The NAIST English Speech Recognition System for IWSLT 2015

Michael Heck, Quoc Truong Do, Sakriani Sakti, Graham Neubig, Satoshi Nakamura

Augmented Human Communication Laboratory,
Graduate School of Information Science,
Nara Institute of Science and Technology,
Nara, Japan

{michael-h,do.truong.dj3,ssakti,neubig,s-nakamura}@is.naist.jp

Abstract

The International Workshop for Spoken Language Translation (IWSLT) is an annual evaluation campaign for core speech processing technologies. This paper presents Nara Institute of Science and Technology's (NAIST's) contribution to the English automatic speech recognition (ASR) track for the 2015 evaluation campaign. The ASR systems presented in this paper make use of various front-ends, varying deep neural net (DNN) acoustic models and separate language models for decoding and rescoring. Recognition is performed in three stages: Decoding, lattice rescoring and system combination via recognizer output voting error reduction (ROVER). We discuss the application of a rank-score based weighting approach for the system combination. Also, a Gaussian mixture model hidden Markov model (GMM-HMM) based speech/non-speech segmenter makes use of said combination scheme. The primary submission achieves a word error rate (WER) of 9.5% and 10.1% on the official development set, given manual and automatic segmentation respectively.

1. Introduction

The 2015 evaluation campaign of the 12th International Workshop on Spoken Language Translation (IWSLT) offers participants the opportunity to advance the state-of-the-art in core tasks of spoken language translation. This involves the tasks of automatic speech recognition (ASR), machine translation (MT) and the combination of ASR and MT, the task of spoken language translation (SLT) itself. All tasks are performed and evaluated on multi-topic TED (short for Technology, Entertainment, Design) and TEDx (licensed spin-off) conference talks (`http://www.ted.com`). This paper describes Nara Institute of Science and Technology's contribution to this year's evaluation campaign by participation in the ASR track for the English language. The goal of this track is the automatic transcription of unsegmented talks, thus the task is two-fold: automatic segmentation followed by speech recognition. We describe the development and application of a Gaussian mixture model (GMM) based speech/non-speech segmenter using the Janus speech recognition toolkit [1] (see Section 4) and the ASR system development and decoding utilizing the Kaldi speech recognition toolkit [2] (see Sections 2 and 5 respectively). Our speech-to-text system makes use of various front-ends, deep neural net (DNN) acoustic models and several language models for decoding and rescoring.

High performance speech recognition often makes use of system combination approaches, especially if recognition in real-time is not a major concern. Recognizer output voting error reduction (ROVER) [3] and confusion network combination (CNC) [4] are among the most popular methods. With confidence scores in hand, both techniques allow for some form of weighting, and studies [5, 6] have affirmed the advantages of confidence based weighting strategies. However, it is common practice that systems that contribute to a combination do so with equal shares: Besides the commonly applied word or segment based weighting, e.g. during lattice combination, systems usually contribute equally to the final output. This strategy however can fail in cases where system performances are unbalanced and better hypotheses might simply be overpowered by suboptimal alternatives. In previous work [7] we were able to show the positive effects of a weighted system combination method that makes use of weights on the system level. In this work, we expand this weighting technique to automatic segmentation by combining multiple models for the segmentation task, in addition to using the system combination for decoding.

2. Overall system

In this section, we describe the components of our framework and the details of the system development. We elaborate our usage of several acoustic front-ends, acoustic modeling, and language modeling. The general framework is illustrated in Fig. 1. The automatic segmentation is explained in the following section.

2.1. Acoustic features

We utilized three different kinds of acoustic features: *a)* Mel-frequency cepstral coefficients (MFCC) [8]; *b)* perceptual linear prediction (PLP) [9]; *c)* log Mel-filter bank (FBANK). All feature vector types are 40-dimensional (raw output without dimension reduction), and are extracted for every 10 ms with a window length of 25 ms.

Additionally, in order to enhance the input features, we also adopt i-vector features [10], which were originally proposed for the speaker identification task. The distribution of an utterance super-vector M can be modeled by

$$M = m + Tw \tag{1}$$

where m is the mean distribution vector, T is a total variability matrix, and w is the i-vector. By having m and T fixed for all utterances, w would be affected by speaker and channel characteristics. We utilized i-vectors because they are able to capture speaker and channel informations that might be helpful for speech recognition, but are not represented in standard features such as MFCC, PLP, and FBANK.

2.2. Acoustic model training

We tested several acoustic model training strategies during development. GMM- and DNN-based acoustic models were trained with

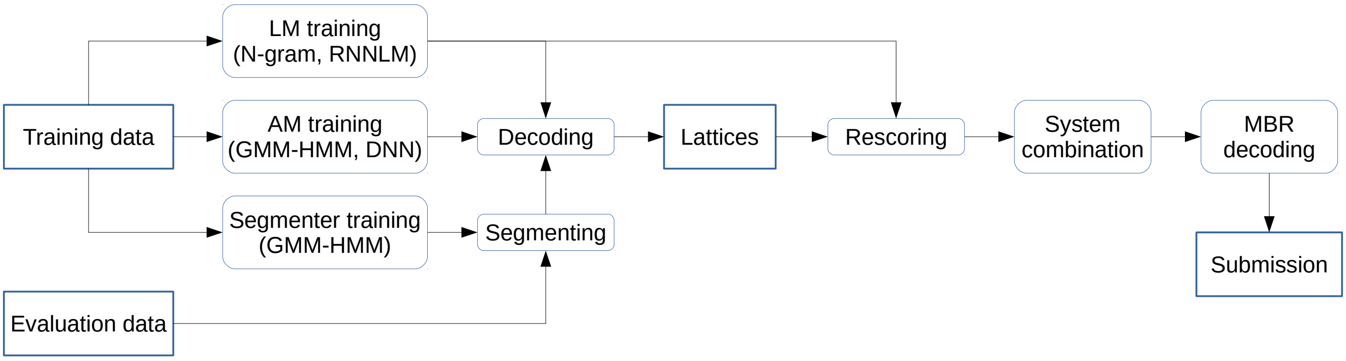

Figure 1: *General overview of our framework.*

different types of input features, as shown in Table 1. Models using speaker adaptive training (SAT) use standard features + feature-space maximum likelihood linear regression (fMLLR) [11], while all but one of the DNN-based acoustic models are trained with stacked standard and i-vector features. We investigated DNN architectures using *sigmoid*, *rectified linear* (ReLU), or *p-norm* [12] units, and also perform training using *state-level minimum Bayes risk* (sMBR) [13, 14]. The models are all implemented using the Kaldi speech recognition toolkit [2], and details are described in the following subsections.

2.2.1. Architectures

The *sigmoid DNN* model can be considered a standard DNN acoustic model with 6 hidden layers, where each layer consists of 2048 nodes. The sigmoid activation function is applied in each hidden layer, and the softmax function is applied in the output layer. The input features are generated by linear discriminant analysis (LDA) + maximum likelihood linear transform (MLLT) + fMLLR performed on top of MFCC. The feature frames are also stacked with 5 preceding and 5 succeeding frames, resulting in the final 440 dimensional DNN input feature vector covering 11 frames of context. First, we performed the pre-training with a restricted Boltzmann machine (RBM) deep belief network [15]. After that, the DNN was trained using the back-propagation algorithm and stochastic gradient descent with frame cross-entropy (CE) criterion as implemented by the Kaldi speech recognition toolkit [2].

We trained a *ReLU DNN* because it has been reported in [16] that rectified linear units can show better performance than sigmoid units for large vocabulary continuous speech recognition (LVCSR) tasks. We utilized a ReLU DNN with 6 hidden layers, where each layer consists of 1024 nodes, and the ReLU activation function is applied in each hidden layer. The input features are a raw 40 dimensional standard feature vector and a 100 dimensional i-vector stacked on top. Further, we do not perform pre-training as for the sigmoid DNN model, but instead we train for a fixed number of epochs and average model parameters over the last few epochs of training [17]. The parameters are also optimized according to the frame CE criterion.

The *p-norm DNN* [12] was adopted as the third type of model. The p-norm is a "dimension-reducing" non-linearity that is inspired by maxout

$$y = ||\mathbf{x}||_p = \left(\sum_i |x_i|^p \right)^{1/p}, \tag{2}$$

where here the vector $\mathbf{x}$ represents a bundled set of 10 feature vectors, p is the normalized parameter and is set to 2 as it showed the best performance as described in [12]. The model architecture is the same with ReLU DNN with 6 hidden layers, each has 1024 nodes. The input features are also the same as for the ReLU DNN. The parameters are trained by using frame CE.

2.2.2. sMBR training

To further enhance the DNN model, we continued training the model according to the state-level minimum Bayes risk (sMBR) criterion. This DNN is a p-norm DNN model but it is optimized according to sMBR instead of cross entropy. We only attempted to optimize the p-norm DNN this way because the training with sMBR is quite complicated and time-consuming, and more importantly, the p-norm DNN outperformed other models on "tst2013" test set during our experiments.

The training procedure is as follows: We first perform forced alignment, followed by a decoding on the training data to derive training samples, this process took 2 days on a cluster machine with 80 CPUs to produce 80 lattices. Then, we merge all lattices down to 4, which is equal to the number of GPUs we utilize. Finally, we perform parallel sMBR training as implemented in Kaldi.

2.3. Dictionary

We utilized a modified CMU pronouncing dictionary (`http://www.speech.cs.cmu.edu/cgi-bin/cmudict`) consisting of about 100k words as a base dictionary. We also employed grapheme-to-phoneme (G2P) conversion using the Sequitur G2P toolkit [18] trained on the CMU dictionary to generate pronunciations for unknown words in the training data. As a result, the total number of words in our dictionary is about 210k words. This dictionary is used for training as well as decoding.

2.4. Language model training

2.4.1. N-gram

N-grams have long been a standard language modeling technique for ASR, where $N - 1$ words are used as context to predict the next word. The larger the context, the more data is required to avoid the data sparsity problem. For the experiments described here, two N-gram language models (LMs) were trained with Kneser-Ney smoothing [19] implemented in the SRILM toolkit [20], a 4-gram LM pruned with probability 10^{-8} for decoding purposes, and a full 5-gram model for rescoring in a second pass.

Front-end	Model type				
	GMM-HMM (SAT)	Sigmoid DNN (CE)	p-norm DNN (CE)	ReLU DNN (CE)	p-norm DNN (sMBR)
MFCC	✓	✓	✓	✓	✓
PLP	✓	-	✓	✓	-
FBANK	✓	✓	✓	✓	✓

Table 1: *The list of all trained acoustic models.*

2.4.2. RNNLMs

Recurrent neural network language models (RNNLMs) have shown to have an advantage over the standard N-gram language model. There are several reasons for this, perhaps the most notable being that RNNLMs can capture the context of entire utterances, which is difficult to do with standard N-grams. [21, 22] have also shown that RNNLMs can significantly improve the performance of speech recognition, especially when RNN models are interpolated with N-gram language models. However, the drawback of RNNLMs is the computational complexity. Therefore, this type of language model is usually used for rescoring in two-pass decoding systems.

The systems that we developed for the IWSLT challenge adopt a class-based RNNLM [21], which consists of 1 hidden layer with 150 hidden nodes and 400 classes. The model is trained using the threaded version of the RNNLM toolkit. It took about 1 day to finish the training process.

2.5. Decoding strategy

For the test evaluation period we had 3 Gaussian mixture model hidden Markov model (GMM-HMM) systems and 12 DNN systems at hand for decoding that made use of 3 different front-ends. The GMM-HMM systems are trained using SAT. The DNNs use 3 different types of activation functions and 2 training criteria (see table 1). The GMM-HMM based SAT systems serve as basis for the sigmoid DNN systems, since their neural nets were built on top of the fMLLR transforms from these systems. We trained all systems on the same data, and they use the same lexicon and language models during decoding and rescoring. We run the decoding with a pruned 4-gram language model. Subsequent lattice rescorings make use of a 5-gram language model and an RNNLM language model. Given the lattices, we apply minimum Bayes risk (MBR) [23] decoding for all systems to minimize the expected word error rate (WER). After rescoring, we perform system combination using ROVER. To benefit from the individual system strengths, we attempted to apply a rank-score based weighting scheme that was first introduced in [7]. System weights during combination are conditioned to their respective rank-score. Let $\mathrm{rank}(s_n) \in \{1, \ldots, |\mathcal{S}|\}$ be the rank of a system $s_n \in \mathcal{S}$, where the system s^* with the highest accuracy $\mathrm{acc}(s^*)$ has rank 1. The rank-score of a system s_n is

$$\mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n)) \qquad (3)$$

A numerically lower rank indicates a system with higher performance. Weighting is performed according to:

$$\mathrm{weight}(s_n) = \frac{\mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n))}{\sum_{s_n \in \mathcal{S}} \mathrm{acc}(s_n) \cdot (|\mathcal{S}| + 1 - \mathrm{rank}(s_n))} \qquad (4)$$

Since for ROVER implicit weighting according to Equation (4) was not possible, we used an approximate method where hypotheses are taken into consideration multiple times for the combination, according to their respective ranks: In a combination of 4 systems, the best system enters ROVER 4 times, the second best 3 times and so on.

Corpus	Amount
BN 1996	81.79 h
BN 1997	72.36 h
TED-LIUM	200.00 h
TIMIT	3.92 h
WSJ	81.01 h
Total	439.08 h

Table 2: *Training data for acoustic modeling.*

Corpus	Word count
EUROPAL	49.13 M
GIGA	567.76 M
NC	1.17 M
TED-LIUM	2.25 M
WSJ	36.98 K

Table 3: *Training data for language modeling.*

3. Data resources

For the IWSLT 2015 evaluation, the regulations regarding the permissible training data are less restrictive, with no explicit cut-off date for data set. Data for language modeling is generally unrestricted, whereas acoustic modeling has to exclude a number of selected TED and TEDx talks that are not permitted to be used for training.

3.1. Acoustic model training data

For the ASR acoustic modeling no training data is provided, in contrast to the other evaluation tracks. Since data selection is unrestricted with the above mentioned exceptions, we were able to freely choose our speech corpora. The data we used for training acoustic models is selected from various resources including TED-LIUM corpus release 2 [24], Broadcast News [25], WSJ [26], and TIMIT [27], as listed in Table 2. We utilized TED-LIUM instead of the original downloadable TED talks because TED-LIUM is an already cleaned, noise-free corpus, and provides a good basis for training a full-fledged speech recognition system [24]. Although TIMIT is a relatively small corpus, it is suitable for training an initial monophone acoustic model.

3.2. Language model training data

The data for training language models comes from different sources including WSJ, EUROPARL, GIGA, NC, and TED, as shown in Table 3. The data is cleaned by removing all punctuation, and removing case sensitivity by uppercasing all characters.

3.3. Evaluation data

With regards to the test corpora, the data set “tst2013” used in past editions as either an evaluation set (2013) or a progressive test set (2014) was provided by the organizers as the official development

Data	Amount
Speech (TED)	343 min
Noises (TED)	342 min
Noises (Soundsnap)	12 min
Total	697 min

Table 4: *Training data for the GMM segmenter training.*

set for this year's evaluation. "tst2014" is used as a progressive test set, and a newly released test set "tst2015" consisting of 28 talks serves as the official test set for the final evaluation of all systems. Automatic segmentation of the raw audio data prior to decoding is a mandatory sub-task of the ASR track since 2013. We describe our approach for generating an automatic segmentation of the evaluation data in the following section.

4. Automatic segmentation of evaluation data

Given our observations regarding the effectiveness of neural net based and GMM based approaches for speech segmentation in previous work [7], we picked GMM-based segmentation as our method of choice for the IWSLT evaluation. This method uses a Viterbi decoder and GMM-HMM models to classify consecutively observed feature vectors into several sound categories. The mechanics of the general framework are comparable to the one presented in [28]. To improve segmentation quality, we experimented with data selection and model selection. We also tested the effectiveness of model combination to improve the final segmentation accuracy.

4.1. Segmentation training data

We used about 11.6 hours of data for model training, consisting of the official IWSLT "dev2010", "dev2012" and "tst2010" spoken utterances, noises extracted from the TED portion of the data used in [29, 30] and hand picked and manually trimmed noise samples downloaded from Soundsnap (`http://www.soundsnap.com`). Instead of keeping the detailed transcriptions, each spoken utterance in the test sets was labeled with a single *speech* token. A noise utterance is either labeled as *applause*, *laughter*, *music* or general *noise*. Table 4 lists the data for segmenter training.

4.2. Segmentation model

The general GMM segmentation framework is essentially a speech recognizer that is capable of discriminating several classes of sounds. Consecutive frames of the same sound are modeled as being generated by multi-state feed forward HMMs without skip states, where the minimal segment lengths are directly modeled by the HMM topology. Each GMM consists of 128 Gaussian components. The input is 42 dimensional LDA transformed MFCCs after stacking with a context of 7. The acoustic model is trained according to the maximum likelihood criterion, where the GMMs grow incrementally in several iterations of "split-and-merge" training [31]. The system is configurable by several parameters, one of which is a padding factor that expands hypothesized speech segments on both sides by a certain amount of milliseconds. This factor is tuned on the segmentation of this year's official development set. Segment coverage is computed on frame level and evaluated in terms of accuracy (ACC), true positive rate (TPR) and true negative rate (TNR).

Classes	Pad	ACC	TPR	TNR
[s],[sil+a+l]	0.325	88.9%	97.6%	45.6%
[s],[sil],[a+l]	0.475	90.1%	95.7%	62.2%
[s],[sil],[a],[l]	0.575	89.6%	95.8%	58.5%
[s],[sil],[a],[l],[n]	0.6	89.4%	95.9%	57.2%
[s],[sil],[a],[l],[n],[m]	0.8	82.6%	86.0%	65.7%

Table 5: *Segmenter performance dependent on the amount of classes. In column "Classes", the abbreviations stand for* speech, silence, applause, laughter, *(general)* noise *and* music, *respectively. Brackets delimit the individual classes formed by the data. Padding factors are in msec.*

Data (types)	Pad	ACC	TPR	TNR	WER
a+l+n+m	0.65	88.9%	95.5%	56.2%	27.3%
a+l+n	0.8	88.1%	95.3%	52.0%	28.8%
a+l	0.475	90.1%	95.7%	62.2%	26.5%
a	0.4	90.2%	96.1%	61.0%	26.0%
-	0.475	89.4%	96.5%	53.9%	26.7%
combined		90.4%	97.5%	55.2%	25.7%

Table 6: *Segmenter performance dependent on the amount of data. Padding factors are in msec.* combined *is the weighted combination of segmentations.*

4.3. Sound class selection

We evaluated the impact of the amount of target sound classes. The most simple system is discriminating speech from non-speech, the most complex system separates the distinct noises into individual classes. Silence in the speech recordings was detected via a simple power threshold during the sample extraction step of the training pipeline and where silence is a class of it's own, these features are used as samples for a *silence* class. Table 5 lists the details of the systems subject to comparison.

It is noteworthy that the 5 class and 6 class models were trained on more data, since they model additional classes for specific sounds. For the 2 class and 3 class models several noise types were simply mapped to one broad noise class. We interpret the results in the following way: It seems 2 classes are less suited to properly discriminate non-speech from speech, given the relatively low TNR, whereas 6 classes make significantly more errors in classifying speech correctly. The adding of samples for music obviously leads to a better noise classification, but also to more confusions in classification of speech. The 3 class model segmentation yields the highest accuracy, showing a comparatively good TNR with little loss in TPR given the alternatives. All further experiments were undergone with the 3 class segmentation model.

4.4. Sound class combination

We trained several models to test the impact of including or or excluding data of distinct noise types during training. The speech data and the hand picked Soundsnap samples were kept fix, and different portions of the TED noises were added. For each set generated this way, a segmenter was trained, tuned and evaluated. The results in table 6 show that it is the original data set that leads to optimal performance. If more noises are added, the performance deteriorates. If less noises are seen during training, the speech classification performance increases, while at the same time noise classification suffers. The table also lists the decoding performance of the SAT models, when decoded given the respective segmentations. The baseline performance on the provided segmentation is 25.0%

	Segmentation →	manual			automatic		
	Features →	MFCC	PLP	FBANK	MFCC	PLP	FBANK
Model	GMM-HMM (SAT)	23.9%	23.9%	24.8%	24.4%	24.8%	25.4%
	Sigmoid DNN (CE)	14.4%	-	-	15.1%	-	-
	ReLU DNN (CE)	11.2%	10.9%	12.7%	12.0%	11.7%	13.5%
	p-norm DNN (CE)	10.8%	10.5%	12.6%	11.4%	11.4%	13.5%
	p-norm DNN (sMBR)	**9.8%**	-	11.2%	**10.5%**	-	11.8%

Table 7: *Individual system performances of all recognizers in WER after rescoring.*

Systems								Weights	
ReLU DNN (CE)			p-norm DNN (CE)			p-norm DNN (sMBR)			
MFCC	PLP	FBANK	MFCC	PLP	FBANK	MFCC	FBANK	equal	rank-score
✓	✓	✓	✓	✓	✓	✓	✓	10.0%	9.7%
✓	✓		✓	✓	✓	✓	✓	9.8%	9.7%
✓	✓		✓	✓		✓	✓	9.8%	9.5%
	✓		✓	✓		✓	✓	9.6%	9.7%
			✓	✓		✓	✓	9.6%	9.7%
				✓		✓	✓	**9.5%**	9.6%
				✓		✓		10.0%	9.8%

Table 8: *Comparison of the 8 best ROVER combinations with equal and rank-score based weighting. Performance is measured in WER.*

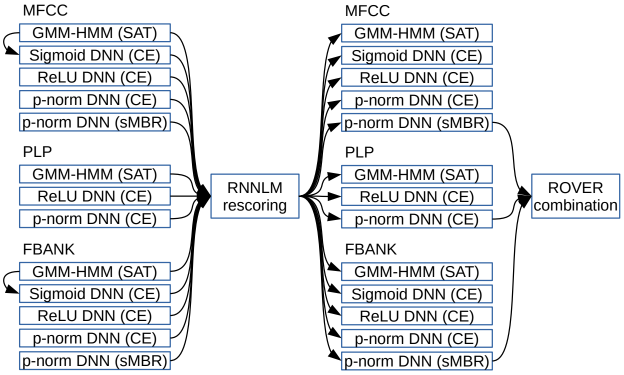

Figure 2: *Decoding pipeline of the primary submission. The left-most arrows symbolize the dependency of the sigmoid DNNs on the fMLLR transforms of the GMM-HMM systems.*

WER. To benefit from the individual model strengths, we successfully applied the rank-score based weighting scheme of subsection 2.5 to combine segmentations on frame level.s Since combination is performed frame-wise, artifacts in form of extremely short segments may be introduced at positions where the models greatly differ in their prognoses. To counter-act this phenomenon, segments are merged according to the heuristic

$$\text{from}(seg_1) - \text{to}(seg_2) \leq \delta \wedge \text{to}(seg_2) - \text{from}(seg_1) \leq \theta \quad (5)$$

with δ being subject to tuning (40 msec during our experiments) and θ set to 30000 msec. The weighted combination improves segmentation accuracy as well as speech recognition performance, reducing the WER to 25.7%. Combination with equal weights yielded similar results, but was inferior to our proposed method.

5. ASR evaluation

We evaluated our ASR systems on the "tst2013" development set, given the manual segmentation as well as our own, automatically generated segmentation. In preliminary experiments we found that RNNLM rescoring consistently outperformed rescoring with the 5-gram LM, producing WERs that were 0.4% absolute better on average. Thus, the results presented in this section only cover the results after RNNLM rescoring.

5.1. Single system performance

Table 7 lists the single system performances of all successful decodings on the development set. PLP features generally helped to achieve the best performance, followed by MFCC features. The gap between the MFCC and FBANK features is fairly large. It can also be seen that DNNs utilizing the p-norm activation function exceed the other nets' classification capabilities. Finally, the nets trained with the sMBR training criterion led to better accuracy than the ones built according to the cross-entropy criterion. The apparent inferiority of the sigmoid DNN might be due to several reasons, one of which is the differing activation function, given that ReLU seems to have an advantage on large data, according to previous work [11]. Another reason might be the differing network layout. Our assumption however is that the main difference is caused by the fact that this model is using standard features only, without the i-vectors stacked on top. This matches our observations in [7], where we used the same layout for all NNs and still observed a large gap between the system's performance. This thus re-confirms our assumption regarding the role of the features.

Decoding for the final submission had to be run on the automatic segmentation. Table 7 therefore also lists the recognition performance in WER for our own segmentation, created with the framework described in Section 4. Assuming that the scoring is identical or almost identical – given that we used the evaluation's default toolkit NIST SCTK (`http://www.nist.gov/itl/iad/mig/tools.cfm`) – our single best system (p-norm DNN sMBR) already outperforms last year's winner in the ASR track by 0.1% absolute on "tst2013".

5.2. System combination performance

Table 8 lists the performance of weighted system combination using the rank-score function compared to the default combination with equal weighting of all systems. To guarantee that the systems are diverse enough to benefit from the combination, each combination of more than 2 systems covered all three front-ends. Experiments confirmed that failing to do so indeed leads to sub-optimal combinations that are not even able to beat the single best system.

The results are interesting insofar as it seems that improvement by weighting is not possible if the standard ROVER already leads to a better performance than the single best system involved in the combination. In cases where unweighted ROVER produces a sub-optimal result, weighting is able to boost the positive effects of combination and achieves a better result. This observation is consistent with the combination results of our segmentation in Subsection 4.4 as well as in [7]. Given the results on "tst2013" we performed the ROVER combination with equal weights on the automatically segmented set and achieved a WER of 10.1%. The system design of our primary submission is highlighted in Fig. 2.

6. Conclusion

This paper described the structure and development of NAIST's English ASR system for the English ASR track of the IWSLT 2015 evaluation campaign. We evaluated different architectures of deep neural network based models as well as various types of input features such as MFCC, PLP, FBANK and i-vector. The results show that a p-norm DNN trained on combined MFCC + i-vector feature vectors following the sMBR training criterion achieves the best performance for a single system, yielding a WER of 9.8% on the official development set. After system combination with ROVER, where the outputs of the best systems for each front-end were combined, the WER can be further reduced to 9.5%.

We trained several simple GMM models for speech/non-speech classification for the purpose of automatic segmentation prior to decoding. To exploit the benefits of multiple models we performed a rank-score based weighting in a segmentation hypothesis combination scheme on frame level. The combined segmentation outperforms the single best segmentation in terms of segment coverage accuracy and WER after actual decoding. Our best decoding on the automatically segmented development set achieves a 10.1% WER, which outperforms last year's winning system by 0.5% absolute WER on this set. This setup was used for producing our primary submission for the evaluation campaign. The official scoring of our primary submission on the "tst2015" evaluation set yields 12.0% WER.

7. Acknowledgements

Part of this research was supported by JSPS KAKENHI Grant Numbers 26870371 and 24240032, and the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

8. References

[1] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Proceedings of ASRU*, 2001.

[2] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *Proceedings of IEEE workshop*, 2011.

[3] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings of ASRU*, Dec. 1997, pp. 347–354.

[4] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[5] B. Hoffmeister, D. Hillard, S. Hahn, R. Schlüter, M. Ostendorf, and H. Ney, "Cross-site and intra-site ASR system combination: Comparisons on lattice and 1-best methods." in *Proceedings of ICASSP*, 2007, pp. 1145–1148.

[6] K. Audhkhasi, A. M. Zavou, P. G. Georgiou, and S. Narayanan, "Empirical link between hypothesis diversity and fusion performance in an ensemble of automatic speech recognition systems." in *Proceedings of INTERSPEECH*, 2013, pp. 3082–3086.

[7] Q. T. Do, M. Heck, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "The NAIST ASR system for the 2015 multi-genre broadcast challenge: On combination of deep learning systems using a rank-score function," in *Proceedings of ASRU*, 2015.

[8] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pp. 357–366, 1980.

[9] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE*, 2010.

[11] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians." in *Proceedings of INTERSPEECH*, 2006.

[12] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proceedings of ICASSP*, May 2014, pp. 215–219.

[13] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proceedings of ICASSP*, vol. 4, April 2007, pp. IV–321–IV–324.

[14] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition." in *Proceedings of INTERSPEECH*, 2006.

[15] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[16] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proceedings of ICASSP*, 2013, pp. 8609–8613.

[17] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv*, 2014.

[18] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434 – 451, 2008.

[19] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, 1996, pp. 310–318.

[20] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, USA, 2002, pp. 901–904.

[21] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *Proceedings of ICASSP*, 2011, pp. 5528–5531.

[22] S. Kombrink, M. K. T. Mikolov, and L. Burget, “Recurrent neural network based language modeling in meeting recognition,” in *Proceedings of INTERSPEECH*, 2011, pp. 2877–2880.

[23] H. Xu, D. Povey, L. Mangu, and J. Zhu, “Minimum bayes risk decoding and system combination based on a recursion for edit distance,” *Computer Speech & Language*, vol. 25, no. 4, pp. 802 – 828, 2011.

[24] A. Rousseau, P. Delglise, and Y. Estve, “Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks,” in *Proceedings of LREC*, 2014.

[25] D. Graff, “The 1996 broadcast news speech and language-model corpus,” in *Proceedings of DARPA Speech Recognition Workshop*, 1996, pp. 11–14.

[26] D. B. Paul and J. M. Baker, “The design for the wall street journal-based CSR corpus,” in *Proceedings of the DARPA SLS Workshop*, 1992, pp. 357–362.

[27] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and L. D. Nancy, “DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1,” *NICT*, vol. 93, p. 27403, 1993.

[28] M. Heck, S. Mohr, C. Stüker, M. Müller, K. Kilgour, J. Gehring, Q. Nguyen, V. Nguyen, and A. Waibel, “Segmentation of telephone speech based on speech and non-speech models,” in *Speech and Computer*, ser. Lecture Notes in Computer Science, M. Železný, I. Habernal, and A. Ronzhin, Eds. Springer International Publishing, 2013, vol. 8113, pp. 286–293.

[29] C. Saam, C. Mohr, K. Kilgour, M. Heck, M. Sperber, K. Kubo, S. Stüker, S. Sakti, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, “The 2012 KIT and KIT-NAIST English ASR systems for the IWSLT evaluation,” in *Proceedings of IWSLT*, 2012.

[30] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stüker, C. Saam, K. Kilgour, C. Mohr, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, “The KIT-NAIST (contrastive) English ASR system for IWSLT 2012,” in *Proceedings of IWSLT*, 2012.

[31] T. Kaukoranta, P. Fränti, and O. Nevalainen, “Iterative split-and-merge algorithm for VQ codebook generation,” *Optical Engineering*, vol. 37, no. 10, pp. 2726–2732, 1998.