NATURAL LANGUAGE UNDERSTANDING AND SPEECH RECOGNITION

George M. White

Natural language understanding must be an integral part of any automatic speech recognition system that attempts to deal with interactive problem solving. The methods for representing and integrating knowledge from different sources may be valuable for the understanding process as well as speech recognition.

Natural language understanding (NLU) refers to computer understanding of *human* language, which includes spoken as well as typed communication. Most of the techniques developed for NLU over the last 25 years are largely concerned with syntax analysis of grammatically correct *typed* sentences. These techniques may not be useful in dealing with the nongrammatical nature of normal speech. This article focuses on speech understanding and the marriage of NLU techniques with speech recognition techniques needed to achieve *speech understanding.*

Automatic speech recognition (ASR) as a field of research has proceeded on a parallel but separate track from NLU for more than 20 years. For most of this time, ASR research has been preoccupied with translating acoustic information into computer commands or text without involving formal NLU techniques. (We use the term "recognition" as distinct from "understanding" when *no* linguistic or semantic analysis is involved.) It might appear reasonable for speech *understanding* to be achieved by acoustic phonetic analysis followed by NLU analysis. However, the subtleties of normal speech render it impractical to create a simple feed-forward interface between an acoustic analysis stage and current NLU techniques that expect reliable word-sized units as input.

For normal spontaneous speech, acoustic-phonetic analysis techniques by themselves will never be able to produce an unambiguous stream of text equivalent to the typed input expected for NLU systems. Thus, at the very least, an NLU stage would have to be modified to handle errorful input. In addition, a feedback mechanism should be provided between the NLU and acoustic phonetic stages. Otherwise, the acoustic-phonetic stage must increase the data processed by an order of magnitude in order to produce a sufficient set of alternatives to be passed on the NLU stage. In other words, *both* NLU and acoustic analysis stages must be modified to achieve a computationally efficient system.

Historically, NLU research has focused on grammatically correct typed queries of databases and automatic language translation of textbook documents, (e.g. technical journals, legal briefs, financial statements). The most successful commercial examples of NLU systems are computerized information retrieval systems from typed queries (e.g. Symantec's Q&A [9]). There are only a few language translation systems and they work with restricted sources of text [16]. There are no commercially successful NLU systems built specifically to work with speech input. (There are dozens of firms offering ASR, most are for personal computers, and none contain NLU technology to assist in the recognition of utterances.)

It is widely believed that NLU techniques must be used in conjunction with acoustic analysis techniques to achieve recognition of continuous speech. This is because continuous speech is normally filled with acoustic ambiguity which can only be resolved through the use of higher sources of knowledge, principally semantic and pragmatic sources. The Defense Advanced Projects Research Agency (DARDA), which provides the major funding for nonproprietary ASR research as well as natural language understanding, is sponsoring interdisciplinary projects explicitly to combine NLU and ASR disciplines. The VOYAGER system, developed at the Laboratory for Computer Science at MIT, is an example of this type of interdisciplinary project. VOYAGER is a speaker-independent, continuous speech-understanding system that employs a natural language component to understand and answer queries concerning the locations of public buildings in the Harvard/MIT area. The system is able to handle about 66 percent of the spontaneous questions asked of it in spoken English within its task domain.

Even though DARPA first began sponsoring speech-understanding research (SUR) in 1970, DARPA contractors are just beginning to successfully merge NLU and ASR research disciplines. Earlier SUR work produced systems that had semantic and prosodic components that theoretically were integrated with acoustic phonetic analysis components. However, these integrated systems were not successfully demnstrated because of inadequate computing power. Most ASR research has not focused on NLU but rather on the recognition of isolated utternces, or continuous utterances from highly constrained vocabularies (such as the digits); or speaker-dependent input instead of speaker-independent input; or carefully articulated continuous speech in narrow task domains with artificial language syntax constraints.

Meanwhile, the need for integrating NLU and ASR techniques has continually increased as the opportunity for spoken language communication with personal computers has increased, fueled by the increasing computational power of personal computers (and "workstations") for interactive problem solving.

Natural language understanding plays two roles in translating speech queries or statements into useful computer commands. The first role is imputing the correct meaning of the speech so the computer gets the right message...is of little concern here. The second, more subtle role (and main focus of this article) is reducing acoustic phonetic ambiguity in normal speech based on "understanding" of meanings.

Without NLU techniques, ASR systems using a purely acoustic-phonetic analysis perform well only on slowly pronounced isolated words. This is not the way we usually talk to one another, and pronouncing words in isolation is slow and tedious compared to continuous speech. However, it is useful and is the basis of a 30,000 word recognition system, DragonDictate [2] from Dragon Systems. This is the largest vocabulary system commercially available in the world. The automatic recognition of continuous speech for thousands of words is much more difficult because continuous speech is filled with acoustic-phonetic shorthand.

The extent of acoustic-phonetic shorthand can be understood from research performed by Dennis Klatt [11]. After recording normal continuous speech, Klatt would splice out individual words and play them back to listeners in random order. Typically, listeners could only understand about 70 percent of the words. Yet listeners got 100 percent of the words right when the words were played in the correct order. The conclusion of this experiment is that it is the grammatical rules, prosodics, and semantic content of a message that allow humans to overcome acoustic ambiguity in at least 30 percent of the words in normal conversation.

ASR system builders have long been aware that semantic and syntactic knowledge must be integrated with acoustic phonetic sources to deal with most forms of continuous speech. Integrating diverse knowledge sources has been at the heart of DARPA-sponsored ASR research for nearly 20 years. An early solution was put forth in the blackboard model of HEARSAY [7] developed at CMU (circa 1974) which simply states that all knowledge sources (KSs) should work in parallel. In addition, KSs should exchange results via a common memory scheme or *blackboard.* This solution, while general enough to cover all situations of interest, does not actually guide the integration of KSs. The HEARSAY model was eclipsed by a system based on Hidden Markov Models (HMMs) that implicitly addressed the issue of combining scores from different KSs.

The technique used with HMM systems for integrating KSs was to represent knowledge as finite state networks with transition probabilities between states, with integration being achieved by allowing different KSs to influence the transition probabilities. The HMM technique was *not* conceived as a technique for integrating KSs, but rather as a technique for applying sound stochastic modeling techniques to the problem of decoding strings of elemental sound symbols. However, HMM techniques did succeed at integrating information from at least three KSs. The KSs were

1. phone temporal duration,
2. phone level elemental sound similarity, measurements and,
3. word pair statistics.

The most notable and recent of these HMM systems was Sphinx [12] (circa 1988)—the world's first speaker-independent, continuous speech recognition system operating on a large vocabulary (1,000 words or more). Today, the HMM approach is the most widely used pattern-matching technique in ASR systems worldwide.

Despite the widespread use of the HMM approach, it has not had much impact on NLU or integration of NLU techniques into ASR systems. In fact the potential of HMM methodology to guide integration of diverse KSs is not at all obvious. It was first mentioned by Jim Baker in his doctoral thesis [3] which was also the first published explanation of HMM applied to ASR. Unfortunately, the Baker thesis only suggested that the HMM approach might apply to NLU.

In fact, it appears that it is not the hidden Markov process itself that is relevant; rather it is the state machine representation, and the methodology of processing the time evolution of the state machine that are relevant.

The Nature of Understanding

Computerized *understanding* of any subject raises philosophical issues. We operationally avoid these issues by defining understanding as the ability to respond appropriately to directives or queries in normal human language based solely on information we already possess.

NLU by machines is the decoding of messages encoded with the symbols and conventions that humans use among themselves. Extensive use of *context* is required since human communication is so extensively conditioned by the world models of the sender and receivers of messages. If the communication is written, the symbols are restricted to text and a few punctuation and highlighting symbols. Speech, on the other hand, contains more symbol carriers, namely pitch, volume and segment duration which together are said to supply prosodic information. To understand the conventions for imparting meaning to sequences of utterances, we need to understand the roles of syntax, pragmatics, and semantics. These are cornerstone issues in human communication.

Communication is the exchange of messages that describe the state of a model or a change in state of a model. These models are typically mental models but they can be computer models equally well. The messages are serial encodings that serve to *compress information and distribute it over ordered sets of symbols.* Serial encoding, and the possibilities for contextual information thus created are powerful information compression methodologies. They are fundamental not only to efficient communication, but probably to intelligence itself.

Contextual information is the local environment provided by nearby symbols that literally redefine a symbol. The fact that any given symbol can take different meanings depending on context means that communication can proceed with fewer symbols than would otherwise be needed.

To appreciate the extraordinary power of contextual information to build meaning into simple sequences, consider two extreme forms, providing the most powerful compression known: information encoding in chromosomes and fractals.

Chromosomes in the germ cells in

no way specify directly how many cells of any sort should appear in an adult being. Rather, they specify how each cell should react to differing contexts provided by other cells through chemical and electrical interchanges as the body matures. The collective behavior is striking as new cell characteristics are created according to the ***sequences*** of cell development. The full message encoded in a chromosome cannot be read directly, but must be allowed to create all the intermediate contexts in order to give expression to the final form.

Fractals [8] encode information in much the same way. As pointed out in *Scientific American* [13], fractals provide the most powerful techniques for picture data compression invented to date. They work by creating irregular geometric shapes that reproduce themselves on smaller and smaller scales, using the "context" of their previous larger structures to guide the smaller ones.

These are examples of extreme data compression using recursive contextual information encoding, in which context begets new context, which begets new context, etc. While not directly applicable to natural language understanding, such examples serve to focus attention on the data compression aspect of contextual information processing and the extraordinary power of context to reveal information to decoders with sufficient memory to utilize it.

If spreading information over sequences makes an efficient coding scheme, we could argue on principles of least effort that it would be used widely in human communication. And we see empirically that context is everywhere in human discourse.

Returning to the core issues of syntax, pragmatics, and semantics, it would appear that they address different levels of contextual information encoding provided by strings of words.

From among these three KSs, the shortest strings and most local context is the realm of language syntax. Syntax rules are typically concerned only with phrases and sentences rather than information spread over paragraphs or books.

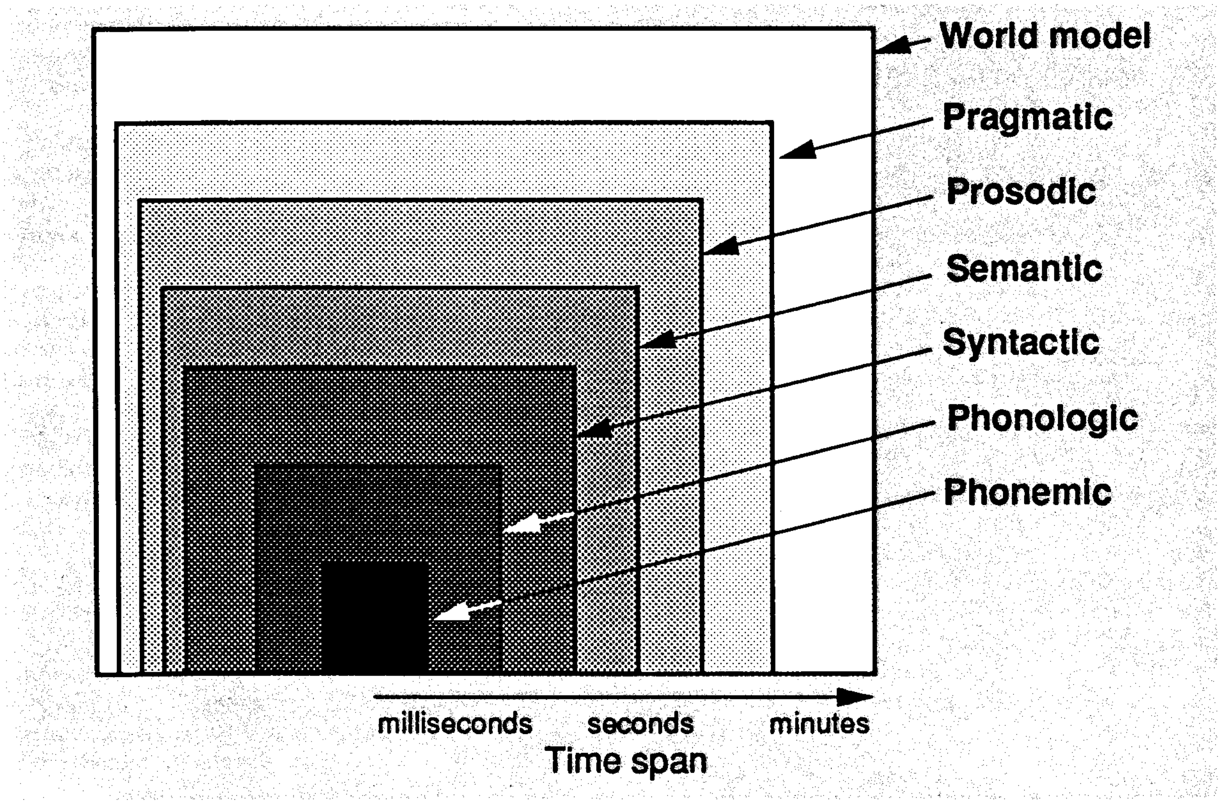
Pragmatic and semantic models deal with larger contexts than encoded in isolated sentences. They generally cover encodings that span several sentences.

Pragmatic models reflect the intermediate states of mind that might be put in place by the progress of discourse or exposition.

Semantic models impart meaning to messages. Semantic models are the objects that messages describe the state of or the changes in. They are also called world models and are the ultimate repository of intelligence in a system. This is the true frontier of NLU and artificial intelligence in general, which will properly be the focus of research for years to come.

The underlying notion behind NLU semantic models is that meaning can be derived from models of the domain of discourse rather than

FIGURE 1 Knowledge Sources (KSs) shown in hierarchy according to the span of discourse covered



exhaustive enumeration of word sequences.

This reduces the problem of recognizing word strings, from a possibly infinite number of variations, to a problem of paraphrasing or transforming a finite number of manipulations that can be performed on finite world models.

In other words, there may be an infinite number of ways of saying the same thing; and the way to handle this situation is to encode the basic information once and use procedures to generate the different ways of saying it. Furthermore, in many domains of discourse, the basic underlying information may be small and represented well in a semantic model with a relatively small number of facts and properties. In this case, the problem of modeling human discourse is largely an issue of finding the right semantic models and combining them with the right linguistic transformation rules.

Roger Schank's [13] suggestion that the world of normal everyday discourse might be reduced to a few hundred generic situational models with 30 or so underlying actions that might be performed on the models, is another example. Sentences that could be parsed were interpreted as a paraphrase of these basic actions and the paraphrase was to be guided by conceptual dependency networks.

Noam Chomsky's [6] famous transformational grammars which transform a finite set of basic sentence structures, (deep structures), into an infinite number of ways of actually composing sentences (surface structures) provide another example. Semantics are encoded at the deep structure level. Transformational grammars play the role of conceptual dependency networks by translating underlying information embodied in the deep structures into the expressive forms actually found in normal human communication.

In summary, it is not the goal of NLU to prerecord all reasonable statements that might be made about any nontrivial subject and attach intended meanings. The goal is to create semantic models from which meanings may be generated and then to develop pragmatic and syntactic encoder/decoders (and prosodic encoder/decoders for speech) to spread the meaning over a serial stream of words.

The Nature of Speech

Speech is remarkable for the variety of rules it follows and even more remarkable for the rules it violates. Written sentences may be expected to obey most grammatical rules most of the time. However, spontaneous speech has at least some grammatical probabilistic errors most of the time; [4] it contains false starts, dangling phrases, mixed cases, mixed tenses, etc. I have a foreign friend who simply leaves out verbs and nouns at random: "Children making kites. All day running in the park. Hot. Tired. Sweaty. We going home now." Makes perfect sense, doesn't it? But it certainly is not grammatically correct.

In his scholarly book, "On Human Communication" Colin Cherry [5] points out that we can strip off all grammatical clues and still communicate (p122): "woman, street, crowd, traffic, noise, thief, bag, loss, scream, police... ."

While spontaneous speech may not be as "nongrammatical" as this, it still violates many of the rules on which typical NLU systems are based. On the other hand, most users of computerized NLU systems, whose goal is presumably to help the dumb machine understand, could be expected to obey most of the rules most of the time. Since most talkers obey many more grammatical rules than they disobey, syntactic information is certainly present. When the speakers fail to be perfect, however, it must not stop the NLU processes, just slow them down.

NLU systems need to extract meaning from partially completed parses (i.e. when the parse has gone as far as it can and it is not yet finished); they must shed light on the potential completions and what each would mean if completed. At the very least, this will require a change from deterministic to a probabilistic parsing strategy. In fact, probabilistic parsing is essential and one of the more important messages of this article.

NLU systems use many grammars for parsing (e.g. transformational, ATN, chart, bottom up, top down, semantic, conceptual dependency) and most may be gracefully applied to imperfections of real speech, . ..*but only if parsing is modified to be probabilistic.* In other words, decisions to follow particular branches in a parse tree must be treated as tentative rather than final (i.e. the decision to pursue any particular branch must be handled as a probabilistic decision); and provisions must be incorporated to retrace steps to follow alternative branches if the probability of the processed branch grows too low. Pioneering work in this area has been done by Seneff [15]. Such search strategies are well developed in the world of information-coding theory and automatic speech recognition. An appropriate approach for NLU is probably *stack decoding* [10].

While it might be universally agreed that probabilistic parsing is a

SPEECH IS REMARKABLE FOR THE VARIETY OF RULES IT FOLLOWSAND EVEN MORE REMARKABLE FOR THE RULES IT VIOLATES, great goal and that ASR systems would be more accurate if they incorporated syntactic, prosodic, pragmatic, and semantic knowledge sources, there is little agreement on how to encode this knowledge and how to apply it. However, it is clear that a uniform method for representing and communicating between syntactic, prosodic, pragmatic, and semantic knowledge sources would help to promote information sharing between these different levels.

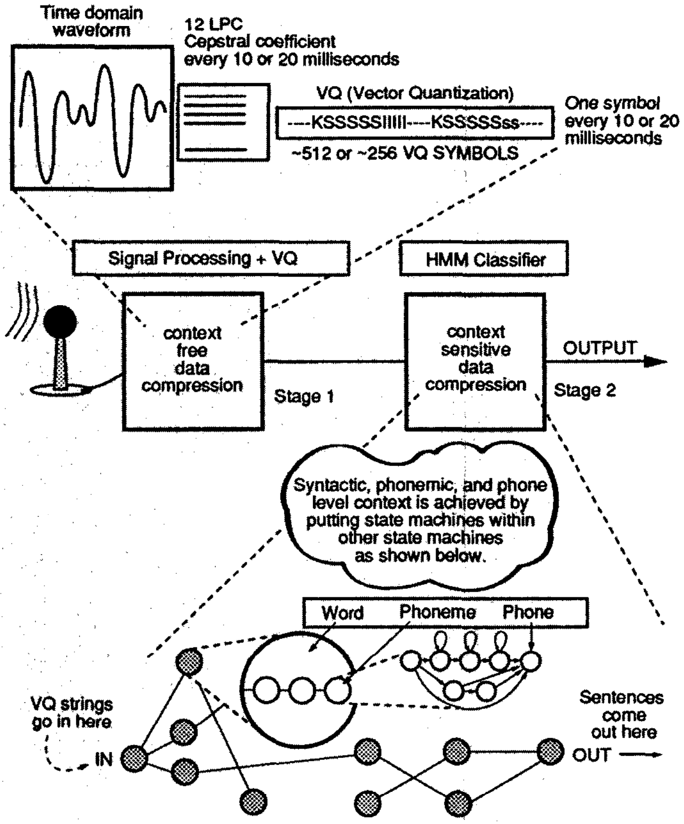
A proposal was put forth by Jim Baker [3] to provide a uniform representation in the form of "finite state machines." The idea was to represent all knowledge sources (semantic, prosodic, pragmatic, syntactic, phonemic and phonologic) as networks (or finite state machine equivalents) and then to determine what states are most probable using the mathematics of the HMM approach. The approach has been applied with great success to the lower sources of knowledge, namely phonologic, phonemic, and word order syntax based on simple word pair statistics.

Figures 2 thru 7 show popular finite state machine models associated with the HMM approach. Figure 2 shows how finite state machines might be used in an operational ASR system. This description is applied to the current leading ASR systems worldwide and is similar to the one described in Kai Fu Lee's paper on Sphinx [12].

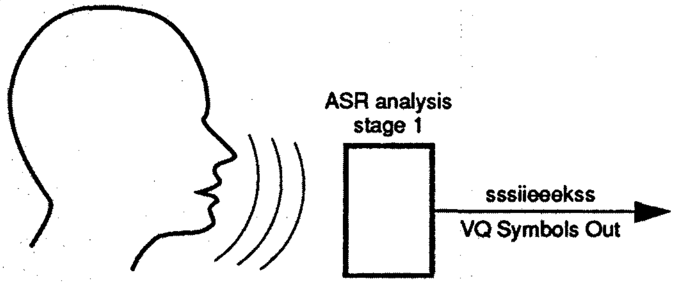
It is important to note that figures 2 through 7 do not directly address the use of NLU techniques. They do not address semantic, prosodic, or pragmatic knowledge sources. Instead, they only show one way to integrate the *lower* sources of knowledge (phone to phoneme, phoneme to word, and word to word syntax). This integration has been implemented by embedding the lower sources explicitly within higher ones (see Figure 2).

Unfortunately Baker's thesis did not actually combine *higher* sources of knowledge (semantic, pragmatic and prosodic) nor did it explain how to do it beyond suggesting that finite state

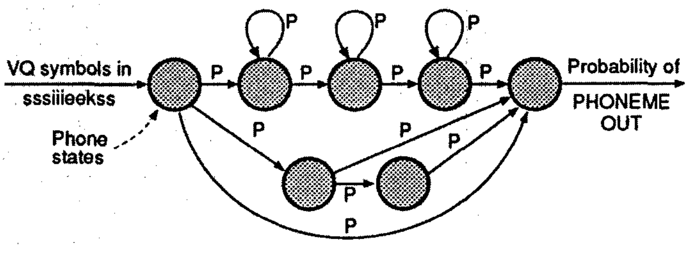
2


Time domain waveform
12 LPC Cepstral coefficient every 10 or 20 milliseconds
VQ (Vector Quantization)
----KSSSSSIIIII----KSSSSSss----
~512 or ~256 VQ SYMBOLS
One symbol every 10 or 20 milliseconds
Signal Processing + VQ
HMM Classifier
context free data compression
Stage 1
context sensitive data compression
OUTPUT
Stage 2
Syntactic, phonemic, and phone level context is achieved by putting state machines within other state machines as shown below.
Word
Phoneme
Phone
VQ strings go in here
IN
Sentences come out here
OUT

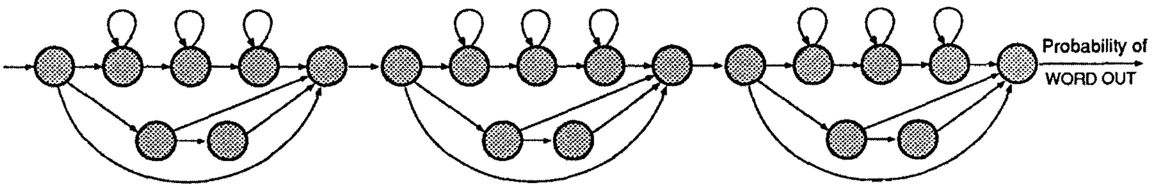
3



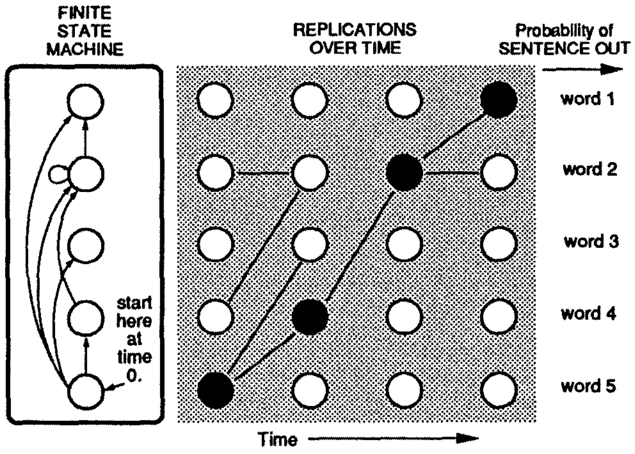
4



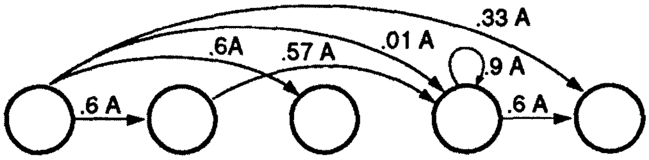
5



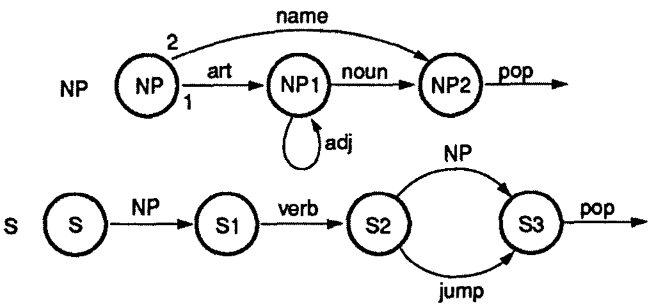
6



7



8



9

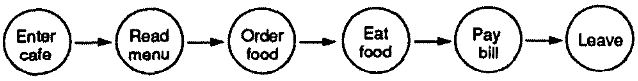


FIGURE 2 A typical ASR System combining KSs from phone, phonemic, and word levels showing production of VQ symbols followed by HMM state machine network

FIGURE 3 Initial stage of a typical ASR system, such as Sphinx, showing the reduction of speech input into a series of phone labels called VQ (vector quantization) symbols

FIGURE 4 Phones to Phonemes. Phones (VQ symbols) are processed by state machines for "phonemes." These phoneme models are described by [12]. The phoneme provides the context for the phone states within.

FIGURE 5 Phonemes to words. Three phonemes are shown concatenated to form a word. The transition probabilities inside the phonemes can be trained using the local context, namely the identity of the word. Context can alternately be provided by the two adjacent phonemes. This context was dubbed "triphones" by Lee [12].

FIGURE 6 Words to sentences. Sentences are created by replicating the states in a finite state machine over time. Each state represents a word. Links between those words with high probabilities are shown by lines in the word matrix. Words that form the sentence are shown in black.

FIGURE 7 Generic finite state machine showing compound probabilities associated with transitons, e.g. ".9A" = .9*A. When the nodes represent the words, the first probability is the language model. The second probability, "A," can be determined by acoustic measurements alone. However, it can also contain information from language syntax, prosodics, pragmatics, etc. as shown in Figure 11.

FIGURE 8 Syntactic knowledge encoded as ATN, (Augmented Transition Network) [16].

FIGURE 9 Roger Schank's restaurant script as example of state machine encoding of task domain knowledge. The key idea here is that the act of eating at a restaurant includes well-defined stages. The stages are small in number. Each stage provides scripts of probable dialogues.

10

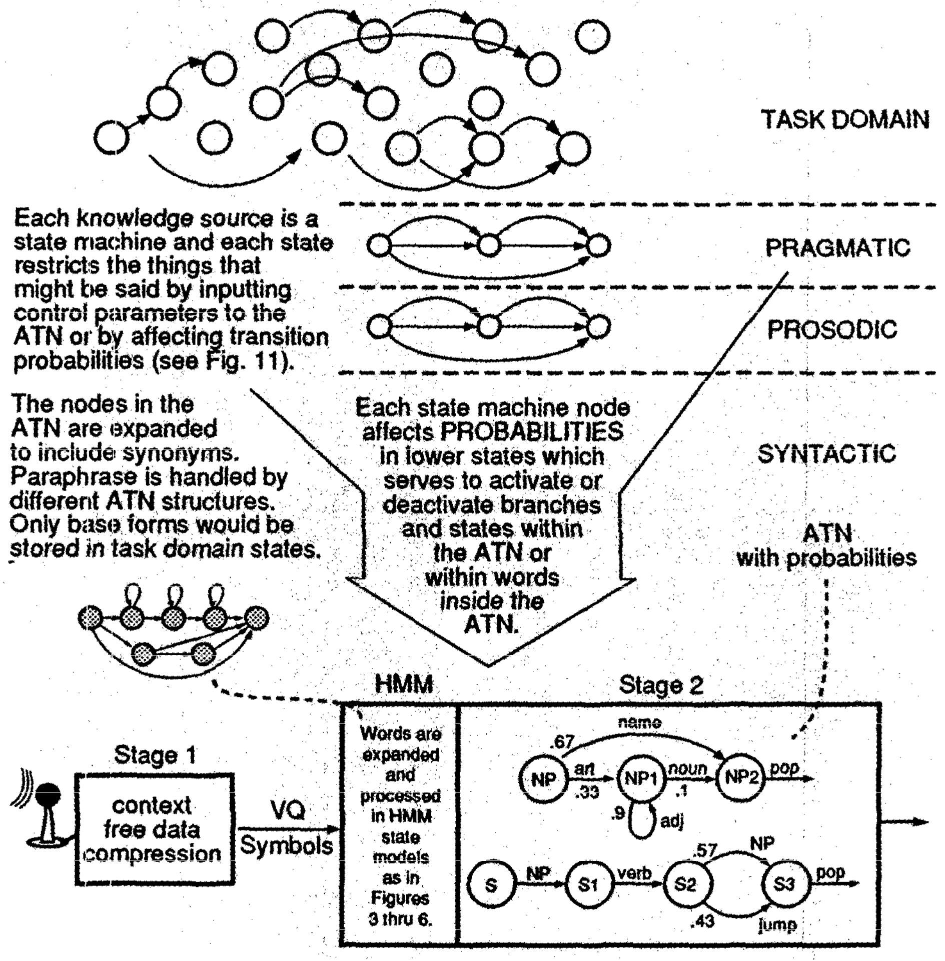

TASK DOMAIN
Each knowledge source is a state machine and each state restricts the things that might be said by inputting control parameters to the ATN or by affecting transition probabilities (see Fig. 11).
PRAGMATIC
PROSODIC
The nodes in the ATN are expanded to include synonyms. Paraphrase is handled by different ATN structures. Only base forms would be stored in task domain states.
Each state machine node affects PROBABILITIES in lower states which serves to activate or deactivate branches and states within the ATN or within words inside the ATN.
SYNTACTIC
ATN with probabilities
HMM
Stage 2
Stage 1
context free data compression
VQ
Symbols
Words are expanded and processed in HMM state models as in Figures 3 thru 6.
name
.67
NP
art
.33
NP1
noun
.1
NP2
pop
.9
adj
.57
NP
S
NP
S1
verb
S2
S3
pop
.43
jump

FIGURE 10 State Machine Networks for task domain, pragmatic, prosodic, and syntactic knowledge sources. Base forms for phrases and sentences originate in the task domain and are modified—expanded or encoded—by the pragmatic and prosodic domains before taking effect ATN. The ATN expands into a graph or network that can be searched to yield a parse of speech input. Higher knowledge sources in the form of state machines control the operation of another state machine, an ATN (or other grammar) by influencing transition probabilities.

FIGURE 11 Integration of indifferent KSs achieved by using conditional probabilities to link transitions in any given context to larger context provided by more global KSs

machines and the HMM formalism should be helpful. It is only now, 15 years later, that we have enough experience with state machines to appreciate why they might be useful for integrating higher sources of knowledge.

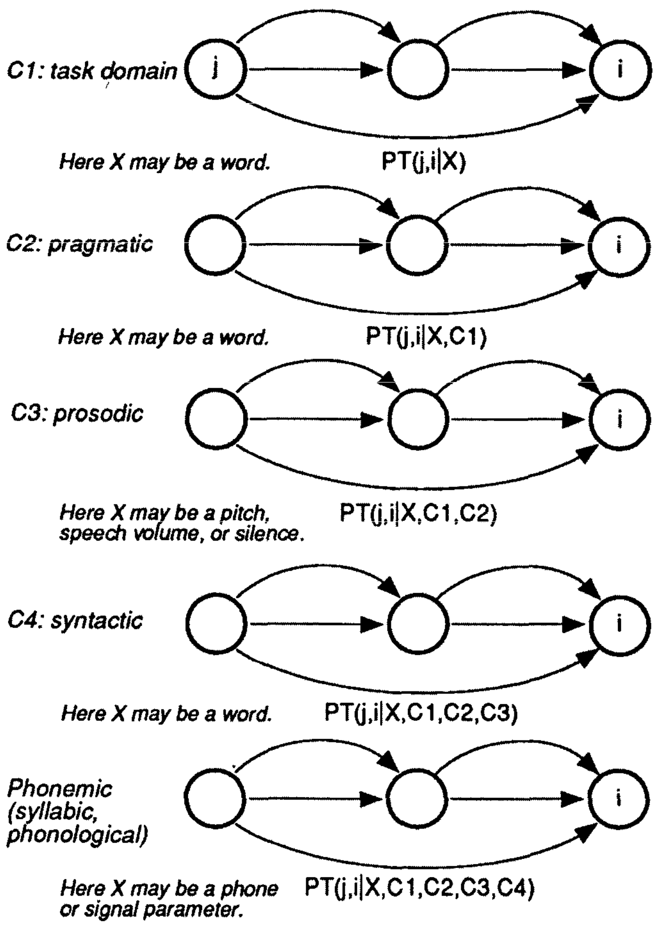
The main purpose of this article is to explore the elements of current HMM ASR methodology and show how they might be extended to encompass the integration of NLU with ASR. These techniques have not been put forth to solve this problem before but they are nonetheless well known techniques.

The main thesis of this article is, then, that NLU knowledge sources can and should be integrated with ASR sources by: first, developing probabilistic state machine networks for semantic, pragmatic and prosodic types of knowledge; and then developing a scheme for interconnecting different knowledge domains.

Once this is done, the time evolution of the probabilities of states can be treated in the same way that the lower model states are treated. We can see from Figure 2, that one way is to simply embed some finite state models within the states of other finite state models. We seek a more general answer here.

The state machines shown in Figure 11 are symbolic structures used simply to illustrate how conditional probabilities increase as contextual scope changes. The bottom state machine with generic transition prob-

11



ability PT (j,i|X,C1,C2,C3,C4) says that the probability of transition between elemental sound units is conditioned by (dependent on) the states occupied at the syntactic (C4), prosodic (C3), pragmatic (C2), and semantic (C1) levels. In practice, the different domains may be treated as independent. For example: PT(j,i|X,C1,C2,C3,C4) = PT(i,j|X, C4)*P(C4|C3)*P(C3|C2)*P(C2|C1) where P(Ci|Cj) is the probability of being in state, Ci, in the ith level given that the state Cj is occupied in the jth level.

Operationally, the conditional probabilities may be interpreted to mean that lower level models should be trained on data labeled by higher level states, and the same set of transition probabilities so gathered should be used during the recognition phase when the specified higher level states are active.

Recall Figure 1 and the hierarchy of contexts provided by task domain, pragmatic, prosodic, syntactic, phonemic, and phonological domains. The lower their position in the hierarchy, the more conditional are the probablities. More precisely, the longer the time span, the more global the context; and domains of lesser extent should be conditioned by the context of more global ones.

Figures 8, 9, and 10 show higher level knowledge sources represented as state machines. Figure 11 summarizes the method for integrating the NLU and ASR knowledge sources discussed in this article.

I believe the correct way to combine NLU and ASR is to specify task domain states first and then proceed to specify the lower states. In other words, first determine the different system states that might occur for each application. Next, determine the questions the user might ask and the commands the user might give for each state. This information would be encoded in grammars that are customized for particular situations. Grammars are preferred to simple lists of statements, even with paraphrase and morphological rules to achieve data compression, because grammars take up less memory.

The challenge is to assign probabilities to the various statements, and to pass these probabilities along to the lower domains of knowledge. The single most fruitful area is likely to be probability assignments to task domain states and to the sentences that would issue from these states. Initially, the key statistic may be PT4(i,j|X,C1)*_P(C4|C1)_ (in place of the current probabilities that are used by ASR systems that would correspond to PT 4(i,j|X)). Later, it will be possible to add prosodic and pragmatic knowledge sources when they are better understood.

Conclusion

I have noted that NLU as a research discipline has not given great consideration to the needs of speech recognition per se. However, it is widely believed that NLU techniques will someday significantly improve performance of ASR systems. An obstacle to fruitful collaboration between ASR and NLU research areas has been the lack of a good theory combining information from different knowledge sources, a fundamental question in artificial intelligence research. I have a tentative answer to this question.

The solution I propose requires that knowledge be represented as networks of states. Communication as the time evolution of a network of states is envisioned. The time evolution is treated as a first order Markov process in a state machine, (i.e., the

SUMMARY

To Integrate Different Knowledge Sources:

1. Encode each knowledge source as a finite state network of relationships.

2. Treat the act of communicating as movement in the network driven by the encoding and decoding of messages.

3. Model the time evolution of the network as a message decoded as a stochastic process in a state machine, (i.e. assign transition probabilities to all legal transitions between states and update all probabilities after each time increment). Let the probability of being be a state i at time t, P[i,t], be the sum of all path probabilities that come to state i where each path probability is the product of a precursor state probability, P[j,t-1], and its transition probability, PT[j,i|X]. Then P[i,t] = P[j,t-1]*PT[j,i|X] where X is the observation variable. It is the input that drives communication. It can be a phone, a word, or an acoustic measurement.

4. Cause one KS to affect another by treating the transition probabilities as conditional upon the context in which they occur. For instance PT[j,i|X,C1,C2,...] is the probability of going from state j to i given the context specified by X, C1, and C2...

states are linked together with transition probabilities with new state probabilities being computed from old state probabilities multiplied times the transition probabilities ...with summation over all products that connect old states to new ones). A significant element in the proposal is that the transition probabilities be conditioned both by the states (context) of more global state machines and also by the new acoustic information driving the communication.

This is not a simple extension of the HMM approach. There is no need to treat a hidden process in the model proposed here. On the other hand, most of the background work that has a direct bearing on this approach comes from the literature on HMM. It is expected that readers will want to refer to the HMM literature to see specific examples of the use of probability to combine knowledge sources.

It should be noted that knowledge sources at different levels may be processed asynchronously, in parallel, instead of serially. This has important consequences for improving the computational underpinnings for machine intelligence in general since parallel processing avoids the Von Neumann bottleneck by promoting the use of multiple hardware processors to increase the power of computing hardware.

I believe the approach described in this article will be immediately applicable to natural language understanding for well defined domains of discourse with limited vocabularies, and a manageable set of semantic alternatives that are characteristic of the task domains within text editors, page layout programs, spreadsheets, telecommunications systems, and databases. These are the areas of practical interest to personal computer manufacturers and significant progress can be expected in these areas. The time has come in the evolution of our society for us to start talking to our computers. The NLU community can play a key role in ushering in this new modality, possibly by using the techniques suggested.

Acknowledgments.

I would like to thank Victor Zue, Gary Hendrix, and Bill Luciw for reviewing the paper and contributing constructive criticism. ■

References

1. Allen, J. *Natural Language Understanding.* The Benjamin Cummings Publishing Company, Inc., 1988.
2. Jim Baker. Dragon Dictate. *Speech Tech. J.*, Media Dimensions, New York, N.Y., (Spring 1989), 20-25.
3. Baker, J., Stochastic modeling for speech recognition. Doctoral thesis, Dept. of Computer Science Carnegie Mellon University, Pittsburgh, PA 1976.
4. Bodmer, F. *The Loom of Language.* W.W. Norton & Company, New York, 1944.
5. Cherry C. *On Human Communication.* MIT Press, Cambridge, Mass. April 1975.
6. Chomsky, N., *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Mass., 1965.
7. Erman, L.D., Lesser, V.R. *Trends in Speech Recognition.* W. Lea, Ed. Prentice Hall, Englewood Cliffs, NJ, 1980, pp. 361-381.
8. Feder, J. *Fractals.* Plenum Press, New York and London, 1988.
9. Harvey, Hendrix, G. *Mastering Q&A.* Sybex, 1988.
10. Jelinek, F. Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech, *Automatic Speech and Speaker Recognition*, Dixon and Martin, IEEE Press, 1979, 231-237.
11. Klatt, D. personal communication.
12. Lee, K.F., Reddy, R., Hsiao-Wuen Hon. An overview of the SPHINX speech recognition system. *IEEE Trans. Acous. Speech, and Sig. Process. 38,* 1, (January 1990), 34-45.
13. Schank, Roger
14. Not just a pretty face; compressing pictures with fractals. *Sci Am.* (March, 1990), 77-78.
15. Seneff, S. TINA: A probabilistic syntactic parser for speech understanding systems. Tech. Rep. LCS Laboratory for Computer Science, MIT, Cambridge, MA, 1989.
16. Slocum, J. *Mach. Transl. Syst.* Cambridge Union Press, 1988.
17. Zue, V. et. al. The voyager speech understanding system: Preliminary development and evaluation. In *Proceedings of ICASSP* 90.

About the Author:

GEORGE M. WHITE began research in Automatic Speech Recognition (ASR) as the Stanford Artificial Intelligence Project in 1969. Today, he is manager of speech recognition research at Apple Computer. Author's Present Address: Apple Computer, 20525 Mariani Ave., Cupertino, CA 95014.