LANGUAGE MODEL ADAPTATION FOR SPOKEN LANGUAGE SYSTEMS

Giuseppe Riccardi, Alexandros Potamianos and Shrikanth Narayanan

AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932-0971, USA
{dsp3,potam,shri}@research.att.com

1. ABSTRACT

In a human-machine interaction (*dialog*) the statistical language variations are large among different stages of the dialog and across different speakers. Moreover, spoken dialog systems require extensive training data for training adaptive language models. In this paper we address the problem of open-vocabulary language models allowing the user for any possible response at each stage of the dialog. We propose a novel off-line adaptation of stochastic language models effective for their generalization (open-vocabulary) and selective (dialog context) properties. We outline the integration of the finite state dialog model and the language model adaptation algorithm. The performance of the speech recognition and understanding language models are evaluated with the *Carmen Sandiego* multimodal computer game. The new language models give an overall understanding error rate reduction of 44% over the baseline system.

2. INTRODUCTION

In the standard speech recognition paradigm, language models exploit the lexical context statistics (word tuples) observed in a training set to predict word sequences probabilities on a held-out set (test set). In the last decade, this has been the framework for many DARPA projects (e.g. ATIS, Wall Street Journal, etc.) that did *not* consider directly the statistical language variation in a human-machine interaction. In contrast with this scenario, spoken dialog systems for restricted domains provide a negotiation-oriented approach to task automation (e.g., flight/train travel planning, automated call routing, computer games etc.) [1, 2, 3, 10]. 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 completing successfully the task successfully.[1] Learning language models that adapt to different events along a spoken dialog session is tightly coupled with the state sequence associated to the human-machine interaction. Without loss of generality, we can assume that each user's response corresponds to a state of the dialog model. 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 $P(w_1, w_2, \ldots, w_N|s_k)$:

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

The computation of the probability $P(w_j|w_1, w_2, \ldots, w_{j-1}; s_k)$ can be decomposed into two subproblems. The first addresses the problem of predicting the word sequence probability computation given the state s_k. The second involves the estimation of $P(w_j|w_1, w_2, \ldots, w_{j-1}; s_k)$. In previous research reports, the dialog model has been used to partition the whole set of utterances spoken in the dialog sessions into subsets (first sub-problem) and then train standard n-gram language models (second sub-problem) [2, 4]. This way, the user can only utter words that he has previously spoken in a specific dialog state. Such language model design does not allow for on-line error recovery from speech understanding or dialog prediction errors. Thus, the main disadvantages of this approach 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 [4] or maximum entropy models [7], speaker backoff models [6] 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. In order to boost the expected probability of any event at state s_k we propose a novel algorithm for stochastic finite state machine adaptation. In the following section we outline the stochastic finite state machine representation of the language model and the novel adaptation algorithm. Then, we describe the system components (understanding and dialog model) as applied to a computer game application. In the last section we discuss the performance of the novel adaptation paradigm along with the speech recognition and understanding evaluations.

[1] In this paper we will perform experiments with an off-line adaptation scheme, while the algorithms proposed are applicable in an on-line scheme.

3. LANGUAGE MODELING

Our approach to language modeling is based on the Variable Ngram Stochastic Automaton (VNSA) representation and learning algorithms first introduced in [8, 9]. 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.) [9].

3.1. Language Model Adaptation

In spoken language system design, the state of the dialog s_k is used as predictor for the most likely user response. For example, if in a particular state s_k the computer asks a confirmation question (YES-NO) the most likely response will be in the YES-NO equivalent class. However, due to dialog model error predictions and to speech understanding errors, we want to let the user move from one state to any state of the dialog. We achieve this goal by building language models that are open in vocabulary for each state s_k. At the same time we adapt language models for each stage on the expected users' responses.

The set of all user's observed responses at a specific stage i of the dialog is split into training $\mathcal{T}_k$ ($\bigcap_k \mathcal{T}_k = \emptyset$), development ($\mathcal{B}_k$) and test ($\mathcal{E}_k$) sets. We train 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 dialog 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:

$$\lambda_k^* = \underset{\lambda_k^A}{\operatorname{argmax}} \log P(\mathcal{B}_k|\lambda_k^A) \qquad (2)$$

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. 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 discount techniques discussed in [9]. The transition probabilities for the model λ_k^A are then computed as follows:

$$P_k^A(q_j|q_{j-1}) = \alpha_k P^T(q_j|q_{j-1}) + (1-\alpha_k)P_k(q_j|q_{j-1}) \qquad (3)$$

The solution to equation 2 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. The complete block diagram, describing the adaptation scenario and training algorithm steps is shown in Fig. 1.

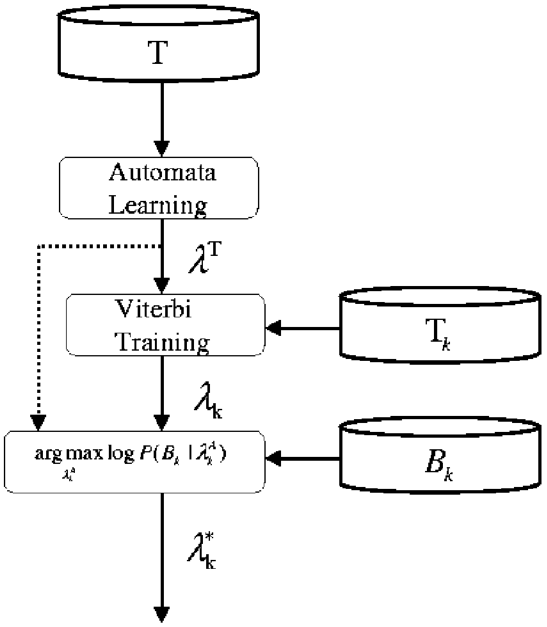

Figure 1: Block Diagram for the Language Model Adaptation Algorithm

4. DIALOG FLOW MODEL

In this section, a formal representation of the "dialog flow" of a general human-machine interaction with multimodal input and output is introduced. A user-initiated "dialog" is assumed which is typical for gaming applications. Further, it is assumed that the user input is interpreted by the application free of context. These assumptions simplify the discussion that follows but can be easily relaxed. The central notion of the dialog flow model is the state s_k that we define in terms of user input and system outputs. If i_t is a multimodal user input to the application and o_t is the multimodal output in response to input i_t, then a typical transaction is

$$\ldots \underbrace{i_{t-1} \to o_{t-1}}_{s(t-1)} \mapsto \underbrace{i_t \to o_t}_{s(t)} \mapsto \underbrace{i_{t+1} \to o_{t+1}}_{s(t+1)} \ldots \qquad (4)$$

where $s(t)$ is the *dialog state* at time t and $s(t) = s_k, \ k = 1, \ldots, K$. Further we define $\mathcal{I}_k$ to be the set of all user inputs that trigger state s_k, and $\mathcal{O}_k$ the set of system outputs that is produced when the system leaves state s_k, i.e., all system responses to input $i_t \in \mathcal{I}_k$. Under the assumption of context-free interpretation of user input: $i_t \in \mathcal{I}_k - o_t \in \mathcal{O}_k$. Thus, we formulate the understanding problem as the mapping from the input i_t into the dialog state s_k. User input class $\mathcal{I}_k$ will be referred henceforth as a *dialog state class*.

Let us now define a *prompt-based* (or output-based) *class* $\mathcal{A}_k = \mathcal{T}_k \cup \mathcal{B}_k \cup \mathcal{E}_k$ as the set of all user inputs that come as a response to a system output from $\mathcal{O}_k$, i.e., $\mathcal{A}_k = \{i_t : \exists \, o_{t-1} \in \mathcal{O}_k, o_{t-1} \mapsto i_t\}$. Note the difference between $\mathcal{A}_k$ and $\mathcal{I}_k = \{i_t : \exists \, o_t \in \mathcal{O}_k, i_t \to o_t\}$. 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 dialog state class $\mathcal{A}_k$ 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 $\mathcal{T}_k$, $\mathcal{B}_k$ and $\mathcal{E}_k$ can be used in an unsupervised language adaptation scheme as proposed in section 3.1.

5. UNDERSTANDING MODEL

As discussed in Section 4 the understanding problem is defined here as determining the dialog state $s(t)$ given the user input i_t. A typical statistical approach to this problem involves constructing a model L_k from the training set $\mathcal{I}_k$ using a maximum likelihood criterion and then determining the dialog state from the user input as:

$$\hat{k} = \operatorname*{argmax}_k P(L_k | i_t) = \operatorname*{argmax}_k \{P(i_t | L_k) P(L_k)\}.$$

If user input is given as a text string then $\mathcal{I}_k$ is a set of transcribed sentences. A simple statistical model for $\mathcal{I}_k$ is the computation of the word sequence probability corresponding to the user's utterance. For this purpose we have used the Variable Ngram Stochastic Automaton [9]. 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 $i_t = w_1 w_2 .. w_N$ is represented as $\bigoplus_n w_n$ then

$$P(L_k | i_t) \approx P(L_k | \bigoplus_{n : w_n \in L_k} w_n) \, [(c_{oov})^{\sum_n \delta(w_n \notin L_k)}] \quad (5)$$

where $w_n \in L_k$ signifies that word w_n is in vocabulary drawn from ${L_k}^2$, $\delta(w_n \notin L_k) = 1$ for out of vocabulary (OOV) word (else 0) and c_{oov} is a task dependent constant penalty for deletion of OOV words from input i_t. The selected dialog state $s_{\hat{k}}$ is the one that maximizes Eq.(5). The existence of OOV words in the transcribed input string i_t is common for closed vocabulary systems. Moreover, OOV words might appear even when i_t is the output of an automatic speech recognizer because in general the training corpus $\mathcal{I}_k$ for understanding model L_k is a subset of the language model training corpus $\mathcal{I}$, i.e., $\mathcal{I}_k \subset \mathcal{I}$. Note that more sophisticated strategies can be used for dealing with OOV words, e.g., by labeling some words in each training set $\mathcal{I}_k$ as OOV (using held out data) and by including the "OOV" label explicitly in L_k. Further, techniques for dealing with sparse data can be borrowed from the language modeling literature, e.g., introduction of concepts or word/phrase super-classes. A detailed discussion of the understanding model is beyond the scope of this paper.

	train	test		
utterance class	utter.	utter.	leng.	oov
prompt-`top`	4320	1499	5.1	2.5%
prompt-`search`	581	204	4.5	7.8%
prompt-`profile`	509	175	4.8	4.1%
prompt-`travel`	629	172	5.0	2.9%
all	6039	2050	5.0	2.0%

Table 1: Corpus statistics: total number of utterances (shown for both training and testing corpora), average sentence length, out-of-vocabulary rate for $\mathcal{A}_k$, where s_k is: `top` (navigation and query), `search` (database), `profile` (database entry) or `travel`.

	trigram grammar			
State	adapt-1		adapt-2	
	PP	WA	PP	WA
prompt-`top`	3.5	81.8%	4.0	82.4%
prompt-`search`	11.1	59.3%	14.6	57.9%
prompt-`profile`	8.2	67.1%	9.5	64.2%
prompt-`travel`	7.0	71.8%	4.0	74.5%
all		77.7%		78.0%

Table 2: Word accuracy (WA) and Perplexity (PP) per prompt-based dialog state for adapted trigram language models across different dialog states s_k and test sets $\mathcal{E}_k$.

6. EXPERIMENTAL RESULTS

The algorithms proposed above have been applied to the "Carmen Sandiego" task. In [1], 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. To successfully complete the game (i.e., arrest the appropriate suspect, two subtasks have to be completed), namely, determining the physical characteristics of the suspect to issue an arrest warrant and tracking the suspect's whereabouts

[2] Symbol L_k is used for both the Ngram model and the set of all utterances produced by this model.

	Word Accuracy		
ASR grammar	baseline	adapt-1	adapt-2
phrase-unigram	71.1%		73.1%
bigram	73.9%	74.8%	74.7%
phrase-bigram	76.2%		77.0%
trigram	77.8%	77.7%	78.0%
phrase-trigram	77.7%		78.1%

Table 3: Word Accuracy before and after language adaptation and for different language models.

(in one of fifty U.S. states). The game is rich in dialog subtasks including: navigation and multiple queries (talk to cartoon characters on the game screen), database entry (filling the suspects profile), and database search (look up clues in a geographical database). Using the dialog flow notation introduced in Section 4 we have defined four dialog states: `top` (navigation and queries), `profile` (database entry), `search` (database) and `travel` (to a U.S. state). For a better understanding of the semantic description of the dialog states see [1]. All collected utterances i_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 $\bigcup_k \mathcal{T}_k$ consists of 6039 utterances collected from 51 speakers and the test set consists of 2050 utterances from 20 speakers. In Table 1 the differences in out-of-vocabulary rate and test set perplexity is shown for the prompt based states. Note the uneven distribution and sparseness of both training and testing data.

Context independent hidden Markov Models (HMMs) using three states and sixteen Gaussians to model each phone were trained. VNSAs were used for language modeling with $N = 1, 2, 3$; 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 constructed prompt-based language models λ_k (referred to as "adapt-1"). The second algorithm used *all training data* to estimate λ_k^* (referred to as "adapt-2"). The speech recognition results are shown in Table 2 and 3. In Table 2, we compare the Word Accuracy for the two adaptation schemes "adapt-1" and "adapt-2" for a trigram language model. The open-vocabulary model λ_k^* gives 3-10% error rate reduction for the most populated dialog state classes. The `search` and `profile` dialog states are the most difficult test sets due to the high OOV rates (see Table 1) and lack of training/adaptation data. Word accuracy has increased due to better probability estimates (all data is used for adaptation) and larger language coverage across different states of the dialog. The speech understanding task has been carried over the pre-defined four dialog states according to the model delivered by equation 5. Understanding accuracy is computed as the number of correctly classified state labels over the total number of state labels. The overall ($\bigcup_k \mathcal{E}_k$) understanding accuracy from speech using a closed-vocabulary trigram language model ($P(L_k|i_t)$ in equation 5) is 91.8%, achieving a 44% error rate reduction over the baseline system (85.4%). The understanding performances per state s_k are not uniform and range between 4% (`top`) and 19% (`profile`). We expect, that a more accurate understanding model based on λ_k^* could outperform the model estimates based on the OOV factor (c_{oov}) factorization.

7. CONCLUSION

In this paper we have proposed a novel adaptation scheme for language modeling in the framework of spoken language system. Data sparseness is a serious problem for stochastic language modeling for large vocabulary systems. Moreover, the sparseness problem is even more acute in presence of data fragmentation, and that is the case of spoken dialog systems. For these reasons, a major challenge in stochastic language modeling is to exploit all available data while providing reliable probabilities conditioned on the dialog state. In this work, we have shown, that our adaptation scheme is effective in delivering statistically reliable probability estimates and increasing the language coverage at any state s_k.

8. REFERENCES

[1] A. Potamianos and S. Narayanan, *Spoken dialog Systems for Children*, Proc. ICASSP, pp. 197-201, Seattle, 1998.

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

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

[4] H. Sakamoto and S. Matsunaga, *Continous Speech Recognition using Dialog-Conditioned Stoachstic Language Model*, Proc. ICSLP, pp. 841-844, Yokohama, 1994.

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

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

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

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

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

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