The TNT team system descriptions for IARPA OpenASR20

Zhiqiang Lv*, Jinghao Yan, Pengfei Hu, Jian Kang
Ambyera Han and Shen Huang
TEG AI
Tencent Inc, Beijing 100193, China
springhuang@tencent.com

Jing Zhao*, Guixin Shi, Guan-Bo Wang
and Wei-Qiang Zhang
Department of Electronic Engineering
Tsinghua University, Beijing 100084, China
wqzhang@tsinghua.edu.cn

Abstract*—This paper represents our architecture and a series of experiments on ASR for OPENASR 20. We both describe the system wi/wo constrained conditions and our post evaluation analysis, whereas the main acoustic model is trained by various shape of models in combination with CNN-TDNN-F-A, which aggregates Convolution Neural Network (CNN), Factored Time Delay Neural Network (TDNN-F) and self-attention (SAN). Such techniques as data cleanup, language tailored features, multi-band training, data perturbation, speaker adaptation, language model adaptation & rescore, pre-training and system fusions are incorporated. For un-constrained condition our end to end ASR systems with conformer in optimized loss and long sequence encoders are adopted. For Cantonese and Mongolian, we also adapt this challenging PSTN conditions using publicly available data in shape of wideband dictated speech with similar accent, respectively. Finally, series of system are submitted for this challenge. The results of our submitted system for constrained condition is between 0.4-0.6 and for unconstrained condition, most of the languages could be below 0.4, in terms of WER. We did NOT manage to submit all the systems so left results are summarized in this report.

Keywords—automatic speech recognition, low resource languages, OPENASR, speech pretraining

I. Introduction

Due to the lack of speech data, language script, lexicons, building an applicable ASR system for low resourced language is very challenging. The goal of the OpenASR20 Challenge is to assess the state-of-the-art ASR technologies under low-resource language constraints. It consists of performing ASR on audio datasets in up to ten different low resource languages, producing the recognized written text. For constrained condition, participants are only given 10 hours subset of labelled acoustic data but extra text data is unlimited. For unconstrained condition, teams may use speech data outside of the 10-hour subset marked for the Constrained condition for the language being processed, as well as additional publicly available speech and text training data from any languages. The evaluation dataset is provided a week before the system submission deadline.

The collaborated team consists of THU and MMT, hence its name TNT. The two teams work closely in model building and system fusion. We participate in all the 10 languages in *Constrained condition* and only two languages in *Unconstrained condition*, i.e. Cantonese and Mongolian. Unfortunately, due to the time limits, we didn't manage to submit all the systems before the deadline, so analysis of system fusion will only be illustrated based on scoring server reopened later while.

II. Constrained System

For our hybrid acoustic model, we propose the CNN- TDNN-F-A network as the main structure, which is trained with lattice-free maximum mutual information (LF-MMI) criterion [1]. The model introduces self-attention mechanism [2] to the combination of CNN and TDNN-F [3] in order to learn more positional information from the input.

Since the major challenge is low-resource condition, various kinds of data augmentations are combined to get additive improvement, such as speed perturbation [4], volume perturbation [4], Spec-Augment [5], Wav-Augment [6] as well as reverberation and noise [7]. These are proved to be effective to ASR performance especially under low-resource conditions.

Besides, systems' diversity is important for the final fusion. We have trained more than four systems for each language to make further use of the differences of single system by system fusion.

A. WORKFLOW

Due to the lack of resources (only 10 hours). NN-HMM hybrid acoustic model proves to be more promising in terms of performances for ASR than end-to-end structures in this particular under resourced condition [17], this hybrid structure is adopted throughout the Constrained condition. The main workflow is illustrated in Fig. 1, which consists of feature extraction, pre-processing, data augmentation, training, decoding and system fusion roughly.

*The first authors *Jing Zhao* and *Zhiqiang Lv* contributed equally to this work
The corresponding authors are *Wei-Qiang Zhang* and *Shen Huang*

This work was supported by the National Natural Science Foundation of China under Grant No.U1836219.

First a Gaussian Mixture Model (GMM) is trained by several training and force-aligning iterations, which includes Speaker Adaptation Trainings (SAT) GMM. Two-stage data clean-ups by both GMM-HMM and NN-HMM are applied to the original speech and text data. Then, various types of augmentation method can process the raw audio with additional diversity and enrich the quantity of training data, such as speed perturbation [4] and reverberation [7]. 20 times of the original quantity of data are obtained for the following training, where high-resolution MFCC, pitch and lower rate i-vectors speaker features are extracted with shared phoneme boundary produced by the aforementioned clean-up results on original data. Finally, system fusion operates more on acoustic model level, i.e. acoustic models are trained with various types of neural networks, which are then composed to build separate ASR systems that output multiple hypotheses in form of lattices, which are then fused in the final stage.

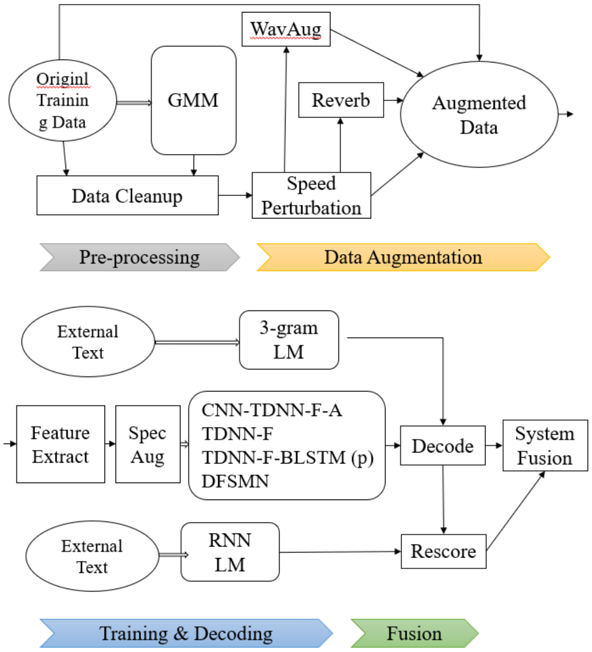

Fig. 1 Workflow of the ASR systems: The whole system process can be roughly divided into data feature extraction, cleanup, pre-processing, data augmentation, training, decoding and system fusion.

Besides, Spec-Augment [5] is applied to the combined acoustic features to augment data in feature level. As for language model, we build a N-gram LM by SRILM [8] with some extra text data from IARPA Babel program [9], and a Recurrent Neural Network (RNN) LM is also used to rescore lattices after decoding. Finally, several different system outputs are fused in lattice and 1-best levels, respectively. The details are described in the following sections.

B. SPEECH PRETRAINING

It is a common way to utilize bottle neck layer (BN) or posterior features (PF) for multi-lingual acoustic model training from a pre-trained phoneme classification task. Unfortunately, in this level of low resource data (10*10=100hours), it is observed that the contributions of multi-lingual BN or PF features are subtle, even inferior to model trained by 10 hours mono-lingual data.

Speech pretraining using Transformer Encoder Representation for Speech (Tera) [20] is explored both on constrained and unconstrained conditions. As far as we know this may be the first system in ASR competition that adopts pre-training. We trained a Tera transformer using constrained speech, with Tera feature extractor, we could get a new 768-dimension feature representation. CNN-TDNN-F-A systems trained using Tera features have shown much faster convergence than traditional features. However, as in Tab. 1 for Cantonese DEV set. Tera features extracted CNN-TDNN-F-A system don't outperform significantly than those trained on traditional features due to a lack of speech data (only 10 hours) in Constrained condition. However, it is still fortunate to discover that Tera-based systems provided quite good compensation for traditional features at system fusion phase.

TABLE 1
Results for hires-mfcc and pre-trained features for Cantonese (DEV set, our computation)

Features	**WER**	**CER**
Hires-MFCC	0.487	0.456
Hires-MFCC+Pitch	0.485	0.444
Tera	0.510	0.476
1+2+3 fusion	0.468	0.424

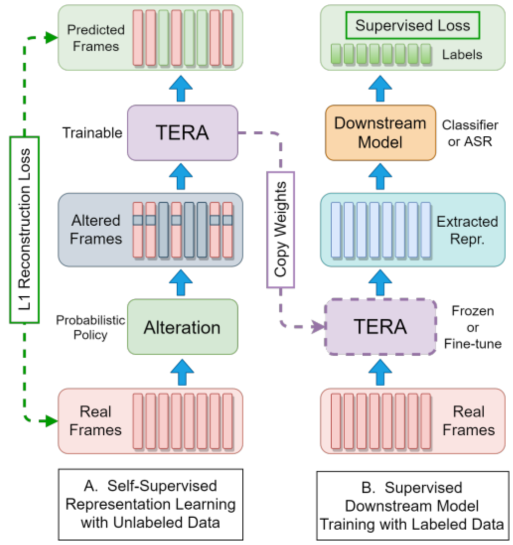

Fig. 2 Pre-training using Transformer Encoder Representation for Speech

C. ACOUSTIC MODEL

1） CNN-TDNN-F-A architecture：

We propose CNN-TDNN-F-A network as the main neural network acoustic model, which combines Convolution Neural Network (CNN), Factored Time Delay Neural Network (TDNN-F) [3] and Self-Attention Network (SAN) [2]. The architecture is displayed in Fig. 3. The popular TDNN-F networks are the basic part of our acoustic model, which is structurally the same as TDNN whose layers have been compressed via SVD, but are trained from a random start with one of the two factors of each matrix constrained to be semi-orthogonal in order to prevent instability in back propagation. A regular TDNN-F block consists of a linear layer, an affine component, an ReLU nonlinearity component, and batch normalization operation followed by dropout. The CNN-TDNN-F-A network contains 11 TDNN-F blocks (9 before SAN and 2 after) in total with hidden dimension of 768 and a bottleneck dimension of 160. For different system settings, the bottleneck dimension is also set to be 120 or 256.

CNN has been applied to the ASR task successfully by introducing three extra concepts over the simple fully connected NN: local filters, max-pooling, and weight sharing [11]. Previous experiments have showed the efficiency of CNN-TDNN [12], whereas TDNN is replaced with TDNN-F which proves to be better in low resource scenario. In our model, the convolutional block is composed of a convolutional layer and an ReLU component followed by batch normalization. We adopt 6 convolution blocks at the beginning with concatenation of i-vectors and hires-MFCCs (with pitch) as input.

SAN layer has been successful with multi-head attention which allows the networks to jointly attend to information from different subspaces at different positions [13]. Besides, In [2] the self-attention layer was adopted in a time-restricted fashion, which is more suitable for ASR. We combine SAN with CNN-TDNN-F, and thus obtain final CNN-TDNN-F-A architecture. The SAN is composed of an affine component, an attention nonlinearity component, and an ReLU non-linearity component followed by batch normalization. The location of the layer should be close to the end of the network. SAN block the third layer from the bottom. In detail, the multi-head attention component has 20 attention heads along with a key-dimension of 8 and a value-dimension of 16.

The CNN-TDNN-F-A structures are used prevalently in all of the systems. Besides, other neural network acoustic models such as TDNN-F, TDNN-F-BLSTM(p), DFSMN [18], are also explored as additional fused system. From final results, although combined system strikes the best performance with additional 0.02-0.03 gains in terms of WER, the most complementary and critical part is still CNN-TDNN-F-A.

2） Features and Speaker adaptation：

Apart from hires MFCC, for tonal languages such as Cantonese and Vietnamese, 3 pitch features, i.e. POV, mean subtracted log pitch and delta of raw pitch are added additionally, which proves to be more compatible with tonal languages.

i-vectors are fixed-length vectors containing speaker information and have become a common technique for speaker-id related speech recognition in fused with non i-vector system [17] in that only one pass decoding is needed. In our speaker aware training, i-vectors are trained based on a diagonal UBM derived extractor [14]. In order to better adapt to CNN structure, the extracted 100-dim i-vectors are mapped to 200-dim by linear transformation before concatenating with MFCC features. Preliminary results show that 0.01-0.03 absolute gains are observed by using speaker aware training.

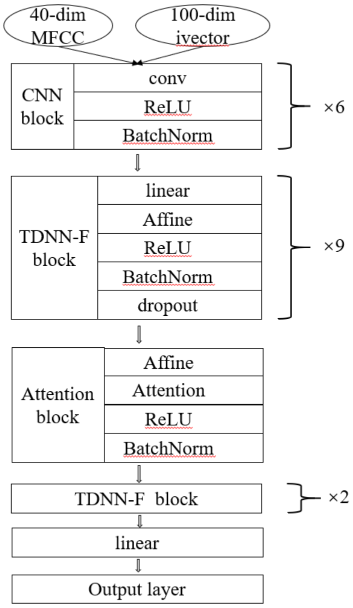

Fig. 3 CNN-TDNN-F-A architecture

3） Training pipeline：

Following the hybrid system training pipeline, we first train GMM-HMM models. To produce high-quality alignments to force-align the training dataset for the NN-based acoustic model. We use Perceptual Linear Prediction (PLP) feature with pitch features. The routine iterative training process is applied, including mono-phone, and triphone model training, Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) based model training and speaker adaptive training (SAT). Alignments and numerator lattices generated from the HMM-GMM model are used for NN acoustic model, which generate second-pass refined alignments.

For training procedure, the CNN-TDNN-F-A acoustic model is trained with chain model using LF-MMI criterion with cross-entropy (CE) regularization [1]. The batch size is set to 128 or 64 with 6 epochs training in total. The initial learning rate is 0.0005 and the final learning rate is 0.00005.

D. LANGUAGE MODEL

N-gram language models are trained by SRILM [8], with some extra text data from IARPA Babel program [9] in addition to transcripts of the provided training data. Language scripts in “training” part related to IARPA Babel program is adopted. e.g. Cantonese refer to the package “IARPA-babel101b-v0.4c-build”. Among the 10 test languages, we realized only 9 languages are available in IARPA babel except for Somali, for which LDC2018T11 is chosen as extra text data.

Besides, we also employ lattice rescoring with NN-based LMs, which are composed by several TDNN-LSTM networks [15]. The RNN language model is trained using a linear approximation of the standard cross-entropy objective as denoted in Equation (1):

$$O_{LM} = z_j + 1 - \sum_i \exp z_i \quad (1)$$

Where z denotes the neural-network output before soft-max layer, and j is the index of the correct word. Actually this objective is a lower bound on the standard CE, which allows model to self-normalize during inference and thus saves computational time. For word feature representation, besides a one-hot representation for the most frequent words, letter n-grams feature is also used in generating word-embeddings. This allows us to utilize the sub-word information of word and shapes better embedding to words that appear relatively rare in the corpus. At last, pruned lattice-rescoring is implemented to combine the weights with the original n-gram weights.

Two stages of LM rescoring using original and reversed text are adopted. In each stage a heuristic score for each arc to be expanded in the output lattice is computed by which arc keep or deletion decided considering both historic and future information in the lattice [21]. The training data for the NN-based LMs are the same as above.

The final weights of RNNLM is chosen to be 0.5, unlike other reported systems, it is found that the factor of weight is trivial unless it is not set to be too biased.

E. DATA PROCESSING

Data quality is critical for removing noises while data robustness is also critical for adding diversities in the trained models. In our work both strategies are considered in the system.

1) Clean-up：

Data clean-up is performed to remove the bad portions of the training transcripts by using a biased ASR which is built by N-gram biased language model with garbage state projecting the nuisance part of the speech. Other minor modifications of transcripts, such as allowing repetitions for disfluencies, and adding or removing non-scored words are also incorporated in this stage. The clean-up relying on SAT-GMM against NN based models proves to be similar in terms of WER in DEV sets for most languages. However, NN based model is believed to be a better choice for the system.

The operation seems to be effective for some languages such as Cantonese than other languages such as Mongolian. It is plausible that Sino-Tibetan language is much vulnerable to label noise whereas label mistake is not completely fatal in spelling languages such as Pashto or Mongolian. Furthermore, a more biased LM for building biased ASR is preferred in our evaluation set in order not to excessively delete correct labels.

2) Augmentation：

Since the major challenge is to deal with low-resource condition, it’s essential to adopt appropriate methods of data augmentation. In order to make full use of the limited training data, we adopt several popular techniques at the same time to enhance the data robustness and make our system more invariant to properties of the evaluation data.

- Speed and Volume Perturbation: In [4], speed and volume perturbation are proposed as effective data augmentation methods by processing the raw signal with 3 versions of volumes and speed;
- Reverb and Noise Perturbation: Given that the DEV set doesn’t contain much music, it is appropriate to enrich the data with reverberation and noise [7], noise from MUSAN database is chosen as stationary noise and noise from other speakers of the training corpus is chosen as babble noise. Random high SNR (>10dB) and noise types are added in each sample. Then reverberation is applied on top of them using the simulated RIRs with room sizes uniformly sampled from 1 to 30 meters. Since that this is a close mic scenario, the reverberation ratio should be trivial;
- Wav-Aug: The recent Wav-Aug is a time domain data augmentation library, which integrates 5 augmentations [5]. Since most of the augmentations have already been implemented by the methods mentioned above, we only adopt pitch modification to deal with pitch changes, band rejective filtering and time masking to enhance the robustness of frequency and time domains, respectively;
- Spec-Augment: Different from the aforementioned methods, Spec-Augment [4] is applied directly to the feature level before input to the acoustic model. The policy consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps;

The alignments for these augmented data are obtained from their clean counterparts. We don’t have time to get an apple-to-apple comparisons among the four methods. However, we empirically discovered three rules: 1) Don’t apply multiple augmentations for a single utterance; 2) Don’t overuse data simulation as convergence of results will be observed at some point; 3) Speed augmentation is the most complementary approach that can be safely utilized.

All the approaches are randomly chosen to enrich speech data in 20x.

F. PRE-AND-POST PROCESSING

Unlike DEV set, in EVAL set there is NO “segments” file provided, which means VAD is a necessary part of the system and it is also essential to fuse an enhanced result from various types of system.

1) Speech Activity Detection：

For the evaluation period, Speech Activity Detection (SAD) is a necessary operation to segment the audio appropriately so that we can decrease the loss of useful speech clips and improve the decoding efficiency and accuracy. In our SAD algorithm [16], we combine a convolutional recurrent neural network (CRNN) and a recurrent neural network (RNN) to make the system more robust. In addition, we add a speech-enhancement module and a one-dimensional dilation-erosion module. For each audio input, “fbank” features are extracted and speech existence is explored in every frame. Finally, the output is processed by post smoothing and hold on&off module to form a final SAD boundary. In DEV set we found that SAD is critical in that using a machine learning based SAD degrades WER performance in

Cantonese from 0.456 to 0.483 in CNN-TDNN-F-A single system, against labeled segments.

2) Decoding

In our systems, we use a WFST-based method for decoding in KALDI. For the first pass, we simply use N-gram as the decoding language model. The decoding beam is set to 15.0. while the beam used in lattice generation is 8.0. The LM weight is chosen from 8 to 12. Besides, a two-layer TDNN-LSTM language model is trained for lattice rescoring as illustrated in section D [15].

3) System Fusion

Setups with different features, augmentations, acoustic network architectures are chosen for knowledge sources to be fused. After lattices are obtained, lattice fusion [30] is chosen rather than ROVER [9], which fuses hypothesis in more paths from resulting graph that yields better results than 1-best.

4) Results Filtering

Final ASR results are obtained from lattice, we filter the word lists by their corresponding degree of confidence. The threshold is set to 0.3, which means the recognition results with confidence value below 0.3 would be abandoned. The operation is effective to reduce the insertion error of WER, especially miscues and verbal pauses.

G. RESULTS

Our systems' performance of the constrained condition on the evaluation set is shown in Tab.2, which are released by NIST OpenASR scoring server.

The results on the left-hand side of the arrow are the results by team TNT. Note that the WER on DEV set is a first pass main system with TDNN-F-A acoustic model without any other pre-post processing, pretraining, fusion, etc. However, in EVAL set the final results are generated from fused systems, that's no wonder why the results on EVAL are much better than DEV.

TABLE 2

WER OF ASR SYSTEMS ON DEV AND EVAL SET (CONSTRAINED)

(For Mongolian, the results are submitted after close time)

	WER on DEV	**WER on EVAL**
Amharic	0.487->N/A	0.458->N/A
Cantonese	0.483->**0.412**	0.436->**0.402**
Guarani	0.499->N/A	0.461->N/A
Javanese	0.571->N/A	0.521->N/A
Kurmanji-Kurdish	0.671->N/A	0.669->N/A
Mongolian	0.524->**0.461**	0.454->**0.449***
Pashto	0.496->N/A	0.486->N/A
Somali	0.592->N/A	0.591->N/A
Tamil	0.691->N/A	0.661->N/A
Vietnamese	0.488->N/A	0.460->N/A

The results on the right-hand side of the arrow are the results fused by TNT (mainly CNN-TDNN-F-A) and MMT (pretraining and other architectures). Note that the WER reduction ranges from 0.03-0.07 in terms of absolute WER. However, we didn't manage to submit Mongolian EVAL on time due to the time limits. A post submission shows that the result WER is 0.449.

III. UNCONSTRAINED SYSTEM

For unconstrained condition, due to the time limit, we only participate in Cantonese and Mongolian. The main fused system is roughly the same with constrained system, except that end to end (e2e) ASR training, hybrid bandwidth acoustic modeling, language optimization and hybrid-e2e fusions are explored additionally.

A. SPEECH PRETRAIN

In unconstrained condition, it is allowed to use extra data for pretraining acoustic model using publicly available data. Since that speech pretraining is language universal, in our experiment, unlike constrained condition, all the available 25 IARPA babel training data provided are used to train a 768 dim feature extractor for the downstream task. From preliminary results, the gain of speech pretrain in unconstrained condition is much larger than constrained condition, which is 0.04-0.05 in terms of WER.

B. END TO END SYSTEM

It is shown that end to end ASR strikes better performance than hybrid system in rich resourced condition. Also, end to end ASR has a good compatibility to legacy system [31]. In our system different fusion strategies are tailored for each system.

The end to end ASR in our system is based on recent conformer [28] structure. As transformer, Conformer model is composed of two parts: encoder and decoder. The encoder part is composed of a convolution subsampling layer and several conformer blocks. The role of conformer blocks is similar as that of transformer, which is composed of four modules: a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module in the end. The decoder part in conformer is also the same as in transformer.

A simplified version of conformer block consisting of Rpe+XL positional encoding, CNN and FNN are proposed without WER degradation, which is implemented in ESPNET. In order to deal with long utterance and data sparsity in low resource telephony condition, such techniques are proposed in the system as is shown in Fig 4:

- Rpe+XL transformer: Relative positional encoding to deal with repetition and instability brought by transformer in long utterances;
- FL: Using focal loss to deal with unbalanced distribution of tokens;
- Ss: Using scheduled sampling to deal with inconsistency during training and inference;

In our preliminary experiments, conformer achieve 3 to 10% relative improvements over traditional transformer (0.45->0.43) in Cantonese. The number of conformer blocks in encoder and decoder of our system is 12 and 6, respectively. The encoder and decoder dimension are both 2048. The attention layer contains 4 heads and 256 units per head.

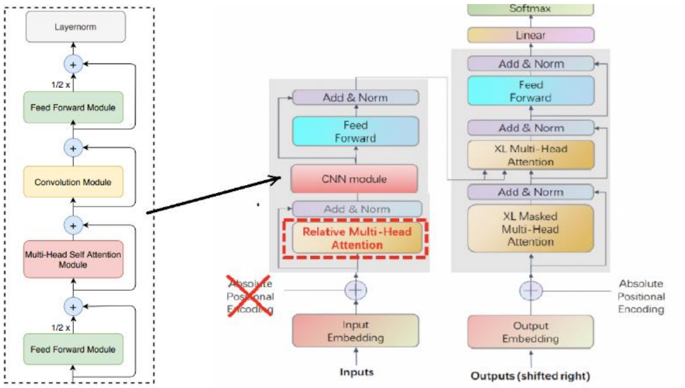

Fig. 4 end to end structure of conformer model

C. ACOUSTIC MODEL FOR HYBRID SYSTEM

For acoustic model, we have accumulated additional training data for some languages, such as Cantonese and Mongolian. Unfortunately, most of the OPENASR DEV data is recorded in telephony channel with a sampling rate of 8khz and we don't have any data that either matched for the target PSTN telephony condition or with the accent. Most of the extra data at our hands are wideband 16khz speeches, in order to utilize these data, we first train a mixed bandwidth acoustic model [22] with non-overlapping set of band-wise filters in 0-4khz and 4-8khz as illustrated in Fig.5. For 8khz data, Spectral Band Replication (SBR) is involved both in training and testing to fill out high frequencies, resulting in a feature extractor for 16khz.

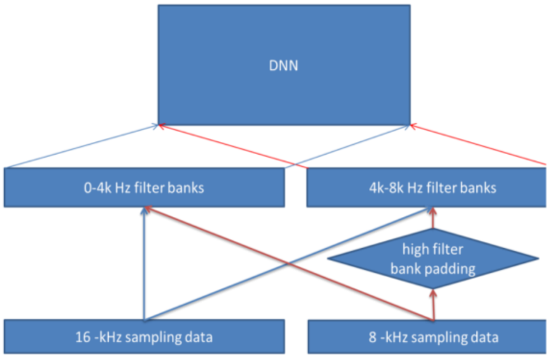

Fig. 5 Multi-band speech models for 8-16khz hybrid recognition

For Cantonese, 2000h wideband dictated speeches from Speech Ocean Inc. and Huiting Tech Inc. [23][24] and 140h narrow band (8khz) speeches from IARPA Babel are applied [9]. For Cyrillic Mongolian, only 10h wide band speeches from Mozilla [25] and 50h narrow band speech from IARPA Babel are available. However, we believe that rich resources of Inner Mongolian speech in China is also beneficial for Cyrillic Mongolian speech recognition, as a result, an extra 500h wideband dictated speech from Speech Ocean [26] and 100h dictated speech from M2ASR project [27] are incorporated. The goal is to utilize automatic Traditional to Cyrillic Mongolian conversion to make them compatible with each other. Details are illustrated in Tab 3.

TABLE 3

Extra datasets for Cantonese and Mongolian

	Narrowband(8khz)		Wideband(16khz)	
Language	**Data source**	**Duration**	**Data source**	**Duration**
Cantonese	IARPA Babel	~140h	Speech Ocean	~1000h
		~1000h	Huiting Tech	~1000h
Mongolian	IARPA Babel	~50h	Mozilla	10h
Inner Mongolian			M2ASR	~100h
			Speech Ocean	~500h

All the acoustic model is trained with lexicon from IARPA Babel program. Three-stream system is proposed:1) The first way system is trained by original 8kh Babel data using hybrid model; 2) For the second way system all the 8khz and 16khz speeches are trained using the above mixed bandwidth training, followed by weight transfer fine-tuned on SBR 16khz OPENASR training data for the target language; 3) For the third way system, the above mixed wideband e2e model is trained and tuned towards same SBR 16khz OPENASR data;

Notice that for each steam of the above two hybrid system the result is also a combined version of various acoustic models. The whole procedures for the main streams of system are illustrated in Fig. 6. The first two-stream hybrid system output lattice that can be rescored and fused using lattice combine [30], the final lattice is timed aligned to form a CTM result, which is fused with e2e CTM via ROVER on 1-best sequence level;

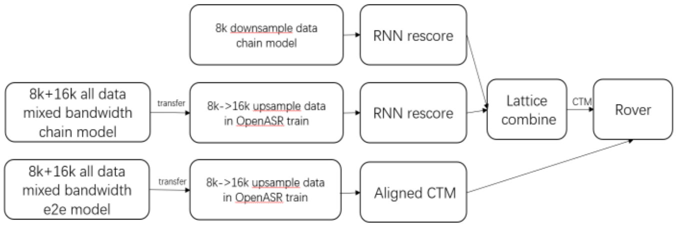
8k downsample data chain model
RNN rescore
8k+16k all data mixed bandwidth chain model
transfer
8k->16k upsample data in OpenASR train
RNN rescore
Lattice combine
CTM
Rover
8k+16k all data mixed bandwidth e2e model
transfer
8k->16k upsample data in OpenASR train
Aligned CTM

Fig. 6 3-way multi bandwidth hybrid & 16khz end to end system fusions

D. LANGUAGE MODEL

It is easy to acquire large amount of text data using crawled engine from publicly available news. However, we realize that the domain mismatch is severe for extra language model and performance on DEV is deteriorated considerably with extra crawled text. Therefore, only the crawled text for Cantonese (about 2GB) is used for word segmentation result correction in order to have a better WER. (not CER)

The main setup is the same with constrained condition, except that speech transcripts in the above extra acoustic data are incorporated in the training resources for the two participated languages.

E. LANGUAGE OPTIMIZATION

Cantonese is a language within the Chinese language family. Since the Cantonese vernacular text data is irregular, we obtain certain additional text data through web crawling, and use regular methods to correct common errors in the text data, such as abbreviations and typos.

Text segmentation is needed to word sequence to calculate the word error rate in the evaluation. For e2e system, characters are used as the modeling unit rather than word. In order to solve this, we use all the crawled and Babel text data to train a text segmentation model through Cantonese BERT pre-training, as is shown in Fig. 7. Our UER tool for multi-lingual Bert model (*https://github.com/dbiir/UER-py*) is applied to train a multi-lingual encoder from mono-lingual data of Mandarin and Cantonese; Then a word-segmentation model is trained using Mandarin segmented data from the above pre-trained Bert model; Finally, low resource Cantonese word segmentation model is fine-tuned by Babel Cantonese dataset;

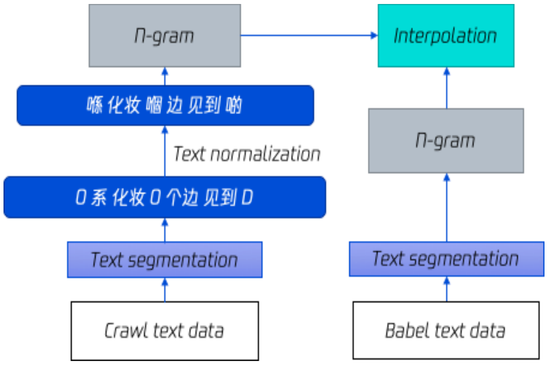

Fig. 7 Process for training Cantonese word segmentation model

Mongolian is a typical agglutinative language that relies on suffix chains in the verbal and nominal domains. To begin with, we check the words out of vocabulary in the Mongolian speech transcription and find some prefix word. For example, in the case of "Алтайг- Алтайгаас", "Алтайг" is the prefix of the word "Алтайгаас" and not found in the dictionary. Pronunciations are expanded into training dictionary for about 30 common OOV words by looking up to the original linguistic dictionary in Mongolian.

In the set of original speech files, more than 10 files (almost 1/10) are formed as ".wav" (44.1khz) while most of files are formed as ".sph" (8khz) in some languages such as Mongolian. In our experiments 44.1khz speech is treated as 16khz for mixed bandwidth recognition while 8khz speech is up-sampled.

As known, there are several Mongolic languages or dialects which are roughly intelligible with each other. The speech data in OPENASR Mongolian is *Halh Mongolian* collected in Mongolia. Here we apply 500hrs dictated *Chakhar Mongolian* speech data is provided by Speech Ocean, which is spoken in the Inner Mongolia region of China and sampled at 16khz. While *Halh Mongolian* in Mongolia is texted as Cyrillic character, the traditional Mongolian character is used in the Inner Mongolia region as is shown in Fig. 8. Besides the text, there are also differences in phoneme types, pronunciations even for the same word in dictionary. For acoustic model, we train the base model using Inner Mongolian speech and do transfer learning using the perturbed IARPA Babel *Halh Mongolian* speech data. During the transfer stage, the soft-max layer is substituted, and the intermediate layers are updated and tuned. The initial experiments show that transfer learning is very helpful for acoustic modeling and reduce the WER by 15% relatively.

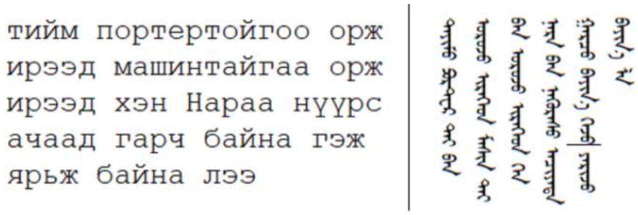
тийм портертойгоо орж
ирээд машинтайгаа орж
ирээд хэн Нараа нүүрс
ачаад гарч байна гэж
ярьж байна лээ

Fig 8. Left figure: the Mongolian Cyrillic text used in Mongolia; Right figure: Traditional Mongolian text used in the Inner Mongolia region of China.

For language model, besides *Mongolian Cyrillic* text in Babel transcription, a seq2seq transformer model is trained to transfer *traditional Mongolian* text to *Mongolian Cyrillic* text to enrich the corpus for *Cyrillic* language model. The performance for language text transfer in terms of CER and WER are 0.0750 and 0.196, respectively. However, because that functional characters in *traditional Mongolian* text are still an unsolved problem, there are considerable errors in the transformed *Cyrillic text*. In final ASR results, small gains can still be observed in DEV set for about 0.02-0.05 in in terms of absolute WER;

F. SYSTEM FUSION

For better performance, we build several different systems for system fusion as in Tab 4. The Cantonese results of all the fused systems in DEV set are as below, system 1 and 2 are results by hybrid model, whereas 1 is trained by 8kh Babel data, 2 is trained by extra 16kh data with mixed bandwidth training, it can be observed that extra data and high frequency bring WER from 0.431 to 0.410. Meanwhile, e2e ASR system outperforms hybrid Model in single system with WER of 0.386.

Using lattice combine, fused system 1 and 2 strikes WER of 0.404 against single system, meaning that unconstrained 16khz data has a good compensation with Babel 8kh data. The final results fused by 1-2 and 3 with ROVER achieve best performance from 0.386 to 0.370. With extra post confidence filter in Section 2F, WER can be further dropped to 0.361.

TABLE 4
Unconstrained System Fusion Results for Cantonese
(DEV set, our computation)

System	Data&Model	Bandwidth	WER	CER
1	8k Babel chain	8k	0.431	0.398
2	8+16kall chain	Mixed band	0.410	0.370
3	8+16kall e2e	Mixed band	0.386	0.347
1+2	Fusion		0.404	0.370
1+2+3	Fusion		0.370	0.343
1+2+3 filter	Fusion		0.361	0.332

G. RESULTS ON EVAL SET

Our systems' performance of unconstrained condition on the evaluation set is shown in Tab.5, which are released by NIST OpenASR scoring server, notice that the results computed by dashboard in NIST OpenASR scoring server is much better than our scoring results in Tab. 4 (WER from 0.361->0.335), we realize that it is because we count language miscues, pauses, and other non-verbal speech as errors.

TABLE 5

WER OF ASR SYSTEMS ON DEV AND EVAL

	WER on DEV		**WER on EVAL**	
	Constrain	**Unconstrain**	**Constrain**	**Unconstrain**
Cantonese	0.412	**0.335**	0.402	**0.320**
Mongolian	0.461	**0.381**	0.449*	**0.406***

It can be observed that by using extra data, an absolute 7-8% WER reduction can be achieved such as Cantonese and the CER is even much lower than 0.300 (0.264). In practice, Speech recognition accuracy for Sino-Tibetan languages relies much on CER rather than WER, WER for these types of languages is largely dominated by word segmentation error, which may incur a biased result.

Notice that for EVAL of Mongolian unconstrained condition, we didn't manage to submit our fused system, so 0.406 in the dashboard is actually a single HMM-NN hybrid system performance using CNN-TDNN-F-A and other data pre-post processing techniques above (the fused system results maybe below 0.4). As for intelligibility, for these types of Arabic & Altai languages such as Mongolian, same word may has many suffix and majorities are correct, we also believe that WER will also under estimate the actual performance.

IV. Hardware And Time Description

The hardware of our proposed system is shown in Tab.6. As for the required time for Constrained condition, the elapsed wall clock time for training is approximately 3 hours for one system of each language, whereas corresponding total CPU time is about 2.5 hours, and the total GPU time is 3 hours; For unconstrained condition the elapsed wall clock time for training is about 170 hours (7 days), where most of the consuming time is spent on conformer e2e model training, the GPU time is about 150 hours and CPU time is about 10 hours;

TABLE 6

HARDWARE DESCRIPTION

OS	CentOS 7.4 64-bit
CPU num	40
CPU description	112, Intel(R) Xeon(R) CPU E5-4650 v4 @ 2.20GHz
GPU num	16
GPU description	Tesla V100 SMX2 16GB
RAM	256GB
RAM per CPU	128GB
Disk storage	About 3TB

Acknowledgment

This work is supported by research fund of Tsinghua Universiy -Tencent Joint Laboratory for Internet Innovation Technology. And special thanks are given to all who supported TNT team during this challenge.

References

[1] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in Interspeech. San Francisco, CA, USA: ISCA, Sep 2016, pp. 2751–2755.

[2] D. Povey, H. Hadian, P. Ghahremani et al., "A time-restricted self attention layer for ASR," in proc. ICASSP. Calgary, AB, Canada: IEEE, Apr. 2018, pp. 5874–5878.

[3] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in Interspeech, 2018, pp. 3743–3747.

[4] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in INTERSPEECH. Dresden, Germany: ISCA, Sep 2015, pp. 3586–3589.

[5] D. S. Park, W. Chan, Y. Zhang et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in proc. interspeech, Graz, Austria, Sep. 2019, pp. 2613–2617.

[6] E. Kharitonov, M. Rivi`ere, G. Synnaeve, L. Wolf, P. Mazar´e, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," CoRR, vol. abs/2007.00991, 2020.

[7] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer and Sanjeev Khudanpur, A study on data augmentation of reverberant speech for robust speech recognition, ICASSP 2017

[8] A. Stolcke, "SRILM - an extensible language modeling toolkit," in proc. ICSLP - interspeech, Denver, Colorado, USA, Sep. 2002.

[9] shttps://www.iarpa.gov/index.php/research-programs/babel

[10] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings. IEEE, 1997, pp. 347–354.

[11] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in ICASSP. Kyoto, Japan: IEEE, Mar 2012, pp. 4277–4280.

[12] A. Georgescu, H. Cucu, and C. Burileanu, "Kaldi-based DNN architectures for speech recognition in romanian," in SpeD. Timisoara, Romania: IEEE, Oct 2019, pp. 1–6.

[13] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30: NIPS, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[14] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, 2013, pp. 55–59.

[15] X. Liu, Y. Wang, X. Chen, M. J. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 4908–4912.

[16] G.-B. Wang and W.-Q. Zhang, "A fusion model for robust voice activity detection," in 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, 2019, pp. 1–5.

[17] The JHU ASR system for VOiCES from a Distance challenge 2019", Yiming Wang, David Snyder, Hainan Xu, Vimal Manohar, Phani Shankar Nidadavolu, Daniel Povey, Sanjeev Khudanpur, Interspeech 2019

[18] Zhang S, Liu C, Jiang H, et al. Non-recurrent Neural Structure for Long-Term Dependency[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(4): 871-884

[19] Haşim Sak, Andrew Senior, Françoise Beaufays, Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, INTERSPEECH, 2014

[20] Andy T. Liu, Shang-Wen Li and Hung-yi Lee. TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech. ArXiv, 2007.06028

[21] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5929–5933.

[22] Jinyu Li, Dong Yu, Jui-Ting Huang, Tifan Gong. Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM, IEEE Workshop on SLT, 2012

[23] http://www.speechocean.com/datacenter/details/709.html

[24] http://www.huitingtech.com/en/dataInfo.action?id=1005

[25] https://pontoon.mozilla.org/mn/common-voice/project-info/

[26] http://www.speechocean.com

[27] Dong wang, et al. M2ASR: Ambitions and first year progress. O-COCOSDA. 2017

[28] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented Transformer for Speech Recognition[J]. arXiv preprint arXiv:2005.08100, 2020.

[29] Zhou P, Fan R, Chen W, et al. Improving Generalization of Transformer for Speech Recognition with Parallel Schedule Sampling and Relative Positional Embedding[J]. arXiv preprint arXiv:1911.00203, 2019.

[30] Xu, H., Povey, D., Mangu, L., & Zhu, J. (2010, March). An improved consensus-like method for Minimum Bayes Risk decoding and lattice combination. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4938-4941). IEEE.

[31] Jeremy Heng, et al, Combination of end-to-end and hybrid models for speech recognition, interspeech 2020.