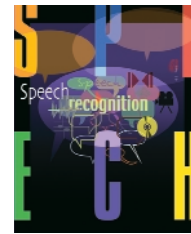


Automated Natural Spoken Dialog



The next generation of voice-based interface technology will enable easy-to-use automation of new and existing communication services, making human-machine interaction more natural.

Allen L. Gorin
Alicia Abella
Tirso Alonso
Giuseppe Riccardi
Jeremy H. Wright
AT&T Labs—Research

Engineers have long sought to design systems that understand and act upon spoken language. Extracting meaning from natural, unconstrained speech over the telephone is technically challenging, and quantifying semantic content is crucial for engineering and evaluating such systems.

Traditional menu-driven speech recognition systems force users to learn the machine's jargon, but many people are unwilling or unable to navigate such highly structured interactions. AT&T's "How May I Help You?" (HMIHY) technology (<http://www.research.att.com/~algor/hmihy>) shifts the burden to the machine by requiring it to adapt to human language and understand what people actually say rather than what a system designer expects them to say.

The intuition underlying our approach is that for a given task, some linguistic events are more *salient*—crucial to recognize and understand—than others. Researchers have already developed methods to automatically extract named entities such as phone and credit card numbers embedded in natural spoken language¹ and to translate utterances into Spanish and Japanese.² Building on these and other early laboratory experiments,³ we have developed algorithms^{4,5} that automatically learn the salient words, phrases, and grammar fragments for a given task far more reliably than other methods.

ACHIEVING NATURAL FUNCTIONALITY

Traditional telephony automation systems offer a list of menu options and prompt users to navigate

to the appropriate destination to obtain service or resolve a problem. Sometimes the machine itself provides the desired service, such as supplying an account balance or billing a call to a credit card, while in other instances only a specially trained person can do so.

Menu systems can be implemented using a touch-tone system ("Press 1 if you want x , press 2 if you want y , ..."), voice labels ("Please say collect, calling card, ..."), or a hybrid of the two ("Press or say 1 if you want x , ..."). Each can be useful when the list of options is short and well understood by customers, but for certain tasks designers must resort to unwieldy hierarchical menus that can bore and frustrate users. On the other hand, callers faced with succinct menu options may have difficulty deciding which of the proffered categories matches what they want. In both cases, users often either bail out of the system by pressing zero or do nothing in the hope of eventually being connected to a person.

Determining the number of options and amount of detail to include in menus can be difficult. In contrast, a human receptionist could simply handle this same routing task by asking, "How may I help you?" and let the caller describe the request or problem. The receptionist should know enough about the task to transfer the caller to the appropriate person or automated module.

Our goal in developing HMIHY was to engineer a call-routing system with this same natural functionality. A caller receives a greeting and makes a request as if talking to a person. The system's job is to recognize and understand what the user wants—

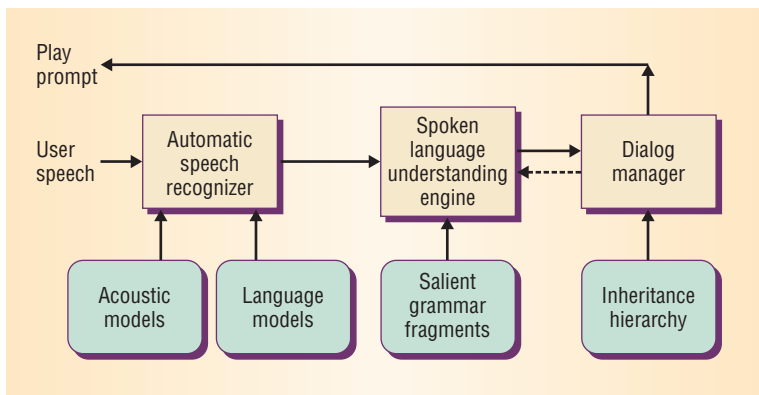


Figure 1. HMIHY architecture. A user receives the open-ended prompt, “How may I help you?” and then makes a request as if talking to a person.

not in an ontological sense, but just sufficiently to properly direct the call. Figure 1 illustrates the system’s basic architecture, which includes an ASR (automatic speech recognition) component that relies on acoustic and language models to extract words from user speech, an SLU (spoken language understanding) engine that uses salient grammar fragments to extract meaning and classify customer requests, and a dialog manager that acts upon SLU output as well as task knowledge in an inheritance hierarchy to complete the call-routing task.

To illustrate HMIHY, we focus on two tasks involving live customer traffic in a telephone network. *Operator services* involves users placing telephone calls and includes specifying billing methods—for example, collect or calling card—as well as requesting information such as rate and area codes for the person making those calls. In the *customer care* application, users ask questions about items on their bill, calling plans, account balances, and so on.

LANGUAGE MODELING

The state-of-the-art approach to recognizing unconstrained spoken language involves training a stochastic language model that predicts word sequence probability. For example, given a sentence $S = v_1 v_2 \dots v_n$, the goal is to estimate the probability of the word v_i given the history of all preceding words: $P(v_i | v_1 v_2 \dots v_{i-1})$. However, data sparseness makes estimating these probabilities for all possible histories intractable. The most familiar method is the n -gram model, which estimates the probability of a word based on only the preceding $n - 1$ words—where typically $n = 3$, denoted a trigram language model. As n increases, data sparseness and the ASR’s memory and computation requirements also increase.

Variable length units

A better alternative is to selectively introduce longer-range history in the form of *variable length units*, which provide the accuracy of high-order n -gram models but have computation and memory requirements similar to low-order models. For ASR language modeling, we select more frequently

occurring sequences based on entropy minimization, which leads to expressions such as “I want to make a,” “collect call,” and “card call” in the operator services task. Using a bigram language model with variable length units would thus lead to terms such as $P(\text{collect_call} | \text{I_want_to_make_a})$, effectively resulting in a seven-gram model. In previous work,⁶ we demonstrated how to embed such acquired phrases into a stochastic ASR language model.

Salient phrases

After recognizing a user’s spoken words, the next step is to understand what the caller said. In our early experiments, we used methods based on a “bag of words” model³ that analyzed text without regard to temporal order—the words could, metaphorically, be put into a bag and shaken up without changing the analysis. We subsequently discovered that exploiting an utterance’s temporal word order has definite advantages that can improve performance—language does indeed have a structure that we can use to derive more reliable understanding.

Further, because what users say changes with respect to context, the ASR language model must vary over the dialog states.⁷ The “Measuring Language Complexity” sidebar illustrates why language recognition and understanding are more difficult in customer care than in operator services.

Our first step was to develop algorithms that automatically acquire salient phrases for a task, in which salience is a mathematical measure of the information content of a linguistic event. For example, “wrong” (associated with requesting a billing credit) is salient in the operator services task, “wrong number” is even more salient, and “dialed a wrong number” is more salient still.⁸ Phrases are preferable to words because they have sharper semantics and because longer events are more reliably recognized in speech.

Salient grammar fragments

The observation that many salient phrases are similar, such as “dialed a wrong number” and “dialed the wrong number,” led to the next step: developing clustering algorithms that exploit a combination of string-edit distances and semantic distortions.⁴ We compactly represent salient-phrase clusters as finite-state machines and denote these as salient grammar fragments, as Figure 2 shows. Phrase clusters are parsimonious, which enables pooling of statistics across multiple low-frequency phrases, and they are robust to ASR errors within

Measuring Language Complexity

Customer care is intuitively a more complex task than operator services. We can quantify this intuition in terms of utterance length, vocabulary, perplexity, and semantic complexity of the classification task.

Utterances are significantly longer in the customer care task. For example, the average number of words in responses to “How may I help you?” is 19 in operator services and 39 in customer care, as the histogram in Figure A shows. Observe that the two distributions have a similar shape—skewed unimodal with a long tail. Our experiments have shown that longer utterances are harder to recognize and understand than shorter ones.

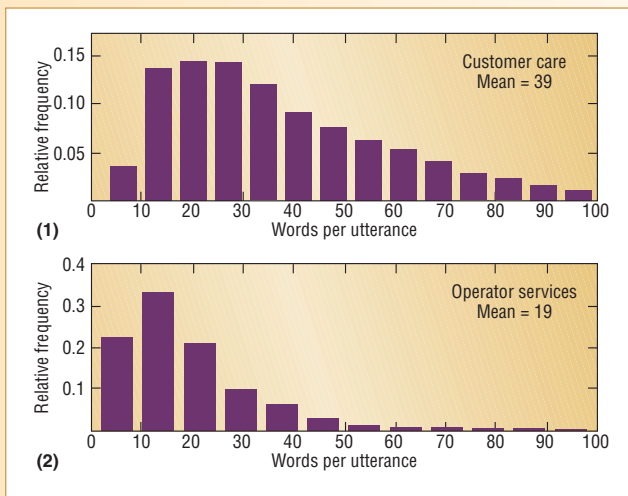


Figure A. Histogram of utterance lengths. On average, callers' responses to “How may I help you?” in the customer care task (1) are more than twice as long as those in the operator services task (2).

grammar fragments. We exploited syntactic distortions to include hierarchical clustering as well.⁵

Embedding salient phrases in the ASR language model improves classification performance while negligibly affecting word accuracy. To classify an utterance, we matched these grammar fragments against the ASR output and applied a decision rule to combine the lattice of detections and their associations.

The following example is the transcription of a customer utterance and ASR output with detected salient fragments boldfaced and errors italicized.

Transcription: Okay I got AT&T Wireless phones and when I got them he told me that I would be switched to 7 cents a minute for all my AT&T long distance calling because I was on 10 10 cents One Rate plan ...

ASR: *yeah I'm not* **AT&T Wireless_Phone** and when I got *and she* told me that I would be switched to **7_Cents_A_Minutes_For_All** my AT&T long distance *on that* I was on 10 10 cents **One_Rate_Plan** ...

For a random sample of 8,000 utterances, the operator services and customer care tasks contain 3,600 and 5,200 distinct vocabulary words, respectively. An out-of-vocabulary event occurs when a word not previously observed in a model's training set appears in its test set. It is well known that the higher the OOV rate, the more difficult the automatic speech recognition (ASR) task. Our measurements indicate that the OOV rate in both tasks is approximately one new word every third utterance. Fewer than half of these OOV words are proper nouns, indicating the high variation in customers' language.

Perplexity is a traditional measure of linguistic complexity that can be loosely interpreted as the average instantaneous vocabulary of a language corpus. Higher perplexity is thus predictive of a harder ASR task. These values are 16 for the operator services and 39 for customer care tasks, again illustrating the latter's greater complexity.

Distribution entropy is a standard measure of classification problem difficulty—for example, a uniform distribution over 2ⁿ classes is an n-bit problem. A skewed distribution has lower entropy—is an easier decision task—than a uniform distribution. We computed the entropy of the skewed distribution over 23 call types in customer care as 3.3 bits per semantic label.

A perplexity value of 39 is equivalent to an entropy value of ~5 bits per word, or a total of ~200 bits in a 40-word utterance that we are attempting to decode via ASR. However, we are only seeking to reliably decode 3.3 bits per utterance. Although this is not a rigorous argument, it suggests why call classification is possible, with high accuracy, while ASR is far more difficult. For example, word accuracy for these tasks is ~65 percent; accuracy on salient phrases is much higher at ~85 percent, and classification accuracy well exceeds 90 percent.

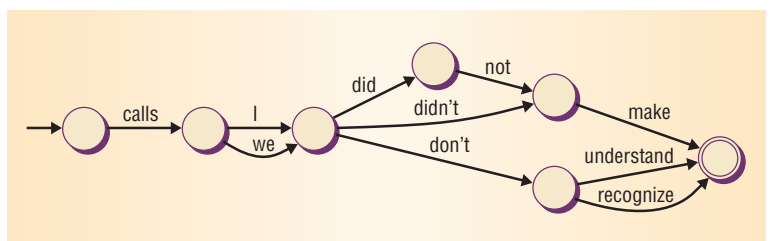


Figure 2. Salient grammar fragment. This fragment from the customer care task is strongly associated with unrecognized number billing queries.

HMIHY detects one fragment associated with the Cellular call type and two others with Calling Plans. Based on strength of associations and coverage, the SLU output is Calling_Plans. The ASR errors are not salient and thus do not impact the SLU.

CALL-TYPE CLASSIFICATION

The SLU engine's primary function in both tasks is to determine which service type a customer is requesting.⁸ We drew on a vast amount of research literature on techniques to classify topics in speech as well as text categorization methods designed to retrieve information and documents.

Dialog Motivators

An interaction with “How May I Help You?” often takes more than one turn—dialog is necessary to confirm low-confidence interpretations, clarify ambiguities, and request missing information to complete a transaction. A *dialog motivator* is one of a small set of general principles that govern a dialog.

The following dialog example contains two invocations of the *missing information* motivator to obtain a caller’s billing method and card number:

Machine: AT&T. How may I help you?
User: I need a credit, I got the wrong number.
Machine: What was the number that you dialed?
User: I dialed four one five two eight ...
Machine: Was the call billed to the phone that you’re calling from now?
User: No, to a credit card.
Machine: What was the card number?
User: Four five six seven ...
Machine: I’ve given you credit for that call. Thank you for using AT&T.

Another example illustrates the *clarification* motivator:

Machine: AT&T. How may I help you?
User: I have a question about my June bill.
Machine: Okay, what is your question?
User: There’s a call to Tallahassee, but I don’t know anyone there.
Machine: I can help you with that. Are you calling from your home phone?
User: I’m calling from my house.
Machine: Please hold on while I transfer your call.

Dialog motivators exist for numerous other functions, including confirmation, error handling, and assumptions.

Call-type classification has several distinguishing attributes:

- It involves speech rather than text, with the inherent difficulties of speech recognition and the disfluencies of conversational-style language.
- Input is from cooperative users who are trying to communicate their needs and make themselves understood.
- The system has the opportunity to ask confirming or clarifying questions of that cooperative user.
- Collateral customer profile information is often available, which can be exploited in understanding a request—for example, cus-

tomers who want to know how to make delinquent payments on their phone bill are routed differently than other callers who want to pay their bills.

From a database we created of what customers say to operators, we observed that, although the variation in vocabulary and language is large, callers usually asked for one of 15 types of services.⁸ For example, “I want to reverse the charges on this call” is a request for a `Collect` call; “Can you tell me what time it is in Tokyo?” asks for `Time` information; and “I was trying to call my sister and dialed a wrong number” is a request for a `Billing Credit`. Automation for each of these call types is straightforward.

Queries that don’t fit into any of the predefined categories, such as “I’ve been trying to dial this number all day and can’t get through,” are classified as `Other` and directed to a human agent for resolution.

The customer care task involves very different types of questions.⁹ For example, “How much money do I owe you?” is a query for an `Account Balance`, while “I don’t recognize this phone call to Tallahassee on October 4” involves an `Unrecognized Number`.

Again, automated responses exist for both of these scenarios. However, “What’s this charge for one dollar and fifty cents?” asking about a `Charge on Bill`, and “I have a question about my bill,” a `Billing Query`, are vague. In these cases, clarification is required to properly classify the call.

The multiclass call classification task provides three traditional measures⁸ for rejection:

- the probability of false rejection—how often the system falsely rejects a request for some service or classifies it as `Other`;
- the probability of correct classification—how often a call-type classification is correct; and
- the true rejection rate—the probability that the system correctly classifies a request as `Other` and thus routes it to a human.

These three measures characterize how well the system classifies and routes customers’ calls.

DIALOG MANAGEMENT

Dialog provides the opportunity to ask confirming and clarifying questions, thus improving call-type classification compared with a single utterance. HMIHY uses a *dialog manager* to confirm the machine’s understanding when its confidence is low, to clarify ambiguities in a customer’s

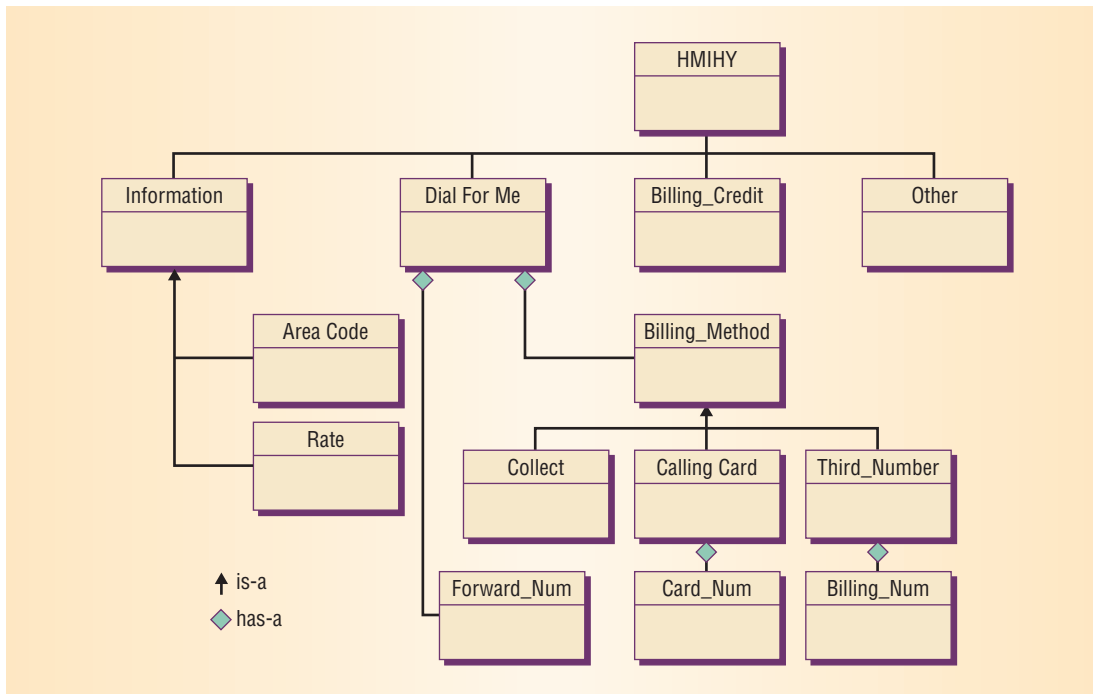


Figure 3. Inheritance hierarchy of task knowledge in operator services. The terminal nodes in the hierarchy provide sufficient specificity for the machine to take action, while the nonterminal nodes require clarifying queries from the dialog manager.

request, and to gather additional information necessary to complete the task.

For example, if someone wants to make a collect call from a train station, ASR confidence might be low because of the noisy background; the machine should therefore confirm its recognition and understanding by asking, “Do you want to make a collect call?” A different scenario arises when a user vaguely says, “Charge this call, please.” The machine needs to resolve the ambiguity regarding payment by querying, “How do you want to charge this call—to a credit card or to a third number?” To verify a customer’s account balance, the machine would need to ask, “What is your home phone number?”

A *call flow*—essentially a long if-then-else specification—defines human-machine interaction in traditional menu systems and strongly prompted dialogs, but this approach does not scale well for complex natural spoken dialogs. Instead, we employ a *construct algebra* framework¹⁰ consisting of a collection of reusable dialog motivators—generic rules that determine what action the dialog manager takes in its next interaction with a user and which are portable over a range of tasks. The “Dialog Motivators” sidebar illustrates two such motivators for missing information and clarification.

The set of semantic labels in call-type classification is not a simple unstructured list. In the case of operator services, *Collect* and *Credit Card* are a kind of *Billing method*, and any call has a

Forward Number (the number being called). Similarly, requests for *Rate*, *Time*, or *Area Code* are all types of request for *Information*.

In HMIHY, task knowledge is based on an object-oriented *inheritance hierarchy*¹¹ that must be encoded and provided to the dialog manager and SLU modules. This inheritance hierarchy defines the relationships among the call types and named entities.

For example, a customer’s question about an unrecognized charge *is a* kind of query about a phone bill and *has a* dollar amount, item number, dialed number, and so on. Representing *is-a* and *has-a* relationships in programming languages such as C++ or Java is fairly straightforward. The dialog manager exploits this task knowledge and the dialog motivators to govern what action to perform at each turn. Figure 3 shows how an object-oriented inheritance hierarchy encodes *is-a* and *has-a* relationships for the operator services task.

The “How May I Help You?” spoken dialog system integrates numerous technologies to offer users a better user experience as well as more accurate routing and increased automation. Although we have achieved call-routing performance exceeding 90 percent success, which is far superior to customers’ ability to self-select and navigate hierarchical menus, research is continuing on ways to improve the system. For example, training the language models for ASR and SLU requires

transcribed and annotated utterances. To resolve this bottleneck, we are developing methods to automatically acquire acoustic morphemes from untranscribed speech. Also, while human agents can easily determine whether an interaction is going well or badly and modify their behavior accordingly, automated techniques that let the dialog manager modify its strategy remain a challenge. ■

Acknowledgment

We gratefully acknowledge the encouragement and support of Larry Rabiner, who has been a friend and mentor to this research for over a decade.

References

1. M. Rahim et al., "Robust Numeric Recognition in Spoken Language Dialog," *Speech Communication*, vol. 34, no. 1, 2001, pp. 195-212.
2. S. Bangalore and G. Riccardi, "Stochastic Finite-State Models for Spoken Language Machine Translation," *Proc. ANLP/NAACL Workshop Embedded Machine Translation Systems*, Association for Computational Linguistics, Brunswick, N.J., pp. 52-59.
3. A.L. Gorin, "On Automated Language Acquisition," *J. Acoustical Society of America*, vol. 97, no. 6, 1995, pp. 3441-3461.
4. J.H. Wright, A.L. Gorin, and G. Riccardi, "Automatic Acquisition of Salient Grammar Fragments for Call-Type Classification," *Proc. 5th European Conf. Speech Communication and Technology*, International Speech Communication Association, Bonn, Germany, 1997, pp. 1419-1422.
5. K. Arai et al., "Grammar Fragment Acquisition Using Syntactic and Semantic Clustering," *Speech Communication*, vol. 27, no. 1, 1999, pp. 43-62.
6. G. Riccardi, R. Pieraccini, and E. Bocchieri, "Stochastic Automata for Language Modeling," *Computer Speech and Language*, vol. 10, no. 4, 1996, pp. 265-293.
7. G. Riccardi and A.L. Gorin, "Spoken Language Adaptation over Time and State in a Natural Spoken Dialog System," *IEEE Trans. Speech and Audio*, vol. 8, no. 1, 2000, pp. 3-10.
8. A.L. Gorin, G. Riccardi, and J.H. Wright, "How May I Help You?" *Speech Communication*, vol. 23, no. 1, 1997, pp. 113-127.
9. A.L. Gorin et al., "Semantic Information Processing of Spoken Language," *Proc. 2000 Int'l ATR Workshop Multilingual Speech Communication*, Advanced Telecommunication Research Institute, Kyoto, Japan, 2000, pp. 13-16.
10. A. Abella and A. Gorin, "Construct Algebra: Analytical Dialog Management," *Proc. 37th Ann. Meeting Assoc. Computational Linguistics*, Association for Computational Linguistics, Brunswick, N.J., 1999, pp. 191-199.
11. A. Abella and A.L. Gorin, "Generating Semantically Consistent Inputs to a Dialog Manager," *Proc. 5th European Conf. Speech Communication and Technology*, International Speech Communication Association, Bonn, Germany, 1997, pp. 1879-1882.

Allen L. Gorin is the technology leader in the Speech Research Department at AT&T Labs—Research. His research interests include spoken language understanding and language acquisition. Gorin received a PhD in mathematics from the City University of New York. He is a senior member of the IEEE and a member of the Acoustical Society of America and the Association for Computational Linguistics. Contact him at algor@research.att.com.

Alicia Abella is a principal technical staff member of the Speech Research Department at AT&T Labs—Research. Her research interests include spoken dialog systems. Abella received a PhD in computer science from Columbia University. She is a member of the ACL. Contact her at abella@research.att.com.

Tirso Alonso is a senior technical staff member of the Speech Research Department at AT&T Labs—Research. His research interests include spoken language processing and signal processing. Alonso received an MSEE in computer science from Lehigh University. He is a member of the IEEE. Contact him at tma@research.att.com.

Giuseppe Riccardi is a principal technical staff member of the Speech Research Department at AT&T Labs—Research. His research interests include language understanding, spoken dialog, and machine translation. Riccardi received a PhD in electrical engineering from the University of Padua. He is a member of the IEEE and the American Association for the Advancement of Science. Contact him at dsp3@research.att.com.

Jeremy H. Wright is a principal technical staff member of the Speech Research Department at AT&T Labs—Research. His research interests include spoken language understanding and dialog. Wright received a PhD in engineering from the University of Southampton. He is a member of the IEEE and a Fellow of the Institute for Mathematics and Its Applications. Contact him at jwright@research.att.com.