

Second Thoughts on an Artificial Intelligence Approach to Speech Understanding

(Japanese title goes here)

Nigel Ward
University of Tokyo

Abstract

A few years ago I undertook a new speech understanding research project, aiming to explore innovative techniques rather than pursue short-term results. My method was to build on the classic 1970s AI approaches to speech, as an alternative to the current mainstream speech understanding research methods. This led to a system that was competitive, both in elegance and performance, with other recent AI-inspired speech understanding systems. However, evaluation of results and prospects led to the realization that the system had no future. This paper analyzes the roots of this failure as a case study in AI methodology gone awry. In particular, it explains why my original, classicly AI goals — namely, be optimal in principle, be well integrated, iteratively refine the interpretation, deal directly with noisy inputs, be linguistically interesting, be tunable by hand, work with clear hypotheses, be architecturally innovative, and relate to general issues in AI — are less important than they seemed.

1 Background

Artificial Intelligence (AI) approaches to speech understanding were actively pursued in the mid-1970s but have been low-profile since. The 1990s seemed like a good time for another try, for several reasons. One reason is the availability of faster computers and nicer Lisps. Another reason is the new tools and techniques from connectionism and from recent linguistics. Yet another reason is that, due to advances in speech *recognition*, for many applications it is the problems of language and meaning that are the limiting factors (Hirschman 1994; Pallett & Fiscus 1995).

So I set to work. After a few years I had a working prototype of a rather novel speech understanding system (Ward 1994a). Prior to scaling this up to a realistic high-performance system, I evaluated the prototype and did another literature review. Doing so led to an unexpected conclusion: the system I had envisioned was not worth building; the task I had set myself was not really meaningful.

This paper analyzes how I misconceived the task of speech understanding. This is worth doing because many of the misconceptions which I suffered from also afflict the classic AI work on speech (Reddy *et al.* 1973; Woods & Makhoul 1973; Klatt 1977; Erman *et al.* 1980; Woods 1980) and other recent AI work (Hayes *et al.* 1987; Kitano *et al.* 1989; Baggia & Rullent 1993; Nagao *et al.* 1993; Kawahara *et al.* 1994; Hauenstein & Weber 1994; Cochard & Oppizzi 1995). Thus, while the paper is written as a collection of self-criticisms, it is intended also to be a high-level critique of AI approaches to speech understanding in general.

This paper is directed less to speech understanding researchers than to AI researchers. The aim is to discuss my experience in a way that highlights some difficulties with the classic AI research methodology. Readers seeking discussion of speech system technology should look elsewhere (Lee 1994; Nguyen *et al.* 1994; Moore 1994; Jurafsky *et al.* 1995; Seneff 1995; Moore *et al.* 1995).

2 Goals and Second Thoughts

This section presents the goals I set for my system, interleaved with second thoughts.

1 It seemed worthwhile to work towards a system that could *produce an optimal interpretation based on an exhaustive understanding of the input*, that is, to *understand as well as people*.

Speech research today is mostly unabashed engineering; most researchers strive to attain measurable, if small, improvements to existing models performing specific tasks.

I didn't want to restrict attention to any specific task, fearing that could result in ideas without wider significance; rather I wanted to work on the general problem. Also, I didn't want to worry about immediate payback, rather I wanted to work towards an ideal and new model.

In particular, I wanted to build a system that would model how people understand speech, believing that this would in the long term lead to a system that would understand as well as people do. For me, human performance seemed essentially perfect: people can understand just about anything you say to them, and even repeat back to you what you said.

¬1 *But the feeling that perception and understanding is complete is an illusion of introspection (Brooks*

⁰nigel@sanpo.t.u-tokyo.ac.jp

1991; Dennett 1991). *The limitations of normal human understanding of speech are easy to observe: if you record a conversation and later listen to it repeatedly, you discover a lot that you missed when hearing it live. The fact that human understanding is limited makes sense from a functional perspective: understanding just enough to react or respond appropriately is good enough for human needs.*

Also, there is no reason to think that people can, under normal conditions, extract the sequence of words reliably, let alone understand completely. Cases where people apparently can, as when shadowing (Cole & Jakimik 1980) or taking dictation, do not tell much about the basic, normal process of recognition and understanding, for two reasons. First, these tasks require a very atypical amount of attention. Second, dictation is not the simple process it seems: it probably involves a subprocess of reconstructing the input from the extracted meaning, compatibly with the key content words and perhaps traces of the intonation contour which remain in short-term memory.

2 To achieve optimal understanding, it seemed necessary to build a system that would be well integrated and, in particular, include feedback.

Most of today's state-of-the-art speech systems rely on a separate "language model" consisting of a little simple knowledge about language, and this is the only higher-level knowledge that the recognizer can access. This seemed inelegant.

I wanted to build a system where the full inventory of higher-level knowledge, including knowledge of syntax, semantics, domain, task and current dialog state, is available and exploited to assist the recognition process, rather than being applied only at the last stages of understanding. That is, I wanted the understander and the recognizer to be tightly coupled (Jurafsky *et al.* 1995; Seneff 1995) rather than operate as independent modules. It seemed obvious that applying more knowledge in the recognition process would enable the recognizer to compute better results.

And it seemed that the human recognition mechanism does employ such higher-level knowledge. This is shown by misperceptions, such as when the phrase "wreak a nice beach" is heard as "recognize speech" in an appropriate context. It is also shown by apparent semantic influences on the phoneme restoration effect, that is, the fact that sounds totally obscured by noise are often perceived as if they were in fact present. Also, listeners sometimes feel that they know what words someone is going to say next and are only listening for the sake of confirmation.

Learning from past AI work, I understood that pure "analysis by synthesis", generating all likely utterances and matching the input to them, is in general too slow to be useful. But it still seemed that top-down knowledge should contribute somehow to help the recognizer.

In particular, my short-term goal was to demon-

strate the use of reasoning to compensate for misrecognitions. Such functionality, while not directly useful, seemed to capture the essence of feedback, independent of the peculiarities of any specific type of recognizer, and independent of whether efficiency and accuracy are best served by feedback in the form of guidance during the initial search or in the form of subsequent requests for verification of hypotheses against the input. And it seemed that a system that could correct for mis-recognitions would be clearly impressive.

Since feedback to the recognizer must be in terms of the words at specific positions (times) in the input, I decided to focus efforts on syntax, the bridge that relates task knowledge and the positions of words.

¬2 *But using more knowledge does not necessarily give better performance; when processing time and memory are finite some knowledge may not be worth considering. Determining whether more knowledge is a good thing in any specific case requires careful experimentation (Nguyen *et al.* 1994).*

There is is no intrinsic need for feedback, since it is in principle possible for a recognizer to spot all possible words in parallel, without using any higher-level knowledge, and to pass all the resulting hypotheses to the understander. Thus the issue of whether and how to use of feedback is purely an engineering matter, not existing as a problem independent of a specific recognizer.

Nor is it obvious that people actually use higher-level knowledge on-line during the process of speech recognition. For example, phenomena like phoneme restoration may be attributable to some post-recognition process. The experimental evidence from psycholinguistics seems inconclusive, as well it might, given the difficulties of imagining experimental results which could be accounted for only by an interactive model (McClelland 1987; Norris 1993).

Nor is it obvious that higher-level knowledge is needed that much; many apparently top-down effects, as in "wreak a nice beach", can perhaps be handled simply by different probabilities for different words or word sequences depending on the domain of discourse.

The purpose of an understander is to understand. It is far from clear that explicitly noticing and correcting mis-recognitions helps understanding. It's easier, and probably better, for an understander to stitch together an interpretation out of whatever hypotheses have high scores, ignoring the rest.

3 To arrive at an optimal interpretation, it seemed necessary to perform many cycles of computation.

Today's mainstream speech understanding systems apply each knowledge source once only.

I wanted to build a system which would produce globally optimal interpretations, as judged using knowledge of all types together. I envisioned a system where knowledge sources would cooperate by taking turns: for example, a partially recognized input could

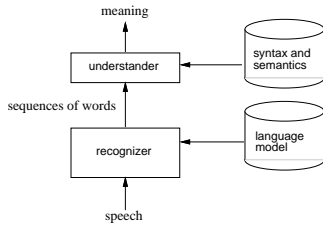


Figure 1: A typical model for speech understanding.

lead to a partial understanding, that understanding could be used to “figure out” more words, leading to a better recognition result, then a better understanding, and so on. Introspection suggested that this is how people understand speech.

This also led me to concentrate work on syntactic hypotheses as mediators of information flow in both directions: up from word hypotheses to conceptual structures, and down from conceptual structures to word hypotheses.

¬3 *But it is not clear that multiple cycles of interaction is the best way to integrate diverse sources of knowledge. One problem is that interactive models are hard to work with. Systems with more controlled flow of information are easier to develop, since it is easier to relate the contribution of each component to the final output of the system. Doing so is important for, among other things, optimally setting the weights to give to the contributions of each knowledge source (Alshawi & Carter 1994).*

Also, there is no real reason for having a hypothesis collect and combine evidence from both bottom-up and top-down sources. The alternative — duplication of hypotheses, for example, having one syntactic hypotheses used for bottom-up information flow, and a doppelganger hypothesis for top-down — allows independent algorithms in each direction, each tailored for its task.

Human language understanding is probably not normally like puzzle solving; the introspective experience of “figuring out” an input is probably not related to normal on-line processing. In those cases where “figuring out” does occur, it may be that the top-down information is applied via an independently existing process, namely the language generator.

4 To achieve tight integration, it seemed that the understander should be tied as closely as possible to perception, and should therefore *deal directly with noisy inputs*.

In many speech understanding systems, the understander’s task is not terribly different from the task of an understander for text. In a typical model it start from a sentence-like sequence of words and works in one direction (Figure 1). Some recent sys-

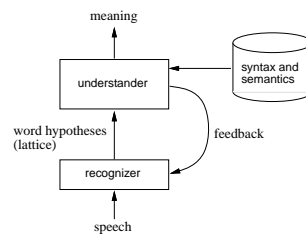


Figure 2: An AI model for speech understanding, where the understander provides feedback (goal 2) and deals directly with noisy inputs (goal 4).

tems include two refinements. First, the recognizer produces several candidate word sequences, in other words the “N-best” sequences (Nguyen *et al.* 1994), sometimes compressed into “word graphs”. Second, the recognizer outputs acoustic scores along with the sequences, and the understander uses these to compute the overall likelihood of various possible interpretations. Nevertheless, in most of today’s systems the understander is still insulated from the uncertainty of the individual word hypotheses; thus much of the problem of dealing with noise and uncertainty is still consigned to the recognizer. This reflects the fact that most parsing models are not as affected as much as they might be by the needs of speech.

I wanted to give the understander access, not to a pruned subset of the information available, but to all the information the recognizer can provide. Learning from past AI work, I realized that it would be a mistake to extend AI techniques down to syllables, phonemes, or acoustic features (statistical models are hard to beat for these), so I drew the line at words.

In particular, I wanted to build a system where the understander starts work directly from the lattice of word hypotheses. That is, I wanted the understander to have access to all the word hypotheses, *including their start and end points*, and their scores based on the acoustic evidence. Figure 2 illustrates this, and also goal 2 above.

Introspectively this seemed a plausible way to modularize the process. Since there are situations where human listeners only catch a word or two of an utterance, it seemed that the human recognizer might operate by spotting words independently of each other, and therefore that the recognizer/understander interface should be in terms of a lattice of such word hypotheses.

I saw two specific needs. First, giving the understander access to the full lattice seemed necessary to allow it to figure out which of the multitudes of word hypotheses were masking regions that in fact contained out-of-vocabulary words, “umm” sounds, or background noise. Second, letting the understander work with the actual times of word hypotheses seemed the key to a way to provide more specific feedback to

the recognizer. Even in today's most tightly coupled systems, syntactic and task knowledge is used only to constrain the transitions between words (that is, to specify follow sets). It seemed that feedback could usefully be more specific, for example, not just providing information like "after word x , word y is more likely" but also "from p milliseconds to q milliseconds, word z is more likely". This could even support the creation of word hypotheses for which the recognizer had no bottom-up evidence. It seemed that a system that could do these things would be clearly impressive.

Reacting against past AI understanders, which could access the whole lattice but whose processing style was to pick and choose which hypotheses to process, I chose to build an understander able to process all word hypotheses and do so in parallel. In slogan form, rather than extending the serial nature of the understander down into lattice processing, I wanted to extend the parallelism of recognition up to higher levels. (This was a challenge I welcomed, since it required developing a mechanism for parallel and evidential syntactic reasoning, something I had experience doing (Ward 1992; Ward 1994b).)

¬4 *It's simplistic to think that word hypotheses need be the only input to the understander. There are many other types of information that can be exploited, including pitch contours, patterns of stress, and variations in speaking rate, and perhaps also the speaker's gestures and environment. If these types of information are available to the understander, then non-words, such as "uh"s, pauses, and word fragments, and background noise, may be explicitly detectable. For example, false starts are identifiable to some extent by using the fact that the corrections that follow them generally are at a higher pitch. In general, if other sources of information are available then it no longer seems so important to give the understander access to every detail of every word hypothesis.*

Computational considerations suggest that word sequences, not unorganized lattices, are the natural output from the recognizer after all. First, there is the need to deal with phonological assimilation across words. The natural place to deal with this is in the recognizer, since it is of course the repository for knowledge about the sounds of words. But to do this the recognizer must internally work with sequences of words, and so its output also may as well be sequences of words. Second, the abutting constraint (the fact that two word hypotheses which abut are both, for that reason, more likely) is very useful and easy to compute in the recognizer. Thus pruning the lattice into a set of sequence hypotheses is something which the recognizer can naturally do.

Computational considerations also suggest that word sequences are the natural input for the understander (Moore et al. 1989; Moore 1994). For one thing, the parsing and understanding of lattices,

rather than sequence hypotheses, seems intrinsically very costly. For another, it is not at all clear that the syntactic aspects of understanding require direct access to the specific times when words occur, although this is vital for interpretation based on prosody.

Also, regarding the argument from introspection, the fact that people sometimes catch only one word of an utterance is scarcely evidence that normal listening is based on word-spotting alone.

5 It seemed worth trying to build a system that would be interesting from a linguistic point of view.

Today's speech systems embody theories of syntax and semantics but are not linguistically very interesting. Many speech systems are conservative — they basically just use standard models, such as context free grammars, unification grammars, and chart parsers. Others are purely practical, incorporating models that are ad hoc by the standards of linguistics, largely in the way they squeeze task and domain knowledge into the grammar formalism, such as augmented transition networks and semantic grammars. Still others apply syntactic and semantic knowledge in the form of statistical models that make no contact with linguistic ideas (Levin & Pieraccini 1995).

Another problem is that few understanders for speech handle interesting grammatical phenomena. Of course this is understandable; if developed as part of a system that can only respond to utterances about simple things, like tuples in a database or items on a computer screen, the demands on syntax are not very great; the semantics of the domain may well map directly to simple SVO sentences. But, looking to the future, there is a need for general (task and domain independent) models of syntax and semantics for speech.

So I set out to treat the syntactic needs of a speech understander in a way that would make the implications for linguistics clear. Doing so would serve to motivate linguists to learn from and eventually contribute to speech understanding research: specifically the development of processing models of language and the development of syntactic models that deal directly with noise (goal 4).

In particular, I set out to develop a computational model of grammar that would be task independent, handle interesting linguistic phenomena, and relate to recent linguistic theories (Goldberg 1995).

¬5 *But "handle interesting phenomena" is a very unconstrained goal. It makes it hard to convince people that what you achieve is important. There really is a lot to say for the idea of setting a concrete goal, even if it is as mundane as handling some percentage of the utterances of some corpus.*

Another problem with a focus on language rather than performance is that it can lead to a system whose syntactic coverage exceeds its semantic coverage. This loses one of the main advantages of natural language processing over linguistics as an approach to language:

that the semantics can be real — grounded in a task and users' needs.

6 To leap ahead of the state of the art, it seemed necessary to work with a system that could *be debugged and tuned by hand*.

Today's systems are trained on data. This enables good performance but is limiting. Current training algorithms assume that the training data is representative of the task the system will perform. This implies that systems have to be task specific. For the long term goal of building general-purpose systems it will be necessary to train with data that is not task limited, perhaps in analogy to the way that children learn to understand language. Looking forward to advances in this field, I decided not to place too much importance on building a system that would be trainable using current methods.

More pressingly, there are currently no algorithms for training heterogeneous interactive systems, and so, in order to get the freedom to use an innovative architecture, I had no choice but to build and tune a system entirely by hand. This, however, was fine by me. As my goal was to explore new ideas, I wanted to work intimately with the system, analyzing its behavior to see if it was working in the ways I had imagined, and to tell whether those ways were in fact viable. In essence I was using experience with the internal workings of a program to leverage introspection about how a cognitive process might operate. This technique was for me the key aspect of AI methodology, and the one that held out the hope of achieving results faster than any normal scientific or normal engineering approach.

The specific development process I used had two aspects. One was to find an input where the lattice, although interpretable by me, had not been handled correctly by the system, and then determine which hypotheses were missing or mis-scored, and fix the system to get it right. Learning from past AI work, I knew that this method alone was insufficient, since an algorithm that cleverly handles one input does not necessarily do well on all the utterances you'd like to handle. So I supplemented this method with periodic checks on performance on a little corpus.

—**6** *When I finally benchmarked my hand-tuned system against the simplest statistical model, bigrams, in terms of ability to provide feedback, I found it to be not just inferior, but completely outclassed. Further, there was no way to retrofit probabilistic reasoning to my system: building a statistics-based version would require a complete rewrite to restructure the knowledge to support probabilistic computation.*

7 It seemed worthwhile to *have easily understandable hypotheses and inferences*.

Today's speech understanding systems are basically implementations of pattern recognition and search algorithms. I wanted to build a system where it would be possible to understand more intuitively

what computation is going on. To achieve this, I wanted to build a system that operates in a way that a human can empathize with. To do this requires use of suggestive metaphors, such as the historically important one of viewing the speech understanding process as a process of problem solving by a committee of intelligent people.

However, learning from past AI work, I realized that systems based on anthropomorphic metaphors can easily become computationally inadequate or overly complex. To avoid getting carried away with such metaphors, I early on adopted five principles based on computational considerations; I planned a system with: 1. an internal state consisting of multitudes of hypotheses, rather than just a few hypotheses, as introspection or a protocol study might suggest, 2. each hypothesis being simple and covering only part of an input, rather than having complex hypotheses that represented full theories of the input, 3. information flow among modules being essentially continuous, rather than communication between modules being a rare event, 4. interaction between modules being simple, rather than communication by information-rich messages, 5. modules operating in parallel, rather than modules taking turns under the control of some scheduler. In sum, I avoided thinking in terms of intelligence or reasoning in favor of thinking in terms of mass-production. This was possible because I planned to use pervasive, cheap, simple parallel computation. More specifically, I envisioned a network representation, where the nodes of the network were hypotheses of various kinds, the links among hypotheses were the paths for information flow, and the final state would be a low-energy one that represented an overall optimal interpretation (McClelland & Elman 1986).

At the same time however, I wanted to design the system so that its behavior would be intuitively understandable, and believed that the key to this was an intuitively understandable representation of internal state. Specifically, this meant explicit hypotheses whose content and current scores would be easy to inspect. This would make it easy to verify that the hypotheses and scores changed on-line as new input came in, a key to cognitive plausibility (goal 1). This would also make it easy to demonstrate it to visitors to illustrate ideas (goal 5). For example, I wanted to be able to say to a linguist: "look, here's how the system is representing the possible syntactic structures of the input; note that it's simpler than an old-fashioned parse tree, but still can do basically the same job." This would also aid development, so that I could look at a module's inputs and outputs and see what it ought to be doing (goal 6).

In particular, making syntactic hypotheses be visualizable seemed the key, since they served as the gathering point for a lot of information. The idea was that the developer would be able to see all the evidence for and against a syntactic construction, from

both bottom-up and top-down sources, and also the implications, both for meaning and the input, if that hypothesis was in fact correct, and to make adjustments on the fly. This was fun to do, in Lisp of course, and I further added an easy-to-use graphical interface, using Garnet (Myers *et al.* 1993).

¬7 *Focusing on the intermediate results of the computation diverts attention from the question of whether the computation produces the correct final result.*

Visualizable hypotheses are not a source of valid insights only. For example, thinking about what the system should do with hypotheses at different times led me to think that right-context effects are different from left-context effects and require special consideration (McClelland 1987), although in fact both can be handled uniformly in the course of search or in a recurrent neural network.

8 It seemed worth trying to build something that would be elegant and interesting in terms of computational architecture.

Today’s typical paper on speech understanding dispassionately reports an idea and its quantitative effects on performance. It is solid but unexciting.

I hoped that the study of speech understanding would have wider implications. Of course, it’s possible to carry this too far; AI researchers working on speech have boldly called for new architectures for computer networks, software systems, and computer hardware, based on the insights obtained from building speech systems. On the other hand, they have often designed their systems to run well on imaginary future computers. Such arguments get out of date very quickly. So the only hardware considerations I indulged in were based on the nature of the human brain, or at least the connectionist image of it.

In particular, I focused on achieving a general-purpose, elegant, and connectionistically plausible method for syntactic processing, namely one that avoided structure-building, specifically the binding of words to constructions and the binding of constructions into sub-trees and then trees.

I also wanted my model to be clean and elegant, according to the tenets of software engineering. This made system-building more challenging. Also, to the extent that I succeeded, it licensed me to make claims based on just a small system, confident that it would scale up.

¬8 *But architectural elegance alone counts for little, if you can actually measure the bottom line performance of your system.*

And architectural elegance does not guarantee that a toy system will scale up.

For real speech tasks, achieving good performance is itself quite enough of a challenge.

9 I wanted to build a system that would cast light on important issues in AI.

Learning from past AI experience, I didn’t expect analogies relating speech understanding to other fields, such as planning, to be very useful. Rather, I was on the lookout for more abstract connections; I wanted to build a system that would have something to say about the Constraint Propagation Problem, the Context Problem, the Disambiguation Problem, the Evidential Reasoning Problem, the Knowledge Integration Problem, the Knowledge Representation Problem, the Noisy Inputs Problem, the Real-World Problem, the Reasoning With Uncertainty Problem, the Sensor Fusion Problem, the Signal-to-Symbol Problem, and a few others.

In particular, my basic need was to conduct a research program that would let me write papers and grant proposals that “positioned and motivated [the] work in the larger context of the general AI community” so I could get published and funded.

¬9 *But thinking in such abstract terms is almost guaranteed to lead to solutions that don’t solve any real problems.*

3 Implementation and Significance

I built a system to meet these goals. Many interesting technical problems arose. Some I partly solved, developing new techniques for the representation of syntactic and semantic hypotheses and their relations, and for system evaluation (Ward 1993; Ward 1995). Others I didn’t solve, including questions of how to effectively integrate different knowledge sources and how best to do the evidential reasoning. This didn’t bother me; on the contrary I was glad to be facing up to some difficult AI problems.

The system actually works, on live speech, although not very well, and meets the goals discussed in §2, more or less (Ward 1994a). Although more a testbed than a usable system, it compares favorably with other recent AI efforts (Hayes *et al.* 1987; Kitano *et al.* 1989; Baggia & Rullent 1993; Nagao *et al.* 1993; Ward 1993; Kawahara *et al.* 1994; Hauenstein & Weber 1994; Cochard & Oppizzi 1995), as discussed elsewhere (Ward 1994a).

Thus the system was a success, judged by my original goals. But as those goals were misguided, it failed the larger goal of serving as a prototype for future high-quality speech understanding systems.

4 Lessons Learned

Most recent work on speech understanding is largely unrelated to AI. Work in the two fields shares little, technically speaking, and there is little exchange between the two communities. This situation seemed inexplicable.

Now I realize that this parting of ways between speech and AI has at least one good reason: AI approaches have attempted to solve the wrong problem. As enumerated in §2, each of my basic goals, which I shared with the classic AI work (Reddy *et al.* 1973; Woods & Makhoul 1973; Klatt 1977; Erman *et al.*

1980; Woods 1980), has turned out to be of questionable importance or downright wrongheaded. For twenty years the classic AI approach to speech has been in limbo, mostly abandoned but never properly laid to rest. I hope that this paper will serve as a proper post-mortem, closing the chapter on 1970s AI approaches to speech, and saving other researchers from the pitfalls they lead to.

5 Further Implications

Why did I, and other AI researchers, misunderstand the nature of the speech understanding problem? Where did the inappropriate goals come from? In my case at least, the initial choice of goals was not entirely conscious — most of them I pursued implicitly as I followed my instincts as an AI researcher. Looking back, I think these boiled down to the fact that I valued:

- ambitious goals (1, 6)
- analogies to introspected human abilities (1, 2, 3, 4)
- systems that can do clever and impressive things (2, 4)
- systems that are easy to understand and empathize with (6, 7)
- computational elegance (2, 8)
- results of wide significance (1, 5, 8, 9)

(The numbers in parentheses refer to the goals discussed in §2.)

These values, I believe, lie at the heart of the classic AI paradigm. Yet for speech they can misguide. Perhaps the real lesson of my experience is that the classic AI paradigm is intrinsically unsuitable for the speech understanding problem.

Thanks to Dan Jurafsky and Jane Edwards for comments.

References

Alshawi, Hiyan & David Carter (1994). Training and Scaling Preference Functions for Disambiguation. *Computational Linguistics*, 20:635–448.

Baggia, Paolo & Claudio Rullent (1993). Partial Parsing as a Robust Parsing Strategy. In *1993 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. II–123–126.

Brooks, Rodney A. (1991). Intelligence Without Reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 569–595.

Cochard, Jean-Luc & Olivier Oppizzi (1995). Reliability in a Multi-agent Spoken Language Recognition System. In *4th European Conference on Speech Communication and Technology (Eurospeech'95)*, pp. 75–78.

Cole, Ronald A. & Jola Jakimik (1980). A Model of Speech Perception. In R. A. Cole, editor, *Perception and Production of Fluent Speech*, pp. 133–163. Lawrence Erlbaum Associates.

Dennett, Daniel C. (1991). *Consciousness Explained*. Penguin.

Erman, Lee D., Frederick Hayes-Roth, Victor R. Lesser, & D. Raj Reddy (1980). The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys*, 12:213–253.

Goldberg, Adele E. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

Hauenstein, A. & H. Weber (1994). An Investigation of Tightly Coupled Time Synchronous Speech Language Interfaces Using a Unification Grammar. In *AAAI Workshop on the Integration of Natural Language and Speech Processing*, pp. 42–49.

Hayes, Phillip J., Alexander G. Hauptmann, Jaime G. Carbonell, & Masaru Tomita (1987). Parsing Spoken Language: a Semantic Caseframe Approach. Technical Report CMU-CMT-87-103, Carnegie Mellon University, Center for Machine Translation. expanded version of a paper in COLING86.

Hirschman, Lynette (1994). The Roles of Language Processing in a Spoken Language Interface. In David B. Roe & Jay G. Wilpon, editors, *Voice Communication between Humans and Machines*, pp. 217–237. National Academy Press.

Jurafsky, Daniel *et al.* (1995). Using a Stochastic Context-free Grammar as a Language Model for Speech Recognition. In *1995 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 180–192.

Kawahara, Tatsuya, Masahiro Araki, & Shuji Doshita (1994). Heuristic Search Integrating Syntactic, Semantic and Dialog-level Constraints. In *1994 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. II–25–28.

Kitano, Hiroaki, Hideto Tomaebuchi, Teruko Mitamura, & Hitoshi Iida (1989). A Massively Parallel Model of Speech-to-Speech Dialog Translation. In *European Conference on Speech Communication and Technology (Eurospeech'89)*.

Klatt, Dennis H. (1977). Review of the ARPA Speech Understanding Project. *Journal of the Acoustical Society of America*, 62:1324–1366. reprinted in *Readings in Speech Recognition*, Alex Waibel and Kai-Fu Lee, eds., Morgan Kaufmann, 1990.

- Lee, Chin-Hui (1994). Stochastic Modeling in Spoken Dialogue System Design. *Speech Communication*, 15:311–322.
- Levin, Esther & Roberto Pieraccini (1995). Concept-Based Spontaneous Speech Understanding System. In *4th European Conference on Speech Communication and Technology (Eurospeech'95)*, pp. 555–558.
- McClelland, James L. (1987). The Case for Interactionism in Language Processing. In Max Coltheart, editor, *Attention and Performance XII: The Psychology of Reading*, pp. 3–36. Lawrence Erlbaum Associates.
- McClelland, James L. & J. L. Elman (1986). Interactive Processes in Speech Perception: the TRACE model. In James L. McClelland & David E. Rumelhart, editors, *Parallel Distributed Processing, Volume 2*, pp. 58–121. MIT Press.
- Moore, Robert, Fernando Pereira, & Hy Murveit (1989). Integrating Speech and Natural Language Processing. In *Speech and Natural Language Workshop*, pp. 243–247. Morgan Kaufmann.
- Moore, Robert C. (1994). Integration of Speech with Natural Language Understanding. In David B. Roe & Jay G. Wilpon, editors, *Voice Communication Between Humans and Machines*, pp. 254–271. National Academy Press.
- Moore, Robert C., Douglas Appelt, John Dowding, J. Mark Gawron, & Douglas Moran (1995). Combining Linguistic and Statistical Knowledge Sources in Natural-Language Processing for ATIS. In *Proceedings of the Spoken Language Systems Technology Workshop*, pp. 261–264. Morgan Kaufmann.
- Myers, Brad A. *et al.* (1993). The Garnet Reference Manuals, Revised for Version 2.2. Technical Report CMU-CS-90-114-R4, Carnegie Mellon University. <ftp://a.gp.cs.cmu.edu/usr/garnet/garnet>.
- Nagao, Katashi, Koiti Hasida, & Takashi Miyata (1993). Understanding Spoken Natural Language with Omni-Directional Information Flow. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp. 1268–1274.
- Nguyen, Long, Richard Schwartz, Ying Zhao, & George Zavalagkos (1994). Is N-Best Dead? In *Proceedings of the Human Language Technology Workshop*, pp. 411–414. Morgan Kaufmann.
- Norris, Dennis (1993). Bottom-Up Connectionist Models of ‘Interaction’. In Gerry Altman & Richard Shillcock, editors, *Cognitive Models of Speech Processing*, pp. 211–234. Lawrence Erlbaum Associates.
- Pallett, David S. & Jonathan G. Fiscus (1995). 1994 Benchmark Tests for the ARPA Spoken Language Program. In *Proceedings of the Spoken Language Systems Technology Workshop*, pp. 5–36. Morgan Kaufmann.
- Reddy, D. Raj, Lee D. Erman, & Richard B. Neely (1973). A Model and a System for Machine Recognition of Speech. *IEEE Transactions on Audio and Electroacoustics*, 21:229–238. reprinted in *Automatic Speech and Speaker Recognition*, N. Rex Dixon and Thomas B. Martin (eds), IEEE Press, 1979, pages 272–281.
- Seneff, Stephanie (1995). Integrating Natural Language into the Word Graph Search for Simultaneous Speech Recognition and Understanding. In *4th European Conference on Speech Communication and Technology (Eurospeech'95)*, pp. 1781–1784.
- Ward, Nigel (1992). A Parallel Approach to Syntax for Generation. *Artificial Intelligence*, 57:183–225.
- Ward, Nigel (1993). On the Role of Syntax in Speech Understanding. In *Proceedings of the International Workshop on Speech Processing*, pp. 7–12.
- 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*, pp. 50–57. <ftp://ftp.sanpo.t.u-tokyo.ac.jp/pub/nigel/papers/integration94.ps.Z>.
- Ward, Nigel (1994b). *A Connectionist Language Generator*. Ablex.
- Ward, Nigel (1995). The Spoken Language Understanding Mini-challenge. <ftp://ftp.sanpo.t.u-tokyo.ac.jp/pub/nigel/lotec2-slum>. (corpus and evaluation software).
- Woods, W. A. (1980). Control of Syntax and Semantics in Continuous Speech Understanding. In *Spoken Language Generation and Understanding*, pp. 337–364. D. Reidel.
- Woods, W. A. & J. Makhoul (1973). Mechanical Inference Problems in Continuous Speech Understanding. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pp. 200–207. revised version appears in *Artificial Intelligence*, 5(1), 1974, pp 73–91.