ELSEVIER

Speech Communication 23 (1997) 113–127

SPEECH COMMUNICATION

How may I help you?

A.L. Gorin *, G. Riccardi [1], J.H. Wright [2]

AT&T Labs-Research, Florham Park, NJ, USA

Received 14 February 1997; revised 23 May 1997

Abstract

We are interested in providing automated services via natural spoken dialog systems. By *natural*, we mean that the machine understands and acts upon what people actually say, in contrast to what one would like them to say. There are many issues that arise when such systems are targeted for *large populations of non-expert users*. In this paper, we focus on the task of automatically routing telephone calls based on a user's fluently spoken response to the open-ended prompt of ''*How may I help you?*''. We first describe a database generated from 10,000 spoken transactions between customers and human agents. We then describe methods for *automatically acquiring language models for both recognition and understanding* from such data. Experimental results evaluating call-classification from speech are reported for that database. These methods have been embedded within a spoken dialog system, with subsequent processing for information retrieval and formfilling. © 1997 Elsevier Science B.V.

Résumé

Nous sommes intéressés par la production de services automatisés par des systèmes de dialogue utilisant la parole naturelle. Nous entendons par *naturel* que la machine comprend et agit selon ce que les personnes effectivement disent, en opposition à ce que l'on aimerait qu'ils disent. Plusieurs problèmes apparaissent quand de tels systèmes sont visés pour une population large d'utilisateurs qui ne sont pas des experts. Dans ce papier, nous focalisons sur la tâche de routage automatique des appels téléphoniques se basant sur la réponse spontanée des utilisateurs à la question ouverte ''*How may I help you?*''. Nous décrivons d'abord la base de données générées par 1000 transactions orales entre des utilisateurs et des agents humains. Nous décrivons ensuite les méthodes pour l'acquisition automatique, à partir des données, des modèles de langage pour la reconnaissance et la compréhension. Les résultats expérimentaux pour l'évaluation de la classification des appels sont rapportés pour cette base de données. Ces méthodes ont été incorporées dans un système de dialogue oral avec des traitements subséquents pour le tri des informations et le remplissage des formes. © 1997 Elsevier Science B.V.

Keywords: Spoken language understanding; Spoken dialog system; Speech recognition; Stochastic language modeling; Salient phrase aquisition; Topic classification

1. Introduction

There are a wide variety of interactive voice systems in the world, some residing in laboratories, many actually deployed. Most of these systems,

* Corresponding author. E-mail: algor@research.att.com.

[1] E-mail: dsp3@research.att.com.

[2] E-mail: jwright@research.att.com. On leave of absence from the University of Bristol, UK.

0167-6393/97/$17.00 © 1997 Elsevier Science B.V. All rights reserved.
PII S0167-6393(97)00040-X

however, either explicitly prompt the user at each stage of the dialog, or assume that the person has already learned the permissible vocabulary and grammar at each point. While such an assumption is conceivable for frequent expert users, it is dubious at best for a general population on even moderate complexity tasks. In this work, we describe progress towards an experimental system which shifts the burden from human to machine, making it the device's responsibility to respond appropriately to what people actually say.

The problem of automatically understanding fluent speech is difficult, at best. There is, however, the promise of solution within constrained task domains. In particular, we focus on a system whose initial goal is to understand its input sufficiently to route the caller to an appropriate destination in a telecommunications environment. Such a call router need not solve the user's problem, but only transfer the call to someone or something which can. For example, if the input is ''*Can I reverse the charges on this call?*'', then the caller should be connected to an existing automated subsystem which completes collect calls. Another example might be ''*How do I dial direct to Tokyo?*'', whence the call should be connected to a human agent who can provide dialing instructions. Such a call router should be contrasted with traditional telephone switching, wherein a user must know the phone number of their desired destination, or in recent years navigate a menu system to self-select the desired service. In the method described here, the call is instead routed based on the *meaning* of the user's speech.

This paper proceeds as follows. In Section 2, an experimental spoken dialog system is described for call-routing plus subsequent automatic processing of information retrieval and form-filling functions. The dialog is based upon a feedback control model, where at each stage the user can provide both information plus feedback as to the appropriateness of the machine's response (Gorin, 1995a). In Section 3, a database is described of 10 K fluently spoken transactions between customers and human agents for this task. In particular, we describe the language variability in the first customer utterance, responding to the prompt of ''*How may I help you*?'' in a telecommunications environment.

In Section 4, we describe the spoken language understanding (SLU) algorithms which we exploit for call classification. A central notion in this work is that it is not necessary to recognize and understand every nuance of the speech, but only those fragments which are salient for the task (Gorin, 1995a). This leads to a methodology where understanding is based upon recognition of such salient fragments and combinations thereof.

There are three main components in our SLU methodology. First is to automatically acquire salient grammar fragments from the data, modeling those parts of the language which are meaningful for the task plus their statistical associations to the machine actions. Second is to recognize these fragments in fluent speech, searching the output of a large vocabulary speech recognizer. The statistical language model which constrains this recognizer embeds automatically-acquired fragments in a stochastic finite state machine, providing an efficient approximation to an *n*-gram model with variable length units (Riccardi et al., 1996). Third, we exploit these multiple recognized fragments to classify the call-type of an utterance. Since the SLU is embedded within a dialog system, the classifier provides both the best (rank 1) and secondary (rank 2, etc.) decisions. Finally, in Section 5, we report on experimental results for call-classification from the above-mentioned speech database, training on 8 K utterances and testing on 1 K.

2. A spoken dialog system

The goal of a call-router is to recognize and understand the user's speech sufficiently to determine the call-type. Dialog is necessary since, in many situations, the call type cannot be determined from a single input. This can be due to an ambiguous request or to imperfect performance of the spoken language understanding (SLU) algorithms.

One important component of dialog is *confirmation*, wherein the machine proposes its understanding of the user's input, receiving reinforcement feedback as to the appropriateness of its response. In previous work, we have described a dialog model for integrating the meaning of initial and secondary user inputs based on a feedback control system (Gorin, 1995a,b). In case the machine's interpretation is in error, the subsequent user input can provide both disconfirmation plus *clarifying* information. Finally, once the call-type has been successfully negotiated and agreed upon, there is often a subsequent form-filling dialog to *complete* the information necessary to provide a service.

To illustrate these issues, we now present several transcripts of spoken human/machine dialogs with this system. The first dialog demonstrates the basic call-routing function where the user explicitly confirms the machine's interpretation, (where M denotes machine and U denotes a human user).

M: How may I help you?
U: *Can you tell me how much it is to Tokyo?*
M: You want to know the cost of a call?
U: *Yes, that's right.*
M: Please hold on for rate information.

The second dialog illustrates an information retrieval stage after call-type classification. Observe that in addition to the call-type, the SLU algorithm must also extract the place-name *Chicago*. Further observe the use of an implicit confirmation request, wherein the user's silence is interpreted as affirmation.

M: How may I help you?
U: *What's the area code for Chicago?*
M: You want an area code.
U: *(silence)*
M: The area code for Chicago is 312.

In some cases, the input is ambiguous or misunderstood, so the device must then engage in a clarifying dialog. There are several possible strategies for such clarification. For the ambiguous user input, ''*Can I charge this call please?*'', the SLU algorithm (cf. Section 4) generates a most likely classification of ''*third number billing*'' with secondary decisions of ''*card billing*'' or ''*collect*''. The dialog illustrated below asks for clarification via proposing the top choice. Alternatively, the machine could provide a short menu, e.g., ''*Do you want A or B?*'', or in this case where there is no clear winner amongst the three choices, ask ''*How do you want to bill the call?*''.

M: How may I help you?
U: *Can I charge this call please?*
M: You want to charge this to another number?
U: *No, to my Universal card.*
M: You want to charge this call to a credit card.
U: *(silence)*
M: Please speak your card number.

In the above example, the device interprets the user's clarifying response to be a combination of disconfirmation (''*no*'') and additional call-type information (''*to my Universal card*'') as per our feedback control model (Gorin, 1995b). The following example illustrates a call-type classification followed by a form-filling subdialog.

M: How may I help you?
U: *I tried calling my Aunt and got a wrong number.*
M: You need a billing credit.
U: *(silence)*
M: Please speak the number that you dialed.
U: *908 582 2778*
M: Was the call billed to the phone that you're calling from?
U: *Yes it was.*
M: You will be given credit, thank you for calling.

There are many open issues in optimizing the user interface of such a system, in particular those of selecting the best prompts to stimulate maximally consistent and informative responses from large populations of non-expert users (Abella et al., 1996; Boyce and Gorin, 1996). In the next section, we discuss the variations in response to a particular open-ended prompt, in the situation where users are talking to a human agent.

3. Database

In order to enable experimental evaluation, we generated a database of 10 K spoken transactions between customers and human agents. First, both channels of the dialog were recorded from the agents' headset jacks onto a digital audio tape (DAT). At the end of each transaction, a control key was manually depressed (by the human agent) to generate a DTMF code, serving both as a segmentation marker and a call-type label. These recordings were then automatically segmented, filtered and downsampled to generate a stereo speech file for each transaction.

We then focused on the first customer utterance, responding to the greeting prompt of ‘‘*How may I help you?*’’. These utterances were endpointed, orthographically transcribed and then labeled as to call-type and quality of the speech and channel. We remark on the distinction between the call-action labels provided by the agents and by the labelers. The agent’s DTMF tag comprised an on-the-spot single label for the entire transaction. The labelers, however, based their decision on the *first customer utterance* only, plus were allowed to select more than one call-label per utterance. We observed that 84% of the utterances were labeled with a single call-type, 16% with two (e.g., *COLLECT* and *PERSON-TO-PERSON*), then a small remainder (0.6%) with 3 labels. It is possible for the agent-generated call-type to not match any of the labeler’s, since sometimes the first utterance is ambiguous, with things becoming clear only after some dialog. An issue for future study is the correlation between these labeling methods, plus an analysis of the reasons for their mismatches. Since the experiments of Section 5 are based on the first utterances only, those are the labels which are used for training and testing.

Several samples of (first) utterances follow, where digits are replaced with the symbol ‘‘x’’.

Examples

I need to make a long distance phone call and charge it to my home phone number
yes how much is it to call the number I just dialed
yes where is area code x x x
yes what time is it in area code x x x right now I’m trying to gauge the time difference
I just I’m trying to get a number from information

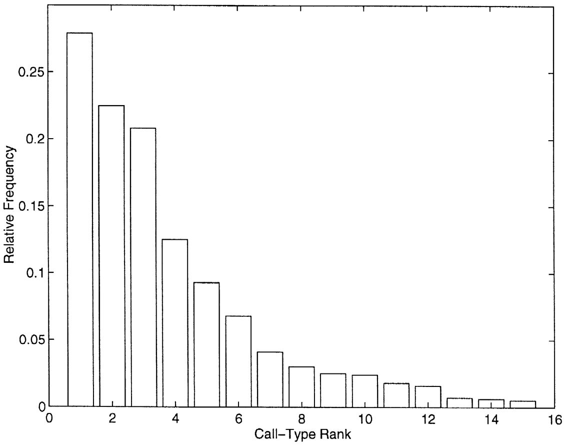

Fig. 1. Rank frequency distribution of call-types.

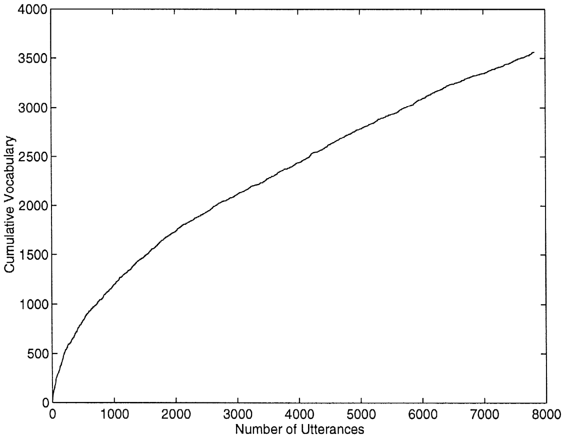

Fig. 2. Vocabulary growth in database.

Although people’s spoken language varies widely, most of the time they are asking for one of a moderate number of services. We selected a subset of 14 services plus an *OTHER* class to subsume the remainder. This distribution is highly skewed, as illustrated in the rank-frequency plot in Fig. 1.

We now discuss the vocabulary in this database. Of the 10 K utterances, 8 K are used for training language models for recognition and understanding, 1 K for testing and the remaining 1 K reserved for future development. Fig. 2 shows the increase in vocabulary size accumulated over the 8 K utterances, with a final value of ∼ 3600 words. Even after 8 K utterances, the slope of the curve is still significantly positive. We examined the tail of the lexicon, i.e., the last 100 vocabulary words accumulated in the training utterances. Approximately half were proper nouns (either people or places), but the other half were ‘‘regular words’’ (e.g., *authorized, realized, necessary*). The out-of-vocabulary rate at the token-level in the test sentences is 1.6%. At the sentence-level, this yields an OOV rate of 30% (which is also observed in the slope of vocabulary growth in Fig. 2). Thus, approximately one out of three utterances contains a word not in the training data. As will be detailed in Section 4, the test-set perplexity using a statistical trigram model is ∼ 16.

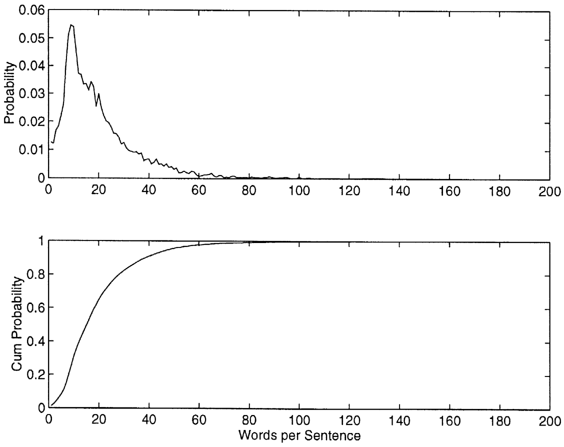

Fig. 3. Words per utterance in the initial utterances.

Utterance length varies greatly, from a minimum of one word (e.g., ‘‘*Hello?*’’) to 183, with an average of 18 words/utterance. We remark on the definition of ‘‘first customer utterance’’. The labelers were instructed that the customer’s first utterance was completed when the human agent began responding. Back-channel affirmations from the agent such as ‘‘*uh-huh*’’ were transcribed and marked, but did not indicate the end of the customer’s utterance. An issue for future research is to understand to what degree such utterances will shorten when people are talking to a machine rather than a human. The distribution of these lengths for the 10 K transcriptions is shown in Fig. 3. In that same figure, the cumulative distribution is also shown. Observe that almost all of the sentences have length less than 60. Observe also that the median is approximately equal to the mean (18), although the distribution is highly skewed. Recall that these utterances are the initial user response to the greeting prompt.

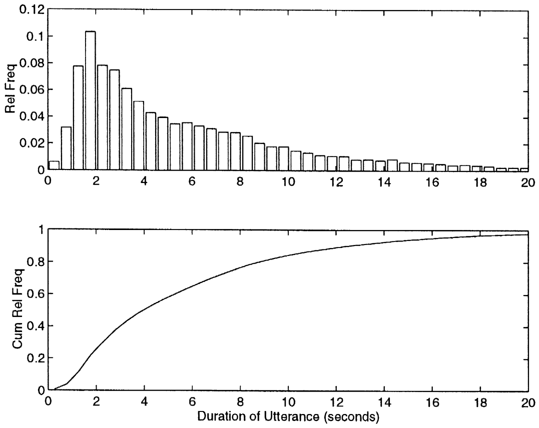

Fig. 4. Duration distribution of initial utterances.

Similarly, one can histogram the duration of these initial utterances, as shown in Fig. 4. The average duration of an utterance is 5.9 sec, so that the speaking rate in this database is approximately 3 words per second.

4. Algorithms

In this section, we describe the algorithms underlying this system and experiments. A key notion is that for any particular task, it is *not* necessary to recognize and understand every word and nuance in an utterance. That is, to extract semantic information from spoken language, it suffices to focus on the salient fragments and combinations thereof. There are three major issues that we address:

- How do we acquire the salient grammar fragments for this task?
- How can we recognize these fragments in fluent speech?
- How do we map multiple recognized fragments to a machine action?

Our technical approach is to avoid hand-crafted models throughout, focusing on machine learning methods which automatically learn the structure and parameters of statistical models for each stage from data. In Section 4.1 we describe algorithms to automatically acquire salient words, phrases and grammar fragments for a task. We recognize these in fluent speech via searching the output of a large-vocabulary recognizer (LVR), whose language model is a stochastic finite state machine (SFSM) with embedded automatically-acquired phrases. The LVR and training algorithm for the recognizer language model are described in Section 4.2. We then formulate a call-classification from those multiple recognized salient fragments as described in Section 4.3.

4.1. Salient fragment acquisition

We are interested in constructing machines which learn to understand and act upon fluently spoken input. For any particular task, certain linguistic events are critical to recognize correctly, others not so. We have quantified this notion via *salience* (Gorin, 1995a), which measures the information content of an event for a task. In previous experiments, salient words have been exploited to learn the mapping from unconstrained input to machine action for a variety of tasks (Gertner and Gorin, 1993; Gorin et al., 1994a,b; Henis et al., 1994; Miller and Gorin, 1993; Sankar and Gorin, 1993). In this work, we build upon the ideas introduced in (Gorin, 1996; Gorin et al., 1996) to automatically acquire salient phrase and grammar fragments for a task, exploiting both linguistic and extra-linguistic information in the inference process. In particular, the input to this inference algorithm is a database of transcribed utterances labeled with associated machine actions. It is these associated actions which comprise the extra-linguistic information in this task. While there is a large literature on automated training of stochastic language models, such efforts have traditionally exploited only the language itself, with the goal of within-language prediction to improve ASR (Jelinek, 1990). Learning from language alone is actually a much harder problem than people are faced with, who acquire language during the course of interacting with a complex environment. This algorithm, following that intuition, exploits both language and extra-linguistic information to infer structure.

4.1.1. Communication and salience

We briefly review the intuitions underlying salience, following Gorin (1995a). Consider devices whose purpose is to understand and act upon fluently spoken input. The goal of communication in such systems is to induce the machine to perform some action or to undergo some internal transformation. The communication is judged to be successful if the machine responds appropriately. We have explored this paradigm in some detail (Gorin, 1995a), in particular contrasting it with traditional communication theory, where the goal is to reproduce a signal at some distant point.

Following this paradigm, we have constructed several devices which acquire the capability of understanding language via building statistical associations between input stimuli and appropriate machine responses (Gertner and Gorin, 1993; Gorin et al., 1994a; Miller and Gorin, 1993; Sankar and Gorin, 1993; Henis et al., 1994). The meaning of an input stimulus (e.g., a word) can be defined (Gorin, 1995a) via its statistical associations to a device's input/output periphery. This has the attractive property of grounding meaning in a device's experiences and interactions with its environment. Viewing this set of associations as a vector enables one to define a semantic distortion between events as the distance between their association vectors. The salience of an event is then defined as its distance from a null-event.

In the case that associations are defined via mutual information between events, then this semantic distortion can be shown to be equivalent to the relative entropy between the a posteriori distributions of the output actions (conditioned upon the two input events). The salience of an event is then the unique non-negative measure of how much information that event provides about the random variable of appropriate machine responses. The reader is referred to the tutorial paper in (Gorin, 1995a) for a detailed discussion of these ideas. We remark that there are related but slightly different salience measures that have been discussed in (Garner and Hemsworth, 1997).

4.1.2. Salient phrase fragments

In previous work, we introduced the notion of a salient word, demonstrating that a rudimentary mapping from input to machine action can be constructed based on only that subset. For example, a salience analysis of the database for this task yields the results in Table 1.

We now search the space of phrase fragments, guided by two criteria. First, within the language channel, a word pair v_1v_2 is considered as a candidate unit if it has high mutual information,

$$I(v_1,v_2) = \log_2[\mathrm{P}(v_2 \mid v_1)/\mathrm{P}(v_2)]. \quad (1)$$

This measure can be composed to recursively construct longer units, by computing $I(f,v)$ where f is a word-pair or larger fragment. We remark that this is an approximation to the mutual information of the full n-tuple (Cover and Thomas, 1991). We then introduce an additional between-channel criterion, which is that a fragment should have high information content for the call-action channel. Following

Table 1
Some salient words

Word	Salience
difference	4.04
cost	3.39
rate	3.37
much	3.24
emergency	2.23
misdialed	1.43
wrong	1.37
code	1.36
dialed	1.29
area	1.28
time	1.23
person	1.23
charge	1.22
home	1.13
information	1.11
credit	1.11

(Gorin, 1995a), where f is a fragment and $\{c_k\}$ is the set of call-actions, denote its salience by

$$S(f) = \Sigma \mathrm{P}(c_k \mid f) I(f, c_k) . \tag{2}$$

This salience measure is a mutual information averaged over the call-actions. It has been shown to be the unique non-negative measure of how much information an event in one channel provides for the random variable of the second channel (Blachman, 1968). We perform a breadth-first search on the set of phrases, up to length four (an implementation artifact), pruning it by these two criteria: one defined wholly within the language channel, the other defined via the fragment's extra-linguistic associations. The within-channel associations are computed via mutual information and/or the ρ measure of Section 4.2. The extra-linguistic associations are computed via the salience of Eq. (2). The following table illustrates some salient and background phrase fragments generated by this algorithm. Three attributes of each fragment are provided. First, the mutual information between the final word in the fragment and the preceding subfragment, denoted MI. Second, the peak of the a posteriori distribution $\mathrm{P}(c_k \mid f)$, denoted $\mathrm{P}_{\mathrm{max}}$. Third, the call-type for which that peak occurs, denoted Call-Type. When the peak is between 0.5 and 0.9, then the fragment is only moderately indicative of that call-type and so is provided within parentheses. When the peak is low (< 0.5), then it is a background fragment not strongly associated with any single call-type, so none is provided.

For example, consider the fragment ‘‘*long distance*’’, which has a strong co-occurrence pattern within the language channel, thus a high mutual information ($\mathrm{MI} = 7.3$). However, it is not a very meaningful phrase in the sense that the most likely call-type (given that phrase in an utterance) is a billing credit query, but only with probability 0.55. Consider on the other hand an extension of that phrase, ‘‘*made a long distance*’’, which both has high mutual information ($\mathrm{MI} = 7.4$) and strongly connotes a billing credit query with probability 0.93. A similar discussion can be made for the fragments ‘‘*area code*’’ and ‘‘*the area code for*’’. There are several background fragments in the list, which have strong co-occurrence patterns but are not indicative of any particular call-type, such as ‘‘*I would like*’’ and ‘‘*could you tell me*’’. Such fragments are useful for creating improved models for speech recognition, as addressed in Section 4.2.

4.1.3. Salient grammar fragments

We now consider a method for combining salient phrase fragments into a grammar fragment. For example, in Table 2, consider the two salient phrases ‘‘*a wrong number*’’ and ‘‘*the wrong number*’’. Clearly, these should not be treated independently, but rather combined into a single unit. The key idea

Table 2
Salient and background phrase fragments

MI	Phrase fragments	$\mathrm{P}_{\mathrm{max}}$	Call-Type
7.4	made a long distance	0.93	Billing credit
7.3	long distance	0.55	(Billing credit)
7.1	I would like	0.24	
6.9	area code	0.65	(Area code)
6.3	could you tell me	0.37	
5.6	the area code for	0.92	Area code
5.3	I'm trying	0.33	
5.0	a wrong number	0.98	Billing credit
4.9	a long distance call	0.62	(Billing credit)
4.8	the wrong number	0.98	Billing credit
4.4	I'm trying to	0.33	
4.3	long distance call	0.62	(Billing credit)
4.3	I just made a	0.93	Billing credit
4.1	I'd like to	0.18	

is that there are two similarity measures, one in the language channel, the other extra-linguistic. Within-channel, there are various measures to compute similarity of word-strings (e.g., a Levenshtein distance). We impose the extra-linguistic constraint, however, that in order for two strings to be clustered, then their meaning must be similar.

For sake of exposition, we restrict attention to a single call-type, focusing on salient fragments for billing credit queries only, based on the transcriptions. Table 3 illustrates the growth of a salient grammar fragment for this call-type. The first pass of the algorithm determines the salient words for billing credits, for which the top choice is ‘‘*wrong*’’. The others are ‘‘*dialed*’’, ‘‘*credit*’’, ‘‘*disconnected*’’, ‘‘*misdialed*’’ and ‘‘*cut*’’.

The word ‘‘*wrong*’’ is strongly indicative of billing credit (denoted Cr), with $\mathrm{P}(\mathrm{Cr}\,|\,\mathrm{wrong}) = 0.92$. The coverage is low, however, with only 48% of those queries containing that word. The local context of this salient word is then evaluated for those elements which sharpen the semantics, i.e., increase the classification rate. The top choice for expanding local context is then ‘‘*wrong number*’’, which sharpens the a posteriori probability to 0.98. Similarly, other left and right contexts are added, leading to the grammar fragment

$$F(\text{wrong}) = (\text{a}\,|\,\text{the}\,|\,\text{was})\ \text{wrong}\ (\text{number}\,|\,\text{eos}\,|\,\text{call}),$$

where *eos* is the end-of-sentence marker, | indicates disjunction (or) and concatenation indicates conjunction in order. The grammar fragment with the kernel ‘‘*wrong*’’ is then denoted F(wrong). At this point, the semantics is quite sharp, with the a posteriori probability being 0.97, although the coverage has dropped to 0.42. This process is then repeated to construct fragments surrounding the other salient words for this call-type, denoted F(dialed), etc. As this expression becomes too long to fit in the table, we indicate the fragment from the previous row by ‘‘—’’. By incrementally adding these fragments, the coverage is increased to 0.64 while maintaining a high classification rate of 0.95.

Table 3
Growth of a salient grammar fragment for distinguishing billing credit queries

Prob correct	Coverage	Fragment
$\mathrm{P}(\mathrm{Cr}\,|\,G)$	$\mathrm{P}(G\,|\,\mathrm{Cr})$	G
0.92	0.48	wrong
0.98	0.41	wrong number
0.95	0.45	wrong (number | eos | call)
0.97	0.42	(a | the | was) wrong (number | eos | call)
0.95	0.50	F(wrong) | F(dialed)
0.95	0.57	F(wrong) | F(dialed) | F(credit)
0.95	0.59	— | F(disconnected)
0.95	0.64	— | F(misdialed) | F(cut off)

Again for the sake of exposition, let’s consider the two-class problem of distinguishing billing credit queries from the others, still restricting attention to transcriptions only. (In Section 5, we will report on a full multi-class experiment from speech.) For any particular salience threshold, a particular set of grammar fragments will be generated. A most rudimentary decision rule would be based simply whether one of these fragments matches a substring of the recognizer output. For example, the following are some illustrative correct detections of a billing credit query, based on such a matching scheme. The substring which matches a grammar fragment is highlighted by capitalization plus connection with underscores. Digit sequences are denoted by ‘‘xxx’’.

Correct detections

i placed a call and i GOT_A_WRONG_NUMBER earlier this afternoon.
yes i MISDIALED a number.
I_WAS_CUT_OFF when trying to call this number.
I_WAS_DIALING 1 xxx xxx xxxx and i got someone else
yes I_JUST_DIALED AN_INCORRECT_NUMBER
yes I would like TO_GET_CREDIT_FOR a number I called

There are two types of errors that occur in such a classifier. First is a *false detection*, i.e., classifying a call as a billing credit when it was not. Second is a *missed detection*, i.e., a billing credit query that was classified as other. The operational costs of such errors can be quite different. For example, a missed detection in a call-router leads to a missed opportunity for automation, while a false detection leads to an incorrect routing. Several examples of such errors are shown below.

False detections
yes i have a number here and i don't know if it's A_WRONG_NUMBER
I was trying to get xxx xxx xxxx and it said it WAS_DISCONNECTED
Missed detections
I am trying to call wooster and the number I have rings to a different number
I'm going to blame this one on my wife I misread her handwriting
I'm dialing xxx xxx xxxx and I keep getting bells and things like that

4.2. Recognizing fragments in speech

In this section, we describe our methodology for recognizing salient fragments in fluent speech. Traditionally, the problem of spotting words or fragments in speech has been approached via constructing models of the those fragments plus a background model to subsume their complement. When there are a small number of fragments, it was sufficient to describe the background via a low-level filler model (Wilpon et al., 1990). As the problem size increases, however, such methods do not scale well. Intuition tells us that the best background model is the rest of the language, leading one to apply large vocabulary recognition and then search the ASR output for the salient fragments. For example, experiments along these lines for keyword spotting using LVR were reported in (Peskin, 1993; McDonough and Gish, 1994).

The ASR engine in our experiments is a research version of AT&T's Watson speech recognizer (Sharp et al., 1997). We use an off-the-shelf acoustic model trained on a separate database of telephone-quality read-speech based on the methods in (Ljolje, 1994) with shared de-correlation matrices across distributions. The lexicon is based on the 8 K training set of Section 2, with a single phoneme-based dictionary pronunciation of each word (Riley et al., 1995b). The language model, pronunciation models and full-context acoustic phone models are composed on-the-fly via the methods of Riley et al. (1995a).

The recognizer is constrained by a stochastic language model which approximates an *n*-gram model on variable-length phrase units. These phrase units are automatically acquired from the database based on their utility for minimizing the entropy of the training corpus. At this point, these phrases are acquired separately and according to a different criterion than the salient fragments of the previous subsection. It is a subject for future research to integrate these two methods, in order to optimize the recognizer to maximize the understanding rate.

4.2.1. Language modeling

For language modeling to constrain the recognizer, we automatically train a stochastic finite state grammar represented via a Variable Ngram Stochastic Automaton (VNSA) (Riccardi et al., 1996). A VNSA is a non-deterministic automaton that allows for parsing any possible sequence of words drawn from a given vocabulary. Moreover, it implements a backoff mechanism to compute the probability of unseen word-tuples. The stochastic automaton is automatically generated from the training corpus according to the algorithm presented in (Riccardi et al., 1996). The order of a VNSA network is the maximum number of words that can be used as left context. That is, if the order is n and w_j denotes the jth word in an utterance, then it utilizes the conditional probabilities $\mathrm{Prob}(w_i \mid w_{i-n+1},\dots,w_{i-1})$. VNSAs have been used to approximate standard *n*-gram language models yielding similar performance to standard bigram and trigram models (Riccardi et al., 1996). Since they are represented as stochastic finite state machines, their incorporation into a one-pass Viterbi speech decoder is straightforward and efficient. Furthermore, they can be exploited in a cascade of transducer compositions for speech processing to include intra and inter-word phonotactic constraints (Pereira and Riley, 1997).

4.2.2. Automatically acquired phrases

Traditionally, *n*-gram language models for speech recognition assume *words* as the basic lexical unit. However, there are several motivations for choosing longer units for language modeling. First, not all languages have a predefined word unit (e.g., Chinese). Second, many word tuples (phrases) are strongly recurrent in the language and can be thought as a single lexical entry, e.g., ''*area code*'', ''*I would like to*'' or ''*New Jersey*''. Third, for any model of a fixed order, we can selectively enhance the conditional probabilities by using variable length units to capture long spanning dependencies. In previous work (Riccardi et al., 1996), the effectiveness of incorporating manually selected phrases in a VNSA has been shown.

In this paper, building upon (Riccardi et al., 1997), we describe an algorithm for automatically generating and selecting such variable length units based on minimization of the language perplexity $PP(T)$ on a training corpus T. We remark that while there has been other research into automatically acquiring entropy-reducing phrases (Giachin, 1995; Matsumura and Matsunaga, 1995; Masataki and Sagisaka, 1996), this work differs significantly in the language model components and optimization parameters.

The phrase acquisition method is an iterative process which converges to a local minimum of $PP(T)$, as illustrated in Fig. 5. In particular, given a fixed model order n and a training corpus T, the algorithm proceeds as follows.

4.2.3. Re-estimation algorithm for the ASR language model

```
Parameters: Let $K$ be the number of candidates generated at each iteration, and $M$ be the number of iterations.
Initialization: Let $T_{11}$ be the initial training corpus $T$, and let $\lambda_{11}$ be the language model of order $n$ trained from that corpus.
Iterate for $m = 1$ to $M$,
  Generate a ranked set of $K$ candidate phrases from symbol pairs in the lexicon of training set $T_{m1}$, denoting these via $(x\_y)_k$. The ranking is via the correlation measure $r$ described below.
  For each candidate phrase, $k = 1$ to $K$
    Filter the current training corpus $T_{m,k-1}$ by replacing each occurrence of the phrase with the phrase unit $(x\_y)_k$. Denote this new filtered set by $T_{mk}$.
    Train a new language model (still of order $n$) from $T_{mk}$, denoted $\lambda_{mk}$.
    Test whether adding this candidate phrase decreases perplexity, i.e., whether $PP(\lambda_{mk},T_{mk}) < PP(\lambda_{m,k-1},T_{m,k-1})$. If so, then continue, else reject this candidate phrase via setting $T_{mk} = T_{m,k-1}$.
  next k
next m
Train a final language model from the filtered corpus $T_{MK}$ plus the original $T$, with lexicon comprising all original words plus the acquired phrases.
```

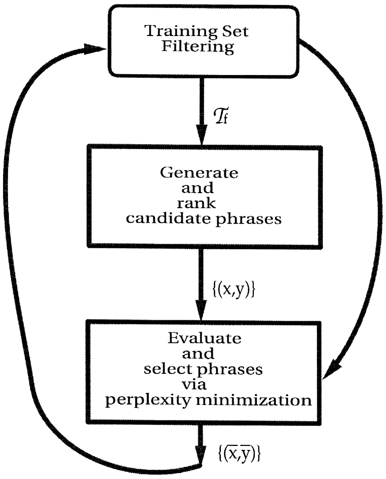

Fig. 5. Phrase selection via entropy minimization.

The algorithm is initialized with the training corpus T, with the initial language model λ_{11} corresponding to a stochastic n-gram model on words. For each iteration, the first step is to generate and rank candidate symbol-pairs (x,y) based on a *correlation coefficient*

$$\rho(x,y) = P(x,y)/[P(x)+P(y)], \tag{3}$$

where $P(x)$ denotes the probability of the event x and $P(x,y)$ denotes the probability of the symbols x and y occurring sequentially. At the first iteration, x and y are both words, in subsequent iterations they are potentially larger units. Observe that $0 \leqslant \rho(x,y) \leqslant 0.5$. We remark that this correlation measure has advantages over mutual information with respect to ease of scaling and thresholding (Riccardi et al., 1997).

Thus, a phrase x_y is selected only if $P(x,y) \sim P(x) \sim P(y)$ (i.e., $P(y \mid x) \sim 1$) and the training set perplexity is decreased by incorporating this larger lexical unit into the model. After the M iterations are completed, there is the final step of retraining the language model from the final filtered corpus T_{MK} plus the original T. This preserves the granularity of the original lexicon, generating alternate paths through the SFSM comprising both the new phrases plus their original word sequences. That is, if the words ‘‘*long*’’ and ‘‘*distance*’’ only occur together in the corpus leading to the acquisition of the phrase ‘‘*long_distance*’’, this final step preserves the possibility of the words occurring separately in some test utterance.

4.3. Call classification

We make a decision as to which of the 15 call-types to classify an utterance in a particularly straightforward manner. The speech recognizer described in Section 4.2 is applied to an utterance, producing a single best word recognition output. This ASR output is then searched for occurrences of the salient phrase fragments described in Section 4.1. In case of fragment overlap, some parsing is required. The parsing algorithm is a simple one, selecting longer fragments over shorter ones, then proceeding left to right in the utterance. This yields a transduction from the utterance s to a sequence of associated call-types. To each of these fragments f_i is associated the peak value and location of the a posteriori distribution,

$$p_i = \max_k P(C_k \mid f_i), \tag{4}$$

$$k_i = \arg\max_k P(C_k \mid f_i). \tag{5}$$

Thus, for each utterance s we have a sequence $\{f_i, k_i, p_i\}$. The decision rule is to select the call-type of the fragment with maximum p_i, i.e., select $C_K(s)$ where

$$i(s) = \arg\max_i p_i, \tag{6}$$

$$K(s) = k_{i(s)}. \tag{7}$$

If this overall peak is less than some threshold, P_T, then the utterance is rejected and classified as *other*, i.e., if $p_{i(s)} < P_T$.

Several examples are given below, listing the transcription then ASR output using the phrase-bigram grammar of Section 4.2 with the detected fragments highlighted via capitalization and bracketed via underscores. The transduction into call-types with associated scores is then given, with the peak fragment indicated via underlining.

Examples 1–4 below demonstrate robustness of the salient fragments in the presence of recognition errors. The fifth example illustrates an ASR error which yielded a salient fragment, where ‘‘*stick it on my*’’ was misrecognized as ‘‘*speak on my*’’ (observing that stick was not in the training data, thus is an out-of-vocabulary word). The final example involves a user who thought they were talking to a hardware store. In this case, the recognizer performs poorly because of a large number of out-of-vocabulary words. However, the call-classification is indeed correct – leading to transfer to a human agent.

Examples of call classification:

1. Transcr:	yes I just made a wrong telephone number
ASR + parse:	*not help me yes I_JUST_MADE_A long telephone number*
Transd + decision:	{$\underline{\textit{CREDIT 0.87}}$}
2. Transcr:	hello operator I get somebody to speak spanish
ASR + parse:	*motel room you get somebody speak SPANISH*
Transd + decision:	{$\underline{\textit{ATT SERVICE 0.64}}$}
3. Transcr:	hi can I have the area code for saint paul minnesota
ASR + parse:	*hi can THE_AREA_CODE_FOR austin for minnesota*
Transd + decision:	{$\underline{\textit{AREA CODE 1.00}}$}
4. Transcr:	yes I wanted to charge a call to my business phone
ASR + parse:	*yes I_WANNA_CHARGE call to distance phone*
Transd + decision:	{$\underline{\textit{THIRD NUMBER 0.75}}$}
5. Transcr:	hi I like to stick it on my calling card please
ASR + parse:	*hi I'D_LIKE_TO_SPEAK ON_MY_CALLING_CARD please*

Transd + decision:	{*PERSON_PERSON 0.78*}
	{$\underline{\textit{CALLING CARD 0.96}}$}

6. Transcr:	I'm trying to find out a particular staple which fits one of your guns or a your desk staplers

ASR + parse: *I'm trying TO_ FIND_OUT they've CHECKED this is still call which six one of your thompson area state for ask stay with the*
Transd + decision: *{RATE 0.25} {OTHER 0.86}*

5. Experiment results

The database of Section 3 was divided into 8 K training and 1 K test utterances. The remainder of the 10 K database has been reserved for future validation experiments. Salient phrase fragments were automatically generated from the training transcriptions and associated call-types via the methods of Section 4.1. In particular, the length of these fragments was restricted to four or less and to have training-set frequency of five or greater. An initial filtering was imposed so that the peak of the a posteriori distribution for a fragment is 0.6 or greater. We have observed in previous experiments (Gorin, 1995a) that the numerical value of salience is influenced by the fragment frequency, as is typical for information-theoretic measures. Fig. 6 shows a scatter plot of salience versus within-channel information content $i(f) = -\log_2[\mathrm{P}(f)]$. It is thus advantageous to introduce a frequency-tilted salience threshold of the form

$$\mathrm{sal}(f) \geqslant \alpha\, i(f) + \beta. \tag{8}$$

The values of α and β can be varied and evaluated empirically. In the scatter plot of Fig. 6, two thresholds are also shown: the vertical line for the frequency threshold, the other for the frequency-tilted salience threshold. In this experiment, we select the values of α and β via a statistical significance test. For any particular fragment, we evaluate the null hypothesis that its observed a posteriori distribution $\mathrm{P}(C_k \mid f)$ occurs as a random sample of the prior distribution $\mathrm{P}(C_k)$. Computed via a multinomial distribution, a significance level of 1% is imposed, yielding the tilted salience threshold shown in Fig. 6. This reduces the total number of phrase fragments by about 20%. There are approximately 3 K such salient fragments, with length distributed as in Table 4.

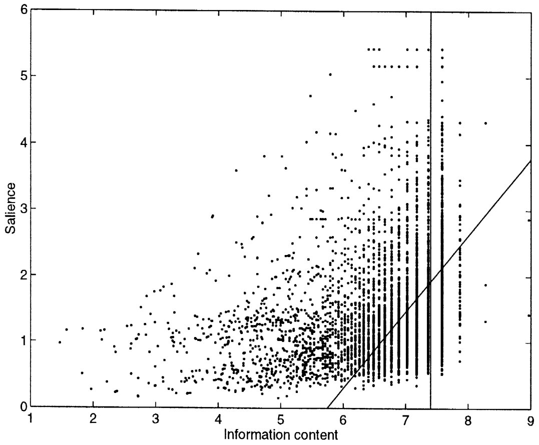

Fig. 6. Salience versus information for phrase fragments.

Table 4
Length of distribution of salient phrases

Length of salient fragment	1	2	3	4
Relative frequency	0.04	0.18	0.43	0.35

The speech recognizer is as described in Section 4.2. The VNSA language model is trained via 20 iterations of the algorithm in Section 4.2, with 50 candidates per iteration. For the phrase-bigram model, this yields 783 phrases in addition to the original 3.6 K word lexicon. The length of these fragments varies between 2 and 16, distributed as shown in Table 5.

We first compare word accuracy and perplexity as a function of the language model. Table 6 shows the word accuracy (defined as probability of correct detection minus probability of insertion) as the language model is varied. Recall that phrase units comprise both the original lexicon of words plus variable-length phrases induced by entropy minimization. Bigrams and trigrams are 2nd and 3rd order models respectively on whichever lexicon is specified. Fig. 7 shows the test set perplexity as a function of language model units and order. We observe that in both of these within-language performance measures, phrase-bigrams fall between word-bigrams and word-trigrams, but with the computation and memory requirements of word-bigrams.

Table 5
Length distribution of VNSA phrase fragemnts

Length of VNSA fragment	2–3	4–5	6–7	8–16
Relative frequency	0.88	0.07	0.03	0.02

Table 6
Word accuracy versus language model

Unit type	Bigram	Trigram
Words	49.5%	52.7%
Words + phrases	50.5%	52.7%

Table 7
Number of salient fragments recognized in an utterance

Salient fragments per utterance	0	1	2	3	4	5	6–11
Relative frequency	0.14	0.25	0.28	0.18	0.07	0.04	0.04

The ASR output is then searched for salient fragments, as described in Section 4.3. If no fragments are found, then the utterance is rejected and classified as other. The number of salient fragments per utterance found in the 1 K test set varies between zero and 11, distributed as shown in Table 7.

5.1. *Performance evaluation*

The salient fragments recognized in an utterance are then rank-ordered, as was described in Section 4.3. We now measure the performance on the test data in terms of true classification rate and false rejection rate. For each test utterance, the decision between *accept* and *reject* is based on the top-ranked call type. If this is *other*, or if the associated probability fails to reach a designated threshold, then the call is rejected. Otherwise, the call is accepted and the accuracy of the attempted classification (at rank 1 and rank 2) is determined using the label set for that call. The desired goal for calls labeled ‘‘*other*’’ is that they be rejected. The *false rejection* rate is the proportion of calls not labeled *other* that are rejected. At rank 1, the *true classification rate* is the proportion of accepted calls for which the top-ranked call type is present in the label set.

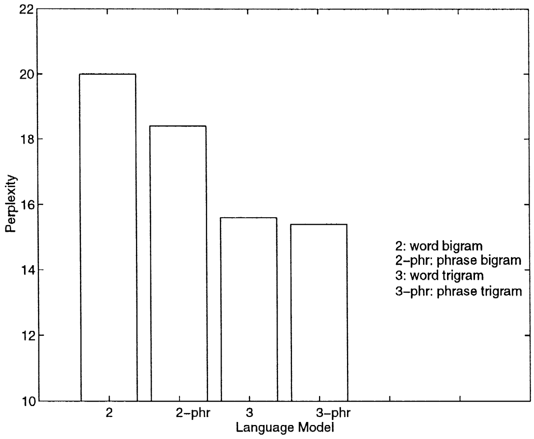

Fig. 7. Test set perplexity versus language model.

At rank 2, the true classification rate is essentially the proportion of calls for which either the first or second highest ranked call type is present in the label set. However, for a small number of calls the label *other* is paired with another call type, and a rejection at rank 2 is then counted as a correct outcome for such a call. We include such cases in the true classification rate at rank 2 because at that point the call has been accepted for handling by the dialog system and contributes to the measure of success appropriate to it.

With these definitions we can plot the ROC curve of true classification rate against false rejection rate. Let’s introduce the following notation and definitions for a particular utterance:

- C is the list of call-type labels (recall that this is typically a single label);
- $\hat{A}_1$ denotes the decision to accept the call at rank 1;
- $\hat{R}_1$ and $\hat{R}_2$ denote the decision to reject the call at rank 1 and rank 2 respectively;
- $\hat{C}_1$ and $\hat{C}_2$ denote the rank 1 and rank 2 call types from the recognized fragments.

We then measure, over the set of 1 K test utterances, the following probabilities:

$$\text{False rejection rate} = \mathrm{P}\left(\hat{R}_1 \,|\, \text{other} \notin C\right), \tag{9}$$

$$\text{True classification rate (rank 1)} = \mathrm{P}\left(\hat{C}_1 \in C \,|\, \hat{A}_1\right), \tag{10}$$

$$\text{True classification rate (rank 2)} = \mathrm{P}\Big(\left(\hat{C}_1 \in C\right) \cup \left(\hat{C}_2 \in C\right) \cup \left(\hat{R}_2 \cap \text{other} \in C\right) \,|\, \hat{A}_1\Big). \tag{11}$$

We generate a performance curve by varying the rejection threshold from 0.6 to 0.95. Fig. 8 shows the rank 1 performance curves for several different ASR language models. As a baseline for comparison, the performance on transcribed output (i.e., error-free

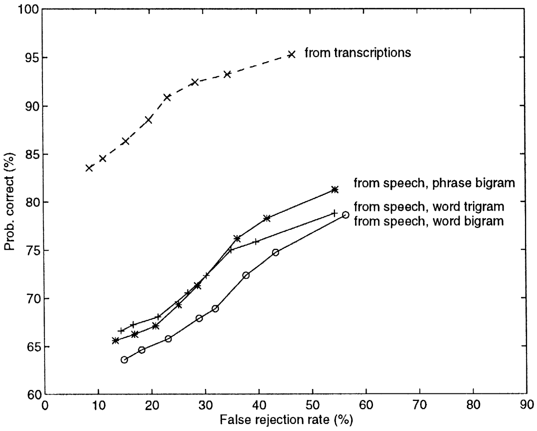

Fig. 8. Call-classification performance for varying ASR language models.

ASR) is also shown. It is interesting to note that call-classification performance is significantly higher than word accuracy – confirming the intuition that some events are crucial to recognize for a task, others not so. It is also worthwhile noting that while the phrase-bigram language model for ASR performs worse than word-trigrams with respect to word accuracy, it performs better with respect to call-classification rate. This reinforces the intuition that optimizing recognition for understanding is an important research issue.

We now compute both rank 1 and rank 2 performance using the phrase-bigram model for ASR, with performance shown in Fig. 9.

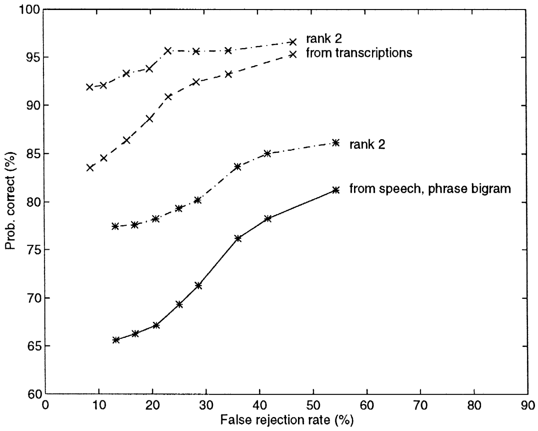

Fig. 9. Rank 1 and rank 2 performance.

6. Conclusions

We have described progress towards a natural spoken dialog system for automated services. By *natural*, we mean that the machine understands and acts upon what people actually say, in contrast to what one would like them to say. A first stage in this system is call-classification, i.e., routing a caller depending on the *meaning* of their fluently spoken response to ''*How may I help you?*'' We have proposed algorithms for automatically acquiring language models for both recognition and understanding, experimentally evaluating these methods on a database of 10 K utterances. These experiments have shown that understanding rate is significantly greater than recognition rate. This confirms the intuition that it is not necessary to recognize and understand every nuance of the speech, but only those fragments which are *salient* for the task.

Acknowledgements

The authors wish to thank Larry Rabiner, Jay Wilpon, David Roe, Barry Parker and Jim Scherer for their support and encouragement of this research. We also thank Mike Riley and Andrej Ljolje for many hours of useful discussion on the ASR aspects of this effort. We finally thank our colleagues Alicia Abella, Tirso Alonso, Egbert Ammicht and Susan Boyce for their continued collaboration in the creation of a spoken dialog system exploiting the methods of this paper.

References

Abella, A., Brown, M., Buntschuh, B., 1996. Developing principles for dialog-based interfaces. In: Proc. ECAI Spoken Dialog Systems Workshop, Budapest, April 1996.

Blachman, N.M., 1968. The amount of information that y gives about x. IEEE Trans. Inform. Theory 14, 27–31.

Boyce, S., Gorin, A.L., 1996. User interface issues for natural spoken dialog systems. In: Proc. Internat. Symp. on Spoken Dialog (ISSD), Philadelphia, October 1996, pp. 65–68.

Cover, T.M., Thomas, J.A., 1991. Elements of Information Theory. Wiley, New York.

Garner, P.N., Hemsworth, A., 1997. A keyword selection strategy for dialog move recognition and multi-class topic identification. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Munich, 1997.

Gertner, A.N., Gorin, A.L., 1993. Adaptive language acquisition for an airline information subsystem. In: Mammone, R. (Ed.), Artificial Neural Networks for Speech and Vision. Chapman and Hall, London, pp. 401–428.

Giachin, E., 1995. Phrase bigrams for continuous speech recognition. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Detroit, 1995, pp. 225–228.

Gorin, A.L., 1995a. On automated language acquisition. J. Acoust. Soc. Amer. 97 (6), 3441–3461.

Gorin, A.L., 1995b. Spoken dialog as a feedback control system. In: Proc. ESCA Workshop on Spoken Dialog Systems, Denmark, June 1995.

Gorin, A.L., 1996. Processing of semantic information in fluently spoken language. In: Proc. Internat. Conf. on Spoken Language Processing (ICSLP), Philadelphia, October 1996, pp. 1001–1004.

Gorin, A.L., Levinson, S.E., Sankar, A., 1994a. An experiment in spoken language acquisition. IEEE Trans. Speech and Audio 2 (1), 224–240, Part II.

Gorin, A.L., Hanek, H., Rose, R., Miller, L., 1994b. Spoken language acquisition for automated call routing. In: Proc. Internat. Conf. on Spoken Language Processing, Yokohama, Japan, September 1994, pp. 1483–1485.

Gorin, A.L., Parker, B.A., Sachs, R.M., Wilpon, J.G., 1996. How may I help you?. In: Proc. IVTTA, Basking Ridge, September 1996, pp. 57–60.

Henis, E.A., Levinson, S.E., Gorin, A.L., 1994. Mapping natural language and sensory information into manipulatory actions. In: Proc. 8th Yale Workshop on Adaptive and Learning Systems, June 1994.

Jelinek, F., 1990. Self-organizing language models for speech recognition. In: Waibel, A., Lee, K. (Eds.), Readings in Speech Recognition. Morgan-Kaufmann, Los Altos, CA, pp. 449–456.

Ljolje, A., 1994. High accuracy phone recognition using context clustering and quasi-triphonic models. Comp. Speech Lang. 8, 129–151.

Masataki, H., Sagisaka, Y., 1996. Variable-order *N*-gram generation by word-class splitting and consecutive word grouping. Proc. Internat. Conf. Acoust. Speech Signal Process., 1996, Vol. 1, pp. 188–191.

Matsumura, T., Matsunaga, S., 1995. Non-uniform unit based HMMs for continuous speech recognition . Speech Communication 17 (3–4), 321–329.

McDonough, J., Gish, H., 1994. Issues in topic identification on the switchboard corpus. In: Proc. ICSLP, Yokohama, 1994, pp. 2163–2166.

Miller, L.G., Gorin, A.L., 1993. Structured networks for adaptive language acquisition. Internat. J. Pattern Recognition Artif. Intel. 7 (4), 873–898.

Pereira, F., Riley, M., 1997. Speech recognition by composition of weighted finite automata. In: Roche, E., Schabes, Y. (Eds.), Finite-State Devices for Natural Language Processing. MIT Press, Cambridge, MA.

Peskin, B., 1993. Topic and speaker identification via large vocabulary speech recognition. In: Proc. ARPA Workshop on Human Language Technology ARPA, Washington, DC, 1993.

Riley, M., Pereira, F., Chung, E., 1995a. Lazy transducer composition: A flexible method for on-the-fly expansion of context-dependent grammar networks. In: Proc. ASR Workshop, Snowbird, 1995.

Riley, M.D., Ljolje, A., Hindle, D., Pereira, F., 1995b. The AT&T 60,000 word speech-to-text system. In: Proc. EUROSPEECH'95, Madrid, 1995, pp. 207–210.

Riccardi, G., Pieraccini, R., Bocchieri, E., 1996. Stochastic automata for language modeling. Comp. Speech Lang. 10 (4), 265–293.

Riccardi, G., Gorin, A.L., Ljolje, A., Riley, M., 1997. Spoken language understanding for automated call routing. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Munich, 1997.

Sankar, A., Gorin, A.L., 1993. Visual focus of attention in adaptive language acquisition. In: Mammone, R. (Ed.), Artificial Neural Networks for Speech and Vision. Chapman and Hall, London, pp. 324–356.

Sharp, R.D., et al., The WATSON speech recognition engine. In: Proc. Internat. Conf. Acoust. Speech Signal Process., Munich, 1997.

Wilpon, J.G., Rabiner, L.R., Lee, C.H., Goldman, E.R., 1990. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE Trans. Acoust. Speech Signal Process. 38 (11), 1870–1878.