The CMU-InterACT 2008 Mandarin Transcription System

Roger Hsiao, Mark Fuhs, Yik-Cheung Tam, Qin Jin and Tanja Schultz

InterACT, Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{wrhsiao,fuhs,yct,qjin,tanja}@cs.cmu.edu

Abstract

We present our Mandarin BN/BC transcription system recently developed for the GALE07 evaluation. The system employs a 3-pass decoding strategy trained with over 1300 hours of quickly transcribed audio. We successfully apply discriminative training, dynamic unsupervised language model adaptation, and system combination techniques in our system. We furthermore achieve improvements by combining an Initial-Final system with a genre dependent phone system. On the GALE07 phase 2 retest evaluation, our system achieves a character error rate(CER) of 13.3% on dev07 test set and 13.5% on eval07 unsequestered test set. Our system also allows combination with other sites and in this paper, we investigate different system combination strategies which significantly improve the final recognition performance.

Index Terms: Mandarin transcription system, broadcast news, broadcast conversation, GALE evaluation

1. Introduction

This paper describes our effort on the development of the Mandarin transcription system for broadcast news (BN) and broadcast conversation (BC). The CMU system was evaluated under the Global Autonomous Language Exploitation (GALE) program. Our target is to provide more accurate automatic transcriptions of radio and TV shows for the GALE translation and distillation tasks. During the first two years of the program, we have received over 1300 hours of transcribed speech data and used them to improve our Mandarin system. Our system development has multiple direction of efforts. We investigate better audio segmentation and clustering, acoustic modeling and language modeling methods, and apply them on a large scale evaluation. In this paper, we describe the areas we have explored, including discriminative training algorithms like maximum mutual information estimation (MMIE) and boosted maximum mutual information estimation (BMMIE), unsupervised LM adaptation using latent semantic analysis (LSA) as in [1, 2], and confusion network and lattice-based system combination techniques.

The paper is organized as follows. In Section 2, we describe our effort to improve the audio segmentation and clustering algorithm. In Section 3, we give a detailed description of acoustic and language modeling and how we apply discriminative training and language model adaptation to improve our system. In Section 4, we present the system performance on GALE evaluation test sets and analyze the performance of each system component. In Section 5, we describe our system combination strategy and evaluate different approaches. In Section 6, we conclude our work and discuss future work.

2. Audio Segmentation and Clustering

Audio segmentation is realized by an HMM segmenter with four classes: Speech, Noise, Silence, and Music. The speech features used are 13-dimension MFCCs plus their first and second derivatives. Each class is represented by a GMM with 64 Gaussians. The system is trained on 3 hours of manually annotated HUB4 shows.

The resulting speech segments (the Noise, Silence, and Music segments are ignored) are then grouped into several clusters, each cluster ideally corresponding to an individual speaker. A hierarchical, agglomerative clustering technique with Bayesian Information Criterion (BIC) stopping criteria is used [3]. A tied Gaussian mixture model (TGMM) is built on the whole set of speech segments. A GMM for each cluster is trained by adaptation of the TGMM. Each segment is considered as a cluster at the initial step. We define the distance between two clusters by the Generalized Likelihood Ratio (GLR):

$$D(C_1, C_2) = -\log \frac{P(X|\theta)}{P(X_1|\theta_1)\ P(X_2|\theta_2)} \tag{1}$$

where X_1, X_2, and X are feature vectors in cluster C_1, in cluster C_2, and in the merged cluster of C_1 and C_2, respectively. θ_1, θ_2, and θ are statistical models built on X_1, X_2, and X, respectively.

We can see from (1) that the smaller the distance, the closer the two clusters are to each other. At each step, the two closest clusters are merged and a model of the new cluster is reestimated. The clustering is done until the BIC stopping threshold is exceeded.

There is a new feature in the GALE 2007 evaluation compared to previous evaluations. In the 2006 evaluation, the test data has one snippet per show with a duration of 3-5 minutes. In the 2007 evaluation, snippets are 1-2 minutes, and some shows contain multiple snippets. We therefore conduct clustering across snippets on the same show, which means all speech segments from different snippets on the same show are pooled together for clustering and a unique speaker label is shared across different snippets on the same show. We use two different BIC thresholds depending on the number of snippets per show to ensure that we do not underestimate the number of speakers in multiple-snippet shows.

3. Acoustic and Language Modeling

The feature extraction employs standard 13-dimension MFCC features extracted using a 16ms window with 10ms frame shift. We concatenate 15 adjacent feature vectors and apply linear discriminant analysis to reduce the feature dimension to 42. We apply standard cepstral mean/variance normalization and vocal-tract length normalization per speaker.

We choose the Initial-Final (I-F) and phone models and build separate systems similar to what we did on our legacy RT04 system [4]. We found the I-F and phone systems useful to improve recognition performance by cross-adaptation. We cluster the states using a quinphone decision tree with tonal questions incorporated such that a single tree implicitly models all tonal variants of the same base phone/syllable. The number of Gaussian mixtures per state is determined using merge and split (MAS) training with a maximum of 100 mixtures per state. We then apply a global semi-tied covariance (STC) matrix [5] on all the acoustic models.

The CMU Mandarin transcription system is a 3-pass system. It consists of three sets of acoustic models, namely AM1, AM2, and AM3. AM1 is a speaker independent (SI) I-F model using multi-style training by uniformly mixing the BN and BC shows together. AM2 is a phone model using speaker-adaptive training (SAT) with feature space adaptation (FSA). AM2 uses hypotheses from AM1 for cross adaptation. Instead of using multi-style training as with AM1, AM2 is genre dependent: its senone tree considers whether the incoming show is BN or BC. We have a simple rule to decide the genre based on the show name. AM3 is also speaker adaptive, based on the output of AM2, but AM3 uses I-F models. In addition, we perform maximum mutual information estimation [6] (MMIE) and boosted maximum mutual information estimation (BMMIE) training to improve recognition accuracy. Table 1 gave a summary of our acoustic model settings.

Model	AM1	AM2	AM3
modeling unit	I-F	phone	I-F
model type	SI	SAT-FSA	SAT-FSA
training	ML	ML	MMIE+BMMIE
# codebooks	6K	10K	10K
genre-dep	no	yes	no
algorithms	-	STC/VTLN/SAT	STC/VTLN/SAT

Table 1: Acoustic model configurations.

The acoustic model training set consists of over 1300 hours of transcribed audio data released from the GALE program containing BN and BC shows. The BN sources are mainly from CCTV, NTDTV, PhoenixTV and VOA, while the BC sources are mainly from CCTV and PhoenixTV.

3.1. Discriminative Training on Acoustic Models

We apply MMIE [6] and BMMIE [7] on AM3 in order to improve recognition accuracy. MMIE aims to maximize the posterior probability of the reference compared to the competitors which are often encoded in a lattice. The objective function of MMIE is:

$$F_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{P_\lambda(X_r|M_{s_r})P(s_r)}{\sum_s P_\lambda(X_r|M_s)P(s)} \qquad (2)$$

where λ represents model parameters to be optimized; X_r is the r-th training utterance; s_r is the reference and M_s represents the corresponding HMM state sequence of sentence s.

Maximizing F_{MMI} improves the posterior probability of the reference in the lattice. This function can be optimized using the extended Baum-Welch (EBW) algorithm, and the update equations of Gaussian means and covariances, without the smoothing parts, are:

$$\hat{\mu}_i = \frac{x_i^{num} - x_i^{den} + D_i\mu_i}{\gamma_i^{num} - \gamma_i^{den} + D_i} \qquad (3)$$

$$\hat{\Sigma}_i = \frac{S_i^{num} - S_i^{den} + D_i(\Sigma_i + \mu_i\mu_i')}{\gamma_i^{num} - \gamma_i^{den} + D_i} - \hat{\mu}_i\hat{\mu}_i' \qquad (4)$$

where x_i and S_i are the weighted sums of features x_t and $x_t x_t'$ for the i-th Gaussian, respectively; γ_i represents the occupancy count; D_i is a constant which controls the learning rate and it is necessary to control the value of D_i to ensure Σ_i is positive definite. The subscripts **num** and **den** specify the statistics belonging to the numerator or denominator of F_{MMI}. For MMIE, the numerator statistics are the same as the statistics of maximum likelihood, while denominator statistics are collected from the lattice.

Boosted MMIE (BMMIE), is an extension to MMIE proposed by D. Povey et. al. [7]. Unlike MMIE, competitors are not considered equally important in BMMIE. While all paths in the lattice represent competitors, some paths may contain more error than the others. Hence, BMMIE boosts the importance of the competitors with larger error and aims to improve the confusable parts. The BMMIE objective function is:

$$F_B(\lambda) = \sum_{r=1}^{R} \log \frac{P_\lambda(X_r|M_{s_r})P(s_r)}{\sum_s P_\lambda(X_r|M_s)P(s)\exp(-bA(s, s_r))} \qquad (5)$$

where $A(s, s_r)$ is the raw phone accuracy of sentence s mentioned in [8]; b is the boosting factor which is larger than or equal to zero. Hence, the likelihood of the competitors with higher error is boosted.

The update equation of BMMIE is the same as MMIE, but the denominator statistics are altered when we compute the forward-backward scores on the lattices: the likelihood score of each word arc in the lattice is substracted by $b \times A(s, s_r)$.

In our system, the boosting factor, b, is 0.5 and we apply I-smoothing [8] with $\tau = 100$ for both MMIE and BMMIE training. We backoff to ML estimate for I-smoothing.

3.2. Language modeling

The LM training corpora has over one billion word tokens obtained from the Mandarin Gigaword V2, audio training transcripts and the web data released from the GALE program. Documents are first divided according to their news sources and a 4-gram LM is built for each source using the modified Kneser-Ney smoothing scheme using the SRI LM toolkit [9]. A background LM is generated by linearly interpolating the source-dependent language models using the DEV07 development set. Dynamic LM adaptation is applied using correlated latent semantic analysis (LSA) [10] to model topic correlation. In LSA training, a set of topic-dependent unigram LMs are built. In testing, the topic weights θ_k are incrementally adapted using the previously decoded utterances within the same show via the variational E-step $p_{lsa}(w) = \sum_{k=1}^{K} \hat{\theta}_k \cdot p(w|k)$ with $\hat{\theta}_k = \frac{\gamma_k}{\sum_{k=1}^{K} \gamma_k}$ where γ_k denotes the fractional posterior count of topic k inferred from the word context. The adaptive LSA unigram is then log-linearly interpolated with the background LM to decode the next utterances. The adaptive LSA unigram works like a cache-based LM. But instead of caching the word counts in the word history, latent topic counts are cached. Thus, this approach is shown to be robust against speech recognition errors.

Marginal LM adaptation [11] is an effective technique grounded by information theory where the background LM is adapted so that its unigram marginals match the in-domain unigram marginals. In our work, the in-domain marginals are estimated by $p_{lsa}(w)$. The adapted LM has the form: $Pr_a(w|h) = \frac{\alpha(w) \cdot Pr_{bg}(w|h)}{Z(h)}$ where $Z(h)$ is a normalization term to guarantee that the probability sums to unity. $\alpha(w)$ is a scaling factor which is commonly approximated as $\alpha(w) \approx \left(\frac{Pr_{lsa}(w)}{Pr_{bg}(w)}\right)^{\beta}$. This technique is expensive for decoding due to computing the normalization term $Z(h)$. On the other hand, it is cheap to apply for lattice rescoring since only the outgoing word links of the same context node are considered to compute $Z(h)$. The LM score of a word link (i, j) can be adapted analogously as follows: $lm_a(i, j) = Mass(i) \cdot \frac{\alpha(w(i,j)) \cdot lm_{bg}(i,j)}{\sum_{o \in Out(i)} \alpha(w(i,o)) \cdot lm_{bg}(i,o)}$ where (i, j) denotes a link from node i to node j and $w(i, j)$ is the word label associated to this link. $Mass(i)$ denotes the total probability mass of the outgoing links in $Out(i)$ coming from the same context node i. Using our GALE-2006 Mandarin evaluation system, 0.7% absolute reduction in character error rate (CER) is observed on the eval06 development set via dynamic LSA-based LM adaptation for decoding and lattice rescoring compared to the unadapted LM baseline with CER of 20.4%.

4. Evaluation Setup

During the GALE 2007 phase 2 retest evaluation, the dev07 set and the unsequestered portion of eval07 were used as the test sets. Dev07 consists of 2.5 hours of audio from various channels, including CCTV, NTDTV, and PhoenixTV. Eval07 unsequestered portion has 1.2 hours of audio from CCTV, NTDTV, PhoenixTV and AnhuiTV. Both dev07 and eval07 unsequestered portion consists of roughly 50% BN and 50% BC shows. The evaluation provides manual segmentation but not clustering on these two test sets. Hence, we can evaluate the performance of our segmentation approach and see how it impacts the overall performance. For clustering, our approach is compared with IBM's clustering procedure on a IBM preliminary ASR system, and we found our clustering performs better by 0.2% and 0.6% on dev07 and eval07 respectively. Thus, the results reported in this paper use CMU clustering.

Table 2 shows the performance of the final pass of our system using manual and automatic segmentations. For both segmentations, we performed speaker clustering using the approach discussed in section 2. The results show that, overall, the manual segmentation is around 1% absolute better than automatic segmentation, which is not a very big difference and it suggests the automatic segmentation of our system performs quite well. If we look at the performance break down, we can see that the difference on the BN shows is smaller, while the gap on the BC shows is significantly larger. This is expected, since BC shows contain complicated speaker turns and less fluent speech.

Figure 1 shows the performance at different stages of our system. Our system benefits from cross adapting I-F and genre dependent phone systems. After we obtained the ML model, we performed MMIE until the improvement converges on dev07. Due to time constraints leading up to the evaluation, we were only able to run one iteration of BMMIE on top of the MMIE model. However, BMMIE still improved the system slightly. Our system has a CER of 13.3% on dev07 and 13.5% on the eval07 unsequestered portion. We expect further improvement by starting BMMIE training from the ML model.

5. System Combination

Combining multiple systems together via cross-adaptation or confusion network combination (CNC) is an established method of improving performance [12]. We have found that our system can be most beneficial as part of a group of systems when both techniques are applied serially.

Model	dev07	eval07unseq
BN-manual	7.3	5.3
BN-auto	7.7	6.0
BC-manual	18.0	24.1
BC-auto	19.5	25.7
All-manual	13.3	13.5
All-auto	14.4	14.5

Table 2: Comparison on using manual and automatic segmentation in CER.

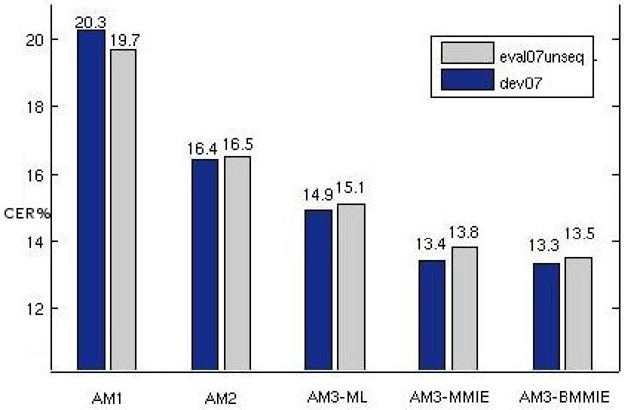

Figure 1: CMU Mandarin system performance at different development stages on the dev07 and eval07 (unsequestered portion) test set.

In addition, instead of combining CNs generated from each system's lattices, a method was developed for directly combining the lattices, resulting in similar improvements as from CNC but with a final hypothesis that is time-coherent, which may be more appropriate for post-STT sentence segmentation [13]. Lattice combination proceeds in six steps:

1. Calculate posterior probabilities of links in each lattice for use as confidence scores.
2. Scale confidence scores based on assigned system weights.
3. Group lattice links from all lattices into equivalence classes (as if for a CN).
4. Agglomeratively cluster lattice nodes in all lattices together by their similarity in time. Merge each cluster into a single node.
5. For links in each equivalence class, combine scores of multiple links with the same word.
6. Find the path that maximizes the minimum confidence over all sub-paths.

For terminological clarity, "links" in a lattice denote words, and "nodes" indicate time markers (word boundaries).

Significant improvements were observed only when the lattice nodes were aggressively merged, permitting paths through the combined lattice that were composed of links from multiple source lattices. Since the starting and ending times of the same (or very similar) words in different lattices invariably differed, nodes with similar times were agglomeratively clustered together until any further cluster joins would result in a cluster whose temporal span (node with highest time minus node with lowest time) exceeded a specified maximum. Increasing the maximum temporal span increases merging of the lattices, which permits a larger hypothesis space to be considered; however, it also decreases the temporal precision of the resulting hypothesis. For the results presented here, a maximum temporal span of 100ms was used, so word boundaries of the resulting hypothesis were guaranteed to be within 50 ms of their original time in the lattice.

Very short links (short words, silence tokens, partial words, etc.), i.e. those that occupy less time than the clustering procedure's maximum temporal span, can cause a variety of abnormalities in the merged lattice. Most obviously, some form of loop detection is required to assure that a very short link's start and end nodes are not merged together. More generally, the global ordering of links provided by the equivalence classes may be violated by the clustering procedure. This often occurs when one node with an incoming link in equivalence class i is merged with another node with an outgoing link in equivalence class i. To prevent these problems, each cluster is assigned values indicating the Latest Equivalence Class Before (LECB) among all incoming links and the Earliest Equivalence Class After (EECA) among all outgoing links. In order for two clusters to be merged, the following conditions must be satisfied: Cluster 1 LECB $<$ Cluster 2 EECA and Cluster 2 LECB $<$ Cluster 1 EECA. When two clusters are merged, the higher LECB and lower EECA is adopted for the new cluster.

In a traditional word lattice, the log acoustic and LM scores of different links are treated as (conditionally) independent and summed. In a merged lattice, a link is scored with a sum of posterior probabilities, which has a strong mutual dependence with other link scores before and after. In a CN, this mutual dependence is leveraged to maximize the confidence within each equivalence class independently, sacrificing any temporal constraints between classes. To overcome this mutual dependence in a merged lattice, the path was selected that maximized the minimum confidence score along all sub-paths. A divide-and-conquer strategy was used:

1. Given a start node i and end node j somewhere in the lattice, find the link (with start node m and end node n) limiting the single-path max flow of confidence from node i to node j.
2. Recursively call this procedure to find the best path from i to m and n to j.
3. Return a path consisting of the best path from i to m, the limiting link, and the best path from n to j.

This procedure was initiated with the initial and final nodes of the lattice. Table 3 compares the results of CNC and lattice combination (LC) on three systems. Two of the systems were built by IBM, and all three systems were cross-adapted against one another prior to combination. The performance of CNC and LC is very similar, and this parity was observed on Arabic language systems as well.

6. Conclusions and Future Works

We present our recent system development on the Mandarin BN/BC transcription system for the GALE evaluation. Our new system achieved good performance on a large-scale evaluation by using MMIE- and bMMIE-based discriminative training, unsupervised LM adaptation using LSA and lattice-based system combination. Our future development includes feature space discriminative training, improved LSA modeling, better system combination and tight coupling between the ASR and machine translation systems.

7. Acknowledgment

We would like to thank Thomas Schaaf and Mohamed Noamany for the development of the HMM segmenter. We would like to thank Kelvin Yong Qin, Qin Shi, Hong-kwang Kuo, Stephen Chu, and all those involved in building the IBM Mandarin systems at the IBM T. J. Watson and China Research Labs. This work is partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-2-0001. 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.

	CNC		LC	
System	dev07	eval07	dev07	eval07
CxTxB	10.32	10.66	10.31	10.68
BxTxC	9.44	9.74	9.50	9.76
TxBxC	9.34	9.51	9.34	9.41
TxBxC + BxTxC	9.03	9.28	9.03	9.29
TxBxC + CxTxB	9.07	9.18	8.94	9.17
TxBxC + CxTxB + BxTxC	8.92	9.14	8.86	9.03

Table 3: Results (CER) comparing various combinations of three Mandarin systems: C = CMU, B = IBM non-tonal (base), T = IBM tonal. CxTxB indicates the CMU system cross-adapted on output of the IBM tonal system, which was cross-adapted on the output of the IBM non-tonal system; others are similar.

8. References

[1] Y. C. Tam and T. Schultz, “Language model adaptation using variational bayes inference,” in *Proceedings of Interspeech*, 2005.

[2] Y. C. Tam and T. Schultz, “Unsupervised language model adaptation using latent semantic marginals,” in *Proceedings of the Interspeech*, 2006.

[3] Q. Jin and T. Schultz, “Speaker segmentation and clustering in meetings,” in *Proceedings of the International Conference on Spoken Language Processing*, 2004.

[4] H. Yu, Y. C. Tam, T. Schaaf, S. Stüker, Q. Jin, M. Noamany, and T. Schultz, “The ISL RT04 mandarin broadcast news evaluation system,” in *EARS Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[5] M. Gales, “Semi-tied covariance matrices for hidden Markov models,” *IEEE Transactions Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[6] V. Valtchev, J. Odell, P. Woodland, and S. Young, “MMIE training of large vocabulary speech recognition systems,” *Speech Communication*, vol. 22, pp. 303–314, 1997.

[7] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, “Boosted MMI for model and feature-space discriminative training,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[8] D. Povey, “Discriminative training for large vocabulary speech recognition,” Ph.D. dissertation, Cambridge University Engineering Dept, 2003.

[9] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proceedings of ICSLP*, 2002.

[10] Y. C. Tam and T. Schultz, “Correlated latent semantic model for unsupervised language model adaptation,” in *Proc. of ICASSP*, 2007.

[11] R. Kneser, J. Peters, and D. Klakow, “Language model adaptation using dynamic marginals,” in *Proc. of Eurospeech*, 1997, pp. 1971–1974.

[12] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus in speech recognition: Word error minimization and other applications of confusion networks,” *Computer Speech and Language*, no. 4, pp. 373–400, October 2000.

[13] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.