INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Data selection by sequence summarizing neural network in mismatch condition training

Kateřina Žmolíková[†], *Martin Karafiát*[†], *Karel Veselý*[†], *Marc Delcroix*[‡], *Shinji Watanabe*[§], *Lukáš Burget*[†] *and Jan "Honza" Černocký*[†]

[†] Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
[‡]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[§]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

xzmoli02@stud.fit.vutbr.cz,{karafiat,burget,iveselyk,cernocky}@fit.vutbr.cz
marc.delcroix@lab.ntt.co.jp,watanabe@merl.com

Abstract

Data augmentation is a simple and efficient technique to improve the robustness of a speech recognizer when deployed in mismatched training-test conditions. Our paper proposes a new approach for selecting data with respect to similarity of acoustic conditions. The similarity is computed based on a sequence summarizing neural network which extracts vectors containing acoustic summary (e.g. noise and reverberation characteristics) of an utterance. Several configurations of this network and different methods of selecting data using these "summary-vectors" were explored. The results are reported on a mismatched condition using AMI training set with the proposed data selection and CHiME3 test set.

Index Terms: Automatic speech recognition, Data augmentation, Data selection, Mismatch training condition, Sequence summarization

1. Introduction

Automatic speech recognition is being deployed extensively in mobile devices and home appliances. These devices are often used in noisy conditions or employed with distant microphones. Consequently, there is a great interest for making ASR systems robust to noise and reverberation. In the last summer, researchers in the field gathered at Second Frederick Jelinek Memorial Summer Workshop on Speech and Language Technology (JSALT) to tackle robust speech recognition in mismatched training and testing conditions.

The target application investigated during the workshop was simulated by training on relatively clean AMI headset microphone data (IHM) [1] and testing on noisy and reverberated data. The Single Distant Microphone (SDM) recordings from CHiME3 2015 [2] and REVERB 2013 [3] challenges were taken for this purposes.

One of the approaches investigated during the workshop was training the acoustic model on multi-condition training data created by artificially augmenting the data. This method has been shown to significantly improve the robustness of the system [4]. Using this approach, we can generate large amount of augmented data from which we can select our training set. This paper presents a new method for data selection which aims to select training data with the most similar noise conditions to test data.

Unlike the existing data selection approaches [5][6][7][8][9], we propose using a fixed-length "summary vector" representing the acoustic conditions to select the utterances within the training data that are the most similar to the test conditions. Previous work [10] has shown effectiveness of using i-vectors for data set characterization and data selection. The proposed summary vector extraction exploits the neural network framework instead of using i-vectors. A special neural network is used to compensate the mismatch between clean and noisy conditions. This is realized by appending a compensation network to a neural network trained on clean speech. The compensation network is trained to perform utterance level bias compensation. Consequently, the output of the compensation network summarizes the information about the noise conditions of an entire utterance and can thus be used to select useful training data.

The extracted "summary-vector" has the desired property of representing a specific noise type distinctly, which we prove by visualizing the vectors. Moreover, we also confirm experimentally that the proposed "summary-vector" can be used to select training data and that it outperforms random training data selection and i-vectors based data selection.

2. Sequence summarizing neural network

In order to select training data similar to the test data, we describe each utterance using a fixed-length vector summarizing the acoustic conditions of the utterance. In other words, a vector is extracted from each utterance that is in some sense similar to i-vectors known from speaker recognition field [11]. However, instead of relying on a conventional i-vector extraction, we train a special neural network which is able to produce the "summarizing vector" for each utterance. We expect that with the proposed approach we may obtain a "summary-vector" that can better represent the noise conditions than i-vectors that mostly represent speaker information.

To extract summary-vectors, we train a composite architecture combining two neural networks as sketched in figure 1. A similar architecture was previously used by Vesely in [12] for speaker adaptation. It consists of a main and a sequence summarizing neural network, both sharing the same input features.

The work reported here was carried out during the 2015 Jelinek Memorial Summer Workshop on Speech and Language Technologies at the University of Washington, Seattle, and was supported by Johns Hopkins University via NSF Grant No IIS 1005411, and gifts from Google, Microsoft Research, Amazon, Mitsubishi Electric, and MERL. BUT researchers were also supported by EU FP7 project No. 605141 "A-PiMod" and by by Technology Agency of the Czech Republic project No. TA04011311 "MINT".

http://dx.doi.org/10.21437/Interspeech.2016-741

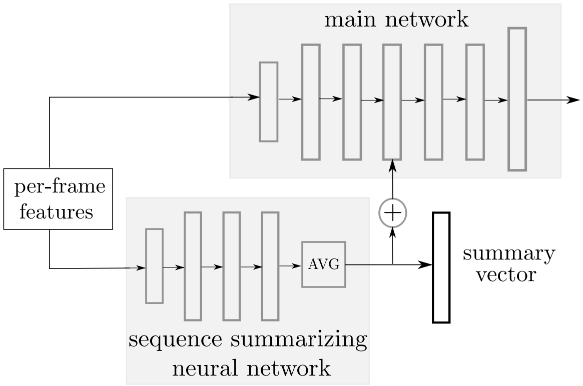

Figure 1: Training of acoustic condition estimator.

To train this scheme for the extraction of summary-vectors, we proceeded as follows:

1. First, the main network (DNN$(\cdot)$, upper part of figure 1) is trained on clean data $\mathcal{X}^{\text{clean}}$ as a standard DNN classifier with triphone state targets $\mathcal{Y}$ and cross-entropy criteria (XE$[\cdot]$), as follows:
$$\hat{\Theta}^{\text{clean}} = \arg\max_{\Theta} \text{XE}\left[\mathcal{Y}, \text{DNN}\left(\mathcal{X}^{\text{clean}}; \Theta\right)\right] \quad (1)$$
The estimated parameters of the main network $\hat{\Theta}^{\text{clean}}$ then stay fixed for the rest of the training.
2. The sequence summarizing neural network (SSNN) is added to the scheme (SSNN$(\cdot)$, lower part of figure 1), which also receives frame-by-frame speech features (same as the main network) as its input. The last layer of the SSNN involves the average operation so that it produces one fixed-length vector for each utterance, which is then added to the activations of a hidden layer of the main network. The whole architecture is now trained on noisy data $\mathcal{X}^{\text{noisy}}$ with the same objective as used in the first step, as follows:
$$\arg\max_{\Phi} \text{XE}\left[\mathcal{Y}, \text{DNN}\left(\mathcal{X}^{\text{noisy}}, \overline{\text{SSNN}(\mathcal{X}^{\text{noisy}}; \Phi)}; \hat{\Theta}^{\text{clean}}\right)\right] \quad (2)$$
Note that only the parameters of the sequence summarizing neural network Φ are trained at this point.

The idea of the training procedure is that the SSNN should learn to compensate for the mismatch caused by presenting noisy data $\mathcal{X}^{\text{noisy}}$ to the main network, which was previously trained only on clean data $\mathcal{X}^{\text{clean}}$. Thus, the vector which is extracted by the SSNN should contain important information about the acoustic conditions to characterize a noise component in an utterance. In the final application (data selection), we discard the main network and only use the SSNN to extract the summary-vectors.

3. The data and its augmenting

Our proposed method assumes a large amount of training data covering various noise conditions. Therefore, it will be possible to find training data similar to the test conditions. In this section, we describe the training and testing datasets and how we augment the training data to cover more noise conditions.

The following datasets were used:

- **Train - AMI** The Independent Headset Microphone (IHM) recordings from AMI meeting corpus[1] were used for acoustic model training. It contains 79 hours of meeting conversation recorded with 16 kHz sampling rate. This dataset is relatively clean.
- **Test - CHiME3** [2] contains simple utterances recorded live in noisy everyday environments. The utterances are based on the “no verbal punctuation” part of the Wall Street Journal (WSJ0) speaker-independent 5k vocabulary development set [13, 14]. They were re-spoken on 6-channel tablet devices in four various environments. Two types of data are employed: ‘Real data’ – speech data that is recorded live in the four noisy environments (on a bus, cafe, pedestrian area, and street junction); ‘Simulated data’ - noisy utterances that are generated by artificially mixing clean speech data with noisy backgrounds.

  Our test set consists of development (2.75 hours) and evaluation (2.2 hours) data. We only used ‘Real data’.

3.1. Noising

Our training data was processed by artificially adding the following types of noises:

- real background noises: 170 samples (4 minutes long) from Freesound[2]. These samples belong to categories: city, fan, AC, restaurant, shop, crowd, library, office and workshop. The noise characteristics are mainly stationary, with minor portions of transient noises and babbling.
- babbling noises: 30 samples (over 4 minutes long), each created by merging speech from 25 random speakers from AMI database selected using speech activity detector.

3.2. Reverberation

We generated artificial room impulse responses (RIR) using ”Room Impulse Response Generator” tool from E. Habets[3]. The tool can model the size of the room (3 dimensions), reflectivity of each wall, type of microphone, position of source and microphone, orientation of microphone toward the audio source and number of bounces (reflections) of the signal. Each of our room model consists of a pair of RIR. One is used to reverberate (convolution with RIR) the speech signal and the other is used to reverberate the noise. Both are mixed into a single recording. Note that only the coordinates of audio sources (speech/noise) differ for each of the RIRs in such a pair. We randomly set all parameters of the room for each room model.

3.3. Composition of the training set

We created the following training datasets with artificially corrupted speech:

3.3.1. Dataset 1

This dataset is used for the experiments with automatic selection of training data described in section 5.3. Various distortion types were investigated:

[1] http://corpus.amiproject.org/

[2] www.freesound.org

[3] http://www.audiolabs-erlangen.de/content/05-fau/professor/00-habets/05-software/01-rir-generator/rir_generator.pdf

- Noising only: one noise (stationary or babbling, see section 3.1) was randomly selected and added to speech signal at a specific SNR. Each copy is created for one of the -5, 0, 5, 10 or 15 dB SNRs which leads to 10 copies of the AMI training corpus.
- Reverberation only: we generated two classes of RIRs described below
  - *Small Rooms* - Artificial RIRs described in section 3.2 (Reverberation time - RT60 is around 0.3 s)
  - *Large Rooms* - Artificial RIRs described in section 3.2 (Reverberation time - RT60 is around 0.7 s)

  One of RIRs (corresponding to the *reverberation type* above) was randomly selected and used to reverberate the speech signal (i.e. speech signal is convolved with the selected RIR). This leads to 2 copies of the AMI training corpus.
- The third option is the combination of the previous two. We combined *Small rooms — Stationary noise*, *Small rooms — Babbling noise*, *Large rooms — Stationary noise* on the all chosen SNR levels. It led to 15 copies of the AMI training corpus.

The result is a large dataset comprising 27 corrupted copies of the AMI corpus.

3.3.2. Dataset 2

This dataset was created to train a summary-vector extractor. For each AMI clean recording, noising parameters similar to Dataset 1 were randomly selected. Three random copies of AMI corpus were created to test the effect of the amount of training data on the accuracy of summary-vector extractor.

4. Speech recognition system

The acoustic models (both GMM-HMM and DNNs) are trained on features that are obtained by splicing together 7 frames (3 on each side of the current frame) of 13-dimensional MFCCs (C0-C12) and projecting down to 40 dimensions using linear discriminant analysis (LDA) [15]. The MFCCs are normalized to have zero mean per speaker. We also use a single semi-tied covariance (STC) transform [16] on the features obtained using LDA. The combined features are referred to as LDA+STC. Moreover, speaker adaptive training (SAT) [17] is done using a single feature-space maximum likelihood linear regression (fMLLR) transform estimated per speaker.

4.1. GMM-HMM systems

The GMM-HMM systems were based on cross-word tied-states triphones. They were trained from scratch using standard maximum likelihood training where the last stage produces LDA +STC+fMLLR features. Boosted maximum mutual information training [18] followed to estimate more accurate models for generation of state alignment. It was taken for training of the final DNNs hybrid system.

4.2. DNN system

The DNNs were trained on the same LDA+STC+fMLLR features as the GMM-HMM baselines, except that the features were globally normalized to have zero mean and unit variance. The fMLLR transforms were the same as those estimated for the GMM-HMM system during training and testing. The network had 7 layers (that is, 6 hidden layers), where each hidden layer has 2048 neurons; the DNN has about 4 thousand output units. The input to the network is an 11 frame (5 frames on each side of the current frame) context window of the 40 dimensional features. The DNN was initialized with stacked restricted Boltzmann machines (RBMs) that were pretrained in a greedy layer-wise fashion [19].

After pre-training, we added the output layer with random weights and we performed frame-classification training (we classify frames into triphone tied-states). We used mini-batch Stochastic Gradient Descent (SGD) to minimize per-frame cross-entropy between the labels and network output.

Table 1: Training summary-vector extractor with various positions of the connection layer.

Connection Layer	XE [%WER]
None	47.72
1	39.89
2	**39.32**
3	40.09
4	41.08
5	40.70

Table 2: Tuning the dimensionality of summary-vector extractor.

Size of 2nd layer	256	512	1024	2048
Clean DNN	49.03	48.41	47.83	47.72
Joint NN	49.39	42.70	40.15	39.32
abs. improvement	-0.36	5.71	7.68	**8.40**

5. Experiments

5.1. Configuration of NN

To find the optimal configuration of the summarizing neural network for extracting the summary-vectors, we performed a set of experiments varying the hidden layer where the vector is added, size of this layer (thus the size of the extracted vector) and amount of data used to train the network. Although the whole composite network as seen on Figure 1 was not intended to be used for decoding in the final application, for these experiments we used it directly to test on the noisy data. This allowed us to find the best configuration without the final time consuming procedure — data selection and system rebuild.

First, the optimal hidden connection layer was evaluated. In these experiments, the size of the hidden layer was 2048 and the amount of noised training data was equal to the original clean set. The results on CHiME3 are shown in Table 1. It seems that the second hidden layer is the most suitable for the adaptation. Moreover, the results present over 8% absolute WER reduction by adding summary-vector extractor to clean DNN. It is a nice improvement taking into account that the extractor performs just simple per-utterance bias compensation in the hidden layer of the clean DNN classifier.

The second hidden layer was taken as the connection layer and the optimal size of summary-vector was evaluated. To be able to train summary-vector extractors of different sizes, we had to re-train the original DNN classifier with various sizes of second hidden layer on the clean data. The table 2 shows WER reduction as a function of the dimensionality of the summary-vector. It degrades with dimensionality, therefore we decided to keep its original dimensionality of 2048.

Finally, the effect of adding data for the training of the sum-

Table 3: Various data sizes for summary-vector extractor training.

summary-vector-dim	Data Size 1xTrain	 2xTrain	 3xTrain
1024	40.15	38.99	37.30
2048	39.32	40.24	37.15

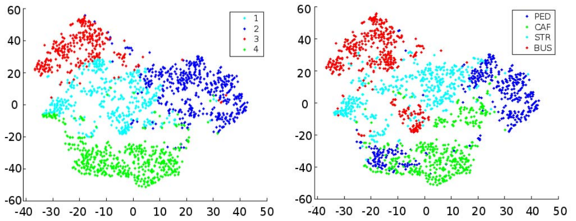

Figure 2: t-SNE plot of summary-vectors estimated from CHiME3 data. The colors correspond to the clusters obtained by k-means (left figure) and the actual noise conditions (right figure).

marizing network was evaluated. We generated several random selections of noised data and trained the extractor on them. Table 3 shows positive effect of sufficient amount of training data (3x Train) although the result of 2x Train was unstable.

5.2. Properties of extracted vectors

To see whether the method generates vectors reflecting the noise conditions in the data, we extracted the vectors for CHiME3 utterances and observed their properties. CHiME3 test set contains 4 different recording environments - bus (BUS), cafe (CAF), street (STR) and pedestrian area (PED). We performed clustering of the extracted vectors in 4 clusters using k-means and compared these clusters to the real environments in the data. Figure 2 shows two plots created by t-SNE [20] technique — the right one shows the 4 real environments in the data and the left one the clusters created by k-means. Although the clusters were created by an unsupervised technique, there are clear similarities with the real clusters.

It is also worthwhile to compare the newly proposed summary-vector with i-vectors [11] as i-vectors are also known to capture information about channel. Note that i-vectors were also recently used for adapting DNN in speech recognition tasks [21, 22]. Figure 3 shows CHiME3 data projected into the first two LDA basis. The recording environment labels were used as classes for LDA. It seems that the environments are better separated in the summary-vector space compared to i-vector space[4]. It shows that summary-vectors are containing information suitable for CHiME 3 recording environment clustering, even though the extractor was trained on different data (corrupted AMI corpus).

5.3. Data selection experiments and results

To perform data selection we extract summary-vectors for each generated training utterance and each test utterance. We select a subset of the generated training data by selecting the utterances that are closer to the test set conditions. For this, we compute the mean of all summary-vectors of the test set and measure its distance to the summary vectors of each utterance of the training data. Only the closest utterances are kept in the training subset. By this, we aimed to select such type of added noise which matches best the noise in the test data. The amount of selected data is equal to the size of the clean training set.

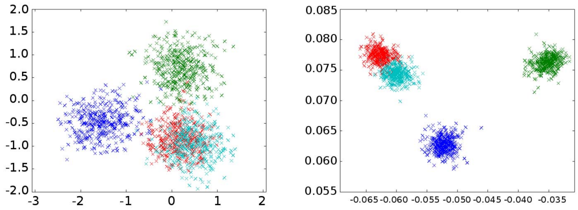

Figure 3: Plot of the first and second LDA basis on CHiME3 data for i-vectors (left) and summary-vectors (right).

As few different types of noises are present in the test data, computing mean of summary-vectors from the whole test set may not be the best way to represent it. Therefore, further experiments were performed where we clustered the test summary-vectors and computed means of these clusters. The training utterances were then selected to have the shortest distance to the summary-vector centroid of randomly chosen cluster.

For measuring the distance between vectors, we used cosine and Euclidean distance. Table 4 shows results obtained using these two measures and different numbers of clusters of test data. Results indicate that using cosine distance leads to better improvement. The best results are obtained using 4 clusters of test data which corresponds to the fact that there are 4 real recording environments in the test set.

Table 5 shows the best result obtained by summary-vector data selection compared to random data selection and selection using i-vectors. Note that the results are obtained from the mismatched condition. About 1% absolute improvement on dev-set and 2% on eval-set was obtained with the proposed data selection method compared to random data selection. Thus we show the effectiveness of the proposed data selection method based on the summary-vector extraction.

Table 4: Comparison of different selection methods.

distance measure / # clusters	1	4	10
cosine	25.09	24.72	24.98
Euclidean	26.75	26.55	26.59

Table 5: Comparison of random vs. automatic selection results in mismatched condition.

Dataset	Selection [%WER] Random	 i-vector	 summary-vector
dev	25.8	25.61	**24.72**
eval	45.58	44.02	**43.23**

6. Conclusion

This work proposed new promising approach for selecting data with respect to similarity of acoustic conditions. The method is based on neural network which extracts vectors containing information about noise and reverberant environments in an utterance. We explored several configurations of this network and different methods of selecting data using these vectors. On CHiME3 test set, we observed 1% absolute improvement over random data selection. In future, we would like to verify our findings on other databases.

[4] Brno University of Technology open i-vector extractor `http://voicebiometry.org` was used for these experiments.

7. References

[1] S. Renals, T. Hain, and H. Bourlard, "Recognition and understanding of meetings the ami and amida projects," in *Proceedings of ASRU*, 2007, pp. 238–247.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Proceedings of ASRU*, 2015, pp. 504–511.

[3] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proceedings of WASPAA*, 2013, pp. 1–4.

[4] M. Karafiát, F. Grézl, L. Burget, I. Szőke, and J. Černocký, "Three ways to adapt a cts recognizer to unseen reverberated speech in but system for the aspire challenge," in *Proceedings of Interspeech*, 2015, pp. 2454–2458.

[5] F. Beaufays, V. Vanhoucke, and B. Strope, "Unsupervised discovery and training of maximally dissimilar cluster models," in *Proceedings of Interspeech*, 2010.

[6] Y. Wu, R. Zhang, and A. Rudnicky, "Data selection for speech recognition," in *Proceedings of ASRU*, 2007, pp. 562–565.

[7] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proceedings of Interspeech*, 2009.

[8] K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes, "Submodular subset selection for large-scale speech training data," in *Proceedings of ICASSP*, 2014, pp. 3311–3315.

[9] T. Asami, R. Masumura, H. Masataki, M. Okamoto, and S. Sakauchi, "Training data selection for acoustic modeling via submodular optimization of joint kullback-leibler divergence," in *Proceedings of Interspeech*, 2015.

[10] O. Siohan and M. Bacchiani, "ivector-based acoustic data selection," in *Proceedings of INTERSPEECH*, 2013, pp. 657–661.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[12] K. Veselý, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. Černocký, "Sequence summarizing neural network for speaker adaptation," in *Proceedings of ICASSP*, 2016, (accepted).

[13] D. B. Paul and J. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1992, pp. 357–362.

[14] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *Proceedings of ICASSP*, 1995, pp. 81–84.

[15] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proceedings of ICASSP*, 1992, pp. 13–16.

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

[17] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proceedings of ICSLP*, vol. 2, 1996, pp. 1137–1140.

[18] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Proceedings of ICASSP*, 2008, pp. 4057–4060.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[20] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[21] M. Karafiát, L. Burget, P. Matějka, O. Glembek, and J. Černocký, "ivector-based discriminative adaptation for automatic speech recognition," in *Proceedings of ASRU*, 2011, pp. 152–157.

[22] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013, pp. 55–59.