ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414725

ANALYSIS OF X-VECTORS FOR LOW-RESOURCE SPEECH RECOGNITION

Martin Karafiát[1], *Karel Veselý*[1], *Jan "Honza" Černocký*[1], *Jan Profant*[2], *Jiří Nytra*[2],
Miroslav Hlaváček[2], *and Tomáš Pavlíček*[2]

(1) Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia
(2) Phonexia s.r.o, Brno, Czechia

ABSTRACT

The paper presents a study of usability of x-vectors for adaptation of automatic speech recognition (ASR) systems. X-vectors are Neural Network (NN)-based speaker embeddings recently proposed in speaker recognition (SR). They quickly replaced common i-vectors and became new state-of-the-art technique. Here, the same approach is adopted for ASR with the hope of similar outcome. All experiments were done on ASR for the latest IARPA MATERIAL evaluation running on Pashto language. Over 1% absolute improvement was observed with x-vectors over traditional i-vectors, even when the x-vector extractor was not trained on target Pashto data.

Index Terms— speech recognition, adaptation, x-vectors, data augmentation, robustness

1. INTRODUCTION

Deep Neural Network (DNN) adaptation is an important technique in most of the state-of-the art automatic speech recognition (ASR) systems as it allows to "adapt" the DNN model to particular conditions such as channel, noise, and speaker. Moreover, model adaptation is crucial in training/test data mismatch. On contrary to classical Gaussian Mixture Models (GMM), the adaptation of DNN is more difficult due to huge number of independent parameters. Several methods have been proposed in tha past: *Constrained adaptation* adds a regularization term (for example KL-divergence [1]) into training on target adaptation data. *Feature normalization* considers NN as a black-box and leverages on independent feature processing to suppress the mismatch. Common feature normalization and enhancement approaches adopted from the GMM-based ASR can be used as well, such as cepstral mean normalization or Constrained Maximum Likelihood Linear Regression (CMLLR) [2]. *Structured DNN Parameterization* imposes adaptable structures in the DNN hidden layers with a relatively small number of adaptation parameters associated with a speaker and/or noise type; notable examples are Linear Hidden Unit Contribution (LHUC) [3], or Cluster Adaptive Training (CAT) for DNN [4]. Currently, the most common adaptation method—*Feature augmentation*— fits into this last category. It incorporates a compact representation of speaker or noise information into a fixed-dimensional vector appended to the input features. The i-vectors [5] are nowadays commonly used for this task.

The i-vectors are speaker embeddings developed originally for speaker recognition (SR). They provide an elegant way of encoding a sequential input with variable length into a single vector with fixed-dimension. The i-vectors became state-of-the-art in SR and quickly found a way into ASR. The first attempt incorporated i-vectors as additional input features to discriminatively trained Region Dependent Linear Transform (RDLT) for GMM-HMM (Hidden Markov Models) ASR system [6]. Soon after, input feature vectors were augmented by i-vectors also for DNN-HMM [7] ASR systems with promising results. Later [8], the optimal size of i-vectors for ASR stabilized on 100-dimensional vector on contrary to SR, where 512 dimensions are commonly used. Further modifications have shown positive effect of online i-vector extraction on the ASPiRE challenge data [9].

The i-vectors are estimated in Maximum Likelihood fashion using a Universal Background Gaussian Mixture Model (UBM-GMM), therefore, there have been numerous efforts to use NNs to generate discriminatively-trained speaker embeddings. The main problem for NN-based speaker recognition systems were variable-length feature sequences, while common NN classification structures expected fixed-length feature vectors. Recently proposed x-vectors [10] have solved this problem in an elegant way by introducing a "statistical" layer. This layer typically sits close to the end of the NN processing pipeline, and is followed by a standard dense layer and a softmax layer with speakers IDs as targets. The statistical layer aggregates frame-level representations and produces a single set of statistics (means and variances) for the whole speech segment. These statistics are mapped into single vector — an x-vector — which serves as speaker embedding for

The work was supported by Czech National Science Foundation (GACR) project "NEUREM3" No. 19-26934X, European Union's Horizon 2020 project No. 833635 ROXANNE and Czech Ministry of Interior project No. VJ01010108 "ROZKAZ". It was also supported by by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) MATERIAL program, via Air Force Research Laboratory (AFRL) contract # FA8650-17-C-9118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, AFRL or the U.S. Government.

further processing.

Recent attempts of using the x-vectors as the embeddings for adaptation of ASR systems [11, 12] showed only slight or no improvement of performance. In this paper, we provide a detailed analysis of suitability of x-vectors for ASR adaptation, and we show significant gain over standard i-vectors.

Initial experiments in section 3.1 run with stand-alone x-vector extractor implemented by Phonexia and trained on proprietary data. Next, in section 5.3, we analyze suitability of various publicly available data for x-vector extraction.

2. DATA

All experiments run on Pashto as part of IARPA MATERIAL evaluations[1]. The main task of this project is Cross Lingual Information Retrieval (CLIR) for low resource languages; the goal is measuring relevance of an English query in target Pashto speech recordings (note, that part of the target data are also text documents, which are not used in this work). Consequently, the system involves a complicated pipeline incorporating various ASR, machine translation and information retrieval. This paper covers only the ASR part of the MATERIAL system.

Speech recognition training data (build_train) from BABEL project [13] Pashto collection is used for acoustic model training. 110 h of speech in this set consist mainly of conversational telephone speech with small portion of scripted read speech. On contrary to BABEL project, where target data was coming mainly from matched telephone conversations [13], target MATERIAL recordings are from three different sources: Conversation Speech (CS, 1.8 h), Topical Broadcast (TB, 11.0 h) and News Broadcast (NB, 3.7 h), 16.5 h in total. Therefore, a significant part of the target data is non-conversational broadcast speech naturally causing data mismatch.

x-vector extractor training data: low amount of speaker recognition Pashto data led us to analyzing language dependency and multilingual approaches to create x-vector training data. We considered the following data-sets:

- *Mult_v1* (40k recordings from 20k speakers): 6 languages: Swahili, Tagalog, Somali (MATERIAL collection), Dholuo, Zulu, Igbo (BABEL collection).
- *Mult_v3* (204k recordings from 40k speakers): 27 languages: Swahili, Tagalog, Somali, Bulgarian, Lithuanian (MATERIAL collection), Pashto, Cantonese, Asamese, Bengali, Turkish, Vietnamese, Haiti, Lao, Tamil, Kurdish, Zulu, Tok Pisin, Cebuano, Kazach, Telugu, Guarani, Igbo, Amharic, Mongolian, Javanese, Dholuo, Georgian.
- *sre_v0* (91k recordings from 7k speakers): NIST Speaker Recognition Evaluation (SRE) data from 2004–2010, Mixer 6, Switchboard 2 (phases 1, 2, 3), Switchboard Cellular.
- *phx_v0* (116k recordings from 4k speakers): in-house Phonexia data from various sources, mostly non-English.
- *vceleb* (1.2M short recordings from 7k speakers) from VoxCeleb 1 and 2 [14].

Layer	**Layer Type**	**Context**	**Size**
1	TDNN-ReLU	$[t-2, t+2]$	512
2	Dense-ReLU	t	512
3	TDNN-ReLU	$[t-2, t, t+2]$	512
4	Dense-ReLU	t	512
5	TDNN-ReLU	$[t-3, t, t+3]$	512
6	Dense-ReLU	t	512
7	TDNN-ReLU	$[t-4, t, t+4]$	512
8	Dense-ReLU	t	512
9	Dense-ReLU	t	1500
10	stats (mean+stddev)	whole segm.	2 × 256

Table 1. Architecture of Extended-TDNN based speaker embedding extractor.

Note, that all wide-band data was down-sampled to 8kHz to be consistent with telephone conversations.

3. X-VECTOR EXTRACTOR

All x-vector extractors presented in this paper are based on Extended-TDNN architecture (see table 1) proposed in [15], having superior SR performance than the original TDNN architecture [10].

The rest of training pipeline follows Kaldi toolkit [16] SRE recipe (egs/sre16/v2/run.sh):

1. *feature extraction* is based on 23 dimensional Mel-Filter Cepstral Coefficients (MFCC).
2. *voice activity detection* is based on energy detector.
3. *data augmentation* leverages RIR and MUSAN datasets[2]. It creates 4 augmented versions of the original data by (1) adding reverberation (convolution with RIRs); (2) adding noise (MUSAN); (3) adding music (MUSAN); and (4) adding background speech. The augmented data is randomly sub-sampled by factor of two, and original data is appended, therefore, the final training data has 3× the size of the original data.
4. *x-vector* training.

Note, that the limit for minimum number of recordings per speaker is switched off when using ASR training data, where it is common to have just one recording per speaker.

[1] https://www.iarpa.gov/index.php/research-programs/material

[2] http://www.openslr.org

3.1. Phonexia x-vector extractor

Phonexia x-vector extractor is taken for the initial experiments due to its simplicity and easiness of use. It is a single binary generating speaker based x-vector for each recording. In addition, it can produce a continuous x-vector stream based on floating window. It operates inside 5s (our case) windows producing output every 100 ms (10 frames). In case of no voice activity, it repeats the last active x-vector. These discontinuities can create long stable parts which can have negative effect on adaptation performance, therefore we also analyzed online x-vectors without voice activity detector. Phonexia extractor used in this work was trained using above recipe on *sre_v0*+*phx_v0* data.

3.2. Baseline i-vector extractor

Standard 100 dimensional online i-vectors [9] were estimated as a baseline. The features were the same as for final acoustic models (40-dimensional MFCC - see section 4). We also experimented with multilingual and speaker-based i-vectors for better comparison with x-vectors.

4. HYBRID ACOUSTICS MODELS

All experiments run with hybrid DNN-HMM ASR trained with the Kaldi toolkit. Factorized Time Delay NN (**TDNNf**) architecture [17] with convolutional layers (CNN) was selected as it was showing similar performance to more complicated recurrent NN types including those based on Long-Short-Term Memory (LSTM) cells.

Our **6CNN-9TDNNf** architecture contains 6 CNN layers (64, 64, 128, 128, 256, 256 filters in L1, L2, . . . , L6) followed by 9 **TDNNf** layers each with 1536 neurons, and bottle-neck factorization to 160 dimensions with stride 3.

The feature extraction is based on 40-dimensional MFCC, where inverse cosine transform is applied before the input of the NN. It re-creates Mel-filter bank outputs more suitable for further CNN processing. The adaptation vectors are transformed by affine transform to 200 dimensions. Both feature streams are concatenated and serve as CNN input. NNs are trained with Lattice Free Maximum Mutual Information (LF-MMI) objective and bi-phone targets as suggested in [18].

5. RESULTS

5.1. Language Model

The output lattices are generated with automatic segmentation. The decoding uses a smoothed 3-gram language model (LM) estimated by linear interpolation from various text sources: *build_train* data transcriptions (1 MWords), machine translation training data (0.9 MW), data crawled from wikipedia (3 MW) and other general sources (224 MW), see [19] for details on LM data creation.

x-vector	mode	analysis WER [%]			
		Total	CS	TB	NB
none	none	47.0	46.0	49.5	40.3
i-vectors	online	46.2	44.8	48.8	39.3
phx	speaker	44.5	**43.3**	46.6	39.0
phx	utterance	**43.9**	44.3	**46.1**	**37.4**
phx	online	44.5	44.1	46.7	38.1
phx	online_novad	44.4	44.8	46.2	38.6

Table 2. Phonexia x-vector extractor in various modes.

5.2. Phonexia stand-alone x-vector extractor

Simple use of Phonexia x-vector extractor allowed for direct testing of suitability of x-vectors for speech recognition. X-vectors with different time granularities (recording, utterance, online) were generated and added as adaptation features to the DNN training. New DNN models were trained and analyzed in table 2. It clearly shows significant 2.3% absolute gain with utterance-based x-vectors over standard i-vectors.

The utterance-based x-vectors show better performance than speaker-based ones due to finer granularity resulting in a possibility to react on speech variations during the recording. Unfortunately, no positive effect from online extraction is observed, probably due to low amount of data for x-vector estimation. On the other hand, it opens a possibility for online ASR.

Based on this outcome, the utterance-based x-vectors were considered for further experiments.

5.3. Training data analysis

Phonexia extractor generalizes well on Pashto data probably due to significant amount of non-English training data coming mainly from *phx_v0* dataset. Unfortunately, this proprietary data makes results non-reproducible, therefore it is removed from further experiments.

We are interested in lowering data mismatch and in producing robust embeddings suitable for ASR. Here, keeping channel information could be important on contrary to speaker recognition task.

Considered data-sets for the following experiments can be categorized with the following attributes:

- **Multilinguality** - *Mult_v1*, *Mult_v3* - multilingual training sets including even matching language (Pashto) data (*Mult_v3*). We routinely use this data for multilingual ASR training [20]. It is not suitable for SR due to low number of recordings per speaker, but it could fit well in case of language mismatch.
- **Channel** - *vceleb* - VoxCeleb data consisting of audio extracted from video recorded originally in wide-band. The embeddings should fit better to target channel than those trained on telephone data, although vceleb was down-sampled to 8kHz.

x-vector data	analysis WER [%]			
	Total	CS	TB	NB
Pashto i-vectors	46.2	44.8	48.8	39.3
sre_v0 i-vectors	47.0	45.3	49.1	41.9
Mult_v1 (speaker mode)	46.9	43.5	49.3	41.5
Mult_v1	45.7	44.6	48.4	38.1
Mult_v3	45.0	**43.7**	47.6	37.9
sre_v0	**44.1**	44.0	**46.3**	37.9
sre_v0 (reco. per spk≥8)	44.3	43.8	46.7	**37.6**
vceleb	44.9	44.1	47.2	38.3
sre_v0 + Mult_v3	44.4	43.6	46.7	38.0
sre_v0 + vceleb	44.6	44.1	47.0	37.9

Table 3. Utterance-based x-vectors from various datasets.

- **Speaker recognition** - *sre_v0* - well balanced data for speaker recognition. The embeddings should describe well speaker characteristics.
- **Combination** of above.

Table 3 shows that having the data well balanced for speaker recognition is more important than the channel and multilingual attributes. Even a combination of the data-sets did not bring any improvement over *sre_v0*.

As a next step, we limited *sre_v0* training set to have a minimum of eight recordings for each speaker (as it is common in x-vector training for SR). It resulted in similar gains as keeping all the data, with only a small degradation of performance in most conditions.

In addition, we experimented with i-vectors estimated on *sre_v0* data for fair comparison with x-vectors. A degradation compared to i-vector baseline was observed probably due to i-vector language dependency (these sets consist mainly of English data).

5.4. x-vector dimensionality

Optimal dimensionality of embeddings for ASR can differ from SR as the role of the vector is to help the system to adapt instead of speaker classification. This outcome was observed for i-vectors already in [8]. To analyze the effect of dimensionality, x-vector extractors with various sizes of statistical layer were trained on *sre_v0* data. Table 4 presents no gains by reducing the layer size. This is probably caused by forcing the NN to squeeze information needed for speaker classification (NN target classes) into too narrow bottle-neck. This may result in high suppression of channel information that is useful for ASR adaptation.

x-vector dim	analysis WER [%]			
	Total	CS	TB	NB
sre_v0 - 1024	44.5	44.1	46.9	37.5
sre_v0 - 512	**44.1**	**44.0**	**46.3**	**37.9**
sre_v0 - 256	45.3	44.0	48.0	38.0
sre_v0 - 100	45.3	44.6	47.7	38.4

Table 4. Various sizes of x-vector statistical layer.

System	x-vector	analysis WER [%]			
		Total	CS	TB	NB
CNN6-9TDNNf	i-vector	37.0	43.2	37.6	32.0
CNN6-9TDNNf	Mult_v3	36.1	42.0	36.7	31.1
CNN6-19TDNNf	Mult_v3	35.6	**41.1**	36.2	30.9
+ specaug	Mult_v3	35.2	41.2	35.8	**30.3**
CNN6-19TDNNf + specaug	sre_v0	**35.0**	41.8	**35.4**	30.4

Table 5. Final evaluation systems.

6. FINAL EVALUATION SYSTEM

Various speech recognition system improvements were explored in parallel to this x-vector analysis. Consequently, the final system for MATERIAL evaluation, run in summer 2020, required complex re-training. In addition to x-vectors, the following enhancements are added: multilingual pre-training [20], new broadcast news Pashto data, increasing number of TDNNf layers from 9 to 19, wide-band feature extraction, spectral augmentation [21], Recurrent Language Model (RNN-LM) [22], and sequence Minimum Bayes Risk (sMBR) training [23]. Only the most important results with x-vectors are shown in table 5. Here, all raws shares wide-band training, multilingual pre-training, further training with sMBR criteria and RNN-LM rescoring. It shows 0.9% gain by "Mult_v3" x-vectors over i-vectors on traditional 6CNN-9TDNNf architecture. Next, 0.2% additional gain from "sre_v0" x-vectors is observed on enhanced architecture with more layers and spectral augmentation.

7. CONCLUSION

The paper presents the first attempt to adopt x-vectors as adaptation vector for speech recognition acoustic model. Extensive analysis shows suitability of this technique for low resource ASR even if the target language is not part of x-vector training data. The x-vectors trained on sufficient amount of well balanced telephone data show robustness to channel and language mismatch. They overcome baseline i-vectors by impressive 2% absolute gain. The obtained improvements are persistent when a significantly more complex ASR system is used.

8. REFERENCES

[1] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recogni-

tion," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2013, pp. 7893–7897.

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

[3] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*, 2014, pp. 171–176.

[4] T. Tan, Y. Qian, and K. Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 03, pp. 459–468, 2016.

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

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

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

[8] A. Senior and I. L. Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, May 2014, pp. 225–229.

[9] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proceedings of Interspeech*, 2015.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*, 2018.

[11] M. A. T. Turan, E. Vincent, and D. Jouvet, "Achieving multi-accent ASR via unsupervised acoustic model adaptation," in *Interspeech 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 1286–1290.

[12] J. Równicka, P. Bell, and S. Renals, "Embeddings for dnn speaker adaptive training," 09 2019. [Online]. Available: https://arxiv.org/abs/1909.13537

[13] M. Harper, "The BABEL program and low resource speech technology," in *Proc. of ASRU 2013*, Dec 2013.

[14] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "Voxsrc 2019: The first voxceleb speaker recognition challenge," vol. abs/1912.02522, 2019. [Online]. Available: http://arxiv.org/abs/1912.02522

[15] D. Snyder, J. Villalba, N. Chen, D. Povey, G. Sell, N. Dehak, and S. Khudanpur, "The JHU speaker recognition system for the VOiCES 2019 challenge," in *INTERSPEECH*, 2019.

[16] 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.

[17] 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 *Proceedings of Interspeech*, 09 2018, pp. 3743–3747.

[18] 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 *Proceedings of Interspeech*, 09 2016, pp. 2751–2755.

[19] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. M. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *INTERSPEECH 2015*, 2015, pp. 839–843.

[20] M. Karafiát, K. M. Baskar, P. Matějka, K. Veselý, F. Grézl, L. Burget, and J. Černocký, "2016 BUT Babel system: Multilingual BLSTM acoustic model with i-vector based adaptation," in *Proceedings of Interspeech 2017*, 2017.

[21] S. H. R. Mallidi and H. Hermansky, "A framework for practical multistream ASR," in *Proc. Interspeech 2016*, N. Morgan, Ed. ISCA, 2016, pp. 3474–3478.

[22] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "RNNLM-recurrent neural network language modeling toolkit," in *Proc. of the 2011 ASRU Workshop*, 2011, pp. 196–201.

[23] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of Interspeech 2013*. International Speech Communication Association, 2013, pp. 2345–2349.