Adaptive Categorical Understanding for Spoken Dialogue Systems

Alexandros Potamianos [1], Shrikanth Narayanan [2] and Giuseppe Riccardi [3]

[1]Dept. of Electronics & Computer Engineering, Technical Univ. of Crete, Chania 73100, Greece

[2] Dept. of Electrical Engineering, Univ. Southern California, Los Angeles, CA 90089, U.S.A.

[3] AT&T Labs–Research, 180 Park Ave, Florham Park, NJ 07932-0971, U.S.A.

Email: potam@telecom.tuc.gr, shri@sipi.usc.edu, dsp3@research.att.com

March 12, 2004

Abstract

In this paper, the speech understanding problem in the context of a spoken dialogue system is formalized in a maximum likelihood framework. Off-line adaptation of stochastic language models is proposed that interpolates the dialogue state specific and general application-level language models. Word and dialogue-state n-grams are used for building categorical understanding and dialogue models, respectively. Acoustic confidence scores are incorporated in the understanding formulation. Problems due to data sparseness and out-of-vocabulary words are discussed. The performance of the speech recognition and understanding language models are evaluated with the "Carmen Sandiego" multimodal computer game. Incorporating dialogue models reduces relative understanding error rate by 15-25%, while acoustic confidence scores achieve a further 10% error reduction for this computer gaming application.

Keywords

Natural language processing, speech understanding, n-gram models, dialogue modeling, acoustic confidence scores, language model adaptation.

EDICS: 1-LANG, 1-RECO

I. Introduction

Recent efforts in natural language understanding have focused on statistically-based approaches. Research is motivated by the increasing complexity of spoken dialogue systems, e.g., user-initiated dialogue, multiple actions and attributes per dialogue turn. Statistical-based approaches have been made possible by the availability of semantically annotated dialogue speech corpora and have mostly concentrated on the travel reservation application domain. Typical statistical understanding systems build statistical models using semantic labels (prior knowledge) and data-driven approaches. Models are bootstrapped from a semantically annotated text corpus. A typical statistical understanding system decodes incoming utterance in three stages: acoustic decoding, semantic parsing of recognizer output (rule-based mapping from text to semantic labels), and context-dependent semantic decoding (statistical mapping from semantic labels to actions and attributes) [8], [10]. In this paper, *actions* are defined as application-specific operations that are independent of the human-computer interface, e.g., input and presentation modalities. *Attributes* are parameters associated with a specific action; (some of) these parameters need to be instantiated to perform the action.

Breaking down the understanding problem into semantic parsing and semantic decoding is appropriate for certain tasks (e.g., travel reservations) where there are few actions and many attributes expressed with short low-perplexity phrase fragments. However, there are many applications, e.g., gaming [12], call-routing [5], where there are many actions with few attributes, and actions are often expressed with high-perplexity phrase fragments. For these applications the emphasis of the understanding system lays on building statistical mapping from user-input to actions. The understanding problem thus becomes mostly a categorical classification problem where the classes are the application-dependent actions. Once the action is recognized, the attributes associated with the action can be identified through semantic parsing.

In this paper, we concentrate on the problem of categorical classification of actions from speech input. Categorical speech understanding has received some attention in the literature especially for the problems of routing telephone calls to the appropriate destinations ("call-routing") [5], [3], [17], [2] and dialog act classification [24]. Call-routing tasks typically involve no spoken dialogue or a limited clarification sub-dialog with the system. A single utterance has to be classified to one of the routing destinations. In this paper, the problem of categorical understanding is investigated for user-initiated and mixed-initiative dialogues. As a result, dialogue context plays an important role in constraining the possible interpretation of user input and enhancing classification performance. Using dialogue models for understanding has also been proposed in [25], [24]. The problem of language modeling for dialogue systems is also investigated in this paper. Various algorithms for adapting language models to a specific dialogue context has been proposed in the literature [4], [23], [15], [16], [17], [21]. In this paper, the effect of a novel language adaptation algorithm is investigated on categorical speech understanding performance for spoken dialogue systems. Finally, a categorical understanding algorithm that utilizes acoustic confidence scores is proposed and evaluated (see [22] for incorporating acoustic confidence scores in a call-routing task).

The organization of the paper is as follows: First the dialogue state formalism is introduced in the context of a user-initiated dialogue. Next the understanding problem is posed in a maximum likelihood framework. In Section IV, the stochastic finite state machine representation of the language model and the novel adaptation algorithm are outlined. The understanding model is proposed in Section V and issues such as incorporating acoustic confidence scores and dealing with out of vocabulary words are discussed. A user (dialogue) model is introduced in Section VI. The algorithms are applied to a computer gaming application, the "Carmen Sandiego" task outlined in Section VII. Word accuracy and understanding accuracy results are presented in Section VIII. Extensions of user modeling to mixed-initiative tasks are discussed in Section IX.

II. Dialogue Flow Model

In this section, a formal representation of the dialogue flow of a general human-machine interaction with multimodal input and output is introduced. A user-initiated finite-state dialogue structure is assumed which is typical for gaming applications. The central notion of the dialogue flow model is the state S_t at turn t that is defined in terms of user input and system output. If W_t is the user input (e.g., speech transcription) to the application and P_t is the output in response to input W_t, then a typical transaction is

$$\ldots \underbrace{W_{t-1} \rightarrow P_{t-1}}_{S_{t-1}} \mapsto \underbrace{W_t \rightarrow P_t}_{S_t} \mapsto \underbrace{W_{t+1} \rightarrow P_{t+1}}_{S_{t+1}} \ldots \tag{1}$$

where $W \rightarrow P$ transitions are determined by the understanding system and dialogue manager, $P \mapsto W$ transitions are determined by the user, and S_t is the *dialogue state* at dialogue turn t. A total of K dialogue states are available $\{s_k,\ k = 1, \ldots, K\}$. In practice, a dialogue state can be associated with no action, e.g., extraneous speech input, or with multiple actions. For simplicity we assume that only one action is allowed per dialogue turn and thus dialogue action and dialogue state are used interchangeably in this paper (generalization of this framework to multiple actions per dialogue state is straightforward; see Section IX).

Given the equivalence between action and dialogue state, we define $\mathcal{I}_k$, $k = 1, \ldots, K$ to be the set of all user inputs that trigger state s_k. User input class $\mathcal{I}_k$ will be referred henceforth as a *dialogue state class*. Note that $\mathcal{I}_k$ contain semantically equivalent utterances W since they all trigger the same action s_k. The understanding problem is formulated here as determining the dialogue state S_t given user input W_t. This is equivalent to the categorical classification of W_t in one of the K classes $\{\mathcal{I}_k\}$. For spoken dialogue systems, at dialogue turn t, only the sequence of acoustic vectors O_t is observable, while the input speech transcription W_t and the dialogue state S_t are hidden variables[1]. Thus the understanding problem in the context of a spoken dialogue system, involves a

[1] For other input modalities, e.g., mouse and keyboard, user input W_t is directly observable.

joint search over the W_t, S_t state space.

Let us also define a *prompt-based* (or output-based) *class* $\mathcal{A}_k$ as the set of all user inputs that come as a response to a system output from $\mathcal{O}_k$, i.e., $\mathcal{A}_k = \{W_t : \exists\, P_{t-1} \in \mathcal{O}_k, P_{t-1} \mapsto W_t\}$. Note the difference between $\mathcal{A}_k$ and $\mathcal{I}_k = \{W_t : \exists\, P_t \in \mathcal{O}_k, W_t \to P_t\}$. One can guarantee that $\mathcal{I}_k$ contains semantically equivalent utterances (since they all trigger the same action s_k) but the same is not necessarily true for $\mathcal{A}_k$. Finally, note that the classification of user input into dialogue state class $\mathcal{I}_k$ requires solving the understanding problem, while mapping the user's input into the prompt based classes $\mathcal{A}_k$ is done automatically by the system (S_{t-1} is known at time t). As a result $\mathcal{A}_k$ can be used in an unsupervised language adaptation scheme as proposed in section IV-A.

III. The Categorical Understanding Problem

As discussed in Section II the understanding problem is defined here as determining the dialogue state[2] S_t given the speech input O_t. The maximum likelihood (ML) approach to this problem is based on maximizing the joint posterior probability[3]

$$\max_{S_t, W_t} P(S_t, W_t | O_t, S_1..S_{t-1}) \tag{2}$$

where S_t is the dialogue state, W_t is the user input (mapped to a sequence of words) and O_t is the acoustic observation sequence at dialogue turn t. This ML problem is equivalent to

$$\max_{S_t, W_t} P(O_t|W_t) P(W_t|S_1..S_t) P(S_t|S_1..S_{t-1}) \tag{3}$$

under the simplistic[4] assumption that the acoustic observations are independent of the dialogue state, i.e., $P(O_t|W_t S_1..S_t) \approx P(O_t|W_t)$. Eq. (3) suggests that acoustic decoding and understanding should be investigated as a single problem. Moreover, dialogue state-dependent language models and understanding models could potentially be merged into a single model that computes

[2] Recognizing the attributes associated with a dialogue state is an equally important part of the understanding problem, however, it is often trivially solved by rule-based parsing of the recognizer output.

[3] Similar results are obtained when starting from the ML formulation $\max_{S_t} P(S_t|W_t, O_t, S_1..S_{t-1})$ as shown in [26].

[4] Acoustics and specifically prosody carry significant semantic information [24].

$P(W_t|S_1..S_t)$. In practice, instead of jointly maximizing Eq. (3) with respect to W_t and S_t it is typical to first maximize the posterior probability with respect to W_t and then with respect to S_t, i.e.,

$$\hat{W}_t = \arg\max_{W_t} P(O_t|W_t)P(W_t|S_1..S_{t-1}) \tag{4}$$

$$\hat{S}_t = \arg\max_{S_t} P(\hat{W}_t|S_1..S_t)P(S_t|S_1..S_{t-1}) \tag{5}$$

where $P(W_t|S_1..S_t)$ was approximated by $P(W_t|S_1..S_{t-1})$ in the first equation since S_t is unknown at decoding time. The two step likelihood maximization although suboptimal is often used in practice because it decomposes the general understanding problem into two simpler, well-studied problems: standard acoustic decoding and understanding from transcription. Indeed, $\hat{W}_t$ is the solution to the decoding problem, i.e., maximizing the product of the acoustic and language likelihood scores[5], while $\hat{S}_t$ is the solution to the understanding problem given a transcription $\hat{W}_t$, i.e., maximizing the product of the understanding and dialogue likelihood scores.

In practice, the probabilities in Eqs. (4),(5) are estimated based on (imperfect) acoustic λ_A, language λ_L, understanding λ_U and dialogue λ_D models. Confidence scores can be attached to the various information streams (acoustic, language, understanding and dialogue) based on the "quality" of the corresponding model. These confidences are typically used to compute exponential weights that adjust the dynamic range of the information sources. These weights γ_A, γ_L, γ_U, γ_D, are task-dependent and can be time-varying, e.g., acoustic confidence scores can be computed at the word level and used to weight the language and understanding model probabilities[6]. Thus, we can rewrite the understanding problem as

$$\hat{W}_t = \arg\max_{W_t} P(O_t|W_t,\lambda_A)^{\gamma_A} P(W_t|S_{t-1},\lambda_L)^{\gamma_L} \tag{6}$$

[5] Note that the language score is conditioned on the dialogue state history, i.e., the language models used are dialogue-state dependent [21].

[6] The choice of γ_A and γ_L is part of the speech recognition decoding problem and beyond the scope of this paper. Typically γ_L/γ_A if referred as the "weight of the language model" and is determined empirically (see [6]).

$$\hat{S}_t = \arg\max_{S_t} P(\hat{W}_t|S_t, \lambda_U)^{\gamma_U} P(S_t|S_{t-1}, \lambda_D)^{\gamma_D} \tag{7}$$

provided that[7]: $P(W_t|S_1..S_{t-1}, \lambda_L) = P(W_t|S_{t-1}, \lambda_L)$, $P(W_t|S_1..S_t, \lambda_U) = P(W_t|S_t, \lambda_U)$, $P(S_t|S_1..S_{t-1}, \lambda_D) = P(S_t|S_{t-1}, \lambda_D)$.

The decoding problem of Eq. (6) is discussed in the next section. Specifically a novel way of computing $P(W_t|S_{t-1}, \lambda_L)$ is proposed, i.e., training state-dependent and state-adapted language models, and compared with state-independent language models $P(W_t|\lambda_L)$. We then address the understanding problem and propose simple Markovian models for the understanding λ_U and dialogue λ_D models. Furthermore, the incorporation of acoustic confidence scores in the exponential stream weights γ_U, γ_D is discussed.

IV. Language Modeling

Our approach to language modeling is based on the Variable Ngram Stochastic Automaton (VNSA) representation and learning algorithms first introduced in [20], [19]. The VNSA is a non-deterministic stochastic automaton that allows for parsing any possible sequence of words drawn from a given vocabulary V. In its simplest implementation the state q in the Stochastic Finite State Machine (SFSM) encapsulates the lexical (word sequence) history of a word sequence. Each state recognizes a symbol $w_i \in V$. The probability of going from state q_{i-1} to q_i (and recognize the symbol associated to q_i) is the state transition probability, $P(q_i|q_{i-1})$. Stochastic finite state machines represent in a compact way the probability distribution over all possible word sequences. The probability of a word sequence W can be associated to a state sequence $\xi_W^j = q_1, \ldots, q_N$ and to the probability $P(\xi_W^j)$. For a non-deterministic finite state machine the probability of W is then given by $P(W) = \sum_j P(\xi_W^j)$. 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.) [19].

[7] The Markovian assumption for the dialogue state sequence is supported from the data for the "Carmen Sandiego" task (see Section 5). However, one can argue that in practice $P(.|S_1..S_t) \approx P(.|S_t)$ for most dialogue interactions.

A. Language Model Adaptation

In spoken language system design, the state of the dialogue s_k is often used as a predictor of the most likely user response. For example, if in a particular state s_k the system asks a confirmation question (YES-NO) the most likely response will be in the YES-NO equivalent class. However, to achieve natural dialogue flow as in human-human communication the user should be allowed to move from one state to any state of the dialogue. To achieve this goal language models that are open in vocabulary for each state s_k are built. At the same time the language models are adapted for each stage of the dialogue based on the expected users' responses.

Without loss of generality, we assume that each user's response corresponds to a state of the dialogue model (see Section II). In this case, the entire transaction is associated to 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 $P(w_1, w_2, \ldots, w_N|s_k)$:

$$P(w_1, w_2, \ldots, w_N|s_k) = \prod_j P(w_j|w_1, w_2, \ldots, w_{j-1}; s_k) \tag{8}$$

In previous work, the dialogue model has been used to partition the whole set of utterances spoken in the dialogue sessions into subsets (first sub-problem) and then train standard n-gram language models (second sub-problem) [11], [23]. This way, the user can only utter words that he has previously spoken in a specific dialogue state. Such language model design does not fully support mixed-initiative dialogues. In other related work, the estimation problem is solved by linear interpolation [23] or maximum entropy models [15], [16], [17], speaker back-off models [1] or MAP training [4]. 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. In order to boost the expected probability of any event at state s_k we use the algorithm for stochastic finite state machine adaptation first proposed in [21], [18].

The set of all user's observed responses at a specific stage i of the dialogue is split into training $\mathcal{T}_k$

($\mathcal{T}_i \bigcap \mathcal{T}_j = \emptyset$ for each $i \neq j$), development ($\mathcal{B}_k$) and test ($\mathcal{E}_k$) sets. A context independent Variable Ngram Stochastic Automaton $\lambda^{\mathcal{T}}$ on the training set $\mathcal{T} = \bigcup_k \mathcal{T}_k$. While, $\lambda^{\mathcal{T}}$ has full coverage over all possible word sequences W at any state s_k, it does not provide a selective model for a given dialogue state prediction. Thus, we build the adapted language models λ_k^* as to maximize the stochastic separation from the generic model λ^T. The model λ_k^* is thus computed as the solution of the log likelihood maximization problem:

$$\lambda_k^* = \operatorname*{argmax}_{\lambda_k^A} \log P(\mathcal{B}_k|\lambda_k^A) \tag{9}$$

where the model λ_k^A is estimated as a linear interpolation of the language model λ^T and a state dependent model λ_k. The transition probabilities for the model λ_k^A are computed as follows:

$$P_k^A(q_j|q_{j-1}) = \alpha_k P^T(q_j|q_{j-1}) + (1-\alpha_k)P_k(q_j|q_{j-1}) \tag{10}$$

For each set $\mathcal{T}_k$ we run Viterbi training starting from the generic model λ^T and estimate the transition probabilities of the SFSM λ_k. In order to account for unseen transitions we smooth the transition probabilities with the standard discounting techniques discussed in [19]. The solution to Eq.(9) with respect to the parameters α_k cannot be given in an explicit form. Hence, we use a greedy algorithm over the development sets $\mathcal{B}_k$ to find the local optimum over a finite number of α_k values. In general there may not be enough data to have sufficient statistics from the training sets $\mathcal{T}_k$. In these cases we replace the Viterbi training estimates $P_k(q_j|q_{j-1})$ with prior distributions.

V. Understanding Model

A typical statistical approach to the problem of estimating $P(W_t|S_t, \lambda_U)$ involves constructing a model L_k from each one of the state classes $\mathcal{I}_k$, $k = 1, \ldots, K$. The understanding model is then defined as $\lambda_U = \{L_k, k = 1, \ldots, K\}$ and

$$P(W_t|S_t = s_k, \lambda_U) = P(W_t|L_k). \tag{11}$$

For spoken dialogue systems, user input W_t is a text string and $\mathcal{I}_k$ is a set of transcribed sentences. A Markovian model for $\mathcal{I}_k$ is the variable n-gram stochastic automaton [19]. Recall that n-grams have been used extensively for language modeling and well-established learning algorithms exist in the literature. If L_k is the n-gram statistical model trained from $\mathcal{I}_k$ and the input utterance $W_t = w_1 w_2 .. w_N$, $W_t \in \mathcal{I}_k$ then the computation of the word sequence probability is done as follows:

$$P(W_t | L_k) \approx \prod_n P(w_n | w_1 .. w_{n-1}, L_k). \tag{12}$$

In many cases, the training corpus has small amounts of data which leads to poor estimates of $P(W_t|L_k)$. Techniques for dealing with sparse training data can be borrowed from the language modeling literature, e.g., introduction of semantic classes (such as cities, dates, digits) back-off techniques etc. A formal evaluation of smoothing techniques in λ_U training in terms of understanding performance remains to be done. Another issue that stems from data sparseness is the high confusability of utterances that lay on understanding class boundaries. In such cases, discriminative approaches could be used to train the understanding models.

A. Out-of-vocabulary Words

Out-of-vocabulary (OOV) words in the transcribed input string W_t is a common problem for large vocabulary systems. Moreover, OOV words might appear even when W_t is the output of an automatic speech recognizer because the vocabulary V_k of understanding model L_k is a subset of the vocabulary used for recognition. To deal with OOV words a simple garbage node is introduced in the understanding model finite state machine with insertion penalty c_{oov}. Specifically, if the input utterance $W_t = w_1 w_2 .. w_N$ is represented as $\bigoplus_n w_n$ then

$$P(W_t | L_k) \approx P(\bigoplus_{n: w_n \in V_k} w_n | L_k) \; [(c_{oov})^{\sum_n \delta(w_n \notin V_k)}] \tag{13}$$

where V_k is the vocabulary drawn from the $\mathcal{I}_k$ training set, $w_n \in V_k$ signifies that word w_n is in V_k, $\delta(w_n \notin V_k) = 1$ for out of vocabulary (OOV) word (else 0) and c_{oov} is a task dependent constant penalty for deletion of OOV words from input W_t.

To avoid deletions from the input utterances that could have deleterious effects when computing n-gram probabilities, OOV words can be modeled explicitly in L_k by labeling a subset of the words in the training set of each class as "OOV". For example a round-robin (a.k.a. hold-one-out) technique can be used for the sentences in each training set $\mathcal{I}_k$ and words in the held-out utterances that don't belong in the state's training dictionary are labeled as OOV. In practice, we have observed improved performance when OOV labels are explicitly modeled as in Eq. (13) rather than included in $\mathcal{I}_k$ training. Thus for our experiments a positive OOV insertion penalty c_{oov} in computed from held-out data. Understanding accuracy as a function of c_{oov} is shown in Fig. 1 for two understanding tasks (see also Results section).

B. Incorporating Acoustic Confidence Scores

Acoustic confidence scores can be generated for each decoded word in an utterance by combining the likelihood ratio scores for the acoustic sub-word units (phones) that make up that word. The likelihood ratio score for each sub-word unit is computed as the ratio of the null (speech recognition) and alternate hypothesis (utterance verification) model score. In [22], the alternate hypothesis model comprises of an "anti-model" (one per sub-word unit) and a generic "background" model. The "anti-model" for sub-word unit j is trained from examples of other sub-word units that are easily confusable with j. The "background" model is a single-state hidden Markov model that provides a broad representation of the feature space. The alternate hypothesis model likelihood is the linear combination of the "anti-model" and "background" model likelihoods. The acoustic confidence scores AC are normalized, i.e., $AC \in [0, 1]$. See [22] for more details on the computation of acoustic confidence scores and typical normalized acoustic confidence scores histograms for a call-routing application.

The acoustic confidence scores can be used to scale the dynamic range of the understanding model probabilities for each word in a sentence or equivalently scale the log probabilities. The argument is that low confidence words should be weighted less in the understanding decision. Specifically, assuming for simplicity that there are no OOV words, the understanding model probabilities can be expressed as

$$P(W_t|L_k)^{\gamma_U} \approx \prod_n P(w_n|w_1..w_{n-1}, L_k)^{\frac{c+1}{c+AC(w_n)}} \tag{14}$$

where c is a smoothing constant experimentally determined from held out data. Note that acoustic confidence scores can also be incorporated in the language model as stream weights γ_L or explicitly as word tags (see [22]).

VI. Dialogue Model

In this section, a statistical dialogue model for computing the dialogue state transition probability, i.e., $P(S_t|S_1..S_{t-1})$ is defined. A simple state n-gram model is used for this purpose. Note that according to the definition of a dialogue state given in Eq. (1) (user-initiated dialogue) a model of the sequence of states $S_1..S_t$ is actually a *user model*, since the user input fully determines dialogue state transitions.

In practice, we have found that for user-initiated dialogues a state bigram $P(S_t|S_{t-1})$ models well the short-time dialogue state dependencies. For example, for the "Carmen Sandiego" task (see next section) the n-gram perplexities of the finite state dialogue model were: state unigram 12.2, state bigram 4.0, state trigram 3.9, state fourgram 4.0 (total of 15 dialogue states, 6039 dialogue turns for training and 2050 for testing). For more complex tasks, that take multiple turns to complete, longer time dependencies are present and higher order dialogue models might be needed (see Discussion section).

VII. Task Description

The algorithms proposed above have been applied to a gaming application, the "Carmen Sandiego" task. In [12], [9], data have been collected and analyzed from 160 children ages 8-14 using voice to interact with the popular computer game "Where in the U.S.A. is Carmen Sandiego?" by Brøderbund Software. A Wizard of Oz scenario was used in the collection. This game was selected because of its richness in dialogue subtasks including: navigation and multiple queries (talking to cartoon characters on the game screen), database entry (filling the suspect's profile), and database search (look up clues in a geographical database). There were no changes to the original (keyboard and mouse-based) game except for allowing the speech input modality and adding spoken clarification and error messages. A brief description of the game follows.

To successfully complete the game, i.e., arrest the appropriate suspect, two subtasks had to be completed: (i) determine the physical characteristics of the suspect and issue an arrest warrant, and (ii) track the suspect's whereabouts in one of fifty U.S. states. In order to complete these two tasks the player can talk to animated characters on the game screen and ask them for clues. The clues can be correlated with information in a geographical database. Information can be obtained from the database either by (single or multiple word) search or by stepping through a hierarchical database structure. The player needs to create the suspect's profile (five features, each with two to five pre-defined attributes) and issue a warrant when all fields are filled. The player has to travel (sequentially) through five U.S. states tracking the suspect, identify him (using the profile information) from among the cartoon characters in the screen and arrest him.

Using the dialogue flow notation introduced in Section II, we have defined *15 dialogue states* for this application. For a better understanding of the semantic description of the dialogue states see [12]. All collected utterances W_t have been manually assigned to the correct state s_k that they trigger according to the definition of $\mathcal{I}_k$. The training set consists of 6039 utterances collected from 51 speakers and the test set consists of 2050 utterances from 20 speakers. A typical dialogue between the user and the system is shown in Table I. Dialogue state labels are shown on the right and attributes are underlined. Note that each user input-output pair is assigned to a dialogue state which is consistent with the formalism introduced in Section II. Four of the fifteen states (or actions) have attributes, e.g., in Table I the "GoToState" dialogue state has "Indiana" as an attribute. The understanding problem is defined here as determining the dialogue state label and attribute(s), e.g., “GoToState” and “Indiana”, given the recognized user input $\hat{W}_t$.

VIII. Experimental Results

In this section, we report results from language adaptation and speech understanding experiments using various configurations of language, understanding and dialogue models. Performance is measured in terms of word accuracy (WACC), dialogue state label accuracy (LACC) and dialogue state attribute accuracy (AACC). LACC is defined as the number of correctly classified state labels over the total number of state labels. Similarly, AACC is defined as the number of correctly identified state attributes over the total number of attributes. A deterministic mapping from words to attributes is used for attribute recognition. Context independent hidden Markov Models (HMMs) using three states and sixteen Gaussians to model each phone were trained from the acoustic data and used in all the experiments described in this paper. The training set consists of 6039 utterances (30276 words, 102 dialogues) and the test set of 2050 utterances (10258 words, 37 dialogues). In Section VIII-A, the effect of language adaptation on WACC and LACC is investigated. The best performing language model is selected (state-dependent trigram) and used for speech understanding experiments in Section VIII-B.

A. Language Adaptation Experiments

VNSAs were used for language modeling with order $N = 1, 2, 3$; specifically, word bigram and trigram, and phrase unigram, bigram and trigram. Finally, word trigrams were trained from $\mathcal{I}_k$ and used as understanding models L_k ($c_{oov} = 10$). Results are reported for speech recognition (labeled “word accuracy”), and sentence classification. The baseline system is based on the context independent language model $\lambda^{\mathcal{T}}$. Two algorithms were used for language adaptation. The first one used data only from $\mathcal{T}_k$ to construct prompt-based language models λ_k (referred to as “state-dependent”). The second algorithm used *all training data* to estimate λ_k^* (referred to as “state-adapted”). The speech recognition results are shown in Table II. In Table II, we compare the word accuracy for the two adaptation schemes “state-dependent” and “state-adapted” for a word and phrase bigram language model. The open-vocabulary model λ_k^* gives 3-10% error rate reduction for the most populated dialogue state classes. Word accuracy has increased due to better probability estimates (all data is used for adaptation) and larger language coverage across different states of the dialogue.

In Table III, the speech understanding results are shown for the same task. The understanding task has been carried over the pre-defined fifteen dialogue states according to the model delivered by Eq. (13). A word bigram understanding model and a unigram dialogue model was used in this experiment (see next section for different understanding and dialogue model configurations). As shown in Table III, the dialogue state label accuracy (LACC) improves by about 2% (absolute) over the baseline when using adapted language models. The relative understanding error rate reduction is 10-15%, which is significantly larger than the relative word error reduction (about 5% in Table II). Similar improvements can be achieved by using state-dependent or state-adapted language models. Finally note that LACC for correct transcription is 93.9% which is significantly better than LACC for the recognizer output.

B. Understanding Model Experiments

In this section, the performance of the speech understanding algorithms are evaluated for various understanding and dialogue model orders. A state-adapted trigram language models was used for all experiments reported in this section. The language model was selected to maximize word accuracy. The word accuracy (WACC) on the 2050 sentence recognition task [8] was 78.0%. Next, results are presented for attribute recognition (AACC), state label understanding (LACC) as a function of the out of vocabulary penalty (c_{oov}) and model order, and LACC when acoustic confidence scores are

[8] The relatively low word accuracy is due to two reasons: the speech data was collected in a spontaneous child-machine interaction setting and the speakers were children ages 8-14. For a review of approaches to improving speech recognition accuracy for children speakers see [13], [14].

incorporated in the understanding algorithm.

Attribute recognition is performed by mapping words or combination of words to attributes, e.g., Indiana → state. The mapping is deterministic and the rules are context-free for this simple application. Thus the attribute accuracy (AACC) is directly comparable to word accuracy (WACC). Indeed, attribute recognition accuracy was at 73%, while WACC was at 78% when using a state-adapted trigram language model. Specifically, four states had attributes associated with them; AACC was: 59% for "Find", 79% for "GoToState", 81% for "Navigate" and 70% for "EnterFeature". The close relationship between AACC and WACC scores should be expected because most attributes are one or two words long (the average word length of an attribute is 1.4). Note that unlike LACC, AACC is not affected by the choice of the understanding or dialogue model.

Word unigram, bigram and trigram models were trained for each of the fifteen dialogue state class $\mathcal{I}_k$ and used as understanding models L_k. The test set perplexity was very different for each of the understanding models L_k, e.g., for word trigram models the perplexity ranged from 1.4 to 12.6. State unigram and bigrams were used for dialogue modeling. In Fig. 1, LACC for correct and recognized transcriptions is shown as a function of c_{oov} for the "Carmen Sandiego" and HMIHY tasks (see [5] for HMIHY task description and state of the art performance). It is interesting to note that for both tasks choosing a very large c_{oov} penalty gives close to best results. However, small OOV penalty values result to large error rates. The OOV word insertion penalty in Eq. (13) was set to $c_{oov} = 10$ for the rest of the experiments in this section.

LACC from correct and recognized transcriptions are shown in Table 1 and Table 2, respectively, for different understanding and dialogue model order. The more complex understanding models (bigram or trigram) perform significantly better (about 10% relative error rate reduction) than the unigram model in the presence of recognition errors. However, when the correct transcriptions are used for understanding the simple unigram model performs almost as well. In both cases, the difference in performance between bigram and trigram understanding models is negligible for this task. By incorporating the dialogue model in the understanding process performance improves significantly. Overall, an additional 15% relative error rate reduction is achieved by incorporating a dialogue model (25% for correct transcriptions). LACC improvements are significant both when adding a state unigram model and when upgrading to a state bigram model.

Finally, in Fig 2 results are shown when incorporating the acoustic confidence scores in the understanding model. Specifically, the label understanding accuracy (LACC) is shown as a function of the smoothing parameter c in Eq. (14). Unigram understanding and dialogue models were used for understanding and a (dialogue state-independent) bigram language model was used for recognition. About a 10% relative label understanding error reduction is achieved when incorporating acoustic confidence scores (LACC: 82.1% for $c = 0.2$ vs LACC: 80.3% baseline performance for $c = \infty$). Overall, the results are comparable with those obtained with the understanding algorithms described in [22] for this task.

IX. Discussion

Language Modeling: One interesting observation is that in our experiments the state-adapted language models outperform the state-dependent ones in terms of word accuracy (see Table III). It should be expected that the WACC improvement obtained by using state-adapted (vs state-dependent) language models depends on the amount of training data. The trend however is reversed for understanding accuracy, i.e., state-dependent language models perform better (see Table V). It is not clear if this is a general trend or it is specific to this task and understanding module. At this time the authors know of no other published results that verify this trend.

Confidence Scores: The proposed method for incorporating confidence scores into the understanding process has not been equally successful when using bigram or trigram understanding models. Alternative formulation for Eq.(14) have to be investigated for higher-order n-gram understanding models. Acoustic confidence scores that are computed both for the current word w_n and the word history w_{n-1} might be more appropriate weights in the exponent of Eq. (14) for a bigram understanding model.

Multiple tags per Utterance: In Sections II-III, a single action was associated with each dialogue turn. This assumption is not critical for formulating Eqs. (3)-(7). However allowing multiple actions per turn does complicates the solution of the understanding from transcription problem in Eq. (7). Dynamic programing has to be used to align the optimal sequence of words in $\hat{W}_t$ (produced by the recognizer) to the (unknown) sequence of actions. The proposed understanding and dialogue models in a maximum likelihood framework can be easily applied to Viterbi decoding. More research is needed to work out the details of integrating Viterbi decoding in the speech understanding problem (see [26] for one approach to this problem).

Dialogue Model–System Initiative: The dialogue flow model introduced in Section II is motivated by a spoken dialogue system where the user has control of the conversation (*user-initiative*). For *system-initiated* dialogue, all actions are determined by the dialogue manager based on user input. In addition, user input is assumed to be a direct reply to the system's requests. Thus, the appropriate dialogue flow model for fully system initiated dialogue maps (system response, user input) pairs to actions as follows:

$$\ldots W_{t-1} \rightarrow \underbrace{P_{t-1} \mapsto W_t}_{S_{t-1,t}} \rightarrow \underbrace{P_t \mapsto W_{t+1}}_{S_{t,t+1}} \rightarrow P_{t+1} \ldots \qquad (15)$$

The sequence of actions ...$S_{t-1,t}$, $S_{t,t+1}$... are determined by the system's dialogue manager. Therefore for system-initiative dialogues the speech understanding problem degenerates to simple attribute recognition and is of little interest. In general, the system and the user share dialogue initiative. Next we discuss a generalization of the proposed framework to such *mixed-initiative* systems.

Dialogue Model–Mixed Initiative: A brief commentary on the order and nature of the dialogue model introduced in Section II follows. The "Carmen Sandiego" task consists of two main subtasks: filling the profile information and identifying the suspects where-abouts. A few tens of dialogue turns are necessary for completing each of these tasks. Such long-term dependencies between dialogue states could be captured by high-order dialogue models. However, as discussed in Section II, almost no reduction in perplexity is achieved by using a trigram (vs bigram) dialogue model. This is possibly due to inadequate amount of training date, variation in the sequence of actions required to complete each task, and task interleaving (working on both tasks in parallel, see [12]). It is possible that a more elaborate dialogue model can capture long-term dependencies more efficiently and achieve significant reduction in perplexity.

It was discussed in Section II, that for user-initiated dialogue the dialogue model is identical to a stereo-typical user model, i.e., a model predicting the user intentions based on current dialogue state information. However, for many spoken dialogue applications the system has most or all of the dialogue initiative. It is important to note, that *even for cases of mixed initiative the proposed dialogue model can improve understanding accuracy.* However, a generalization of the dialogue model is needed to capture the fact that the user and the system share initiative. As a result of mixed initiative user input W_t and system output P_t *don't necessarily belong to the same dialogue state* and the dialogue flow has to be generalized to

$$\ldots \underbrace{W_{t-1}}_{S_{t-1}^W} \rightarrow \underbrace{P_{t-1}}_{S_{t-1}^P} \mapsto \underbrace{W_t}_{S_t^W} \rightarrow \underbrace{P_t}_{S_t^P} \mapsto \underbrace{W_{t+1}}_{S_{t+1}^W} \rightarrow \underbrace{P_{t+1}}_{S_{t+1}^P} \ldots \tag{16}$$

where user input W_t and system output P_t utterances are classified to S_t^W, S_t^P actions, respectively. The dialogue model now computes the probability of user input W_t belonging to action S_t^W given that P_{t-1} belongs to S_{t-1}^P and W_{t-1} to S_{t-1}^W, i.e., $P(S_t^W|S_{t-1}^P, S_{t-1}^W)$. The increased complexity of the dialogue model is the price that has to be paid for having mixed-initiative dialogue. For cases where the dialogue manager is a set of deterministic rules that map from S_t^W to S_t^P a second-order model $P(S_t^W|S_{t-1}^W)$ should still provide adequate dialogue modeling.

Note that the main goal for introducing the dialogue flow formalism and the statistical dialogue models is to *improve understanding performance.* A statistical dialogue model based on Eq. (16) might also be useful for dialogue management. Most practical dialogue management modules consist of a set of deterministic rules and are very different from the simple statistical models proposed above. Having noted the differences between a stereotypical user model and the dialogue management module it should also be said that a good dialogue manager should try and closely follow the intentions of a typical user. Therefore the user model $P(S_t^W|S_{t-1}^W)$ should also be incorporated in the dialogue manager. However, this is beyond the scope of this paper[9].

X. Conclusions

A categorical classification approach was introduced for the problem of speech understanding in dialogue systems. A maximum-likelihood formulation of this problem was proposed in Eq. (3) as a two-dimensional decoding problem. The formulation suggests a unifying approach to language and understanding modeling. State-dependent and state-adapted language models were shown to significantly improve the speech recognition and understanding performance. Language adaptation was used to alleviate the problems of data sparseness for stochastic language modeling in the presence of data fragmentation. Language modeling techniques were successfully applied to the problem of training categorical understanding models and shown to provide results similar to fragment-based understanding models for certain tasks. Significant improvement in understanding accuracy was achieved by incorporating dialogue models and acoustic confidence scores in the statistical formulation of the understanding problem. Overall, relative speech understanding was improved by 5-15% by using language model adaptation, 15-25% from dialogue modeling and by 10% from incorporating acoustic confidence scores.

Acknowledgments: Most of this work was performed while A. Potamianos and S. Narayanan were with AT&T Labs–Research, Florham Park, NJ. The authors would like to express their sincere appreciation to Dr. Jerry Wright for providing baseline results of the “Carmen Sandiego” task using the HMIHY understanding system, to Dr. Rick Rose for his help with the utterance

[9] Such a unified framework is independently proposed and quantitatively explored in [26].

verification algorithms, and to the anonymous reviewers for many useful comments.

References

[1] S. Besling and H. Meier, "Language Model Speaker Adaptation," Proc. Eurospeech, pp. 1755-1758, Madrid, 1995.

[2] J. Bellegarda and K. Silverman, "Toward unconstrained command and control: Data-driven semantic inference," Proc. ICSLP, Beijing, China, 2000.

[3] B. Carpenter and J. Chu-Carroll, "Natural Language Call Routing: A Robust, Self-Organizing Approach," Proc. ICSLP, Sydney, Australia, 1998.

[4] M. Federico, "Bayesian Estimation Methods for N-gram Language Model Adaptation," Proc. ICSLP, pp.240-243, Philadelphia, 1996.

[5] A. L. Gorin, G. Riccardi and J. H Wright, "How May I Help You?", Speech Communication, vol. 23, pp. 113-127, 1997.

[6] X. Huang, A. Acero and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, p. 610, Prentice Hall, New Jersey, 2001.

[7] R. Kneser, J. Peters and D. Klakow, "Language Model Adaptation Using Dynamic Marginals," Proc. Eurospeech, pp. 1971-1974, Rhodes, Greece, Sep. 1997.

[8] W. Minker, "Stochastically-Based Natural Language Understanding Across Tasks and Languages," Proc EUROSPEECH, Rhodes, Greece, Sep. 1997.

[9] S. Narayanan and A. Potamianos, "Creating Conversational Interfaces for Children," IEEE Trans. On Speech and Audio Proc., vol.10, no.2, pp. 65-77, Feb. 2002.

[10] R. Pieraccini and E. Levin, "A spontaneous-speech understanding system for database query applications," Proc. ESCA Workshop on Spoken Dialog Systems - Theories and Applications, 1995.

[11] C. Popovici and P. Baggia, "Specialized Language Models using Dialog Predictions," Proc. ICASSP, pp. 815-818, Munich, 1997.

[12] A. Potamianos and S. Narayanan, "Spoken dialogue Systems for Children," Proc. ICASSP, pp. 197-201, Seattle, 1998.

[13] A. Potamianos, S. Narayanan, and S. Lee, "Automatic speech recognition for children," Proc. EUROSPEECH, Greece, pp. 2371–2374, Sep. 1997.

[14] A. Potamianos and S. Narayanan, "Robust Recognition of Children's Speech," IEEE Trans. On Speech and Audio Proc., to appear, 2003.

[15] P. S. Rao, M. D. Monkowski and S. Roukos, "Language Model Adaptation via Minimum Discrimination Information," Proc. ICASSP, pp. 161-164, Detroit, 1995.

[16] P.S. Rao, S. Dharaniprada and S.Roukos, "MDI Adaptation of Language Models Across Corpora," Proc. Eurospeech, pp. 1979-1982, Rhodes, Greece, Sep. 1997.

[17] W. Reichl, "Language Model Adaptation Using Minimum Discrimination Information," Proc. Eurospeech, Budapest, Hungary, Sep. 1999.

[18] G. Riccardi and A.L. Gorin, "Stochastic Language Adaptation over Time and State in Natural Spoken Dialog Systems," *Trans. on Speech and Audio Processing*, vol. 8, Jan 2000.

[19] G. Riccardi, R. Pieraccini and E. Bocchieri, "Stochastic Automata for Language Modeling," Computer Speech and Language, vol. 10(4), pp. 265-293, 1996.

[20] G. Riccardi, E. Bocchieri and R. Pieraccini, "Non Deterministic Stochastic Language Models for Speech Recognition," Proc. ICASSP, pp. 247-250, Detroit, 1995.

[21] G. Riccardi, A. Potamianos and S. Narayanan, "Language Model Adaptation for Spoken Dialogue Systems," Proc. ICSLP, Sydney, Australia, Nov. 1998.

[22] R. Rose, H. Yao, G. Riccardi and J. Wright, "Integration of utterance verification with statistical language modeling and spoken language understanding," Proc. ICASSP, Seattle, 1998.

[23] H. Sakamoto and S. Matsunaga, "Continuous Speech Recognition using Dialog-Conditioned Stochastic Language Model," Proc. ICSLP, pp. 841-844, Yokohama, 1994.

[24] A. Stolcke et al, "Dialog Act Modeling for Conversational Speech," AAAI Spring Symposium, Stanford, California, Mar. 1998.

[25] P. Taylor et. al., "Using Prosodic Information to Constrain Language Models for Spoken Dialogue," Proc. ICSLP, pp. 216-219, Philadelphia, 1996.

[26] S. Young, "The Statistical Approach to the Design of Spoken Dialogue Systems," Technical Report CUED/F-INFENG/TR.433, Cambridge University Engineering Dept., Cambridge, England, Sept. 2002

List of Figures

List of Tables

User input/System output	Dialogue State
W_{t-3}: Tell me about the suspect? P_{t-3}: She is neither long- nor short-legged	S_{t-3}: TellmeAbout
W_{t-2}: Her height is average P_{t-2}: ... [updating suspect's drawing]	S_{t-2}: EnterFeature
W_{t-1}: Where did the suspect go? P_{t-1}: She is picking peonies in Bloomington	S_{t-1}: WhereDid
W_t: Go to Indiana P_t: ... [travel theme]	S_t: GoToState

TABLE I

Typical user-system interaction in the “Carmen Sandiego” task.

recogn. grammar	baseline	state-dependent	state-adapted
bigram	73.9%	74.8%	74.7%
phrase-bigram	76.2%	76.8%	77.0%

TABLE II

Word Accuracy (WACC) for baseline and adapted language models.

recogn. grammar	baseline	state-dependent	state-adapted
bigram	81.4%	83.2%	82.9%
phrase-bigram	84.1%	86.7%	86.1%
correct transc.	93.9%		

TABLE III

Dialogue state label accuracy (LACC) for baseline and adapted language models (λ_U is a bigram and λ_D is a unigram model).

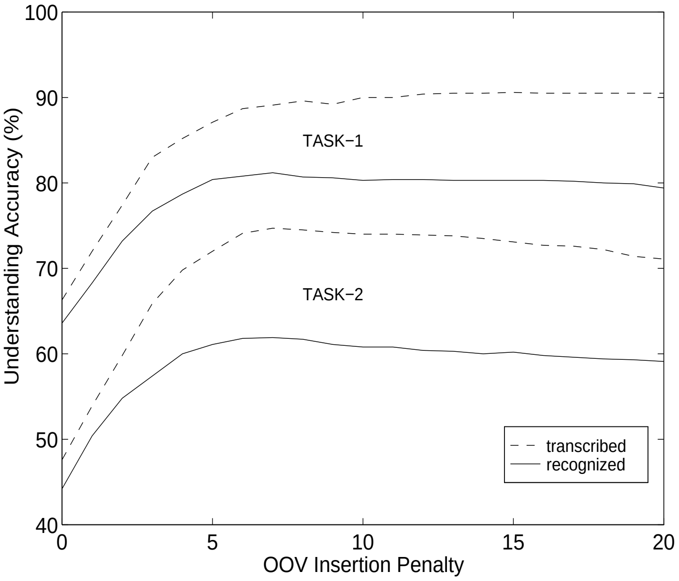

Fig. 1. Dialogue state label understanding accuracy (LACC) as a function of the OOV insertion penalty for two classification tasks (TASK1 = 'Carmen Sandiego', TASK2 = 'HMIHY') for transcribed and recognized utterances.

	Dialogue Model		
Understanding Model	none	unigram	bigram
unigram	91.8%	93.2%	94.1%
bigram	92.6%	93.2%	94.4%
trigram	92.4%	93.6%	94.3%

TABLE IV

Dialogue state label understanding accuracy (LACC) from correct transcriptions.

	Dialogue Model		
Understanding Model	none	unigram	bigram
unigram	81.5%	82.6%	84.8%
bigram	84.3%	84.0%	86.3%
trigram	84.4%	84.7%	86.3%

TABLE V

Dialogue label understanding accuracy (LACC) from recognized transcriptions

(78% word accuracy, λ_L is a state-dependent trigram language model).

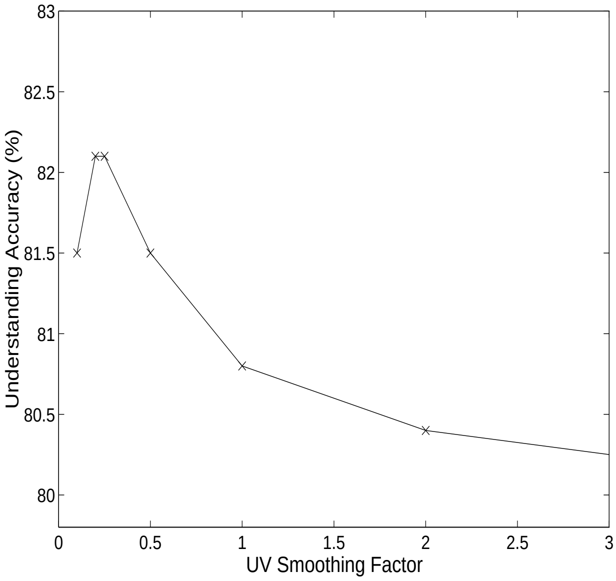

Fig. 2. Dialogue state label understanding accuracy (LACC) as a function of the acoustic confidence score smoothing factor c.