BUILDING A HIGHLY ACCURATE MANDARIN SPEECH RECOGNIZER

Mei-Yuh Hwang[1], *Gang Peng*[1], *Wen Wang*[2], *Arlo Faria*[3], *Aaron Heidel*[4], *Mari Ostendorf*[1]

[1]Univ. of Washington, Dept. of Electrical Engineering, Seattle, WA 98195 USA
[2]SRI International, Menlo Park, CA, 94025 USA
[3]International Computer Science Institute, Berkeley, CA, 94704 USA
[4]National Taiwan Univ, Dept. of CS and IE, Taipei, Taiwan
{mhwang,gpeng,mo}@ee.washington.edu, wwang@speech.sri.com,
arlo@icsi.berkeley.edu, aaron@speech.ee.ntu.edu.tw

ABSTRACT

We describe a highly accurate large-vocabulary continuous Mandarin speech recognizer, a collaborative effort among four research organizations. Particularly, we build two acoustic models (AMs) with significant differences but similar accuracy for the purposes of cross adaptation and system combination. This paper elaborates on the main differences between the two systems, where one recognizer incorporates a discriminatively trained feature while the other utilizes a discriminative feature transformation. Additionally we present an improved acoustic segmentation algorithm and topic-based language model (LM) adaptation. Coupled with increased acoustic training data, we reduced the character error rate (CER) of the DARPA GALE 2006 evaluation set to 15.3% from 18.4%.

Index Terms— Mandarin, character error rates, multi-layer perceptrons, discriminative features, acoustic segmentation, LM adaptation, out-of-vocabulary.

1. INTRODUCTION

Based on the DARPA GALE Project [1], we seek to build a highly accurate automatic speech recognizer (ASR) for continuous Mandarin speech, particularly broadcast news (BN) and broadcast conversation (BC). This paper starts off with a description of the acoustic and text data used in building the system, followed by a description of the major differences between our two AMs. Section 3 illustrates our decoding structure, including improvements in acoustic segmentation to reduce deletion errors, and LM adaptation. Section 4 presents our experimental results, and in the last section we summarize our findings and describe future work.

1.1. Acoustic Data

In this paper, we use about 866 hours of speech data collected by LDC, including the Mandarin Hub4 (30 hours), TDT4 (89 hours), and GALE Year 1 (747 hours) corpora for training our acoustic models. Chronologically, they span from 1997 through July 2006, from shows on CCTV, RFA, NTDTV, PHOENIX, ANHUI, and so on.

We test our system on three different test sets for various studies: DARPA EARS RT-04 evaluation set (eval04), DARPA GALE 2006 evaluation set (eval06), and GALE 2007 development set (dev07). [1] Each test set is selected from segments of different shows, as summarized in Table 1.

[1]The dev07 set used here is the IBM-modified version, not the original LDC-released version.

Table 1. Acoustic test data.

Data	year/month	#shows	duration
eval04	2004/04	3	1 hr
eval06	2006/02	24	2 hr
dev07	2006/11	74	2.4 hr

1.2. Text Corpora and Lexicon

Our text corpora come from a wide range of data, in addition to the transcriptions of the acoustic training data. Other sources of text include the LDC Mandarin Gigaword corpus, all GALE-related Chinese web text releases, other web text collected by National Taiwan University and Cambridge University, and the conversational telephone text described in [2]. The source text underwent a few passes of cleaning to remove HTML tags, punctuation, corrupted GB2312 codes, the normalization of numbers from digits to spoken forms, and so on, before being segmented into "word" units. All together, there are over 1 billion words in this training collection.

Our Mandarin ASR system is based on "word" recognition. We start from the BBN-modified LDC Chinese word lexicon, and manually augment it with a few thousand new words (both Chinese and English words) over time. We end up with a lexicon of 70,000 words or so. For word segmentation (to insert space between sequences of Chinese characters), we start off with a simple longest-first match to segment our training documents and then train a unigram LM. The most frequent 60,000 words are then selected as our decoding lexicon and the unigram LM is trimmed back to these 60 K words.

Given this initial unigram, we then use the unigram to do maximum-likelihood (ML) word segmentation on the training text. Having done this, the only possible out-of-vocabulary (OOV) words are OOV English words and OOV single-character Chinese words. We do not add these OOV single-character words in our decoding lexicon because (a) adding them would only increase n-gram perplexity and acoustic confusability among the existing vocabulary, and (b) they are so rare that it is not worth the increased recognition difficulty. In our experience, ML word segmentation results in only slightly better perplexity and usually translates to no further improvement in recognition. However, we believe that the ML word segmentation more often offers a semantically better segmentation which helps downstream applications such as machine translation (MT).

After the ML word segmentation, we then re-train our n-gram LMs using the modified Kneser-Ney smoothing method [3].

Table 2 lists the sizes of the full n-grams with different frequency cutoffs, and their pruned versions. For example, our full 4-gram LM has 316 million 3-gram probabilities and 201 million 4-gram probabilities. Section 3 will describe how these n-grams are used in our system.

Table 2. Numbers of entries of n-gram LMs.

#entries	full LM	pruned LM
lexicon size	60421	60421
3-gram		
n2	58 M	6.6 M
n3	108 M	3.3 M
4-gram		
n2	58 M	19 M
n3	316 M	24 M
n4	201 M	6 M

2. TWO ACOUSTIC SYSTEMS

A key component of our system is cross adaptation and system combination between two subsystems. We seek to create two subsystems having approximately the same error rate performance but with error behaviors as different as possible, so that they will compensate for each other. The differences between our two acoustic systems are summarized in Table 3.

Table 3. Differences of our two acoustic models.

	System-ICSI	System-PLP
feature dim	74 (MFCC+MLP)	42 (PLP)
fMPE	no	yes
phones	72	81

2.1. System-ICSI

The first system uses 70 phones for pronunciations, inherited from the BBN dictionary. Additionally, there is one phone designated for silence, and another one for noises, laughter, and unknown foreign speech. Both the silence phone and the noise phone are context-independent.

The front-end features consist of 74 dimensions per frame, including

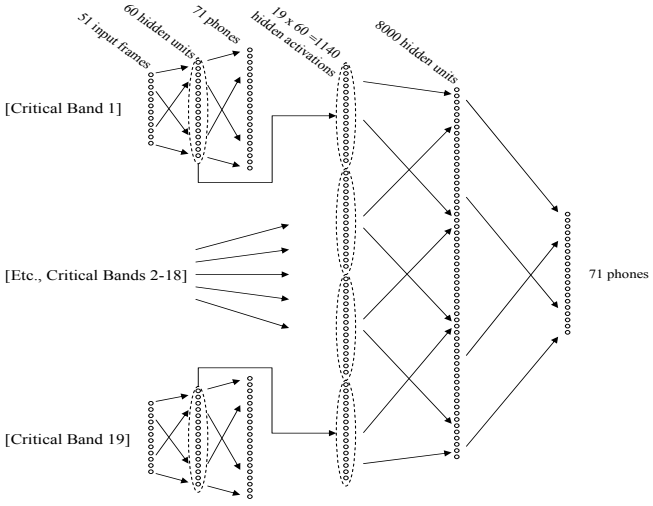
- 13-dim MFCC, and its first- and second-order derivatives;
- spline smoothed pitch feature [4], and its first- and second-order derivatives;
- 32-dim phoneme-posterior features generated by multi-layer perceptrons (MLP) [5, 6].

The MLP is designed to provide discriminative phonetic information at the frame level. The MLP feature generation involves three main steps. We first, for each input frame, concatenate its neighboring 9 frames of PLP and pitch features as the input to an MLP. Each output unit of the MLP models the likelihood of the central frame belonging to a certain phone, given the 9-frame intermediate temporal acoustic evidence. We call this vector of output probabilities the *Tandem* phoneme posteriors. Particularly in our case, there are 42*9 input units, 15000 hidden units, and 71 output units in the Tandem MLP. The noise phone is excluded from the MLP output because it is presumbly not a very discriminable class. It is aligned to many kinds of noises and foreign speech, not any particular phonetic segment. The acoustic training data are Viterbi aligned using the best existing model, to identify the target phone label for each frame. The data for the noise phone are excluded from MLP training.

Next, we separately construct a two-stage MLP where the first stage contains 19 MLPs and the second stage one MLP. The purpose of each MLP in the first stage, with 60 hidden units each, is to identify a different class of phonemes, based on the log energy of a different critical band across a long temporal context (51 frames ~ 0.5 seconds). The second stage of MLP then combines the information from all of the *hidden* units (60*19) from the first stage to make a grand judgment on the phoneme identity for the central frame. This merger MLP has 8,000 hidden units. The output of the second stage is called the *HATs* phoneme posteriors (*h*idden *a*ctivation *t*emporal pattern*s*) and is illustrated in Figure 1.

Finally, the 71-dim *Tandem* and *HATs* posterior vectors are combined using the Dempster-Shafer [7] algorithm. Logarithm is then applied to the combined posteriors, followed by Principal component analysis (PCA) to (a) make each dimension independent, as our HMM models use Gaussian mixtures with diagonal co-variances, and (b) reduce the dimensionality from 71 to 32. The 32 dimensions of phoneme-posterior features are then appended into the MFCC and pitch features. This system with 74-dim features is thus referred to as System-ICSI because of the use of the MLP features produced at ICSI, Berkeley.

Fig. 1. The *HATs* feature, computed using a two-stage MLP. Notice the output from the hidden units of the first-stage MLPs is the input to the second-stage MLP.



MLP feature extraction as described in this paper differs from our 2006 system [8] in three principal ways: (1) the amount of training data is doubled, (2) pitch features are added to the Tandem MLP input layer, and (3) to combine the two posterior feature streams (Tandem and HATs), we use the Dempster-Shafer theory of evidence rather than inverse-entropy weighted summation.

While it is clearly advantageous to use more training data, this

introduces considerable practical complications for MLP training since the online training algorithm cannot be easily parallelized across multiple machines. To address this, we optimized our multi-threaded QuickNet code to run on an 8-core server in a quasi-online batched mode: each network update is performed using the feed-forward error signal accumulated from a batch of 2048 randomized input frames. Additionally, we decrease the training time by partitioning the training data and applying the learning schedule described in [9].

A cross-word triphone model with the ICSI-feature is trained with an MPE [10, 11] objective function, and an SAT feature transform, based on 1-class constrained MLLR [12]. Decision-tree based HMM state clustering [13] is applied. There are 3500 shared states, each with 128 Gaussians. This model size is denoted as 3500x128.

2.2. System-PLP

The second acoustic system contains 42-dimension features with static, first- and second-order derivatives of PLP features. Similarly, a 3500x128 cross-word triphone model with the PLP-feature is trained with an MPE objective function and an SAT feature transform, using decision-tree based state sharing. Moreover, to compete with the ICSI-model which has a stronger feature representation, an fMPE [14] feature transform is learned for the PLP-model. The fMPE transform is trained by computing the high-dimension Gaussian posteriors of 5 neighboring frames, given a 3500x32 cross-word triphone ML-trained model with an SAT transform. Therefore, in

$$y_t = x_t + Mh_t,$$

the dimension of h_t is bigger than $3500 * 32 * 5 = 560K$ (including context-independent Gaussians), and M is on the order of 42 x $560K$.

To tackle spontaneous speech which occurs more often in BC shows than BN, and which tends to be spoken more quickly than narrative speech, we introduce a few diphthongs in the PLP-model as shown in Table 4, where phone names are case-sensitive and vowels with no tone represent all four tones. The addition of diphthongs naturally removes the need for the syllable-ending `Y` and `W` sounds. Combining two phones into one reduces the minimum duration requirement by half and hence is likely better for fast speech. Additionally the 72-phone set does not model the neutral tone, or tone 5, but instead the third tone is used as a replacement. As there are a few very common characters that are 5-th tone, we add three neutral-tone phones for them. Furthermore, we add the context-independent phone `/V/` for the `v` sound in English words, as this phone is missing in Mandarin but not difficult at all for Chinese people to pronounce accurately. For the two common filled-pause characters (呃,嗯), we use two separate context-independent phones to model them individually, so that they are not sharing parameters with regular Chinese words. In addition, to keep the size of the new phone set manageable, we merge `/A/` into the `/a/` sound, and both `/I/` and `/IH/` into `/i/`. We rely on triphone modeling to distinguish these allophones of the same phoneme. Finally, the 72-phone set does not model the somewhat-rare phone `/I2/` as in 词; instead we use `/I1/`. Thus with `/I2/` represented by `/i2/`, the second tone of the non-retroflex `/i/` sound is now modeled correctly. To sum up, the new phone set has 81 base phones, including three extra context-independent phones.

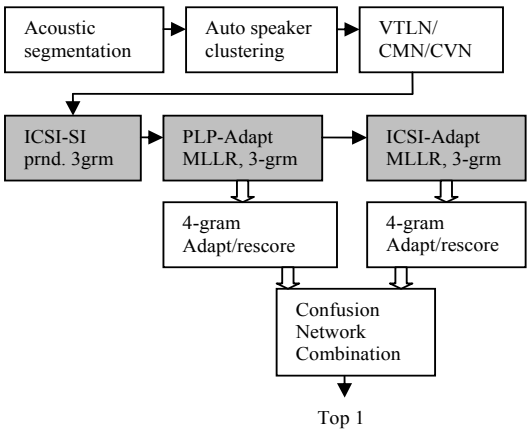
Table 4. Difference between the 72-phone and 81-phone sets. Asterisks indicate context-independent phones.

example	phone-72	phone-81	#additions
要	a W	aw	+4-1
北	E Y	ey	+4-1
有	o W	ow	+4
爱	a Y	ay	+4
安	A N	a N	-4
次	I	i	-3
尺	IH	i	-4
了	e3	e5	+1
吗	a3	a5	+1
子	i3	i5	+1
victory	w	V^*	+1
呃	o3	fp_o^*	+1
嗯	e3 N	fp_en^*	+1
Total			+9

3. DECODING ARCHITECTURE

Figure 2 shows a simplified representation of our recognition architecture.

Fig. 2. System decoding architecture. Single arrows represent top 1 word sequence output, while block arrows represent top-n best word sequences.



GALE test data comes in as per-show recordings; however, only specified segments (usually a few minutes long per segment) in each recording need to be recognized. Instead of feeding the whole show into our decoder, we segment each 0.5–1 hour of recording into utterances of a few seconds long, separated by long pauses, and run utterance-based recognition. Next we perform speaker clustering using Gaussian mixture models of static MFCC features and K-means clustering. We call these speakers *auto* speakers. Vocal tract length normalization (VTLN) is then performed for each auto speaker, followed by utterance-based cepstral mean normalization (CMN) and cepstral variance normalization (CVN).

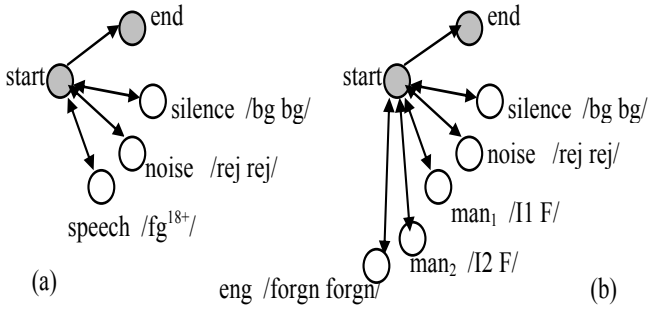
3.1. Acoustic Segmentation

In error analysis of our previous system, we discovered that deletion errors were particularly serious. Deletion errors not only degrade ASR performance, but are particularly bad for downstream processing such as machine translation; their effect is worse than insertion errors. Our system has a noise phone to model garbage/noises. We found that some of the deletion errors were caused by false alarm garbage words. To control these garbage false alarms, we introduce a garbage penalty into the decoder, which is successful in removing some deletion errors. However, most of the deletion errors came from dropped speech segments due to faulty acoustic segmentation. Therefore, we attempt to improve acoustic segmentation such that not only fewer speech segments are dropped, but new insertion errors are simultaneously avoided [15].

3.1.1. Previous Segmenter

Our previous segmenter is a speech-silence detector given the whole-show recording. A recognizer is run with a finite state grammar as shown in Figure 3(a). There are three words in the vocabulary of the decoder: silence, noise, and speech, whose pronunciations are shown in the figure. Each pronunciation phone (`bg`, `rej`, `fg`) is modeled by a 3-state HMM, with 300 Gaussians per state. The HMMs are ML trained on Hub4, with 39-dimension features of MFCC and its first-order and second-order differences. The segmenter operates without any knowledge of the underlying phoneme sequence contained in the speech waveform. More seriously, due to the pronunciation of "speech", each speech segment is defined as having at least 18 consecutive `fg`s, which forces any speech segment to have a minimum duration of 540 ms.

Fig. 3. Acoustic segmenters: (a) previous segmenter, (b) new segmenter.



After speech/silence is detected, segments composed of only silence and noises are discarded. The segmenter then attempts to merge short speech segments into at most 9 seconds long per utterance, if it judges that two consecutive segments are from the same speaker (based on MFCC Gaussian-mixture models) and the pause in between is under a certain threshold.

3.1.2. New Segmenter

Our new segmenter, shown in Figure 3(b), makes use of broad phonetic knowledge of Mandarin and models the input recording with five words: silence, noise, a Mandarin syllable with a voiceless initial, a Mandarin syllable with a voiced initial, and a non-Mandarin word. Thus there are 6 distinct HMMs for speech-silence detection and the minimum speech duration is reduced to 60 ms. Except for the finite state grammar and the pronunciations, the rest of the segmentation process remains the same. As shown later in Section 4.1, we are able to recover most of the discarded speech segments via the new finite state grammar and the new duration constraint.

3.2. Search with Trigrams and Cross Adaptation

The decoding is composed of three trigram recognition passes:

1. ICSI-SI: We begin with a quick search using a speaker-independent (SI) within-word triphone MPE-trained ICSI-model and the highly pruned trigram LM. This gives us a good initial adaptation hypothesis quickly.
2. PLP-Adapt: Next we use the ICSI hypothesis to learn the speaker-dependent SAT transform and to perform MLLR adaptation [12] per speaker, on the cross-word triphone SAT+fMPE MPE trained PLP-model.
   After the acoustic model is adapted, we then run full-trigram decoding to produce an N-best list for each utterance.[2]
3. ICSI-Adapt: Similar to PLP-Adapt, we run cross adaptation first, using the top 1 PLP hypothesis to adapt the cross-word triphone SAT MPE trained ICSI-model, followed by full-trigram decoding to produce N-best lists.

3.3. Topic-Based Language Model Adaptation

We perform topic-based language model adaptation using a Latent Dirichlet Allocation (LDA) topic model [16, 17]. The topic inference algorithm takes as input a weighted bag of words $\boldsymbol{w}$ (e.g. in one topic-coherent story) and an initial topic mixture $\boldsymbol{\theta}^0$ and returns a topic mixture $\boldsymbol{\theta}$. During training, we label the topic of each individual sentence to be the one with the maximum weight in $\boldsymbol{\theta}$, and add the sentence to this topic's corpus. We then use the resulting topic-specific corpora to train one n-gram LM per topic [18]. The general LMs trained in Table 2 are called topic-independent (TI) background LMs.

During decoding, we infer the topic mixture weights dynamically for each utterance; select the top few most relevant topics above a threshold, and use their weights in $\boldsymbol{\theta}$ to interpolate with the TI n-gram background language model.

In order to make topic inference more robust against recognition errors, we weight the words in $\boldsymbol{w}$ based on an N-best-list derived confidence measure; additionally we include words not only from the utterance being rescored but also from surrounding utterances in the same story chunk via a decay factor, where the words of distant utterances are given less weight than those of nearer utterances. As a heuristic, utterances that are in the same show and less than 4 seconds apart are considered to be part of the same story chunk. The adapted n-gram is then used to rescore the N-best list.

4. EXPERIMENTAL RESULTS

4.1. Acoustic Segmentation

Tables 5 and 6 show the CERs with different segmenters at step ICSI-SI and step PLP-Adapt, respectively, on eval06. The error distributions and our manual error analysis both show that the main benefit

[2]For various legacy and computation reasons, the actual implementation is to use a pruned bigram to dump word lattices quickly first, and then expand the bigram lattices into full trigram lattices, from which we then extract N-best lists.

of the new segmenter is in recovering lost speech segments and thus in lowering deletion errors. However, those lost speech segments are usually of lower speech quality and therefore lead to more substitution and insertion errors. For comparison, we also show the CERs with the oracle segmentation as derived from the reference transcriptions. These results show that our segmenter is very competitive.

Table 5. CERs at step ICSI-SI on eval06 using different acoustic segmenters.

Segmenters	Sub	**Del**	Ins	Overall
Previous segmenter	9.7	**7.0**	1.9	18.6
New segmenter	9.9	**6.4**	2.0	18.3
Oracle segmenter	9.5	**6.8**	1.8	18.1

Table 6. CERs at step PLP-Adapt on eval06 using different acoustic segmenters.

Segmenters	Sub	**Del**	Ins	Overall
Previous segmenter	9.0	**5.4**	2.0	16.4
New segmenter	9.2	**4.8**	2.1	16.1
Oracle segmenter	8.8	**5.3**	2.0	16.1

4.2. MLP Features

Since Mandarin is a tonal language, it is well known that adding pitch information helps with speech recognition [19]. For this reason, we investigate adding pitch into the input of the Tandem neural nets. For quick verification, we used Hub4 to train within-word triphone ML models. Table 7 shows the SI bigram CER performance on eval04. Pitch information obviously provides extra information for both the MFCC front end and the *Tandem* front end.

Table 7. SI bigram CERs on eval04, using 30 hours of acoustic training data for within-word triphone ML models.

HMM Feature	MLP Input	CER
MFCC	—	24.1
MFCC+F0	—	21.4
MFCC+F0+Tandem	PLP	20.3
MFCC+F0+Tandem	PLP+F0	19.7

To compare the impact of different phoneme posterior combination methods, we trained all neutral networks with the 866 hours of training data. Then to have a fast turnaround, we trained two within-word triphone ML models with 98 hours of data (Hub4 and a subset of TDT4). One model was trained using the MLP feature combined by the inverse entropy method, the other by Dempster-Shafer. Table 8 suggests the superiority of the Dempster-Shafer approach, where the first column is speaker independent recognition and the second column with unsupervised MLLR adaptation using the first-pass output.

Table 8. CERs on eval04, using different methods of combining Tandem and Hats features. The acoustic models were within-word triphones, ML trained on 98 hours of data.

Combo Method	First-pass	Spkr-adapt
Inverse Entropy	17.6	16.5
Dempster-Shafer	17.0	16.4

4.3. Cross Adaptation Using Outside Regions

To increase the amount of unsupervised adaptation data, we also decode those speech segments outside the specified testing range. Particularly, we decode 60 seconds before and after each specified testing range. If any utterance in these outside regions is classified as one of the auto speakers in the "inside" region, it is then added into MLLR adaptation.

The top three rows of Table 9 show how CERs change as we increase the amount of unsupervised acoustic adaptation data, on dev07. dev07-0 means no outside region is used in either acoustic segmentation or adaptation. dev07-60-inside means the $\pm 60s$ outside regions are used during acoustic segmentation, but not during AM adaptation. As our acoustic segmenter is affected by the surrounding context, the acoustic segmentations from dev07-0 and that from dev07-60-inside can be different. dev07-60 means the outside regions are used in both acoustic segmentation and AM adaptation. Note that dev07-60-inside and dev07-60 thus have identical acoustic segmentation.

Unfortunately, it seems that most of the improvement comes from better acoustic segmentation rather than from more AM adaptation data, perhaps because the speaker boundary information is not very accurate, or maybe because the MLLR regression classes need to be learned using a more sophisticated approach. [3]

Table 9. Decoding progress on dev07. All AMs are adapted. The top three rows use the full trigram.

Step	PLP-Adapt	ICSI-Adapt
dev07-0	12.4	–
dev07-60-inside	12.1	–
dev07-60	12.0	11.9
adapted pruned 4-gram	(a) 11.7	(b) 11.4
static full 4-gram	(c) 11.9	(d) 11.7
(a)+(b) CNC	11.2	
(c)+(d) CNC	11.4	
(a)+(b)+(c)+(d)	11.2	

4.4. Pronunciation Phone Sets

Table 10 shows the CERs on dev07 with the two different phone sets, using dev07-60 acoustic segmentation from Table 9. To perform a fair comparison, two within-word triphone PLP models were ML trained with the 866 hours of data: one with the 81-phone set and

[3] Currently the MLLR regression classes are fixed 3 or 4 classes (silence/noises, consonants, and vowels).

the other with the 72-phone set. These comparisons were conducted with the SI models and the pruned trigram in Table 2.

Table 10. CERs on dev07 using different phone sets. The AMs are SI PLP-feature ML trained within-word triphones. The LM is the pruned trigram.

	BN	BC	Avg
phone-81	7.6	27.3	18.9
phone-72	7.4	27.6	19.0

A careful analysis reveals that the improvement in the BC portion from the 81-phone set is completely due to the reduction in deletion errors. Therefore, despite the modest overall improvement, the new phone set achieves our goal of generating different error patterns.

4.5. Adaptation on Language Models

Due to memory constraints, we are unable to adapt the full 4-gram LM. Instead, we train 64 topic dependent 4-grams and interpolate them with the TI pruned 4-gram in Table 2.

During decoding, the N-best lists of both the adapted PLP system and the adapted ICSI system are used to compute the topic mixture weights $\boldsymbol{\theta}$, and the most relevant topics (those whose weights in $\boldsymbol{\theta}$ are above a threshold) are then selected and interpolated with the TI pruned 4-gram, on a per-utterance basis. An adapted 4-gram is finally applied to rescore the N-best list of each utterance. The result is shown in the fourth row in Table 9. Compared with the full static TI 4-gram in the next row, the adapted 4-gram is slightly albeit consistently better.

4.6. System Combination

Finally, a character-level confusion network combination of the two rescored N-best lists yields a 11.2% CER on dev07, as shown in the row of "(a)+(b) CNC" in Table 9. When the entire system of Figure 2 is applied to eval06, we reduce the CER from 18.4% a year ago to 15.3%.

5. FUTURE WORK

This paper presents a highly accurate Mandarin speech recognizer. We have made significant progress over a one year time frame, including improving our MLP discriminative features, different pronunciation phone sets, acoustic segmentation, language model adaptation and increased training corpora.

Anecdotal error analysis on dev07 shows that diphthongs did help in examples such as 北大 (/b ey3 d ay4/, Beijing University), and merging /A/ and /a/ was not harmful. But merging /I/ and /IH/ into /i/ seemed to cause somewhat more confusion among characters such as (是,至,地)=(shi,zhi,di). Perhaps we need to reverse the last decision.

The topic-based LM adaptation is simple and fast. However, we are not satisfied with the current degree of improvement. Further refinement in the algorithm and in the implementation is needed to adapt the full 4-gram and obtain greater significance. Our previous study [8] showed that full re-recognition with the adapted LM offered more improvement than N-best rescoring. Yet the computation is expensive. A lattice or word graph re-search is worth investigating.

Acknowledgment

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

6. REFERENCES

[1] "Global Autonomous Language Exploitation (GALE)," http://www.darpa.mil/ipto/programs/gale/.

[2] T. NG, M. Ostendorf, M.Y. Hwang, M. Siu, I. Bulyko, and X. Lei, "Web data augmented language models for Mandarin conversational speech recognition," in *Proc. ICASSP*, 2005, pp. 589–592.

[3] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Science Group, Harvard University, TR-10-98*, 1998.

[4] X. Lei, M. Siu, M.Y. Hwang, M. Ostendorf, and T. Lee, "Improved tone modeling for Mandarin broadcast news speech recognition," in *Proc. Interspeech*, 2006.

[5] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *Proc. ICSLP*, 2004.

[6] J. Zheng, O. Cetin, M.Y. Hwang, X. Lei, A. Stolcke, and N. Morgan, "Combining discriminative feature, transform, and model training for large vocabulary speech recognition," ICASSP 2007.

[7] F. Valente and H. Hermansky, "Combination of acoustic classifiers based on dempster-shafer theroy of evidence," in *Proc. ICASSP*, 2007.

[8] M.Y. Hwang, W. Wang, X. Lei, J. Zheng, O. Cetin, and G. Peng, "Advances in mandarin broadcast speech recognition," in *Proc. Interspeech*, 2007.

[9] Q. Zhu, A. Stolcke, B.Y. Chen, and N. Morgan, "Using MLP features in SRI's conversational speech recognition system," *Proc. Interspeech, Lisbon*, 2005.

[10] D. Povey and P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

[11] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. Interspeech*, 2005, pp. 2125–2128.

[12] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[13] M.Y. Hwang, X.D. Huang, and F. Alleva, "Predicting unseen triphones with senones," in *Proc. ICASSP*, 1993, pp. 311–314.

[14] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, 2005.

[15] G. Peng, M.Y. Hwang, and M. Ostendorf, "Automatic acoustic segmentation for speech recognition on broadcast recordings," in *Proc. Interspeech*, 2007.

[16] T. Hofmann, "Probabilistic latent semantic analysis," in *Uncertainty in Artificial Intelligence*, 1999.

[17] D.M. Blei, A.Y. NG, and M.I. Jordan, "Latent dirichlet allocation," in *The Journal of Machine Learning Research*, 2003, pp. 993–1022.

[18] A. Heidel and L.S. Lee, "Robust topic inference for latent semantic language model adaptation," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2007.

[19] M.Y. Hwang, X. Lei, T. NG, I. Bulyko, M. Ostendorf, A. Stolcke, W. Wang, and J. Zheng, "Progress on mandarin conversational telephone speech recognition," in *International Symposium on Chinese Spoken Language Processing*, 2004.