1981), 137-160

[Dre87] Dreckschmidt, Gaby, "The Linguistic Component in the Speech Understanding System SPICOS". In: H. G. Tillmann, G. Willeé (ed), Analyse und Synthese gesprochener Sprache, Hildesheim-Zürich-New York, Olms Verlag 1987

[For82] Ford, M., Bresnan, J.W., Kaplan, R.M., "A Competence Based Theory of Syntactic Closure". In: Bresnan, J.W. (ed): "The Mental Representation of Grammatical Relations", Cambridge/Mass, The MIT Press, 1982, 727-796

[Fra78] Frazier, L., Fodor, J.D., "The Sausage Machine: A New Two-Stage Parsing Model". In: Cognition 6 (1878), 291-325

[Geh88] Gehrke, M., Haugeneder, H., "APE User Manual". Siemens Report, to appear.

[Hau86] Haugeneder, H., Gehrke, M., "A User Friendly ATN Programming Environment (APE)". In: Proc. COLING-86, 399-401.

[Hau87] Haugeneder, H., Gehrke, M., "Modelling Heuristic Parsing Strategies". In: K. Morik (ed.), GWAI-87 - 11th German Workshop on Artificial Intelligence, Berlin-Heidelberg-New York, Springer-Verlag, 1987, 84-93

[Hei82] Heidorn, G.E., "Experience with an Easily Computed Metric for Ranking Alternative Parses". In: Proc. ACL-82, 82-84.

[Kap73] Kaplan, R.M., "A General Syntactic Processor". In: Rustin, R. (ed), "Natural Language Processing", New York, Algorithmics Press 1973, 193-241.

[Kay80] Kay, M., "Algorithm Schemata and Data Structures in Syntactic Processing". Xerox PARC Tech. Report No. CSL-80-12, 1980.

[Mar80] Marcus, M., "A Theory of Syntactic Recognition for Natural Language". Cambridge/Mass., The MIT Press 1980.

[Min63] Minsky, M., "Steps Towards Artificial Intelligence". In: Feigenbaum, E. A. and Feldman, J. (eds), "Computers and Thought", New York, McGraw-Hill 1963, 406-450.

[Rob82] Robinson, J., "DIAGRAM: A Grammar for Dialogues". CACM Vol. 25 No. 1 (1982), 27-47.

[Slo83] Slocum, J., "A Status Report on the LRC Machine Translation System". In: Proc. Conference on Applied Natural Language Processing 1983, 166-173.

[Wit86] Wittenburg, K., "A Parser for Portable NL Interfaces Using Graph-Unification-Based Grammars". In: Proc. AAAI-86, 1053-1058.