ATR Symposium on Basic Research for Telephone Interpretation (ASTI),
Kyoto (Japon), 11-13 December 1989.

A Generative View of Ill-Formed Input Processing

Extended Abstract

Bernard LANG
INRIA
B.P. 105, 78153 Le Chesnay, France
lang@inria.inria.fr

The intent of this presentation is to exhibit the commonalities between the following syntactic problems:

1. parsing ambiguous or incomplete input, often known as *word lattice parsing*;
2. parsing ill-formed input, i.e. input that does not belong to the language formally defined by a grammar;
3. syntactic disambiguation of ambiguous sentences;
4. accounting for deviant syntactic structures in grammatical language descriptions.

The key idea behind this work is that of a *weighted grammar*. For simplicity we consider here only a special case of the more general definition given in [Teitelbaum 73], which could lead to other interesting variations (e.g. probabilities as weights, using multiplication instead of addition as below). We define a weighted grammar as a Context-Free (CF) grammar with a numeric weight attached to each of its rules. We attach to any derivation tree a weight that is the sum of the weights of all instances of rules used in that derivation.

Syntactic disambiguation

A first obvious use of such weights is the syntactic disambiguation of ambiguous sentences. If the weights are inversely proportional to the frequency (or some other likelyhood or desirability measure) of the syntactic constructions described by the rules, then disambiguation of a sentence may proceed by choosing the parse tree with the smaller weight. Alternatively, if parse-trees may be rejected on semantical grounds, then one can try them in increasing order of weight, thereby increasing the likelyhood of finding early the most appropriate parse. Of course, ambiguity resolution may be factored by considering it separately for independently ambiguous constituents.

The main advantage of this approach is that it meshes well with the chart/Earley parsing techniques [Earley 1970, Lang 1974] which have themselves been shown to have a wide applicability, even for non-CF formalisms such as TAGs or CCGs [Vijayshankar & Weir 89, Lang 88b]. In fact it just amounts to a less trivial use of the dynamic programming nature of chart parsing.

Lattice parsing

Though syntactic disambiguation, as sketched above, has received some attention in CF parsing, its finite state counterpart is commonly used at the phoneme or lexical level. Dynamic programming and beam search are a standard way of determining the

most likely word sequence interpretation of a sequence or "lattice" of phonemes, on the basis of weights associated with the phonemes [Ney 82].

However, rather than choosing the "best" phoneme sequence in an independent way, the (*regular*) set of possible phoneme sequences is represented by the graph of a finite state automaton (FSA) Φ. This FSA Φ may be used directly for parsing at the phoneme level or it can be translated into the graph of another FSA W representing the regular set of all word sequences. The FSA W is better known as *word lattice*, and various authors have proposed CF parsing of word lattices (e.g. [Nakagawa 87]). These FSA may contain elementary loops corresponding for example to missing input sequences [Lang 88a].

Chart parsing is easily extended to parsing such FSA/lattices. This is not surprising, since CF languages are closed under intersection with regular sets. It can be easily shown, as an extension of the results in [Lang 88a & 89], that the parsing of a FSA A according to a CF grammar G can produce a new grammar T for the intersection of the languages $L(A)$ and $L(G)$, giving to all sentences in that intersection the same structure as the original grammar G. This grammar T may also be understood as the shared forest of all possible parses of the sentences in $L(A) \cap L(G)$.

The phonemes of the FSA Φ are actually associated with weights. If Φ is directly parsed, one can include this weight information in the dynamic programming structure of the chart parser, so as to determine the most likely parse. This is almost identical to ambiguity resolution, except that the weights are associated with the input rather than with the rules. Obviously both techniques could be combined.

Alternatively, the weights of the phoneme FSA may be used to build a weighted word lattice/FSA to be used for weighted parsing [Paeseler 87].

In both cases, rather than buiding only the most likely parse, it is possible to build a weighted grammar T of all possible parses. This grammar generates all sentences in $L(A) \cap L(G)$ with precisely their original weight according to the weighted FSA A. If the grammar G was itself weighted, the weights of the sentences in $L(T)$ depends on those of both A and F. Alternatively, some form of beam search in the parsing process may be used to restrict that grammar T to those words that have a weight exceeding some predetermined value.

Ill-formed input

A frequent way of processing ill-formed sentences is to assume the presence of some standard errors such as missing, extraneous, interchanged or misused words (this may also be at the phoneme or character level). Several variations on Earley's algorithm have been proposed to deal with such errors by (for example) inserting, deleting, transposing or replacing words in the input sequence [Aho & Peterson 72, Teitelbaum 73, Lyon 74].

Aho and Peterson achieve this result by transforming the original grammar G into a weighted grammar G' which extends G with new rules accounting for three forms of ill-formedness (extra, missing or erroneous word). The original rules of G have 0 weight while the new rules have weight 1. Weighted chart parsing is then used to find a parse with the least cost.

Lyon proposes a direct construction as a variation of Earley's algorithm, which is

essentially equivalent to [Aho & Peterson 72] with pruning.

A more general approach is taken in [Teitelbaum 73]. Most error correction techniques can be expressed as finite state transductions (alias GSM mappings) on the input sequence. The Finite State Transducer (FST) encoding them may be weighted according to the likelyhood of the errors being corrected. Standard constructions can merge the original grammar G and the weighted FST into a new weighted grammar G' to be used for parsing, thus generalizing previous constructions, both w.r.t. the kind of errors corrections considered and w.r.t. the weighting scheme.

However, it is not clear that the FST should actually be combined with the grammar, instead of using both at parsing time and hopefully avoiding a combinatorial explosion, particularly when beam search is used. This approach is close to Lyon's proposal, and such a weighted error corrector for insertion, deletion and replacement at the phoneme level has been reported in [Saito & Tomita 88].

One may remark that the construction of the weighted grammar G', incorporating the FST corrections, is very similar to the construction of the weighted grammar T that generates the parsable strings in the input "lattice". Indeed, the FSA represented by the input lattice may be seen as a non-deterministic FST translating the empty string into any of the strings generated/recognized by the FSA. The construction of T may be seen as using G as a grammar for the language reduced to the empty string, with this FST as error corrector.

Inversely, given an input sentence S and an error correcting FST, one can build a word (or phoneme) weighted "lattice" — possibly with loops — which define the regular set of all transforms of the input by the FST. This lattice may then be parsed according to the standard grammar of the language.

Again, these techniques may be freely combined with the previous ones. The essential simple facts on which the above relies are:

- associativity of transducer composition,
- closure of the set of weighted FSTs by composition, and of weighted regular languages by weighted finite state transduction,
- closure of the set of weighted Push-Down Automata by composition with weighted FSTs.

But despite theoretical equivalence, specific strategies still have to be chosen according to practical implementation problems.

Deviant syntactic structures

Rather than limiting ill-formed input handling to those errors that can be expressed as finite state transformations, one can directly modify the CF grammar of the language by adding weighted rules that account for expected ill-formedness. This short-circuits the construction of Aho and Peterson, and allows expressing CF sets of ill-formed structures.

However, one could take a different view of this technique. Rather than defining a language by a very normative fixed set of rules, one could instead consider the

language as it is often used, with great variations in syntactic structures. Many "ill-formed" constructions can then be considered just as being unusual, and are thus given a correspondingly higher weight. This can apply for example to colloquial, dialectal or archaic syntax. A similar weighting can also be applied to the lexicon (which amounts to the same thing when lexicalized formalisms are used), thus turning the language description into a fuzzy structure that corresponds more closely to reality.

Weights can also be dynamically adjusted to adapt to the context in which the language processor is used.

Then syntactic ambiguity resolution may be seen as a choice of the most common syntactic structures (in a given context), or as a choice of the best formed input. Naturally, this mixes comfortably with the other views.

Conclusion

We have shown that several facets of natural language processing are only variants or views of a single problem: parsing with weighted grammars. This approach may be considered as a generative view of ill-formed input processing for at least two reasons:

- standard syntactic ill-formedness may be expressed at the level of the grammar generating the language, to such an extent that sometimes it may even be considered as part of the language description.

- all ambiguities, whether resulting from undefinedness in the input (e.g. word lattice), from expected errors or deviant syntax, or from actual language ambiguities, may be reflected in a weighted grammar that generates exactly the input sentence (or set of sentences in the lattice case) with exactly all these ambiguities. This grammar is actually a weighted shared forest of all possible parses. This weighted grammar/forest gives each parse tree a weight according to the weight schemes of the input, the error processor and the language grammar.

The formal uniformity need not imply (nor exclude) an identical treatment of all aspects of the problem. It is however a guaranty of the compatibility of various techniques, which we can expect to mix without problems.

Though the issues seem clear from a theoretical point of view, some experimenting is probably still needed to understand various practical aspects such as:

- the combination of these techniques with various chart parser optimizations [Billot & Lang 89].

- the various ways of combining with the original grammar the variants or expected ill-formedness defined by finite state devices: should it be done statically and how, or should it be done dynamically as parsing proceeds?

- the determination of the best amount of pruning of the shared parse forest, and the use of weights in the semantic processing of shared parse forests.

- the possible advantages of different weighting schemes.

Acknowledgements: This work has been supported in part by the Eureka Software Factory (ESF) project.

References

[AhoP-72] Aho A.V.; and Peterson T.G. 1972 A Minimum Distance Error-Correcting Parser for Context-Free Languages. *SIAM Journal on Computing*, 1(4) :305-312.

[BilL-88] Billot, S.; and Lang, B. 1989 The structure of Shared Forests in Ambiguous Parsing. *Proc. of the* 27^{th} *Annual Meeting of the Association for Computational Linguistics*, Vancouver (British Columbia), 143-151. Also INRIA Research Report 1038.

[Ear-70] Earley, J. 1970 An Efficient Context-Free Parsing Algorithm. *Communications of the ACM*, 13(2): 94-102.

[Lan-74] Lang, B. 1974 Deterministic Techniques for Efficient Non-deterministic Parsers. *Proc. of the* 2^{nd} *Colloquium on Automata, Languages and Programming*, J. Loeckx (ed.), Saarbrücken, Springer Lecture Notes in Computer Science 14: 255-269.
Also: Rapport de Recherche 72, IRIA-Laboria, Rocquencourt (France).

[Lan-88a] Lang, B. 1988 Parsing Incomplete Sentences. *Proc. of the* 12^{th} *Internat. Conf. on Computational Linguistics (COLING'88)* Vol. 1 :365-371, D. Vargha (ed.), Budapest (Hungary).

[Lan-88b] Lang, B. 1988 *The Systematic Construction of Earley Parsers: Application to the Production of* $\mathcal{O}(n^6)$ *Earley Parsers for Tree Adjoining Grammars*. Unpublished report.

[Lan-89] Lang, B. 1989 A Uniform Formal Framework for Parsing. *Proc. of the International Workshop on Parsing Technologies*, Carnegie Mellon University, Pittsburgh (Pennsylvania), pp. 28-42.

[Lyo-74] Lyon, G. 1974 Syntax-Directed Least-Errors Analysis for Context-Free Languages: A practical Approach. *Communications of the ACM*, 17(1): 3-14.

[Ney-82] Ney, H. 1982 Dynamic Programming as a Technique for Pattern Recognition. *Proc. of the* 6^{th} *Int. Conf. on Pattern Recognition*, München (West Germany), pp. 1119-1125.

[Ney-87] Ney, H. 1987 Dynamic Programming Speech Recognition using a Context-Free Grammar. *Proc. ICASSP 87*, Dallas (Texas), pp. 3.2.1-4.

[Nak-87] Nakagawa, S. 1987 Spoken Sentence Recognition by Time-Synchronous Parsing Algorithm of Context-Free Grammar. *Proc. ICASSP 87*, Dallas (Texas), Vol. 2 : 829-832.

[Päs-87] Päseler, A. 1987 Modification of Earley's Algorithm for Speech Recognition. Proc. of the NATO Advanced Study Institute on *Recent Advances in Speech Understanding and Dialog Systems*, Bad Windsheim (West Germany), H. Niemann, M. Lang and G. Sagerer Eds., Springer Verlag 1988, pp. 466-472.

[Tei-73] Teitelbaum, R. 1973 Context-Free Error Analysis by Evaluation of Algebraic Power Series. *Proc. of the Fifth Annual ACM Symposium on Theory of Computing*, Austin (Texas), pp. 196-199.

[SaiT-88] Saito, H.; and Tomita, M. 1988 Parsing Noisy Sentences. *Proc. of the* 12^{th} *Internat. Conf. on Computational Linguistics (COLING'88)* Vol. 2 :561-566, D. Vargha (ed.), Budapest (Hungary).

[Tom-86] Tomita, M. 1986 An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition. In *Proceedings of IEEE-IECE-ASJ International Conference on Acoustics, Speech, and Signal Processing (ICASSP 86)*, Vol. 3: 1569-1572.

[VijW-89] Vijay-Shankar, K.; and Weir, D.J. 1989 Recognition of Combinatory Categorial Grammars and Linear Indexed Grammars. *Proc. of the International Workshop on Parsing Technologies*, Carnegie Mellon University, Pittsburgh (Pennsylvania), pp. 172-181.