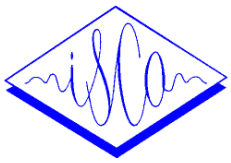

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

A LIGHTWEIGHT PARSER FOR SPEECH UNDERSTANDING

Nigel Ward

Mech-Info Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113 JAPAN

ABSTRACT

To integrate well into a speech understanding system, a parser should be evidential. The current best evidential approach, stochastic context-free grammar parsing, is not completely satisfactory. This paper presents a new model. Here the parser's input is the raw lattice of word hypotheses, and its output is a set of scored clues to a semantic interpretation; and conversely for feedback. The parser is based on "construction hypotheses" each of which maps directly to word hypotheses, on the one hand, and conceptual hypotheses, on the other. These construction hypotheses are mutually independent, which simplifies the parser. These ideas are incorporated in a tiny prototype speech understanding system.

1. BACKGROUND

The goal of building a fully integrated speech system has been with us for a long time. The hope is more robust understanding.

How to build such a system is an open question, but one promising approach is to build a fully evidential system. Making an entire system be evidential should allow fine-grained communication between all components, in the form of evidence for various hypotheses.

For speech recognition, evidential (probabilistic) approaches have for many years been the mainstream. At the syntactic level also, there is a need for evidential models; this paper proposes a new evidential approach to syntax.

§2 and §3 discuss some limitations of existing evidential approaches to syntax, §4 re-analyzes the parsing problem, §5 presents an evidential parser for speech understanding, and §6 and §7 discuss results and future work.

2. ISSUES AND RELATED WORK

This section discuses some shortcomings of "structuralist" approaches to parsing for speech, including the best studied evidential approach, namely parsers based on stochastic context-free grammars.

2.1 The parser/recognizer interface

Traditionally, the input to a parser is a string of words. However, the output of any (unaided) recognizer is at best a lattice of word hypotheses. This mis-match is resolved by adding some mechanism for pruning out word hypotheses or otherwise organizing them into strings (the "sentence hypotheses"). The parser is then able to work from these strings. Thus the recognizer/parser interface is commonly in terms of strings of words.

One problem with this is the potential loss of information entailed by pruning before the application of syntactic and semantic knowledge; that is, the premature resolution of uncertainty (typically using a separate "language model" which incorporates some approximate linguistic knowledge).

Further, this recognizer/parser interface is problematic for providing feedback to the recognizer. One hope for evidential parsing is to allow the full weight of syntactic knowledge (and semantic) to be brought to bear on the recognition problem (and incidentally, removing the need for a language model). But in fact most probabilistic parsers accept only, and rescore only, sentence hypotheses (with the exception of (Jurafsky *et al.* 1994)).

§4 returns to this question, discussing how to build a parser which instead interfaces directly to the lattice.

2.2 The parser/semantic interpreter interface

Traditionally the output of a parser is a parse tree or other structural description of the input sentence. However, this is not necessarily directly useful to the semantic interpreter (back-end). This becomes a significant problem if one is trying to build a "tightly-coupled" architecture, that is, one where semantic considerations, mediated by syntactic hypotheses, are used to rescore word hypotheses (Seneff 1992). The difficulty is in computing the implications of semantic evidence for syntactic hypotheses (which is the first step in the process of using the top-down scores to help the recognizer).

The usual way around this issue is to use semantic grammars, where simple semantic constraints are encoded in the grammar. This does allow some semantic knowledge to be applied early in the speech understanding process, but only certain types of semantic knowledge.

Looking forward to the day when more general semantic interpreters will be available, it seems better to keep semantics and syntax as separate knowledge sources. When doing so, a central problem is that of designing a parser capable of interfacing tightly enough to the back-end to exploit semantic feedback.

2.3 Parsing for spoken language

Structuralist models of syntax arose from early work which focused on written texts and which took grammatical sentences to be the normal case. The latter, though a useful simplifying assumption, is blatantly false for the description of spoken language. When judged as process models also, structuralist models come up short — they are not a very good model of what goes on in human language production (Ward 1994b).

As the focus of speech understanding research moves towards more realistic, spontaneous speech, this mismatch between model and reality becomes more worrying. It will not be long before a main topic of speech research will be how to use rich semantic or domain knowledge to compensate for

poor recognizer performance — and doing so may require different models for parsing.

3. GOALS

It may be that the problems mentioned above can be solved by appropriate refinements and elaborations to structuralist parsers, and indeed there are currently several research efforts in this direction.

Another class of approaches to syntax for speech understanding, also attractive, gives up on the idea of doing a full parse. For example, there have been proposals for 'sketchy' parsing, 'skimming' parsing, or 'partial parsing'. These mostly rely on the easily-obtainable information from syntax, e.g., by parsing noun phrases, while neglecting to extract the information available from larger syntactic constructions.

This paper maps out a third approach: to work towards a non-structuralist, evidential, full-strength parser. It is a preliminary study in this direction, and so takes an idealistic approach, with little attention to engineering considerations. There is no chance that this will give good results in the short term, but it may eventually lead to a better way to handle syntax for speech.

4. THE ROLE OF THE PARSER

This section argues for a new view of the role of the parser in speech understanding.

First, to review an obvious fact, spoken utterances are often fragmentary and or noisy. Even when they aren't, they underdetermine the "meaning", since, of course, real speakers only convey the minimal amount of information needed for the hearer to reconstruct the "message".

Therefore (A) *the responsibility of the recognizer and parser should be only to come up with "clues" that the semantic interpreter can use.* (A similar point has been made by Bates *et al.* (1993).) The clues may be mutually inconsistent and will invariably be incomplete. This proposal implies that all problems of ambiguous or confusing utterances should be left for the semantic interpreter to deal with. This contrasts with the traditional view that syntax should output a complete and consistent interpretation of the input, one that maps into a "meaning".

(Note that this is not an abandonment of the goal of extracting from the input every possible bit of information that syntax can provide; merely a change in the form in which that information should be provided to the interpreter.)

Defining the parser/semantic interpreter interface in this way solves problem 2 above, in that it is easy for the semantic interpreter to provide top-down scores to the clues, and for the parser to use these scores in turn to rescore syntactic hypotheses (as illustrated in §6).

I propose also that (B) *each syntactic hypothesis should relate directly to word hypotheses, on the one hand, and conceptual hypotheses, on the other.* A syntactic mechanism that operates along these lines can be called a "lightweight parser". It is lightweight in the sense that it requires no overhead to invoke, being simply a layer in the network of hypotheses (lexical, syntactic, and conceptual), as suggested by Figure 1. It can be lightweight also in the sense that it requires a minimum of computation and a minimum of working memory. I should note that calling it a "parser" is appropriate in that it applies syntactic knowledge, although, unlike most parsers, it does not output a parse tree.

Proposal B implies that all syntactic hypotheses should be mutually independent — there should be no direct evidential links from one syntactic hypothesis to another. This is in contrast to most parsers, which form complex syntactic hypotheses by coordinating, unifying, assembling, or linking the elements (sub-trees etc) of a syntactic interpretation into trees or other structures. This is just about manageable (using charts etc.) for written input. For spoken input, however, the huge number of word hypothesis in the raw lattice, many overlapping, combined with the huge number of composite syntactic hypotheses, seems computationally overwhelming.

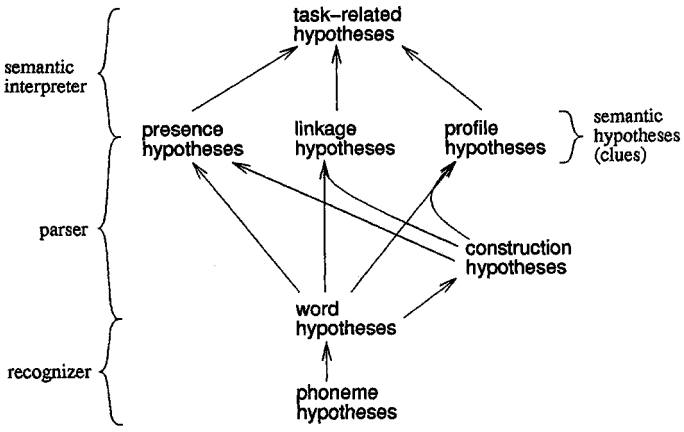

Figure 1: Hypothesis types and their relations.

Adopting Proposal B simplifies the parser, by factoring the representation of the current 'theory' as to the structure of the input into independent syntactic hypotheses. This decreases the number of syntactic hypotheses the parser needs to work with (since there are no composite hypotheses). Therefore it can compute the evidence for all syntactic hypotheses supported by the lattice, without drowning in computation (solving problem 1 above).

5. IMPLEMENTATION

I have implemented a system embodying these ideas (Ward 1994a). It uses some notions developed in a model of the syntactic aspects of human language production (Ward 1994b). Being single-mindedly devoted to the goal of building a simple demonstration of a new approach to integrated evidential understanding, I gave little weight to practical considerations. As a result the resulting architecture is extreme in some ways. Its value, I hope, will lie in the new perspective it gives.

5.1 Linguistic/Conceptual Hypotheses

Proposal A states that a parser should produce "clues", meaning conceptual hypotheses which are incomplete and possibly inconsistent. We want these clues to be "small" and "simple", to simplify the parser (and possibly also the semantic interpreter). Therefore, I propose that *the "conceptual hypotheses" for which language provides evidence are of three types: 1. presence information, e.g. "this input involves john", "this involves a beneficial action", 2. linkage-information, e.g. "in this input john is related to kiss", and 3. relational information, e.g. "john is very active in this input".*

Note that this list decomposes traditional case relations, such as "*john* is the *agent* of *kiss*", into one clue of type 2 and several clues of type 3. This is an refinement of the idea of representing relational information with "profiles", that is, vectors of values over "case

features", instead of Fillmorean deep cases (Ward 1992; Ward 1994b). It is a refinement in that the earlier proposal did not properly segregate evidence; for example it treated 2 units of evidence for *(.8 responsible)* as equivalent to 1 unit of evidence for *(.4 responsible)*. The current scheme computes the evidence for each degree of each case feature independently. However, since the evidence for each degree is not completely independent, there is smoothing; for example, evidence for *(.6 topic)* also counts as (weaker) evidence for *(.5 topic)*.

The value of this technique is that it makes it easy for many construction hypotheses to independently contribute clues to an interpretation; these clues can be combined numerically. In particular, several hypotheses overlapping at a certain time span can contribute synergistically to the interpretation of the semantic role of words in that time span. The fact that these contributions can be combined numerically means that there is no need to coordinate or unify constructions: this is what allows the parser to make do with independent syntactic hypotheses.

This technique also eliminates the need for a "case-slot mapping" to map grammatical relations to the argument structure of predicates — both the parser and the semantic interpreter can use profiles as a representation mechanism.

5.2 Construction Hypotheses

Finally, it is time to describe how the parser works. It turns out to be shockingly simple.

The key idea of this parser is the notion of "grammatical construction", adopted from Fillmore *et al.* (1988). The basic idea is that it is possible to represent syntactic knowledge as an inventory of constructions, analogous to the representation of lexical knowledge as an inventory of words. Each construction is a pairing of form and meaning. The form is a sequence of constituents. For concreteness, each constituent can be visualized as referring to a word, a syntactic category, a profile, or some combination thereof. Examples of constructions are the Subject-Predicate Construction, the Transitive Construction, the Adjective-Noun Construction, and the Passive Construction. It seems that constructions are a representational mechanism adequate for writing complete grammars; and doing so is the enterprise of 'Construction Grammar' (Fillmore 1988).

Thus, the parser uses *syntactic hypotheses of the form 'constituent X of construction Y was present over time span Z'*. Such 'construction hypotheses' are easy to relate to word hypotheses, on the one hand, and to conceptual hypotheses, on the other. The remainder of this section sketchs out how this is done.

A construction hypothesis can be spawned when there is a good match between the constituents of the construction and the hypothesized words in some time range. The current algorithm is roughly as follows. First the system locates timepoints which are likely places for an inter-constituent boundary (e.g. for the Subject-Predicate Construction it looks for places where the end of a noun hypotheses is near the start of a verb hypothesis). Second, for each plausible boundary placement, constituent spans are determined by examining the evidence distributions over time for the relevant syntactic categories (e.g. for the first constituent of Subject-Predicate it looks at the evidence for nouns, summed at each timepoint over all noun hypotheses). Finally, for each construction hypothesis thus instantiated, its evidence is computed by integrating over each constituent span its support, as determined by the scores of the supporting word hypotheses.

A construction hypothesis which spans a certain time range can be used for interpreting that part of the input. To give just one example, suppose that: A. there is a hypothesized occurrence of the Subject-Predicate Construction for which the first constituent spans the time from the 10th to the 22nd frame, and B. an occurrence of the word *"John"* is hypothesized in the time span from the 9th to the 19th frame. From this, since the time spans overlap, there is evidence for *john* being the "subject" (actually, for him having the profile *(.6 topic), (.4 volitional), (.3 active), (.5 responsible), (-.2 affected)* .... This is then a bundle of clues to pass on to the semantic interpreter. (Conversely, for feedback, semantic rescoring of such a clue directly causes rescoring of the associated construction hypothesis or hypotheses.)

Figure 2 illustrates some construction hypotheses.

5.3 Timeline-based Reasoning

As suggested above, the system *relies on overlapping time spans for computing evidence.* This is done to avoid the complexity involved in binding words to the constituents of constructions. (An attempt to build an evidential parser which did binding convinced me that avoiding this was important, especially when the parser works directly from the raw lattice.)

The technique used can be called "timeline-based reasoning". For example, if there is the hypothesis that "the part of the input from frame11 thru frame18 corresponds to direct-object", then there is evidence for *(affected .8)*, and this evidence is stored on positions 11 thru 18 of the *(affected .8)* timeline. If there is also the hypothesis that "the word *"John"* appeared from frame12 thru frame16", then, by using the information on that timeline, the evidence regarding the degree of affectedness of *John* can be easily computed.

Evidence from many construction hypotheses (etc.) is summed onto each timeline. As there is only a small number of timelines, much fewer than the number of hypotheses of various types, this technique makes the parser relatively fast.

6. EXPERIMENTS

This proposal has been implemented as part of a small speech understanding system (Ward 1994a). ("Small" means 4000 lines of C and 6000 lines of Lisp, with tests run with an 11 word vocabulary, 3 grammatical constructions, and inputs of 2 to 5 words, single speaker, continuous speech.)

The parser does in fact provide syntactic feedback to the recognizer; that is, construction hypotheses provide evidence used to re-score the word hypotheses. This is effective: on a 25 utterance micro-corpus, the correctness of the lattice (roughly, the product of the positional accuracy and the score accuracy, averaged over all word hypothesis) was 15% higher after the application of this feedback.

The parser does support the extraction and scoring of clues, as illustrated in §5.2.

The parser does support syntactically mediated semantic feedback to the recognizer. For example, given that A. the conceptual hypothesis *'john-is-active'* is highly ranked, and B. there is a hypothesized occurrence of the Subject-Predicate Construction where the first constituent is positioned around frame12, then there is evidence for any hypotheses that the word *"John"* appears in the input around frame 12. With a little more work here, I hope to measure whether evidential parsing, by allowing a more tightly integrated system, provides the hoped-for improvement in overall system robustness.

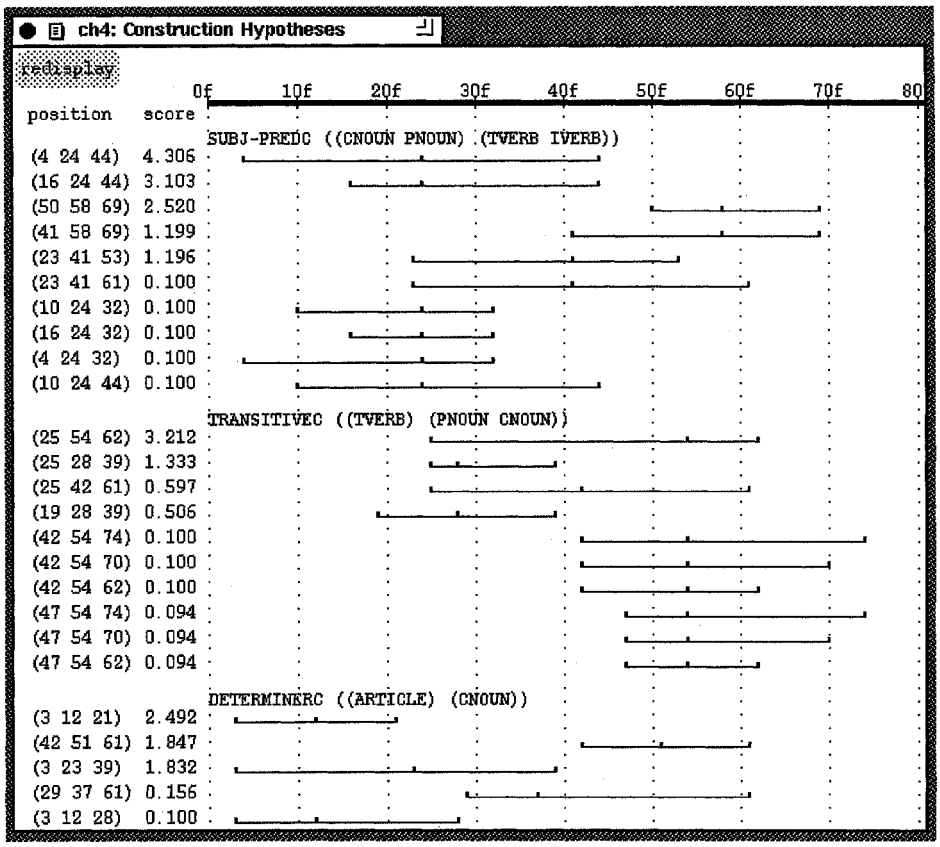

Figure 2: Construction hypotheses for an input consisting of a fairly sloppy pronunciation of *"Batman rescued the mayor"*. The x-axis is time; the dotted lines mark 200ms intervals. The horizontal lines indicate the durations of the constituents for each hypothesis. The first *subject-predicate*, the third *transitive*, and the second *determiner* construction hypotheses most nearly correspond to a correct interpretation.

7. STATUS AND HOPES

So far, I've only sketched out a model of syntax for understanding. The contribution is the analysis of the types of hypotheses involved and how they should relate to the hypotheses in the other components of the system. However the current mechanisms for computing evidence — including the basic rules for scoring construction hypotheses from word hypotheses, and for scoring clues from construction hypotheses — are ad hoc.

The current implementation serves primarily as a testbed to enable experimentation with different algorithms here. It includes a window interface to enable the developer to adjust the scores of hypotheses, open or close various feedback pathways, and observe how the evidence propagates through the system. Having gained more experience with the behavior of the system and some ad hoc algorithms, it will be time to properly analyze the system, put the computations on a firm probabilistic foundation, and develop techniques for obtaining the probabilities from data.

References

Bates, Madeline, Robert Bobrow, *et al.* (1993). The BBN/Harc Spoken Language Understanding System. In *1993 IEEE ICASSP*, pp. II-111–114.

Fillmore, Charles J. (1988). The Mechanisms of "Construction Grammar". In *Berkeley Linguistics Society, Proceedings of the Fourteenth Annual Meeting*, pp. 35–55.

Fillmore, Charles J., Paul Kay, & M. C. O'Connor (1988). Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone. *Language*, 64(3).

Jurafsky, Daniel, Chuck Wooters, *et al.* (1994). Integrating Experimental Models of Syntax, Phonology, and Accent/Dialect in a a Speech Recognizer. In *AAAI Workshop on the Integration of Natural Language and Speech Processing.*

Seneff, Stephanie (1992). TINA: A Natural Language System for Spoken Language Applications. *Computational Linguistics*, 18(1):61–86.

Ward, Nigel (1992). An Alternative to Deep Case for Representing Relational Information. In *Proceedings 14th COLING.*

Ward, Nigel (1994a). An Approach to Tightly-Coupled Syntactic/Semantic Processing for Speech Understanding. In *AAAI Workshop on the Integration of Natural Language and Speech Processing.*

Ward, Nigel (1994b). *A Connectionist Language Generator.* Ablex.