The 2016 KIT IWSLT Speech-to-Text Systems for English and German

Thai-Son Nguyen, Markus Müller, Matthias Sperber, Thomas Zenkel,
Kevin Kilgour, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics, Karlsruhe Institute of Technology
Karlsruhe, Germany
{first.lastname}@kit.edu

Abstract

This paper describes our German and English *Speech-to-Text* (STT) systems for the 2016 IWSLT evaluation campaign. The campaign focuses on the transcription of unsegmented TED talks. Our setup includes systems using both the Janus and Kaldi frameworks. We combined the outputs using both ROVER [1] and confusion network combination (CNC) [2] to archieve a good overall performance. The individual subsystems are built by using different speaker-adaptive feature combination (e.g., lMEL with i-vector or bottleneck speaker vector), acoustic models (GMM or DNN) and speaker adaption (MLLR or fMLLR). Decoding is performed in two stages, where the GMM and DNN systems are adapted on the combination of the first stage outputs using MLLR, and fMLLR.

The combination setup produces a final hypothesis that has a significantly lower WER than any of the individual subsystems. For the English TED task, our best combination system has a WER of 7.8% on the development set while our other combinations gained 21.8% and 28.7% WERs for the English and German MSLT tasks.

1. Introduction

For many years now, the *International Workshop on Spoken Language Translation* (IWSLT) offers a comprehensive evaluation campaign on spoken language translation. The evaluation is organized in different evaluation tracks covering automatic speech recognition (ASR), machine translation (MT), and the full-fledged combination of the two of them into speech translation systems (SLT). The evaluations in the tracks are conducted on TED Talks[1], short 5-25min presentations by people from various fields related in some way to Technology, Entertainment, and Design (TED) [3]. In this years installment, an additional track was added using recordings from Skype (MSLT).

The goal of the TED ASR track is the automatic transcription of fully unsegmented TED lectures. The quality of the resulting transcriptions is measured in word error rate (WER).

[1] http://www.ted.com/talks

This system paper describes our English and German ASR setups with which we participated in the TED ASR and MSLT tracks of the 2016 IWSLT evaluation campaign. Similar to previous years' evaluation [4], we used the Janus Recognition Toolkit (JRTk) [5] which features the IBIS single-pass decoder [6] to build several complementary subsystems and combined them with an additional system based on the TED-LIUM recipe of the Kaldi toolkit [7]. Our Janus-based systems employ different speaker-adaptive features, acoustic models or speaker adaption techniques. While the Kaldi-based system employs the same training database, sequence training and RNN based language models for rescoring.

The rest of this paper is structured as follows. Section 2 describes the data that our system was trained and tested on. This is followed by Section 3 which provides a description of the acoustic front-ends used in our system and Section 7 which describes our segmentation setup. An overview of the techniques used to build our acoustic models is given in Section 5. We describe the language model used for this evaluation in Section 6. Our decoding strategy and results are then presented in sections 8 and 9. We conclude the paper with Section 10.

2. Data Resources

2.1. Training Data

Table 1 and Table 2 show the data sources we used for the acoustic model training of our systems. This year we included more 80 hours of broadcast news which results a total of 483 hours for the English systems. For the German systems, we used the same training data as last year.

Source	# Amount
Quaero from 2010 to 2012	200 hours
Broadcast news [8]	80 hours
TED-LIUM v2 [9] excluding disallowed talks	203 hours
Total	483 hours

Table 1: *English acoustic modeling data.*

Source	# Amount
Quaero from 2009 to 2012	180 hours
Broadcast news	24 hours
Baden-Württemberg parliament	160 hours
Total	364 hours

Table 2: *German acoustic modeling data.*

2.2. Test Data

For this year's evaluation campaign, the evaluation test sets "tst2015" and "tst2016", as well as the development test sets "tst2013" and "tst2014" were provided for the English TED evaluation campaign. All development test sets featured a pre-segmentation provided by the IWSLT organizers. For the test set, automatic segmentation was required. For the MSLT task, development sets were provided as well. In contrast to the TED evaluation data, manual segmentation was provided for both development and test sets.

3. Feature Extraction

Our systems are built using several different front-ends as previously described in [4] including 40-dimensional log scale mel filterbank (lMEL), 20-dimensional mel frequency ceptral coefficient (MFCC), 20-dimensional minimum variance distortionless response (MVDR) and 14-dimensional tonal (T) features. These features can be augmented with i-vectors (Section 3.2) or bottleneck speaker vectors (Section 3.3) to be directly used for acoustic modeling or fed into deep bottleneck networks (Section 3.1) for extracting bottleneck features. The extracted bottleneck features are then transformed using feature-space maximum likelihood linear regression (fMLLR) and augmented with i-vectors to build speaker-adaptive features (Section 3.4). Our detailed feature extraction pipeline is explained in [10].

3.1. Bottleneck Features

We employed the deep bottleneck architecture described by [11], which consists of a stacked denoising auto-encoder of 4-5 layers each containing 1600-2000 units, followed by a 42 unit bottleneck, a hidden layer and the classification layer. The stacked auto-encoder is first pre-trained layer-wise [12], then the whole network is fine-tuned to discriminate target phoneme states. For the extraction of bottleneck features (BN), the layers after the bottleneck were removed and the output activations of the bottleneck layer were used as BN.

3.2. I-vectors

To extract i-vectors, a full universal background model (UBM) with 2048 mixtures was trained on the training dataset using 20 Mel-frequency cepstral coefficients with delta and delta-delta features appended. The total variability matrices were estimated for extracting 100 dimensional i-vectors. We tuned the size of the i-vectors in a series of preliminary experiments for optimal recognition performance. The UBM model training and i-Vector extraction was performed by using the sre08 module from the Kaldi toolkit [7]. I-vectors as well as tonal features were always used in combination with other features.

3.3. Bottleneck Speaker Vectors

In addition to i-vectors, we also used Bottleneck Speaker Vectors (BSVs) [13]. While they serve the same purpose, they are entirely neural network based. We used the same setup as for our hybrid systems, but trained the network to recognize different speakers instead of phonemes using a one-hot encoding of the speaker identities. To extract the BSVs, we used a bottleneck layer as second last layer of the speaker classification network and discarded all layers after this layer after training. For obtaining the final speaker vector, we averaged the output activation of this hidden layer on a per speaker basis or on utterance level if no speaker information was available (MSLT task).

3.4. Speaker Adaptive Features

To build speaker-adaptive features (SAF) for GMM systems, we first train deep bottleneck network from 11 stacked frames of regular features and i-vectors. The extracted BN features are then spliced for 11 consecutive frames and transformed using Linear Discriminate Analysis (LDA) which are known to make inputs more accurately modeled by GMMs.

The speaker-adaptive features for DNN systems are obtained after transforming BN features using fMLLR transformation and then augmented with i-vectors. The process of fMLLR estimation was performed as traditional approach. During the training, we used the adaption data of the same speaker and the reference transcriptions to do the alignment, while the same GMMs were used as first-pass systems to generate transcriptions in the testing.

4. Phoneme and Dictionary

For English, we used the CMU dictionary[2]. This is the same phoneme set as the one used in last year's systems. It consists of 45 phonemes and allophones. We used 7 noise tags and one silence tag. Missing pronunciations were created using the FESTIVAL [14] Text-to-Speech Engine.

Our German system uses an initial dictionary based on the Verbmobil Phoneset [15]. Missing pronunciations are generated using both MaryTTS [16] and FESTIVAL [14].

[2]http://www.speech.cs.cmu.edu/cgi-bin/cmudict

5. Acoustic Modeling

5.1. HMM CD-Phone

All GMM and hybrid models classify context-dependent quinphones with three states per phoneme and a left-to-right HMM topology without skip states. The English acoustic models use 8,156 distributions and codebooks derived from decision-tree based clustering of the states of all possible quinphones. The German acoustic models use 18,016 distributions and codebooks.

5.2. GMM Models

The GMM models are trained by using incremental splitting of Gaussians training (MAS) [17], followed by optimal feature space training (OFS) which is a variant of *semi-tied covariance* (STC) [18] training using a single global transformation matrix. The model is then refined by one iteration of Viterbi training.

For the evaluation, we trained two GMMs using SAF features with different front-ends for the English TED task. The front-ends include standard lMEL and a combination of MFCC, MVDR and tonal features (M2+T). For the English MSLT task, we use the combinations of lMEL+T and lMEL+IVec to build two GMM systems.

5.3. Hybrid Models

All the DNN models also share the same architecture which has 5-6 hidden layers with 2000 units per layer. The input of the DNNs are 11 stacked frames of 42-dimensional transformed bottleneck features or 40-dimensional lMEL, with or without combining i-vectors and tonal features. We used the sigmoid activation function for the hidden layers and softmax for the output layer. DNN systems were trained using the cross-entropy loss function to predict context-dependent states. The same training method is applied for all DNNs which includes pre-training with denoising auto-encoders and followed by fine-tuning with back-propagation. We used an exponential schedule to update the learning during the neural network training.

This year, we built two DNNs using SAF features with different front-ends for the English TED task. The used front-ends are the same as for the GMM systems. For English MSLT task, we employed 4 DNN systems which have different speaker-independent features as listed in Table 6.

The German setup for the MSLT task consists of 5 DNN systems based on different combinations of input features as shown in Table 7

6. Language Models

6.1. Vocabulary and Kneser-Ney Models

For language model training and vocabulary selection, we used the subtitles of TED talks, or translations thereof, and text data from various sources (see Tables 3 and 4). Text cleaning included tokenization, lowercasing, number normalization, and removal of punctuation. Language model training was performed by building separate language models for all (sub-)corpora using the SRILM toolkit [19] with modified Kneser-Ney smoothing. These were then linearly interpolated, with interpolation weights tuned using held-out data from the TED corpus. For German, we split compounds similarly as in [20].

For the vocabulary selection, we followed an approach proposed by Venkataraman et al.[21]. We built unigram language models using Witten-Bell smoothing from all text sources, and determined unigram probabilities that maximized the likelihood of a held-out TED data set. As our vocabulary, we then used the top 150k words for English, and 300k words for German.

6.2. Feed-forward Neural Language Model

During decoding the probabilities of a feedforward neural network language model were linearly interpolated with the baseline language model. Due to performance considerations, the most recent 40k queries for this language model were cached and we constrained the output vocabulary to the 20k most frequent words which appeared in the text corpora. We used 200 dimensional word embeddings trained with the Skip-gram model [22]. Three words were considered as the context, while the rest of the network consisted of three hidden layers followed by a softmax output layer. The training text consisted of 30M words for German and English. The German text was selected from the callhome, HUB5 and newscrawl data, while in English the training data was chosen from the TED and TEDLIUM corpora.

6.3. Recurrent Neural Language Model

We also trained a recurrent neural network language model for n-best rescoring, using 2 layers of long short-term memory [23, 24], with 650 hidden units each. We added shortcut connections as in [25]. We used BPE subword units [26] to handle rare words, the vocabulary size was 50k. We first trained on a large background corpus until convergence, and then fine-tuned parameters via continued training on in-domain data. We extracted 1000-best lists from the lattice or system combination output, and used MERT [27] to find rescoring weights for acoustic model scores, Kneser Ney language model scores, recurrent language model scores, and word and filler penalties. English background data was the TEDLIUM corpus, for German we used the newscrawl corpus. In-domain data and development data were taken from TED for the English TED task, from Fisher for the English MSLT task, and from Callhome and HUB5 for the German task (see Tables 3 and 4).

7. Automatic Segmentation

In this evaluation, the test set for the ASR track was provided without manual sentence segmentation, thus automatic

Text corpus	# Words
TED	3.6m
Fisher	10.4m
Switchboard	1.4m
TEDLIUM dataselection	155m
News + News-commentary + -crawl	4,478m
Commoncrawl	185m
GIGA	2323m

Table 3: *English language modeling data.*

Text corpus	# Words
TED	2,685k
News+Newscrawl	1,500M
Callhome	159k
HUB5	20k

Table 4: *German language modeling data after cleaning and compound splitting*.

segmentation of the target data was mandatory. We utilized an approach to automatic segmentation of audio data that is SVM based. This kind of segmentation is using speech and non-speech models, using the framework introduced in [28]. The pre-processing makes use of an LDA transformation on DBNF feature vectors after frame stacking to effectively incorporate temporal information. The SVM classifier is trained with the help of LIBSVM [29]. A 2-phased post-processing is applied for final segment generation.

We generated the segmentations for both English and German using this SVM based segmentation. The parameters for the SVM segmenter were chosen on a per language basis after preliminary experiments.

8. Systems and Combination

Table 5 shows our systems built for the English TED submission. In the first-pass, we used two GMM and two DNN systems with the acoustic models and 4-gram language model described in Section 5 and Section 6. Their decoded lattices are sent to a consensus decoding system (CNC) to produce combined hypotheses and confidence scores for the adaption in the second-pass. Two GMM systems are fully adapted as transitional approach using both feature space adaption (fMLLR) and model adaption (MLLR). The DNN systems are adapted by training the DNN acoustic models one more epoch on the adaption data of each speaker. The adaption data is obtained by performing alignment of the CNC decoded results with the speaker audio and filtering out the frames with the confidence scores higher than 0.7. In the second-pass, we employed the feed-forward language model described in Section 6 instead of the 4-gram language model. All these systems were built using Janus Recognition Toolkit (JRTK) [15].

Beside that we also trained a different system using the TED-LIUM receipe (s5) with the Kaldi toolkit [7]. The same train database is used for acoustic modeling but we used the Cantab-Tedlium [30] language model for decoding and our RNN language model for lattice rescoring. Our final submission for the English TED task consists of a ROVER of this Kaldi based system and the adapted systems. The results of the single and adapted systems as well the combined system are presented in Table 5.

This year we also participated in the new MSLT task. Since the MSLT data is provided without the speaker information, it was difficult to employ speaker adaptive features or apply our speaker adaption techniques. Only very limited gains could be achieved. In German, the WER decreased from 32.6% to 32.5% using lMEL+T in combination with BSVs, as shown in Table 7.

We archived the recognition improvement by combining several systems and using the RNN language model for rescoring as described in Section 6. Tables 6 and 7 show the results of many systems that we made and their combination.

9. Results

For the English TED task, we gained significant improvements over building speaker adaptive features, DNN model adaption, RNN language model rescoring and CNC combination. On the test set "tst2013" and "tst2014", we archived 9.4% and 7.8% WERs relatively. For the English and German MSLT tasks, the improvements was obtained by combining multiple systems and using the RNN language model. We archived WERs of 21.6% and 28.7% on "dev2016" for the final results.

System	tst2013	tst2014
GMM(SAF-lMEL)	13.5	11.0
GMM(SAF-M2+T)	13.4	10.9
DNN(SAF-lMEL)	12.0	10.4
DNN(SAF-M2+T)	12.3	10.0
CNC	10.5	8.6
GMM(SAF-lMEL) adapted	10.7	8.5
GMM(SAF-M2+T) adapted	10.5	8.6
DNN(SAF-lMEL) adapted	9.8	8.6
DNN(SAF-M2+T) adapted	10.2	8.8
Kaldi-s5 RNN rescored	11.8	8.6
ROVER	9.4	7.8

Table 5: *Results for English talk task on 'tst2013' and 'tst2014' development sets.*

In addition to our experiments on these two English tracks, we also participated in the German MSLT task. The results on the "dev2016" test set with and without neural network language models are shown in Table 7.

System	dev2016
GMM(lMEL+T)	26.7
GMM(lMEL+IVec)	26.6
DNN(lMEL+T)	27.1
DNN(lMEL+IVec)	27.6
DNN(BN-lMEL)	26.6
DNN(BN-M2+T)	26.7
CNC	22.9
CNC rescored	21.6

Table 6: *Results for English MSLT task on 'dev2016' development set.*

System	dev2016	+ NN-LM
DNN(BN-lMEL+T)	33.7	32.6
DNN(BN-lMEL+T+bsv)	33.8	32.5
DNN(BN-M2+T)	33.0	32.1
DNN(BN-M2+lMEL+T)	32.7	31.6
DNN(Mod-M2+lMel+T)	32.3	31.0
CNC	30.8	28.8
CNC rescored	–	28.7

Table 7: *Results for German MSLT task on 'dev2016' development set.*

10. Conclusion

In this paper we presented our English and German LVCSR systems, with which we participated in the 2016 IWSLT evaluation. All systems make use of neural network based front-ends, HMM/GMM and HMM/DNN based acoustics models. The decoding set-up of all languages makes extensive use of system combination of single systems obtained by combing different feature extraction front-ends and acoustic models.

11. References

[1] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, USA, Dec. 1997, pp. 347–354.

[2] 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.

[3] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in *Procedings of the 10th Workshop on Spoken Language Translation (IWSLT 2013)*, 2013.

[4] Markus Müller, Thai-Son Nguyen, Matthias Sperber, Kevin Kilgour, Sebastian Stüker, and Alex Waibel, "The 2015 KIT IWSLT Speech-to-Text Systems for English and German," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2015.

[5] M. Woszczyna, N. Aoki-Waibel, F. D. Buø, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. Rose, T. Schultz, B. Suhm, M. Tomita, and A. Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[6] H. Soltau, F. Metze, C. Fugen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 214–217.

[7] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[8] D. Graff, "The 1996 broadcast news speech and language-model corpus."

[9] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *Proc. of LREC*, 2014, pp. 3935–3939.

[10] Thai Son Nguyen, Kevin Kilgour, Matthias Sperber, and Alex Waibel, "Improved speaker adaption by combining i-vector and fmllr with deep bottleneck networks," in *Submitted to Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference*, 2016.

[11] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, 2013.

[12] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *The 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[13] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4610–4613.

[14] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[15] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 83–86.

[16] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[17] 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.

[18] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[19] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[20] Kevin Kilgour, Christian Mohr, Michael Heck, Quoc Bao Nguyen, Van Huy Nguyen, Evgeniy Shin, Igor Tseyzer, Jonas Gehring, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel, "The 2013 KIT IWSLT Speech-to-Text Systems for German and English," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2013.

[21] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," in *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, pp. 245–248.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *Interspeech*, 2012, pp. 194–197.

[25] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[26] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[27] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 160–167.

[28] M. Heck, C. Mohr, S. 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.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[30] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," *arXiv preprint arXiv:1502.00512*, 2015.