Stochastic Language Adaptation over Time and State in Natural Spoken Dialog Systems

Giuseppe Riccardi, *Member, IEEE,* and Allen L. Gorin, *Senior Member, IEEE*

***Abstract*—We are interested in adaptive spoken dialog systems for automated services. Peoples' spoken language usage varies over time for a given task, and furthermore varies depending on the state of the dialog. Thus, it is crucial to adapt automatic speech recognition (ASR) language models to these varying conditions. We characterize and quantify these variations based on a database of 30K user-transactions with AT&T's experimental *How May I Help You?* spoken dialog system. We describe a novel adaptation algorithm for language models with time and dialog-state varying parameters. Our language adaptation framework allows for recognizing and understanding unconstrained speech at each stage of the dialog, enabling context-switching and error recovery. These models have been used to train state-dependent ASR language models. We have evaluated their performance with respect to word accuracy and perplexity over time and dialog states. We have achieved a reduction of 40% in perplexity and of 8.4% in word error rate over the baseline system, averaged across all dialog states.**

***Index Terms*—Language model, large vocabulary speech recognition, spoken dialog system, stochastic finite state machines, stochastic model adaptation.**

I. Introduction

THERE exist a variety of interactive speech systems in laboratories around the world, some even in actual service [4], [6], [8], [11], [12]. There are, however, many open issues concerning how to provide robustness for large populations of nonexpert users. Peoples' spoken natural language is highly variable. A first and well-studied dimension of variation is difference in language usage among individuals [7]. Different people use different words and sentence structure to convey the same meaning [8]. The second variation is over time. The ensemble user-behavior changes as does the world (e.g., ten years ago nobody asked for "`internet access`"). Plus, there are shifts in language usage as people adapt to speaking with machines. The third variation is over dialog state. Depending on the dialog history, in particular the latest prompt, people will of course respond differently.

In this work, we propose a novel algorithm for stochastic language model adaptation that allows for a *natural* human-machine interaction. By *natural*, we mean that the machine recognizes and understands what people actually say, in contrast to what a system designer hoped they would say. We enable the machine to do this by relaxing the constraints on language coverage at each dialog instant, by estimating time and context varying features for the probability distribution of a large vocabulary speech recognizer (LVSR).

The *naturalness* of our spoken dialog system is quantified by analyzing the language characteristics of human-human and human-machine interactions as a function of time. One direct measure of language complexity is the utterance length distribution in terms of words. We will show that this figure of merit allows for a partial separation between human-human and human-machine distributions. Furthermore, language variations in time and dialog contexts show a need to adapt word probability distributions without constraining the vocabulary size. The algorithm for language model adaptation is defined as log-likelihood maximization in the context of the cross-validation technique. In particular, we show our algorithm outperforms the maximum likelihood estimates in tracking the time variation of the empirical distribution. The underlying framework for the model estimation and adaptation is the stochastic finite state machine representation given by the variable N-gram stochastic automaton (VNSA) [16], [17]. In these cases, given one or more input strings as input, the goal is to reestimate state transition probabilities pertaining only to the input set. Input string matching on a finite automaton is a convenient solution to this problem.

The evaluation of our algorithms has been carried out within the *How May I Help You?* spoken dialog system for a call-routing task [8]. Over three years, we have collected a total of 30K user-transactions at three distinct points in time and for different experimental setups. The language probability distributions have been shown to change over time and context and the predictions of the adaptation algorithm have been tested accordingly.

In Section II, we outline the motivations for building adaptive spoken language systems. In Section III, we describe the language variability over these three databases. The language model adaptation algorithm is described in Sections IV and V. The algorithm is experimentally evaluated in Section VI, which gives the improvements in perplexity and word accuracy resulting from the adapted automatic speech recognition (ASR) language models.

Manuscript received December 22, 1998; revised July 26, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James R. Glass.

The authors are with Shannon Laboratory, AT&T Labs-Research, Florham Park, NJ 07932–0971 USA (e-mail: dsp3@research.att.com).

Publisher Item Identifier S 1063-6676(00)00275-3.

II. Spoken Language Systems

Traditionally, for real-world applications, the approach to handling spontaneous speech is to spot task-specific keywords and implement system-initiated dialog strategies. In this case

grammars are hand-crafted to guess the user's response. Everything that is not recognized with high acoustic confidence (within or out of the grammar) is rejected and the user is reprompted [10], [15]. For example, an automated call routing system using word-spotting technology and close-ended prompt would behave as follows (where M denotes machine and U denotes a human user).

M) Please say collect, calling card or operator.
U) *I would like to reverse the charges to Nancy.*
M) Please say collect, calling card or operator.
U) *collect, please.*
M) Please speak the telephone number now.
U) *The number is 1 2 3 4 5 6 7 8 9 0 area code 1 2 3.*
M) Invalid telephone number. Please speak the telephone number now.
U) *1 2 3 1 2 3 4 5 6 7 8 9 0.*
M) Thank you for calling.

In this example, the system is not able to recognize the user's initial request from unconstrained speech. Upon reprompting, the user uses the menu-speak style and the keyword "collect" is spotted correctly. Subsequently, the system is not able to recognize the spontaneous *numeric* language and simply rejects the second utterance. In order to successfully complete the transaction, the user is required to speak the digit sequence in a highly constrained manner.

In contrast to this approach, this paper addresses the problem of creating *natural* spoken dialog systems for automated services. For a human-machine interaction to be natural, it is crucial to have time and dialog-state varying language model parameters. Within a human-machine dialog a word sequence should be predicted based on the whole dialog history. For example the word `yes` is a reinforcement feedback signal in the case of confirmation questions. However, the word "yes" (or its equivalent) is also used colloquially to mark the beginning of a sentence without any further semantic connotation. As a consequence, the probability distribution of "yes" should be dependent on the dialog context.

A *natural* spoken dialog system should allow for recovering the specific goal of the user by having large language lexical coverage at each stage of the dialog. In our system the available lexicon is uniform throughout the dialog session so that the understanding module is reactive to either user's or system's initiated topic switch [24]. The following is an illustrative example.

M) How May I Help You?
U) *I want to put this on my VISA card.*
M) What is your card number?
U) *Uh, I can't find it. Can I make this a collect call?*
M) What number would you like to call?
U) *Good question. I need John Smith's number in Newark.*
M) Please hold on for directory assistance.

Thus, our goal is to shift the burden from human to machine, so that the system adapts to people's language, in contrast to forcing users to learn the machine's jargon. In the next sections we will examine how peoples' language actually varies in time and dialog context and how language models can be adapted so that a *natural* interaction with the system is possible.

TABLE I
DIALOG STATE AS PROMPT-EQUIVALENCE CLASSES

Prompt Class	Example
GREETING	*AT&T, How May I Help You ?*
BILLING METHOD	*How would you like to bill this call?*
CARD NUMBER	*May I have your card number, please?*
CONFIRMATION	*Do you need me to give you credit?*
PHONE NUMBER	*What number would you like to call?*
REPROMPT	*Sorry. Please briefly tell me how may I help you?*

III. MEASURING LANGUAGE VARIABILITY

A. Databases

In spoken dialog systems, users' utterances depend on the dialog history and should be clustered accordingly. Hence, we partitioned the data based on the notion of dialog state. Each dialog state is associated with a set of users' responses. There are many notions of dialog state in the literature. In fact, the dialog manager in our system [1] has *no* explicit representation of state. But, in these experiments we mapped users' responses into equivalence classes of prompts, which is a first-order approximation to dialog history. Examples from these various classes are shown in Table I.

Over three years, there have been three data collections in the process of training and adapting language models for the *How May I Help You*? spoken dialog system. The users have always been sampled at random and generally used only once our automated system. The three databases will be referred to as HH, HM1, and HM2 sets.

HH) This first collection served as bootstrap for our language models. We transcribed only the user's response to *human* agents' greeting of *How May I Help You*? The training and test set is composed of 7844 and 1000 utterance transcriptions, respectively [8].
HM1) The HH training set was used to train language models for speech recognition and understanding for the first dialog system for *How May I Help You*? For later stages of the dialog where we had no training data, we designed place-holder grammars. We then had the spoken dialog system interact with live customer traffic and collected the HM1 database. The size of the training and test sets in HM1 are, respectively, 8K and 1K.
HM2) The HH and HM1 data sets were used to train and adapt the speech recognizer to the different dialog contexts. We then exposed this new incarnation to live traffic and gathered the third set, namely the HM2 database (12K). This dataset has been used for the system evaluation only.

B. Empirical Word Sequence Distributions

As was observed in[8], the number of words per utterance in HH is unimodal and highly skewed with a long tail. In Fig. 1,

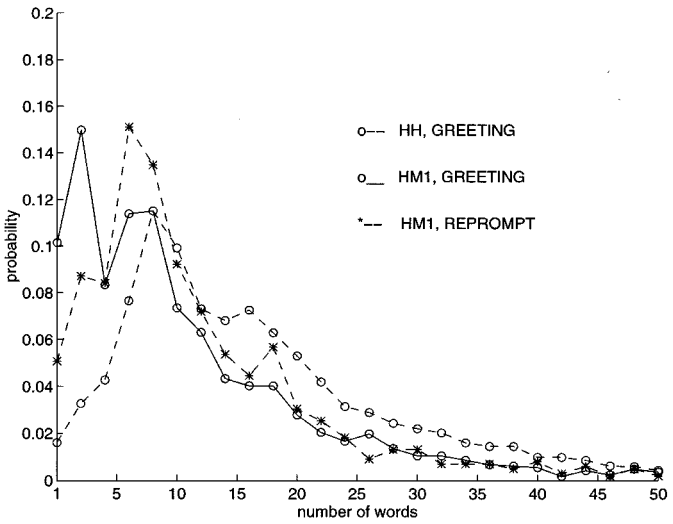


Fig. 1. Utterance length distribution for responses to REPROMPT and GREETING prompt-equivalence class in HH and HM1 databases.

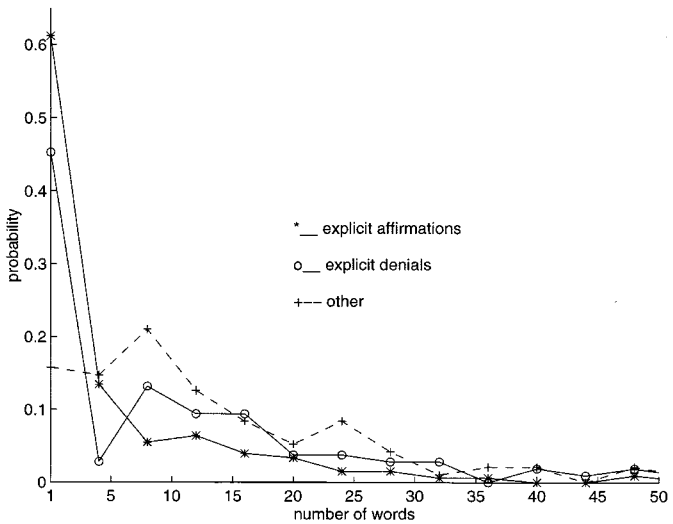


Fig. 2. Utterance length distribution for responses to CONFIRMATION prompt-equivalence class in HM1 database.

we compare that to the length distribution for responses to the GREETING prompt in HM1. First, observe that the HM1 histogram is bimodal. One mode corresponds to menu-speak: when people are aware that they are talking with a machine, then they sometimes speak in short fragments. Interestingly, while some of the menu-speak corresponds to keywords on deployed menus, many do not. Instead, these short phrases often correspond to the salient fragments which were derived from the HH natural language database. Observe also that the second mode of HM1 is almost identical to the single mode of the HH responses. Thus, we can view the HM1 GREETING-responses as a mixture of menu-speak and natural spoken language, with the second component similar to the natural language in HH. Also in Fig. 1, we observe that the HM1 distribution tail falls off much faster than for HH. Upon inspection, we observe that the very long utterances in HH are accompanied by the agent's back-channel utterances such as uh-huh, encouraging the customer to continue talking. In the case of HM1, there is no such back-channel encouragement from the machine, so people don't tell long stories as often. Finally, also in Fig. 1, we plot the length distribution for responses to a reprompt in HM1, observing that it is also unimodal and similar to the HH distribution of natural language responses to a human agent. So, it appears that these people who need reprompting respond in natural language, not menu-speak. This fact reinforces the need for training language models based on dialog history.

We then measure the length distribution for responses to CONFIRMATION prompts, as shown in Fig. 2. The responses are divided into three categories: *explicit affirmations, explicit denials*, and *other*. Explicit affirmation/denials are sentences which contain the words "yes" or "no" or some variant thereof (i.e., the YES-NO equivalence classes). These are sometimes spoken in isolation, or as a prepend to a natural language utterance to provide further task information. For example, responding to the prompt *Do you want to make a credit card call?*, as a user might respond "Yes, the card number is xxxxxxx." The *other* category occurs during context-switching, error recovery or user-confusion (see the second example in

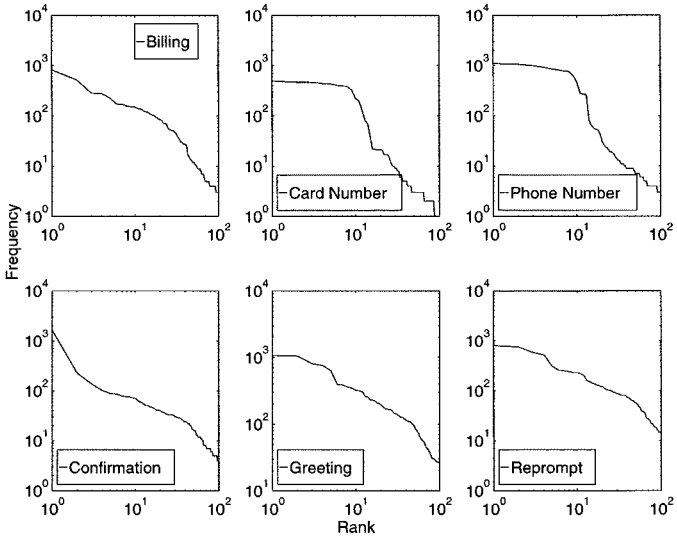


Fig. 3. Word frequency versus rank order plots for the six dialog contexts (log-log scale) in HM1 database.

Section II). Observe that the *affirmation*-length distribution is unimodal and tends to comprise shorter utterances than the denials. The explicit denials are a bimodal mixture of short responses plus a second mode at the same position as for the GREETING prompts. These modes correspond to people answering no or some variant (short utterances) or to people using natural language, with "no" prepended. Thus, we observe that it is more likely for "no" to be followed by additional spoken information than it is for "yes.". Finally, the *other* responses also have their mode at that same position, corresponding to the natural language distribution.

The utterance length distributions over different contexts give us a statistical description of the language complexity as measured by the sentence length. Another measure of the language complexity are the Zipf plots [25] of word relative frequency versus their rank orders. For natural language, Zipf's relation is of the form $fr = K$, where f is the relative frequency, r is the rank order and K is a constant. In Fig. 3

we show the log-log Zipf plots of six different dialog contexts, for the HM1 data. The log-linear dependency fits not only the open-ended prompts (GREETING and REPROMPT) but also the CONFIRMATION and BILLING queries. In the case of CARD and PHONE NUMBER, the curve fitting is composed of a constant piece (all digits are approximatively equally likely) and a log-linear piece which accounts for the carrier phrases within spoken digits and user or system error recovery. All the empirical utterance length and word distributions support the argument for language models with large lexicon coverage at any instant in the dialog. In the next section we describe how language models are trained for large lexicon coverage and specific to each dialog context.

IV. Language Modeling

In the standard speech recognition paradigm, language models exploit the lexical context statistics (word tuples) observed in a training set to predict word sequence probabilities on a test set. In that traditional approach, the underlying assumption is that the information source (the *natural* language) is stationary. As a consequence, this evaluation paradigm does not account for the temporal and contextual language variation in a human-machine interaction. In contrast with this scenario, spoken dialog systems pose a challenge to the traditional view of language model training. In general the word sequence distribution at stage s_k of the dialog is dependent on the entire interaction history. Hence, it is more appropriate to conceive the LVSR as a statistical model that dynamically adapts to the different stages of the human-machine negotiations for successfully completing the task.

Learning language models that adapt to different events in the course of a spoken dialog session is tightly coupled with the state sequence associated with the human-machine interaction. In general, a dialog state s_k should keep track of the entire history. However, we will make a first-order approximation and associate a state s_k to each prompt equivalence class. The word probability computation will apply to any definition of dialog state.

The entire transaction is associated with a state sequence and the model is defined in terms of the states and state transitions. The state s_k is then used as a predictor to compute the word sequence probability $P(w_1, w_2, \cdots, w_N|s_k)$, as follows:

$$P(w_1, w_2, \cdots, w_N|s_k) = \prod_j P(w_j|w_1, w_2, \cdots, w_{j-1}; s_k). \tag{1}$$

The computation of the probability $P(w_j|w_1, w_2, \cdots, w_{j-1}; s_k)$ can be decomposed into two subproblems. The first addresses the problem of computing the word sequence probability given the state s_k. The second involves the estimation of $P(w_j|w_1, w_2, \cdots, w_{j-1}; s_k)$. In previous reported research, such dialog models have been used to partition the whole set of utterances spoken in the dialog sessions into subsets (first subproblem) and then train standard n-gram language models (second subproblem) [11], [21]. A deficiency in that approach is that the user can only utter words that he has previously (training set) spoken in a specific dialog state. Such language model design does not allow for topic switching, or on-line error recovery from speech understanding errors. Thus, the main disadvantages of all previous approaches are the poor language coverage at each state of the dialog and data fragmentation. In other related work, the estimation problem is solved by linear interpolation [21] or maximum entropy models [14], speaker backoff models [2], or MAP training [5]. In this work we take the approach of training language models for each state s_k in such a way that the user can interact in an open-ended way without any constraint on the expected action at any point of the negotiation.

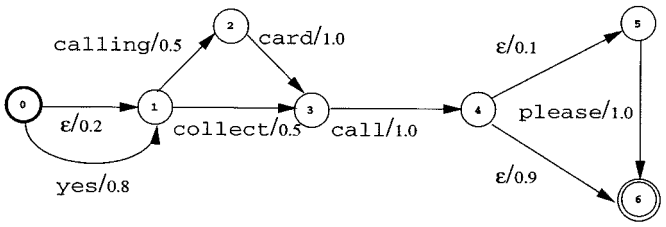


Fig. 4. Example of a variable Ngram stochastic automaton (VNSA).

In order to condition the expected probability of any event at state s_k we propose a novel adaptation algorithm for self-organizing stochastic finite state machines. At the same time, the word probability distribution is estimated to account for any possible event at any instant of the dialog. In the following section, we outline the stochastic finite state machine representation of the language model and the novel adaptation algorithm.

A. Stochastic Finite State Machines

Our approach to language modeling is based on the VNSA representation and learning algorithms first introduced in [16] and [17]. The VNSA is a nondeterministic stochastic finite state machine (SFSM) that allows for parsing any possible sequence of words drawn from a given vocabulary V. In its simplest implementation the state q in the VNSA encapsulates the lexical (word sequence) history of a word sequence. Each state recognizes a symbol $w_i \in V \cup \{\epsilon\}$, where ϵ is the empty string. The probability of going from state q_i to q_j (and recognizing the symbol associated with q_j) is given by the state transition probability, $P(q_j|q_i)$. Stochastic finite state machines represent the probability distribution over all possible word sequences in a compact way. The probability of a word sequence W can be associated with state sequences $\xi^j = q_1^j, \cdots, q_{n_j}^j$ and to the probability $P(\xi^j)$. For a nondeterministic finite state machine ($j > 1$) the probability of W is then given by $P(W) = \sum_j P(\xi^j)$. Moreover, by appropriately defining the state space to incorporate lexical and extra lexical information, the VNSA formalism can generate a wide class of probability distribution (i.e., standard word n-gram, class-based, phrase-based, etc.) [17]–[19]. In Fig. 4, we plot a fragment of a VNSA trained with word classes and phrases. State zero is the initial state and final states are double circled. The ϵ transition from state zero to state one carries the membership probability $P(C)$, where the class C contains the two elements {*collect, credit card*}. Then, the probability of going from state zero to three is the class-based estimate $P_{CB}(\text{``}collect\text{''}) = P(C)P(\text{``}collect\text{''}|C)$. The ϵ transition from state four to state five is a *backoff* transition to a

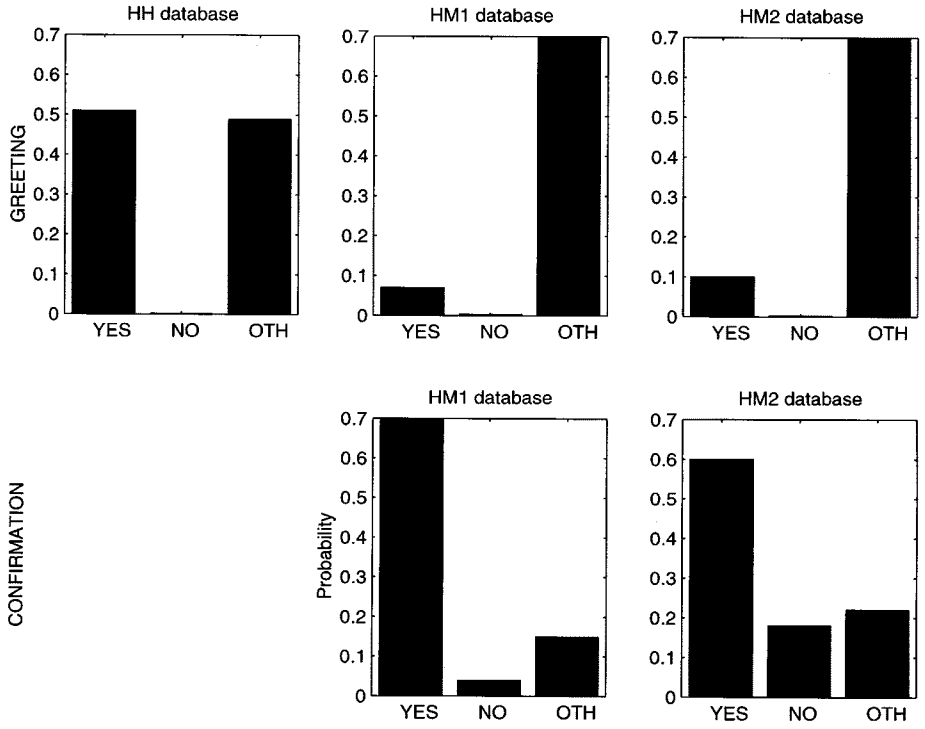


Fig. 5. Probability of the word classes YES, NO, and OTHER for three points in time (HH, HM1, and HM2 databases) and for two dialog states.

lower order n-gram probability. The state two carries the information about the phrase *calling card*. The state transition function, the transition probabilities and state space are learned via the self-organizing algorithms presented in [17].

V. Language Model Adaptation

In spoken language system design, the state of the dialog s_k is used as predictor of the user response. For example, if the computer asks a CONFIRMATION question, then the most likely response will contain language in the YES-NO equivalence class. However, in order to provide robustness to user-initiated context switch, or system errors, we want to enable the system to move from one state to any other state of the dialog without *a priori* defined constraints. We achieve this goal by training language models that recognize unconstrained utterances for each state s_k. At the same time we adapt language models for each stage based on the expected users' responses to open-ended prompts.

In Fig. 5, we plot the word distribution for the first token of a sentence for different dialog contexts. The distributions are computed along two dimensions, time (HH, HM1, and HM2 data sets) and dialog contexts (GREETING and CONFIRMATION).[1] In particular, we define three word classes: YES and NO containing all the equivalent words for "yes" and "no," respectively, and the OTHER (OTH) word class subsuming the remaining words in the dictionary. Note that in the HH database, the word class YES occurs 50% of the times for the GREETING stage, while its occurrence on the HM1 and HM2 GREETING sets is negligible. This is an interesting characterization of language usage in human-human and human-machine interactions (see also menu-speak effect in Fig. 1).

[1]Recall from Section III that for the HH set we transcribed only the responses to the GREETING prompt.

A. Adaptation Algorithm

The first step of the adaptation algorithm consists of partitioning the empirical data into all the available dialog contexts. The set of all user's observed responses at a specific stage k of the dialog is split into training $\mathcal{T}_k$, development ($\mathcal{B}_k$), and test ($\mathcal{E}_k$) sets. We assume that there is an initial model $\lambda^{\mathcal{T}}$ to bootstrap the adaptation algorithm and provide a probability estimate for all words

$$\lambda_k^* = \arg\max_{\lambda_k^A} \log P(\mathcal{B}_k|\lambda_k^A) \tag{2}$$

where the model λ_k^A is the generic adapted language model. The maximum likelihood (ML) solution to the problem in (2) is given by a model exclusively trained on $\mathcal{T}_k$. However, the size of the training set $\mathcal{T}_k$ has generally insufficient statistics for reliable estimates of the model probabilities. Thus, the formulation of the adapted language model is given in terms of a convex interpolation of the language model $\lambda^{\mathcal{T}}$ and a state dependent model λ_k. This formulation is consistent with the Bayesian or maximum *a posteriori* (MAP) training proposed in the literature in the case of adaptation from small data sets [5], [21]. Recall that the language model $\lambda^{\mathcal{T}}$ is composed of the state transition function F

$$F\colon (q, w) \Rightarrow p, \quad p, q \in Q \quad \text{and} \quad w \in V \cup \{\epsilon\} \tag{3}$$

where Q is the set of all the SFSM states, and the state transition probabilities of the kind $P(q_j|q_i)$, where $F(q_i, w) = q_j$ for $w \in V \cup \{\epsilon\}$. In general, the language model $\lambda^{\mathcal{T}}$ has larger coverage than the data represented in $\mathcal{B}_k$. Moreover, the model λ_k^A should be estimated from the statistics drawn from $\mathcal{B}_k$. Then for λ_k^A to have a high coverage language model while modeling

the data source ($\mathcal{B}_k$), the solution is to bootstrap the state transition function F from $\lambda^{\mathcal{T}}$ and compute the ML estimates, over $\mathcal{B}_k$. In practice, we replace the ML estimate with the Viterbi approximation in order to prune the low probability state sequence paths. As for the estimation of the model λ_k, we run Viterbi training over each set $\mathcal{T}_k$ starting from the generic model $\lambda^{\mathcal{T}}$ and estimate the transition probabilities. In order to account for unseen transitions, we smooth the transition probabilities with the standard discount techniques discussed in [17]:

$$P_k(q_j|q_i) = \frac{n_{i,j}+1}{n_i + m_i} \tag{4}$$

where n_i ($n_{i,j}$) is the number of times the state q_i (state transition $q_i \rightarrow q_j$) is selected by the Viterbi decoding and m_i is the number of transitions leaving q_i. The transition probabilities for the model λ_k^A are then computed as follows:

$$\begin{aligned} P_k^A(q_j|q_i) &= \alpha_k P^{\mathcal{T}}(q_j|q_i) + (1-\alpha_k)P_k(q_j|q_i) \\ \sum_j P_k^A(q_j|q_i) &= 1 \\ P_k^A(q_j|q_i) &\geq 0. \end{aligned} \tag{5}$$

Recall that in a nondeterministic stochastic automaton there are transitions where either $w_i \in V$ is recognized or the epsilon symbol gets processed (also known as *epsilon move*). The *epsilon move* is encountered typically for backoff purposes or nonterminal symbol resolution (e.g., class-based language models). Thus, (5) applies to both cases in a unified manner, in contrast to the traditional linear interpolation of n-gram probabilities [9].

The solution to (2) with respect to the parameters α_k cannot be given in an explicit form. However, for the adaptation form in (5), we use the cross-validation paradigm over the development sets $\mathcal{B}_k$ to find the local optimum over a finite number of α_k values. ($\alpha_k^* = 0.8$). Whereas the data is insufficient for CARD and PHONE NUMBER cases, α_k^* is given a fixed value of 0.5. When a model $\lambda^{\mathcal{T}}$ is not available for bootstrap, a context independent variable N-gram stochastic Aautomaton can be trained by pooling together the training sets $\mathcal{T}_k$ [20] and still achieving accurate state-dependent language models. The complete block diagram, describing the adaptation scenario and adaptation algorithm steps is shown in Fig. 6.

An important issue in tracking the variability of a probability distribution is the stochastic separation from a prior or alternate distribution. The problem is to measure the similarity between the model λ_i^* and λ_j^* or $\lambda^{\mathcal{T}}$. If the two distributions are similar the sample data they have been estimated is most likely drawn from the same random source. A classical measure of stochastic separation used in the decision theory for hypothesis testing is the log likelihood ratio (LLR). We extend this notion to the token-level log likelihood ratio computed over the development set $\mathcal{B}_k$:

$$\mathrm{LLR}(k,\, j) = \frac{1}{N} \log \frac{P(\mathcal{B}_k|\lambda_k^A)}{P(\mathcal{B}_k|\lambda_j^A)} \tag{6}$$

where N is the number of tokens in $\mathcal{B}_k$. The $\mathrm{LLR}(k,\, j)$ measure over all possible models $j \neq k$ can be positive or negative. If it is a large positive (negative) number, that means that λ_k^A is modeling the data $\mathcal{B}_k$ better (worse), in the log likelihood sense,

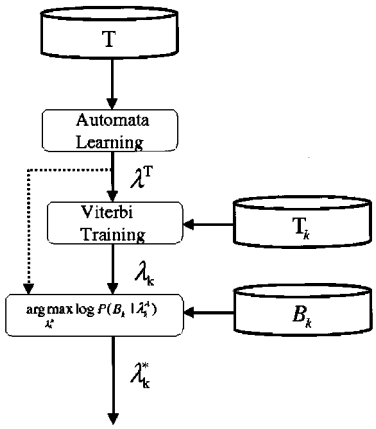


Fig. 6. Block diagram for the language model adaptation.

TABLE II
TEST SET PERPLEXITY FOR STATIONARY AND TIME-VARYING MODEL

Dialog State	Stationary (ML model)	Time-Varying (Adapted Model)
GREETING	13.7	12.8
BILLING METHOD	10.4	6.4
CARD NUMBER	18.2	15.0
CONFIRMATION	9.0	7.1
PHONE NUMBER	16.3	12.8
REPROMPT	14.0	13.2

than λ_j^A. It is interesting to note that $\mathrm{LLR}(k,\, j)$ corresponds to the entropy gain of the model λ_k^A with respect to λ_j^A:

$$\mathrm{LLR}(k,\, j) = H(\mathcal{B}_k|\lambda_j^A) - H(\mathcal{B}_k|\lambda_k^A) \tag{7}$$

where, for the entropy, we have used the ergodic assumption of the information source instantiated in $\mathcal{T}_k$, $\mathcal{B}_k$ and $\mathcal{E}_k$. We will use the $\mathrm{LLR}(k,\, j)$ figure of merit to validate the estimates computed through (2) in the next section.

B. Stationary versus Time and State Varying Stochastic Models

An important characterization of an information source is its stationarity with respect to the parameters of its probability distribution.[2] If we consider the sample data at each instant an instantiation of a stationary process, then the ML estimates can be computed by pooling together the data sampled at that instant. On the other hand, if we wish to model a time-varying source, we need to stochastically update the parameters of the probability distribution. In the case of *spontaneous* spoken language, the most appropriate working hypothesis is to assume the nonstationarity hypothesis for two reasons: not only do the statistics of natural language vary over time and from one dialog state to another, but the interface of the dialog system may be improved from time to time, eliciting different kinds of spoken responses. Moreover, from a practical point view it is not always the case that the databases are available for on-line adaptation.

In Table II, we compare the test set perplexity of the stationary and stochastically adapted models. For the stationary model, we train ML language models by pooling together the HH data and the HM1 training sets.

[2]Here we consider stationarity with respect to the mean.

TABLE III
AVERAGE STOCHASTIC SEPARATION FOR CONTEXT-DEPENDENT LANGUAGE MODELS

Prompt Class	$\overline{LLR}(k)$
GREETING	0.75
BILLING METHOD	1.25
CARD NUMBER	0.50
CONFIRMATION	1.88
PHONE NUMBER	0.49
REPROMPT	0.53

TABLE IV
PERPLEXITY REDUCTION VIA ADAPTATION TO DIALOG STATE

Dialog State	Baseline $\lambda^{\mathcal{T}}$ on HM1	Adapted λ_i^* on HM1	Adapted λ_i^* on HM2
GREETING	17.3	12.8	13.8
BILLING METHOD	17.0	6.4	8.4
CARD NUMBER	21.2	15.0	16.1
CONFIRMATION	27.8	7.14	26.8
PHONE NUMBER	19.8	12.8	15.7
REPROMPT	15.1	13.2	13.8

TABLE V
PERCENT WORD ERROR RATE (WER) FOR STATE-ADAPTED LANGUAGE MODELS. Δ(WER) IS THE WER RELATIVE IMPROVEMENT BETWEEN THE BASELINE (SECOND COLUMN) AND ADAPTED MODELS (THIRD COLUMN)

State	Trial	$(\lambda^{\mathcal{T}})$	(λ_i^*)	(WER)
GREETING	47.6	47.6	43.8	7.9
BILLING METHOD	40.0	40.0	36.0	10.0
CARD NUMBER	27.5	15.5	13.0	16.1
CONFIRMATION	60.4	45.6	41.7	8.5
PHONE NUMBER	30.0	20.8	17.9	13.9
REPROMPT	43.3	43.3	42.6	1.6
Average	48.3	35.5	32.5	8.4

For each dialog state s_k, the stochastic mapping $\lambda^{\mathcal{T}} \rightarrow \lambda^*_{s_k}$ ($s_k =$ GREETING, BILLING, $\cdots$) has been estimated with the algorithm described in the previous section. In Table II, we show the test perplexity measured on the HM1 test sets $\mathcal{E}_{s_k}$. In the case of GREETING and REPROMPT stages, the adapted model slightly outperforms the ML estimates in tracking language variation over time and nature of interaction (human-human *versus* human-machine). In the other cases, the performance of the adapted models show that they are very effective in making the *background* model $\lambda^{\mathcal{T}}$ tailored to the statistics of dialog contexts with relatively little amount of data.

While test set perplexity is a measure of a stochastic model's prediction power, it also useful to quantify the distance between language models with the LLR figure of merit. In fact, users' responses might overlap more in some stages of the dialog than in others. For example, the responses to PHONE and CARD NUMBER requests have similar word distributions (see Section III-B) and in fact their LLR(k, i) is small. In Table III, we report the average LLR(k, j) for a specific dialog state s_k. The average LLR is given by $\overline{\text{LLR}}(k) = 1/5 \sum_i \text{LLR}(k, i)$, where there are five language models λ_i^* competing with λ_k^* on the same development set $\mathcal{B}_k$. As pointed out in (7), the LLR(k, i) figure of merit can interpreted as entropy gain (in bits), so that a one bit entropy gain corresponds to halving the perplexity (equivalent models have LLR $= 0$). In Table III there are two dialog contexts (BILLING and CONFIRMATION) that stand out for their stochastic separation from the other stages of the dialog. Those queries turn out to be the final stages of the human-machine interaction.

Overall, context dependent language models achieve high LLR values for each state of the dialog s_k. Thus, we have shown that the adaptation algorithm achieves effective separation for modeling large-coverage language at a given dialog state.

VI. APPLICATION OF THE ADAPTATION ALGORITHM

Recall that $\lambda^{\mathcal{T}}$ is a language model trained from HH: peoples' responses to a human agent's greeting. The state-conditional model for state s_k was obtained by adapting with the data $\mathcal{T}$ and $\mathcal{B}_k$ from HM1 training sets. One method to evaluate the utility of this adaptation is to compute their test-set perplexities on the test sets $\mathcal{E}_k$ drawn from HM1 database, as shown in Table IV.

Also shown is the perplexity on HM2 whose data has *not* been used to compute λ_i^* and corresponds to a later data collection. As was reported in [8], the test-set entropy of HH was 18.2. The responses to the GREETING prompt in HM1 occurred later in time, with a modified prompt to "tip our hand" that people were talking with a machine [3]. The language variation in both time and state is illustrated by each row of Table IV. The adapted language model provides a significantly lower perplexity for the human/machine data than the human/human data. Observe also that the adapted model does a better job of modeling the GREETING-responses in HM1 and HM2, as compared to HH. This confirms our intuition that people's responses are "simpler" in HM1 (or HM2) than HH, as discussed also in our earlier analysis of utterance-length.

In Table V, we provide corresponding measurements of word accuracy at each dialog state for these adapted models.

The first column gives the speech recognition results of the first second trial (HM1 test sets). In this system, we used place-holder grammars where needed: for the GREETING, REPROMPT and BILLING states we used the $\lambda^{\mathcal{T}}$ model and for the other contexts we designed hand-crafted grammars for digit recognition [15] and CONFIRMATION questions. In the second and third column, each speech recognition language model has a uniform lexicon coverage and a vocabulary of 3.6K words. The word accuracy is improved over the baseline system across all dialog states. We remark that for the CARD and PHONE NUMBER responses, this is the average accuracy over all dictionary words (columns 2 and 3), not just the digits (column 1). A detailed discussion of the language distribution and baseline performance for utterances containing embedded digit sequences is in [15]. We also remark that task accuracy is much higher than the word accuracy, as detailed in [8]. The latest reported result is 91% correct call-classification on the HH GREETING-responses [23]. For the number queries (PHONE and CARD NUMBER), the place-holder grammars in the HM1 trial were digit loops with appropriate constraints and garbage models at each end. Although most of the tokens in those utterances were indeed digits, there were still 15% nondigit tokens. Thus, adapting a large vocabulary grammar improves word accuracy over the digit-only grammars.

VII. Conclusion

In this work, we have addressed the problem of modeling spoken language for adaptive human–machine interactions. We have analyzed the statistical variations of language for human-human and human–machine interactions. We have presented a novel adaptation algorithm for estimating the time and state varying parameters of language models for natural spoken dialog systems. These models allow users to say anything at anytime in the dialog. The adapted language models fit the data better than the ML estimator for nonstationary process. We have quantified the notion of dialog context dependency via the LLR figure of merit and demonstrated the specificity of language models for each dialog stage s_k. Then, this algorithm was evaluated with respect to perplexity and word accuracy on a database of 30K human-machine transactions. We have achieved a reduction of 40% in perplexity and of 8.4% in WER over the baseline system, averaged across all dialog states.

References

[1] A. Abella and A. L. Gorin, "Generating semantically consistent input to a dialog manager," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1879 –1882.

[2] S. Besling and H. Meier, "Language model speaker adaptation," in *Proc. Eurospeech*, Madrid, Spain, 1995, pp. 1755–1758.

[3] S. Boyce and A. L. Gorin, "Designing user interfaces for spoken dialog systems," in *Proc. Int. Symp. Spoken Dialog (ISDD)*, Philadelphia, PA, 1996, pp. 65–68.

[4] B. Carpenter and J. Chu-Carroll, "Natural language call routing: a robust, self-organizing approach," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 76–79.

[5] M. Federico, "Bayesian estimation methods for N-gram language model adaptation," in *Proc. ICSLP*, Philadelphia, PA, 1996, pp. 240–243.

[6] J. R. Glass and T. J. Hazen, "Telephone-based conversational speech recognition in the JUPITER domain," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 593–596.

[7] A. L. Gorin, "On automated language acquisition," *J. Acoust. Soc. Amer.*, vol. 97, pp. 3441–3461, June 1995.

[8] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you?," *Speech Commun.*, vol. 23, pp. 113–127, 1997.

[9] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press, 1998.

[10] E. Lleida and R. C. Rose, "Efficient decoding and training procedures for utterance verification in continuous speech recognition," in *Proc. ICASSP*, Atlanta, GA, 1996, pp. 507–510.

[11] C. Popovici and P. Baggia, "Specialized language models using dialog predictions," in *Proc. ICASSP*, Munich, Germany, 1997, pp. 815–818.

[12] A. Potamianos and S. Narayanan, "Spoken dialog systems for children," in *Proc. ICASSP*, Seattle, WA, 1998, pp. 197–201.

[13] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[14] P. S. Rao, M. D. Monkowski, and S. Roukos, "Language model adaptation via minimum discrimination information," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 161–164.

[15] M. Rahim *et al.*, "Robust automatic speech recognition in a natural spoken dialog," in *Proc. Workshop on Robotic Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999.

[16] G. Riccardi, E. Bocchieri, and R. Pieraccini, "Non deterministic stochastic language models for speech recognition," in *Proc. ICASSP*, Detroit, MI, 1995, pp. 247–250.

[17] G. Riccardi, R. Pieraccini, and E. Bocchieri, "Stochastic automata for language modeling," *Comput. Speech Lang.*, vol. 10, pp. 265–293, 1996.

[18] G. Riccardi, A. L. Gorin, A. Ljolje, and M. Riley, "A spoken language system for automated call routing," in *Proc. ICASSP*, Munich, Germany, 1997, pp. 1143–1146.

[19] G. Riccardi and S. Bangalore, "Automatic acquisition of phrase grammars for stochastic language modeling," in *Proc. ACL Workshop on Very Large Corpora*, Montreal, P.Q., Canada, 1998, pp. 88–196.

[20] G. Riccardi, A. Potamianos, and S. Narayanan, "Language model adaptation for spoken language systems," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 2327–2330.

[21] H. Sakamoto and S. Matsunaga, "Continuous speech recognition using dialog-conditioned stochastic language model," in *Proc. ICSLP*, Yokohama, Japan, 1994, pp. 841–844.

[22] P. Taylor *et al.*, "Using prosodic information to constrain language models for spoken dialog," in *Proc. ICSLP*, Philadelphia, PA, 1996, pp. 216–219.

[23] J. H. Wright, A. L. Gorin, and G. Riccardi, "Automatic acquisition of salient grammar fragments for call-type classification," in *Proc. Eurospeech*, Rhodes, Greece, 1997, pp. 1419–1422.

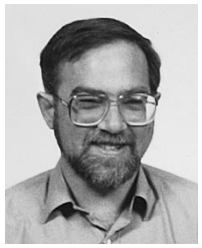
[24] J. H. Wright, A. L. Gorin, and A. Abella, "Spoken Language understanding within dialogs using a graphical model of task structure," in *Proc. ICSLP*, Sydney, Australia, 1998, pp. 385–388.

[25] G. K. Zipf, *The Principle of Least Effort*. Reading, MA: Addison-Wesley, 1949.

Giuseppe Riccardi (M'96) received the Laurea degree in electrical engineering and the M.S. degree in information sciences in 1991 from the University of Padua, Padua, Italy, and CEFRIEL Research Center, Milan, Italy, respectively, and the Ph.D. degree in electrical engineering in 1995 from the University of Padua.

From 1990 to 1993, he collaborated with Alcatel-Telettra Research Laboratory, Milan, Italy, and investigated algorithms for speech and audio coding for medium-low bit rates. From 1993 to 1995, he spent two years as visiting researcher at AT&T Bell Laboratories, working on automata learning for stochastic language modeling. He participated at development of the AT&T spoken language system used in the 1994 DARPA ATIS evaluation. In 1996, he joined AT&T Laboratories-Research, Florham Park, NJ, where he is currently Principal Member of Technical Staff. He has coauthored more than 30 papers in the field of speech and audio coding, speech recognition, and language modeling. His current research interests are stochastic language modeling, language understanding, spoken dialog, and machine translation.

Dr. Riccardi s a member of the Association for Computational Linguistics.



Allen L. Gorin (M'80–SM'92) received the B.S. and M.A. degrees in mathematics from the State University of New York, Stony Brook, in 1975 and 1976, respectively, and the Ph.D. degree in mathematics from the City University of New York Graduate Center in 1980.

From 1980 to 1983, he was with Lockheed Corporation, investigating algorithms for target recognition from time-varying imagery. In 1983, he joined AT&T Bell Laboratories, Whippany, NJ, where he was the Principal Investigator for AT&T's ASPEN project within the DARPA Strategic Computing Program, investigating parallel architectures and algorithms for pattern recognition. In 1987, he was appointed Distinguished Member of Technical Staff. In 1988, he joined the Speech Research Department, Bell Laboratories, Murray Hill, NJ, and is now with AT&T Labs-Research, Florham Park, NJ. He was a Visiting Researcher at the ATR Interpreting Telecommunications Research Laboratory, Kyoto, Japan, during 1994. His long-term research interest focuses on machine learning methods for spoken language understanding.

Dr. Gorin has served as a guest editor for the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He is a member of the Acoustical Society of America.