DOMAIN ADAPTATION AUGMENTED BY STATE-DEPENDENCE IN SPOKEN DIALOG SYSTEMS

He Wei, Li Honglian, Yuan Baozong

Institute of Information Science
Northern Jiaotong University, 100044, Beijing, China
hdavid@263.net

ABSTRACT

In the development of spoken dialog systems, domain adaptation and dialog state-dependent language model are usually researched separately. This paper proposes a new approach for domain adaptation augmented by the dialog state-dependence, which means a dialog turn based cache model decaying synchronously with the dialog state change. Through this approach it's more simple and rapid to adapt a Chinese spoken dialog system to a new task. Two different tasks, the train ticket reservation and the park guide are selected respectively as the target task in the experiments. The consistent reductions of perplexity and character error rate are observed during the adaptation.

1. INTRODUCTION

Although many approaches for language model adaptation to a new task have been proposed [1][2]. Domain adaptation for spoken dialog systems has its own problem. The initial data to bootstrap the adaptation is hard to collect from the new task, because the task-specific data is sparse and usually needs to be obtained in the on-line environment. One solution to this problem is to design the grammars for recognition [3], but it's very difficult for non-expert developers to do.

This paper proposes a dialog turn based decaying cache model for more simple and rapid on-line domain adaptation. Here "simple" means no need of handcrafted grammar and "rapid" means acceptable performance of recognition after training on only a few utterances.

The cache model is proposed as a dynamic component which tracks short-term fluctuations in word frequencies [4][5]. In our system a bigram cache model is used to track the dialog history which contains the task data. As Fig1 shows, the cache bigrams are updated after each dialog turn, which includes the user's utterance and system's response. Then the cache model is interpolated with the generic trigram language model. This adaptation is on-line incremental and could be supervised by the user who corrects the recognizer's output or unsupervised, i.e. the recognizer's output is used directly for adaptation.

We can suppose the incorporation of dialog state dependence in the cache model would further reduce the recognition error rate. Unfortunately during the initial stage of adaptation there is no data to train the state dependent language model [3].

As a substitute for the dialog state dependent language model, the cache is decaying with each dialog turn. We chose dialog turn not utterance as the training data in order to incorporate dialog contexts. A dialog turn is a basic unit to represent a dialog state [6]. As dialog state shifts, the focus and words also change in each dialog turn. That means the bigrams happened in the more distant dialog turn would be less possible to recur than those happened in the recent dialog turn. To predict the forthcoming word pair in the current dialog turn, the cache bigrams in the more distant past is less influential and should be decayed more. It's very different from the standard cache model which considers all the past bigrams same [5]. The decaying process could be simply realized by adding an exponential decay function on the cache bigrams when the cache is updated. The final dialog turn based decaying cache model for domain adaptation is shown in Fig 1.

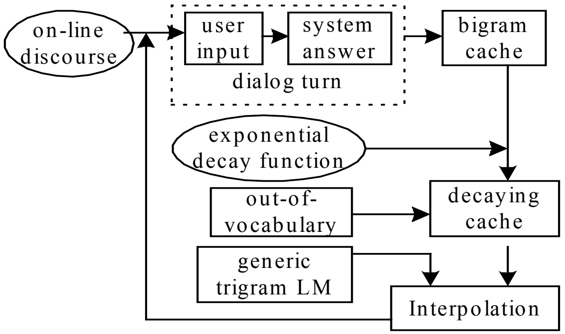

Fig 1 dialog turn based decaying cache adaptation

2. THE GENERIC LANGUAGE MODEL

A Chinese large vocabulary continuous speech recognition system is used as the reference system. The acoustic model was trained on over 200 hours of speech data covering 5 accents using maximum likelihood

estimation. The trigram language model of the reference system is as the generic language model for adaptation.
The text corpus for training the generic language models is about 500 million characters from the encyclopedia and newspaper. The vocabulary contains 47,041 words. The back-off language model is based on Katz smoothing, which is improved by adding 1 count to each unigram to avoid the zero unigram.

3. DIALOG TURN BASED DECAYING CACHE

The standard bigram cache model is represented in formula (1), which considers the history, i.e. the words recognized so far, is same. That's not true. In fact it's different between the far history and the recent history. It's also different between cross-sentences and within a sentence. So the standard cache is improved in two aspects. First the history is divided into different units and second the cache is decaying with dialog turn distances.

$$P_{cache}(w_i / w_{i-1}) = \frac{\sum_{j=1}^{i-1} I(w_{j-1} = w_{i-1}, w_j = w_i)}{\sum_{j=1}^{i-1} I(w_{j-1} = w_{i-1})} \quad (1)$$

Where $I(\bullet)$ is an indicator function which takes the value 1 if $(\bullet)$ occurred, and 0 otherwise.

3.1. Dividing History into Units

In the standard cache the words are inserted into the cache as soon as they are recognized and would predict the next recognition. However a word is unlikely to recur immediately, e.g. occur twice or more in a sentence, that's considered poor usage of language. In fact the cache needs a distance to take effect. So the history is divided into different units such as sentences or dialog turns. Only those bigrams in the past units are inserted into the cache. In the same unit the recognized bigrams have no effect on the next recognition. Now the standard cache becomes the history unit based cache as follows:

$$P_{cache}(w_i / w_{i-1}, H_1^{T-1}) = \frac{\sum_{t=1}^{T-1} \sum_{j \in H_t} I(w_{j-1} = w_{i-1}, w_j = w_i)}{\sum_{t=1}^{T-1} \sum_{j \in H_t} I(w_{j-1} = w_{i-1})} \quad (2)$$

$H_1^{T-1} = (H_1, H_2 \ldots H_{T-1})$ are the history units before the current time *T*. In this paper dialog turn is chosen as history unit, so it's also called dialog turn based cache.

3.2. Decaying with Dialog Turn Distances

Recall the dialog turn could incorporate more dialog contexts and represent the change of dialog states. We will observe the reoccurrence probability of a bigram in the development set (see section 4). The bigram "的车" is chosen from the train tickets reservation. Fig 2 shows the reoccurrence probability of this bigram, which is decaying with the dialog turn distance.

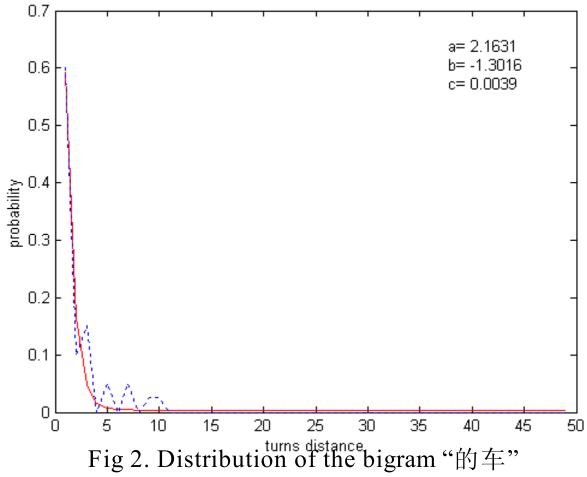

Fig 2. Distribution of the bigram "的车"

In Fig 2 the solid line is a curve fit to the distribution by an exponential function. The phenomena suggest we should add an exponential decay function to the turn based cache. The decay function used in our system is as follows:

$$e^{-\partial (T-t)} \quad (3)$$

Where ∂ is the decay rate, $(T-t)$ is the dialog turn distance to the current turn. For real time recognition the decay rate is constant for all bigrams and those bigrams in the same dialog turn will get the same discount off.

By adding the exponential decay function, the dialog

$$P_{cache}(w_i / w_{i-1}, H_1^{T-1}) = \frac{\sum_{t=1}^{T-1} \sum_{j \in H_t} I(w_{j-1} = w_{i-1}, w_j = w_i) e^{-\partial (T-t)}}{\sum_{t=1}^{T-1} \sum_{j \in H_t} I(w_{j-1} = w_{i-1}) e^{-\partial (T-t)}} \quad (4)$$

turn based decaying cache model is represented as follows.

The above decaying cache model has some similarity to the work in [7], but it differs from the previous decay model in several respects. The first is that it's a bigram cache for spoken language recognition not a unigram cache for dictation. Secondly it is dialog turn based not word based, i.e. it is decaying with turn distance not word distance. Thirdly the decaying cache is used to track the dialog state change. Finally it needs much less time to train, so it could be applied in real time recognition.

3.3. Dialog Turn Based Decaying Cache for Domain Adaptation

The dialog turn based decaying bigram cache is interpolated with the generic trigram language model to adapt the language model as follows.

$$P(w_i / w_{i-2}^{i-1}) =$$

$$\lambda P_{cache}(w_i / w_{i-1}) + (1-\lambda) P_{generic}(w_i / w_{i-2}^{i-1}) \quad (5)$$

λ is the interpolation weight which is empirically estimated on the task development data.

To adapt the spoken dialog system to a new task domain through the dialog turn based decaying cache, the developers needn't build the grammar or use the Wizard OZ approach to collect the initial data. What they need do is just empty the cache and build the out-of-vocabulary words list. Then the supervised or unsupervised adaptation is performed, in the same time the task specific data is collected from the actual dialog.

4. EXPERIMENTS

In the experiments of domain adaptation, two tasks were selected separately as the target task. One is the train tickets reservation task, which is similar to the ATIS task. The other is the park guide domain where the user always takes the initiative. Within each task domain, the cache was not flushed. For both tasks, the cache size was set 20k. When the cache was full those least recently used bigrams would be removed from the cache. The interpolation weight λ was set 0.8 in the park guide and 0.7 in the train tickets reservation.

4.1. Database

For each task a development set and a test set were built. There were total 24 users (16 men and 8 women). Each user was required to use the spoken dialog system in both tasks at least once. The development set was for perplexity experiments and collected in text environment to avoid the influence of recognition errors. Note the development set is only required in this paper and needn't in the practical adaptation. The test set was for recognition experiments and collected by allowing users to talk with the system in the on-line supervised adaptation.

Table 1. Development sets and test sets

	Development Sets		Test Sets	
	Train	Park	Train	Park
Dialogs	60	30	62	56
Turns	308	366	320	598
Utterances	308	366	321	656
Characters	12,455	12,678	15,669	20,058

4.2. The Effect of Decay Rate on Perplexity

Perplexity experiments were conducted separately on the two development sets using the dialog turn based decaying cache model. In the same time a range of decay rates ∂ were investigated, as Fig 3 shows.

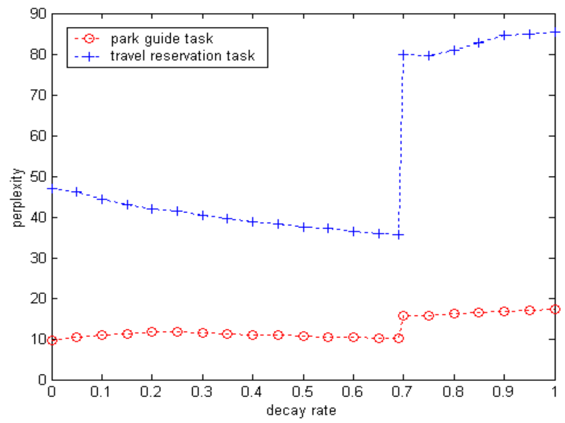

Fig 3. the effect of decay rate on perplexity

From Fig 3, the decay rate ∂ is in the grade $10^{-1} \sim 10^{-2}$ more than we had thought. Where $\partial = 0.65$ the perplexity for two tasks is approximately the lowest. So ∂ is set 0.65 for the decaying cache.

One phenomenon should be noted that where $\partial = 0.7$ the perplexity suddenly increases very much. That's because of the round error. The cache bigram is a decimal with limited digits. When $\partial \geq 0.7$, $e^{-\partial} < 0.5$, a lot of bigrams would be removed for their decayed counts could be rounded off to zero. The rest of the cache bigrams are too sparse to estimate. On the other hand, if $e^{-\partial} \geq 0.5$, it assures the count of any bigram is never rounded to zero, so all the bigrams are retained till the cache full.

When $\partial = 0$ the model turns to be the general incremental cache without decay. The perplexity on the development sets for the generic language model, the general cache and the decaying cache are as follows.

Table 2. Perplexity for the generic LM, general cache and decaying cache

	Park	Train
Generic LM	751.3	373.5
General cache ($\partial = 0$)	10.2	46.9
Decaying cache (∂ =0.65)	9.6	35.9

In the park guide the user inputs concentrate on only several sceneries, so there is a small improvement for the decaying cache compared with the general cache.

4.3. Tracking the Dialog State Change

To testify the decaying cache could track the dialog state change, we chose 20 dialogs from the train tickets reservation and tagged the dialog states. The token-level log likelihood ratio (LLR) [3] is introduced to measure the stochastic separation between the decaying cache and the general cache for each dialog state.

$$LLR(\partial = 0.65, \partial = 0) = \frac{1}{N} \log \frac{P(\mathrm{B}_k / \lambda_{\partial=0.65})}{P(\mathrm{B}_k / \lambda_{\partial=0})}$$
$$= LP^k_{\partial=0} - LP^k_{\partial=0.65} \quad (6)$$

Where B_k is the development set at state k , N is the number of tokens in B_k , LP^k is the *logprob* computed on the data B_k . If the $LLR(\partial = 0.65, \partial = 0)$ is positive, that means that $\lambda_{\partial=0.65}$ is modeling the data B_k better than $\lambda_{\partial=0}$. The *logprob* computed on the dialog state data separately with the general cache and the decaying cache is as follows.

Table 3. Stochastic separation between the general cache and the decaying cache for each dialog state

Dialog State	Turns	*logprob*		*LLR*
		$\partial = 0$	$\partial = 0.65$	
Departure-arrival city	28	7.19	7.15	0.04
Departure date	12	6.98	6.98	0
Trains	8	6.79	6.79	0
Tickets preference	16	7.62	7.49	0.13
Confirmation	20	7.20	6.86	0.34
Others	12	7.39	7.39	0

The improvement by using the decaying cache to track the dialog state change is tiny but indeed exists especially for those frequent states. As the reduction of perplexity on the full development set is 23.9% (see 4.2), we believe the improvement would be more obvious when more dialog state data could be obtained.

4.4. Recognition Experiments

The recognition experiments were conducted on the test sets in both tasks. The generic language model in the decoder was replaced with the decaying cache based language model. The average training time on each dialog turn is about 0.6s. The performance of recognition is measured by character error rate. When character error rate is calculated per 50 utterances, as Fig 4 shows, the reduction of character error rate is consistently observed as the test data increase. The test sets are recognized again by using the general cache. The comparison of character error rate for these two situations is shown in Table 4. From Table 4 we can see the decaying cache would further reduce the character error rate within the same task data.

5. CONCLUSIONS

The dialog turn based decaying cache could track the dialog state change. Through this way it's more simple and rapid to adapt a Chinese spoken dialog system to a new task. When the collected task data are only several hundred utterances, the reduction of perplexity and character error rate is obvious and satisfying.

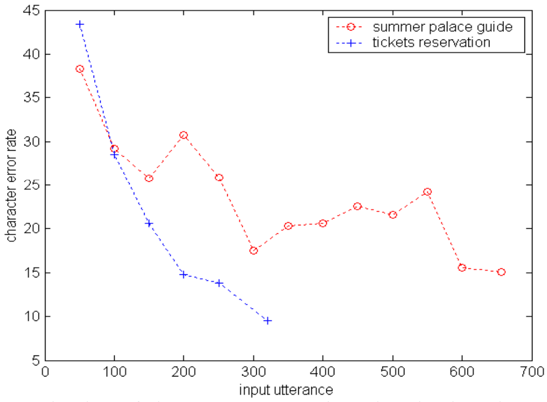

Fig 4.Reduction of character error rate by using the decaying cache

Table 4. Character error rate for two situations: using the general cache or the decaying cache

	The Park Guide		Train tickets	
	General Cache	Decaying Cache	General Cache	Decaying Cache
Sub	11.7%	11.7%	9.4%	8.3%
Ins	2.9%	3.4%	1.2%	1.2%
Del	0.9%	0.1%	0.1%	0.1%
Err	15.5%	15.1%	10.6%	9.5%

6. REFERENCES

[1] M. Federico, "Efficient Language Model Adaptation Through MDI Estimation," *Eurospeech'99*, Budapest, vol.4. pp.1583-1586, 1999.

[2] R. Rosenfeld, "A Maximum Entropy Approach to Adaptive Statistical Language Model," *Compute Speech and Language*, vol.10(3). pp.187-228, 7.1996.

[3] G. Riccardi and A. L. Gorin, "Stochastic Language Adaptation over Time and State in Natural Spoken Dialog Systems," *IEEE Trans on speech and audio processing*, 8(1): pp3-10, 2000

[4] R. M. Iyer and M. Ostendorf, "Modeling Long Distance Dependence in Language: Topic Mixtures versus Dynamic Cache Models," *IEEE Trans. Speech and Audio Processing*, vol.7, No.1. pp30-39, 1999.

[5] R. Kuhn and R. de Mori, "A Cache Based Natural Language Model for Speech Recognition," *IEEE Trans. Pattern Anal. Machine Intell*, vol.14. pp.570-583,1992.

[6] Ruhi Sarikaya. Et al. "Turn-based Language Modeling for Spoken Dialog Systems," *ICASSP'2002*, vol.1. pp.781-784, 2002.

[7] P. Clarkson, "Language Model Adaptation Using Mixtures and an Exponential Decaying Cache," *ICASSP*, Munich, vol. 2.pp799-802, 1997.