Recurrent Neural Network based Language Model Adaptation for Accent Mandarin Speech

Hao Ni[1], Jiangyan Yi[1], Zhengqi Wen[1], Jianhua Tao[1, 2]

[1]National Laboratory of Pattern Recognition, Institute of Automation,
[2]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China

{hao.ni,jiangyan.yi, zqwen, jhtao}@nlpr.ia.ac.cn

Abstract In this paper, we propose adapt the recurrent neural network (RNN) based language model to improve the performance of multi-accent Mandarin speech recognition. N-gram based language model can be easily applied to speech recognition system, but it is hard to describe the long span information in a sentence and arises a serious phenomenon of data sparse. Instead, RNN based language model can overcome these two shortcomings, but it will take a long time to decode directly. Taking these into consideration, this paper proposes a method which combines these two types of language model (LM) together and adapts the RNN based language model to rescore lattices for different accents of Mandarin speech. The architecture of the adapted RNN LM is accent-specific top layers and shared hidden layer. The accent-specific top layers are used to adapt different accents and the shared hidden layer stores history information, which can be seen as memory layer. Experiments on the RASC863 corpus show that the proposed method can improve the performance of accented Mandarin speech recognition over the baseline system.

Keywords: multi-accent, speech recognition, RNN language model, adaptation

1 Introduction

Statistical language model (LM) is a crucial component for automatic speech recognition (ASR) system, which models the distribution of word sequences. The traditional approach for language model is N-gram model, which estimates the word's distribution directly from the relative word frequencies with some smoothing techniques. The N-gram model often suffers from the data sparsity problem and hard to describe the long span information in a sentence. However, neural network language model (NNLM) [1, 2] can represent the non-linear relationship between input words and word probabilities through projecting input words into a continuous space and estimating word probabilities in a softmax layer. It is efficient to overcome the problem of data sparsity. In addition, recurrent neural network (RNN) LM [3, 4, 5] is proposed as a variant of general NNLM, which can break through the limit of fixed context

adfa, p. 1, 2011.

length information and then capture long span information. Therefore, it is superior to the NNLM in ASR system.

It is very crucial to use the large-scale text corpora to train the universal model. But it is still difficult to cover all the domains. So, it often leads to a deterioration of the recognition performance in mismatch condition. In the multi-accent Mandarin speech recognition task, the speakers often come from different regions with different speaking style. It is difficult to cover with all situations since the original corpora and the general language model cannot always meet the requirement. There is a way to solve the problem by adapting LM to improve the performance and make up the deficiency.

Previously, there are several works on multi-domain adaptation. In [6] an NNLM adaptation scheme is proposed by cascading an additional layer between the projection an hidden layer. This scheme provides a direct adaptation of NNLMs via a non-linear, discriminative transformation to a new domain. In [7], a domain dependent element-wise multiplication layer is set between projection and hidden layers of NNLM. The model includes the adaptation layer from beginning, many parameters of the adaptation layer are tied across domains. In [8], the architecture is RNN LM, the adaptation layer is added between recurrent layer and output layer, a domain dependent parameter act as one of the inputs of adaptation layer. They think the architecture only need very small number of domain-specific parameters which enables the model to adapt to domains with little data without the danger of overfitting. On the other hand, in [9], the domain information is fed as an additional input feature to the RNN LM.

The method presents in this paper differ from previous works as follows. An adapted layer or additional feature is needed in previous work. However, it is not necessary in our model. In term of model architecture, inspired by multi-task learning [10, 11], the architecture in our method is accent-specific top layers and shared hidden layer. The accent-specific top layers are used to adapt different accents and the shared hidden layer stores the history information, which can be seen as the memory layer. We adapt our model on top layers only, and improve the performance of accented Mandarin speech recognition over the baseline system.

The reminder of this paper is organized as follows. In Section 2, class based RNN LM is introduced. Section 3 presents the RNN LM adaptation method. The details about our experiments and results are shown in Section 4. We conclude the paper in Section 5.

2 Class based RNN Language Model

RNN is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike feed-forward neural networks, RNN can use their internal memory to process arbitrary sequences of inputs. This makes it applicable in speech recognition.

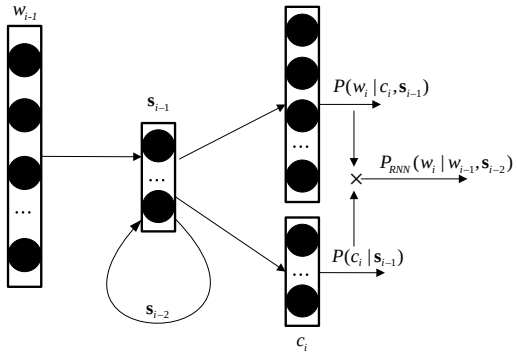

Fig. 1. Architecture *of class based recurrent neural network.*

The class-based recurrent neural network is proposed to compute LM probabilities $P_{RNN}(w_i|c_i, \boldsymbol{s}_{i-1})$ in [4]. Its architecture is described in Figure1. The input layer concatenates the 1-of-*N* representation of the previous word w_{i-1} with the previous state of the hidden layer $\boldsymbol{s}_{i-2}$, which represents the full history vector for word w_i. The hidden layer compresses the information of these two inputs and computes a new representation $\boldsymbol{s}_{i-1}$ using a sigmoid activation to achieve non-linearity. The output layer w_i using a softmax activation to represent the probability distribution of the next word and the state of the hidden layer in the previous time step has the same dimensionality as w_{i-1}.

To reduce the computational cost, a classification layer is added together with the output layer. Each word in the output layer is classified to a unique class based on the frequency counts. In this work, we choose 500 classes, which means the words that correspond to the first 0.2% of the unigram probability distribution would be mapped to class 1 and the next 0.2% would be mapped to class 2. The LM probability assigned to a word is factorized onto two individual terms, as:

$$P_{RNN}(w_i|w_{i-1}, \boldsymbol{s}_{i-2}) = P(w_i|c_i, \boldsymbol{s}_{i-1})P(c_i|\boldsymbol{s}_{i-1}) \qquad (1)$$

Class-based RNN LM can be trained by back-propagation though time (BPTT) algorithm. [12] describes the BPTT algorithm in detail. Training a RNN is more difficult than the feed-forward neural network since gradient explosion maybe occurred [13]. A simple solution to the exploding gradient problem is to truncate values of the gradients. In our experiments, the maximum size of gradients of errors accumulated in the hidden neurons are limited to [-20, 20], which make the training more stable.

3 Proposed RNN Language Model Adaptation Method

In this paper, we use model adaptation method to adapt RNN LM with the accent-specific top layers. This method is motivated by training the RNN with the shared hidden layer in the multi-task learning structure. Multi-accent speech recognition can

adopt the similar method in training the RNN LM. In this paper, four accent-specific tasks learn together with the shared hidden layer. This often leads to a better model for all the four tasks, because it allows the learner to use the commonality among the tasks [11].

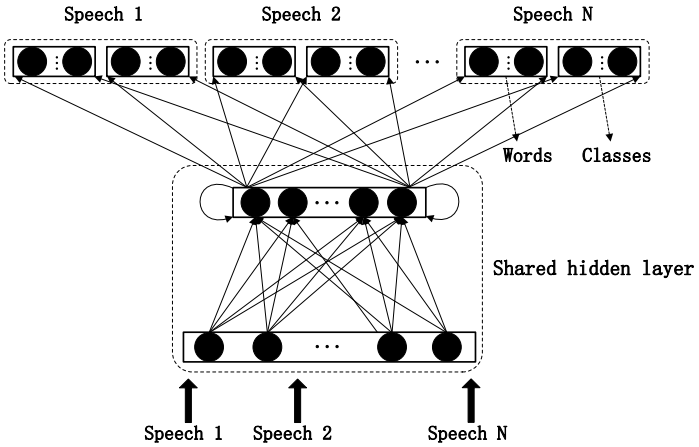

Fig. 2. Architecture of multi-accent RNN LM with region-specific layer and shared hidden layer.

Figure 2 shows the architecture of the proposed multi-accent Mandarin speech adaptation model. The model is builded on the base of trained fig.1 and then copy the top layer N times. In the model, the input is the 1-of-N representation of the previous word. Hidden layer is shared across all the accented speeches, as the global linguistic feature transformation crosses and serves all training accented speeches. Conversely, each accented speech has its own output layer. Each output layer is divided into two parts (words and classes), the classes part is designed to reduce computational cost.

The shared hidden layer, in the multi-accent speech RNN LM, can be treated as a global linguistic feature transformation universal to all training speeches. The shared hidden layer can also be used to transform the linguistic feature for a new accented Mandarin speech. By fixing the hidden layer and only updating the accent-specified layer, supervised adaptation can be achieved with only limited adaptation data.

In our work, we using two pass decoding method. In the first pass, we use a modified Kneser-Ney smoothed 3-gram [14] and in the second pass, RNN LM linearly interpolated with N-grams is used as follows:

$$P(w_i|h_1^{i-1}) = \alpha P_{RNN}(w_i|h_1^{i-1}) + (1-\alpha)P_{NG}(w_i|h_1^{i-1}) \quad (2)$$

where α is used to control the weight, $P_{RNN}(w_i|h_1^{i-1})$ is for RNN LM distribution and $P_{NG}(w_i|h_1^{i-1})$ is for N-gram.

4 Experiments and Result

4.1 Data description

We use the open source Eesen [15] speech recognition toolkit for features extraction, acoustic model training and decoding. N-gram model is trained by SRILM [16]. Our experiments are conducted on RASC863 [17], from which the spontaneous part is selected, about 53.5 hours. The corpus consists of four different kinds of accented Mandarin speech. The speakers of different accents are native residents from Chongqing, Guangzhou, Shanghai, and Xiamen respectively. Moreover, balance in terms of the age, sex, and educational background has considered. The statistics of the training data in the corpus we used lists in Table1.

Table 1. Statistics of acoustic training data.

Accent	#speakers	#utterances	#hours
Chongqing	200	7117	13.0
Guangzhou	200	7659	15.5
Shanghai	200	7401	13.5
Xiamen	200	6115	11.5
Total	800	28292	53.5

For different accent, the corpus partitioned into three parts (training set, validation set, test set) according to 8:1:1 respectively.

We train our N-gram and RNN LM on the same dataset crawled from the Internet, which contains one billion sentences and takes up space of 0.6GB. The dataset is collected almost by written language, such as People's Daily.

4.2 Training

Acoustic model training.

We adopt deep RNNs as the acoustic model, and the Long Short-Term Memory (LSTM) [18] units as the RNN building blocks. Unlike in the hybrid approach, the acoustic model is trained by using frame-level labels with respect to the cross-entropy (CE) criterion. Instead, we adopt the connectionist temporal classification (CTC) [19, 20, 21] objective function to automatically learn the alignments between speech frames and their label (phonemes) sequences. In our experiments, we adopt 3 layers bi-RNNs and each layer contains 320 LSTM units. The input feature is 120 dimensions. The output layer is a softmax layer, which contains 65 units cover with 61 phones and <lau>, <nsn>, <spn>, <blk>.

RNN language model training.

Training RNN LM training includes two phases: general model training and model adaptation. In the general training phase, RNN LM is trained with BPTT. Net-

works are trained in several epochs, in which all data from training corpus are sequentially presented. Weights are initialized to small values (random Gaussian noise with zero mean and unit variance). The input layer contains 240078 units, involved with 239078 words and 1000 hidden units from previous step. The output layer is 239578 units, which contain words and 500 classes. BPTT order is set to 3 which mean the error is propagated through recurrent connections back in time for three steps.

The learning rate starts from 0.1 and remains unchanged until the drop of log-perplexity on the validation set between two consecutive epochs falls below 0.3%. Then the learning rate is decayed by a factor of 0.5 at each of the subsequent epochs. The whole learning process terminates when the perplexity fails to decrease by 0.3% between two successive epochs. The training finished after 11 epochs.

The adaptation process for the RNN LM is a one-iteration retraining on the accent-specific training set. The learning rate is fixed to 0.025, which equal to the last epoch for general model training. In the following, rescoring is performed a second time using the adapted RNN LM, and an improved recognition output is obtained.

Decoding.

Normally, the speech recognition system output the most likely word sequence directly for given acoustic signal, but it is often advantageous to preserve more information for subsequent processing steps. What's more, it is time consuming to decode directly using RNN based language model. In the experiment we use two pass decoding method to get results. The decoding steps as follows:

- Decoding utterances form WFST, produce lattices.
- Extract n-best lists from lattices.
- Compute sentence-level scores using N-gram.
- Perform weighted linear interpolation of log-scores given by N-grams and RNN LM
- Re-rank the n-best lists using the new LM scores

4.3 Baseline model

Different modeling techniques for LM.

To evaluate the performance of single N-gram, single RNN and fusion model (N-gram+RNN), we do experiment on different model. When α=0.0, means N-gram LM is used for lattice rescoring only and α=1.0 means RNN LM is used. Otherwise, fusion model is used. We donate n=10 in n-best lists and choose the highest-score one from it as result.

Table 2. Performance comparison of single N-gram, single RNN and fusion model. (Accuracy %)

α	Chongqing	Guangzhou	Shanghai	Xiamen
0.0	64.51	66.01	74.43	66.33

0.3	66.01	67.82	75.45	67.52
0.4	66.07	67.75	75.43	67.67
0.5	66.12	67.83	75.17	67.63
0.6	66.10	67.76	75.23	67.62
1.0	65.94	67.61	74.98	67.59

- The last row of Table 2 shows, when only RNN LM is used, we achieves 1.43%, 1.50, 0.55%, 1.26% absolute accuracy improves respectively than only N-gram (α=0.0) is used, the result shows that the performance of the RNN LM does better than N-gram in speech recognition.
- Comparing RNN LM with fusion model, we report results for N-gram linear-interpolation with RNN LM with weight 0.5, We achieved 1.61%, 1.82%, 0.74% and 1.30% absolute accuracy improvement respectively for four accent speech.

In summary, fusion model significantly outperforms the other two models while N-gram gets worst results.

Different scale of n-best lists.

Table 3. Performance of different scale n-best lists. (Accuracy %)

n	Chongqing	Guangzhou	Shanghai	Xiamen
0	64.51	66.01	74.43	66.33
10	66.12	67.83	75.45	67.67
100	66.68	68.71	76.04	68.58
1000	67.20	69.11	76.25	69.05

To compare the performance of different scale n-best lists, we select n=0, 10, 100 and 1000 to rescore from lattice. Table 3 report that with the increase of n, we can get higher accuracy, however, the computational complexity will increase rapidly. With the increase of n, search space increase rapidly, we can find better decoding path, but the search time increased dramatically. When the n increases to a certain amount, the accuracy do not improve obviously, this is because the search space is enough, most of high probability paths are included in the search space. The best performance is obtained when we set n to 1000. Therefore, we choose n=1000 in the baseline model.

4.4 Adapted model

Different scale of n-best lists for adaptation.

Table 4. Performance of different scale n-best lists for 1-iteration. (Accuracy %)

n	Chongqing	Guangzhou	Shanghai	Xiamen
0	64.51	66.01	74.43	66.33
10	66.65	68.57	76.21	68.18
100	67.49	69.55	76.80	69.68

1000	68.26	70.12	77.23	70.15

Table 5. Performance of different scale n-best lists for 2-iteration. (Accuracy %)

n	Chongqing	Guangzhou	Shanghai	Xiamen
0	64.51	66.01	74.43	66.33
10	66.72	68.75	76.27	68.75
100	67.56	69.74	76.94	69.84
1000	68.24	70.36	77.31	70.33

For the different scale of n-best lists in adapted model, Table 5 shows 1-iteration adaptation for different scale n-best lists and Table 6 for 2-iteration. From the two experiments, we can get:

- With the increasing of *n*, the performance gets better.
- The best performance achieved when *n* is set to 1000.
- When *n* is set to 10 and 100, all four accents perform better in 2-iteration adaptation than 1-interation. When *n*=1000, except for Chongqing accent, other accents perform better.

In summer, bigger *n* should be chosen to achieve better performance.

Time consumption with different scale of n.

Table 6. Time consumption with different n for rescore.

Accent(test)	Size(h)	n=10(h)	n=100(h)	n=1000(h)
Chongqing	1.29	0.41	0.97	6.61
Guangzhou	1.56	0.44	1.16	7.19
Shanghai	1.40	0.40	0.98	6.80
Xiamen	1.18	0.41	0.88	5.67
Total	5.43	1.66	3.99	26.27

Table 6 shows the time consumption of rescore with different n. The second column is the size of test set. The next three columns are time consumption for n=10, 100 and 1000 respectively. When n increased from 10 to 100, rescore time increased by less than 2 times, but when n increased to 1000, rescore time increased more than 14 times.

To balance the time consumption and accuracy, 100 is a good choice in practical system.

Different iterations for adaptation.

We use fusion model and adapts the RNN based language model to rescore lattices for different accents of Mandarin speech. For model adaptation, we first compare the influence of different adapt-iterations.

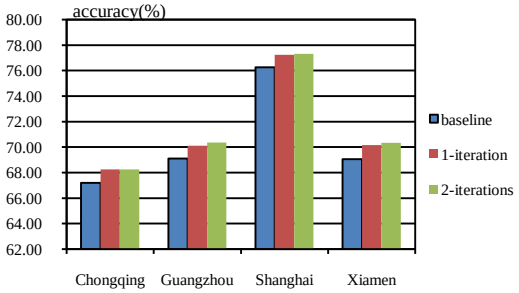

Fig. 3. Performance of different iterations.

Figure 3 shows adapted 1-iteration can get obvious improvement than baseline (without adaptation). However, 2-iterations adaptation gets slightly improvement than 1-iteration. What's more, for Chongqing accented Mandarin, the accuracy becomes worse using 2-iterations adaptation. We can conclude that 1-interation adaptation is enough to achieve good performance and also can avoid overfitting.

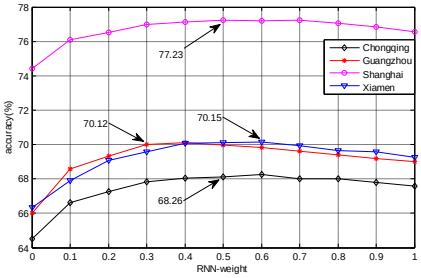

Fig. 4. Performance of RNN LM with one-iteration adaptation.

After RNN adapted for 1-iteration, we use the trained model to compute accuracy for all four kind of accent speeches in the test set. Figure4 shows the accuracy for four different accent speech with RNN LM weight from 0.0 to 1.0. In addition, the experiment is made under the condition that n is set to 1000 in n-best lists.

We can observe that with the RNN LM, the accuracy is higher than without RNN LM for all four accent-speeches. We can conclude that using adapted RNN LM can obviously improve the performance of accented Mandarin speech recognition task, more detail it achieves biggest improvement for 12.09% on Guangzhou accented speech and the least on Chongqing for 10.57% relative improvement on word error rate than N-gram based system. However, for RNN LM based system, we achieved at most 4.12% relative improvement, which is on Shanghai speech. The lowest improvement has only 3.23% on Chongqing speech. One possible reason is the utterances used to adaptation not enough. We have only about 5600 utterances available

for each accented speech. In addition, Chongqing accented speech is similar to standard Mandarin. The improvement of Chongqing accented speech is not obviously than others.

5 Conclusions

We have presented an effective way to improve the performance of multi-accent Mandarin speech recognition system. In our proposed method, we combine N-gram with RNN LM together and adapt the RNN based language model for different accents of Mandarin speech. The architecture of the RNN LM is accent-specific top layers with the shared hidden layer. The accent-specific top layers are used to adapt to different accents and the shared hidden layers stores history information. Experiment results show that proposed method can improve the performance of accented Mandarin speech recognition over the baseline systems.

For future works, we plan to use regularized adaptation methods to avoid overfitting, when the adaptation set is small. Moreover, we want to adopt LSTM-RNN based language model adaptation for other domain text resources (such as spontaneous and written speech) to improve the performance of ASR task.

6 Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) (No.61425017, No.61403386, No.61305003, No.61233009, No.61273288).

7 References

1. Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., & Gauvain, J. L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*(pp. 137-186). Springer Berlin Heidelberg.
2. Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*, *21*(3), 492-518.
3. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S., "Recurrent neural network based language model" in the Proceedings of Interspeech 2010.
4. Mikolov, T., Kombrink, S., Burget, L., Černocký, J. H., & Khudanpur, S. (2011, May). Extensions of recurrent neural network language model. In*Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (pp. 5528-5531). IEEE.
5. Stefan, K., Tomas, M., Martin, K., & Lukas, B. (2011). Recurrent neural network based language modeling in meeting recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association*.
6. Park, J., Liu, X., Gales, M. J., & Woodland, P. C. (2010, September). Improved neural network based language modelling and adaptation. In*INTERSPEECH* (pp. 1041-1044).
7. Alumäe, T. (2013). Multi-domain neural network language model. In*INTERSPEECH* (pp. 2182-2186).

8. Tilk, O., & Alumäe, T. (2014, September). Multi-Domain Recurrent Neural Network Language Model for Medical Speech Recognition. In *Baltic HLT* (pp. 149-152).
9. Chen, X., Tan, T., Liu, X., Lanchantin, P., Wan, M., Gales, M. J., & Woodland, P. C. (2015). Recurrent neural network language model adaptation for multi-genre broadcast speech recognition. *Proceedings of ISCA Interspeech, Dresden, Germany*, 3511-3515.
10. Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41-75.
11. Huang, J. T., Li, J., Yu, D., Deng, L., & Gong, Y. (2013, May). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* (pp. 7304-7308). IEEE.
12. Boden, M. (2002). A guide to recurrent neural networks and back-propagation.*The Dallas project, SICS technical report*.
13. Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, *5*(2), 157-166.
14. Chen, S. F., & Goodman, J. (1996, June). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics* (pp. 310-318). Association for Computational Linguistics.
15. Miao, Y., Gowayyed, M., & Metze, F. (2015). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. *arXiv preprint arXiv:1507.08240*.
16. Stolcke, A. (2002, September). SRILM-an extensible language modeling toolkit. In *INTERSPEECH* (Vol. 2002, p. 2002)
17. "RASC863: 863 annotated 4 regional accent speech corpus," Chinese Academy of Social Sciences, 2003 [Online]. Available: http://www.chineseldc.org/doc/CLDC-SPC-2004-005/intro.htm
18. Graves, A., Jaitly, N., & Mohamed, A. R. (2013, December). Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on* (pp. 273-278). IEEE.
19. Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (pp. 1764-1772).
20. Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376). ACM.
21. Li, J., Zhang, H., Cai, X., & Xu, B. (2015). Towards End-to-End Speech Recognition for Chinese Mandarin Using Long Short-Term Memory Recurrent Neural Networks. In *Sixteenth Annual Conference of the International Speech Communication Association*.