Recent Advances in ASR Applied to an Arabic Transcription System for Al-Jazeera

Patrick Cardinal[1], *Ahmed Ali*[2], *Najim Dehak*[1],*Yu Zhang*[1]
Tuka Al Hanai[1], *Yifan Zhang*[2], *James Glass*[1], *Stephan Vogel*[2]

[1]MIT Computer Science and Artificial Intelligence Laboratory,
Cambridge, Massachusetts 02139, USA
[2]Qatar Computing Research Institute, Doha, Qatar

patrick.cardinal@csail.mit.edu, amali@qf.org.qa, najim@csail.mit.edu, yzhang87@mit.edu
tuka@mit.edu, yzhang@qf.org.qa, glass@mit.edu, svogel@qf.org.qa

Abstract

This paper describes a detailed comparison of several state-of-the-art speech recognition techniques applied to a limited Arabic broadcast news dataset. The different approaches were all trained on 50 hours of transcribed audio from the Al-Jazeera news channel. The best results were obtained using i-vector-based speaker adaptation in a training scenario using the Minimum Phone Error (MPE) criteria combined with sequential Deep Neural Network (DNN) training. We report results for two different types of test data: broadcast news reports, with a best word error rate (WER) of 17.86%, and a broadcast conversations with a best WER of 29.85%. The overall WER on this test set is 25.6%.

Index Terms: Arabic, ASR system, Kaldi

1. Introduction

Arabic speech recognition is a challenging task for several reasons: the number of different dialects spoken around the world, the impressive number of synonyms, the morphological richness of the language and the fact that some uttered sounds are not transcribed in the written texts. With the objective to improve the accuracy of Arabic speech recognition systems, the Qatar Computing Research Institute (QCRI) and MIT have started a collaboration with an initial focus on the broadcast news domain, and a preliminary system developed with 50 hours of broadcast news collected from the Al-Jazeera channel.

The first objective is to build a state-of-the-art broadcast news transcription system that is tuned and optimized for offline broadcast domain. Initially, the focus has been on Modern Standard Arabic (MSA).

Building Automatic Speech Recognition (ASR) systems for Arabic is not new, and has been addressed by many researchers, especially through the DARPA GALE program. GALE MSA data contains more than 1800 hours of recordings (with most of those not transcribed). For this reason, most published results have been accomplished using much more data than are currently publicly available [1, 2, 3, 4]. The most comparable system is described by Rybach *et al.* [3]. Using 450 hours from GALE database, they obtain a WER of 25.9% for broadcast news reports (BR) and 33.9% for broadcast conversation (BC). The combination of several systems with ROVER give them a small improvement over single system performances. Using new state of the art techniques on similar audio (recordings from Al Jazeera) allowed to reduce the WER to 17.86% (BR) and 29.85% (BC). In this experiment, only one system was used.

This paper describes practical aspects in building a Broadcast news transcription system by discussing both Language Model (LM) and Acoustic Modeling (AM) aspects. Results with different state-of-the-art acoustic modeling techniques, including those based on deep neural networks (DNNs), are shown. In addition, the i-vector framework, combined with fMLLR features, is used to improve the efficiency of the speaker adaptation process.

The rest of this paper is organized as follows: Section 2 describes the training data; Section 3 will illustrate the ASR system and report experimental results; Section 4 concludes and suggests future work.

2. Training Data

2.1. The Database

The QCRI automatic Arabic speech recognition corpus consists of broadcast news reports and conversational shows spoken only in MSA. The Al-Jazeera channel is the main source for collecting this data. The recordings have been segmented and transcribed to avoid any non-speech segments such as music and background noise. The recordings were made using satellite cable sampled at 16kHz. The development dataset consists of one hour of speech, and is composed of broadcast news reports. No conversational shows have been included. However, a two hour test set has been designed and collected consisting of both kind of data types used in the training corpora.

2.2. Diacritization

Building an Arabic speech recognition system has several challenges. The most important one consists of the optional use of diacritics in written text, which are absent the majority of time. The diacritics role is to specify short vowels and consonant doubles. For example, the word كُتُب (kutub)[1] is usually written كتب (ktb). Unfortunately, automatically adding diacritics is not an easy task since a given word can have several diacritized versions, depending on the context in which the word has been used. For example, the word كتب (ktb) can be either vowelized كُتُب (kutub), which means book, or كَتَب (katab), which means

[1]Buckwalter transliteration is Romanized Arabic [5]

wrote. This is an extremely important problem when training the acoustic models since several diacritized phonemes will not be available in the lexicon. Diacritics are also helpful for building a good language model, since they result in more accurate n-gram probabilities, since word contexts are more efficiently used. A drawback of diacritized language models is the increase of the vocabulary size, and the resulting need for a larger text corpus.

Another challenge comes from the fact that Arabic is a morphemically rich language, creating lexical varieties and data sparseness. This phenomena leads to huge out-of-vocabulary (OOV) rates, and high language model (LM) perplexities. Rybach *et al.*[3] report an OOV rate of 5.6% on a language model for MSA with a vocabulary of 256K words. A similar language model in English generally has an OOV rate lower than 0.7%.

The richness in the Arabic language can lead to mistakes in the transcription, if a transcriber is more faithful to the speech rather than the language guidelines. For example, consider the following transcription: هذا الإلتزام الأخلاقي القوي (h*A Al<ltzAm Al>xlAqy Alqwy). This transcription is not wrong according to the audio, but from the linguistic point of view, the sentence should be transcribed by: هذا الالتزام الأخلاقي القوى (h*A AlAltzAm Al>xlAqy AlqwY). In this example, the > has been changed for A and y by Y. A text normalization process applied to transcriptions should produces a linguistically correct transcription.

In this work, MADA [6] is used to implement a morphological decomposition to normalize and vowelize the text by retrieving the missing diacritics. Since several vowelizations for a specific word are possible, a confidence score is provided for each candidate. MADA has been widely used for both Statistical Machine Translation [7, 8], and in ASR to address the aforementioned challenges. See [6] for more information regarding MADA operation details.

2.3. Lexicon

The lexicon has been created with rules proposed by Biadsy *et al.* [9]. They describe rules for representing glottal stops, short vowels, coarticulation of the definite article Al, nunnation, diphthongs, word ending p (/t/), and case endings, while ignoring geminates. Table 1 summarizes the rules used to create the lexicon.

3. ASR Systems

3.1. Language Modeling

The language model has been built from manually transcribed, and automatically diacritized transcriptions (430K words) of Al-Jazeera broadcast news, and from automatically diacritized texts (109 million of words) downloaded from the Al-Jazeera web site. The vocabulary contains 400K words, a combination of words in audio transcriptions and the 400K more frequent words in Al-Jazeera web site texts.

The development set discussed in Section 2.1 has been used to choose the interpolation coefficient in the mixing of both text sources. The OOV rate on the test set is 3.1%.

3.2. Acoustic Modeling

This section describes the acoustic modeling techniques that have been studied for this project. All models have been created with Kaldi [10], an open-source speech recognition system. The only exception is the extraction of bottleneck features presented in section 3.2.3 for which our own DNN library has been used.

Rule	Source	Target
Long Vowel (*Dagger Alif*)	/‘/	/ae:/
Long Vowel (*Madda*)	/|/	/ae:/
Nunnations	/AF/	/ae n/
	/F/	/ae n/
	/K/	/ih n/
	/N/	/uh n/
Glottal Stop (*Hamza*)	/['}&<>]/	/q/
p word ending (*tah-marbuta*)	/p/	/t/
Long Vowel (*Alif Maqsura*)	/Y/	/ae:/
Geminates (*Shadda*)	/~/	//
Diphthongs	/u w $\$cons$/	/uw/
	/ih y $\$cons$/	/ih:/
Suffix ’uwoA’ (*Waw Al Jama’a*)	/uh w ae:$/	/uw/
Definite Article (*Al*)	/Al/	/ae l/
Word Middle Long Vowel (*Hamzat Wasl*)	/{/	//
Definite Article *Al* (Sun Letters)	/^aw l $\$sun$/	/^ae $\$sun$/

Additional Variants to Lexicon.

p Word Ending (*tah-marbuta*)	/p {ae, uh, ih, F, K, N} $/	//
Short Vowel Word Ending	/{ae, uh, ih}$/	//

Using regular expressions. ^ means first character. /$/ is end of word.
$\$italics$ indicates variable. {a,b, . . . } is a set.
$\$cons$ = { b, d, dd, t, tt, k, kq, q, f, th, dh, zh, z, s, ss, sh, gh, kh, ai, h, hh, jh, m, n, w, l, r, y}
$\$sun$ = {t, th, d, dh, t, z, s, sh, tt, ss, dd, zh, l, n}

Table 1: Summary of Rules by Biadsy *et al.* [9].

3.2.1. Gaussian Mixture Model (GMM) Systems

Several acoustic modeling approaches have been explored and compared in the context of the Al-Jazeera dataset. The first system is based on a GMM technique applied to model cepstral features. These acoustic features correspond to 12 MFCC components, energy, and their first and second derivatives. The trained speech recognizer contains 4,000 state distributions with a total 128,000 Gaussian components.

The second system is also a GMM-based system where a speaker adaptation approach was applied to make acoustic models more appropriate to a specific speaker. The method used in this work is fMLLR, a widely used technique for speaker adaptation proposed by Povey *et al.* [11] . However, since the only speaker information available from the database is speaker turns, features are adapted on an utterance basis. This system operates on a different feature set compared to the first system. These features consist of stacking 13 MFCC speech frames with a context window of 9 frames. These features were then projected in a new space of dimension 40 using a Linear Discriminant Analysis (LDA) transform.

In the third system, we explore the use of SGMM technique. It was first introduced by Povey *et al.*[12] in the context of modeling low-resource languages. This model is based on a Universal Background Model (UBM) of 700 Gaussian components, which has been estimated on our Al-Jazeera training dataset.

In the last system, we investigated the use of discriminative training in the context of SGMM modeling. This training was based on the Maximum Mutual Information (MMI) criteria.

System	WER		
	Reports	Conversations	Overall
Basic GMM	28.01%	42.62%	37.42%
Basic GMM+fMLLR	23.65%	37.69%	32.70%
SGMM+fMLLR	21.56%	36.05%	30.90%
SGMM+fMLLR+MMI	20.90%	33.67%	29.13%

Table 2: *WER of GMM-Based systems*

The results obtained by the four systems are reported in Table 2. As expected, the results show that Broadcast news reports are easier that conversational speech with 8.46% different in the word error rate (WER). Not surprisingly, the SGMM system provided the best results for both types of data. However, the discriminative training did not help significantly for the report condition.

3.2.2. DNN Systems

Using DNNs for ASR acoustic modeling has become very popular in recent years. In an ASR system, the DNN is used to estimate the posteriors of each state in the HMM model [13, 14]. For the training process, the targets of the DNN are generated by a forced alignment using a GMM-based system. The DNN is then trained using the back-propagation algorithm.

The DNN used in this work has five layers of 2048 nodes. The input features are similar to the ones used on the SGMM which consists of 12 MFCCs plus energy. The resulting speech frames were stacked with a context window of nine frames, and reduced to 40 dimension using LDA. The projected vectors were then expanded by concatenating 11 frames, producing feature vectors of dimension 40x11=440 as described in [15].

Similarly to GMM modeling, a DNN can be trained using a discriminative approach using sequence-based criteria. Veselý *et al.* showed that discriminative training on DNN improves the performance of the ASR system [16]. Discriminative training using the MPE criterion has been used in our experiments in order to train the DNN.

Results are presented in Table 3. As expected, the best results are obtained with the sequentially trained DNN. In these experiments, the results have been improved by 2.25% absolute over the DNN trained with the cross-entropy criterion. The results also show that fMLLR does not improve the results of the DNN, and a plausible explanation is that DNN already implicitly normalizes the speaker effects. This can be also explained by the fact that fMLLR achieved a significant improvement of 2.3% absolute WER in the case of SGMM modeling.

System	WER		
	Reports	Conversations	Overall
DNN	21.05%	34.71%	29.85%
DNN+fMLLR	20.51%	34.03%	29.22%
DNN+fMLLR+MPE	18.93%	30.27%	26.24%

Table 3: *WER using a hybrid DNN/HMM decoder.*

3.2.3. Bottleneck Features

Bottleneck features is another way to incorporate DNN into HMM. The DNN is used to extract more discriminative features from the original set. These features can then be modeled with the GMM framework allowing the use of all optimization technics developed in the last two decades. This is done by training a DNN as before and then using the activation of a narrow hidden ”bottleneck” layer as features for GMMs.

In this work, we used the low rank stacked bottleneck (LrSBN) scheme as proposed by Zhang *et al.*[17]. The LrSBN approach, summarized in Figure 1, is used to extract features for the GMM system. The input of the first layer is made of 23 critical-band energies are obtained from a Mel filter-bank. Each of the 23+2 dimensions is then multiplied by a Hamming window across time, and a DCT is applied for dimensionality reduction. The 0th to 5th coefficients are retained, resulting in a feature of dimensionality (23 + 2) x 6 = 150.

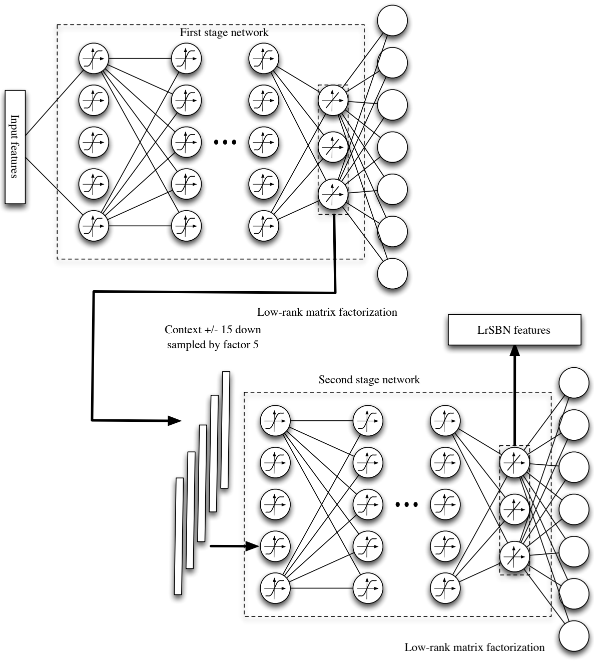

Figure 1: *Diagram of bottleneck feature extraction.*

The input features of the second DNN are the outputs of the bottleneck layer from the first DNN. The context expansion is done by concatenating frames with time offsets 10, 5, 0, 5, 10. Thus, the overall time context seen by the second DNN is 31 frames. Both DNNs use same setup of 5 hidden sigmoid layers and 1 linear bottleneck layer, and both use tied-states as target outputs. The targets are generated by forced alignment from the HMM baseline. No pre-training is used. Finally, the raw BN outputs from the second DNN are whitened using a global PCA and used as features for a conventional CD-HMM system. More details about the architecture can be found in [17].

The LrSBN features are then used to train a GMM system similar to those presented in section 3.2.1. The input feature vector is the concatenation of 13 MFCC features with their corresponding derivative and second derivative and the LrSBN features. As for the system in section 3.2.1, the trained systems contain 4000 distributions for total of 128000 Gaussians. Table 4 shows results with LrSBN features.

The results show that the improvement obtained in experiments of section 3.2.1 using ”LDA” features and speaker adaptation is not as good when the LrSNB features approach is used. This is because the second DNN takes into account a similar

System	WER		
	Reports	Conversations	Overall
Basic GMM	20.04%	34.08%	29.09%
Basic GMM+fMLLR	19.70%	33.62%	28.67%

Table 4: *WER with bottleneck features.*

context window and also because the fMLLR speaker adaptation is implicitly estimated by the DNN. These results are consistent with those obtained in section 3.2.2.

3.2.4. Speaker Adaptation with I-Vector

The i-vector approach [18] is a powerful technique that summarizes all the updates happening during the adaptation of the Universal Background Model (UBM)[2] mean components (named also GMM supervector) to a given utterance sequence of frames. All this information is modeled in a low dimensional space named the total variability space. In the i-vector framework, each speech utterance can be represented by GMM supervector, which is assumed to be generated as follows:

$$M = m + Tw_u$$

where m is the speaker independent and channel independent supervector (which can be taken to be the UBM supervector), T is a rectangular matrix of low rank, and w_u is a random vector having a standard normal distribution prior $N(0, 1)$. The i-vector is a Maximum A Posteriori (MAP) point estimates of the latent variable w adapting the corresponding GMM (supervector m) to a given recording.

The i-vector approach was first introduced in speaker and language recognition. Recently, it has been successfully applied for speaker and channel adaptation in speech recognition [19]. Adding speaker characteristics to the audio features allow the DNN to learn more efficiently how each speaker can produce a specific phoneme. For example, ذلك (*lk) can also be pronounced زلك (zlk) by some people. The DNN will tend to give a better probability to the most used form of phones.

Similarly to [19], we trained two i-vector extractors of dimension 100 each. The two systems are based on two different UBMs of 512 Gaussians each have been trained on two kind of features. The first set of features is the same one as described in Section 3.2.1 which is 40-dimension of LDA transformed feature of nine stacked MFCC frames of dimension 13. The second features consist of applying the fMLLR speaker adaptation transform to the first set of feature vectors. In our experiments, the GMM sizes are much smaller than those used in [19] because the training data set is limited.

For a given utterance, the i-vector w_u is created. Then, the i-vector is concatenated to the feature vector. Figure 2 describes how the feature frames and i-vector are combined in order to be feed to the DNN.

Table 5 shows the results that try to answer the question if the i-vector can provide complementary information to the fMLLR approach since both techniques are used for speaker adaptation.

The results show that i-vector provides different information about speakers, allowing a WER improvement up to 1.12% absolute. It's interesting to see that the improvement is similar with and without speaker adapted features. That means that both techniques are really complementary.

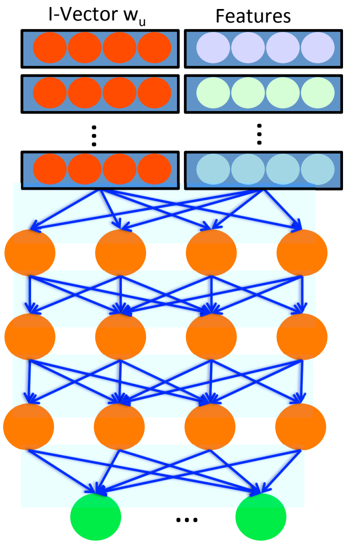

Figure 2: *Using i-vector as a speaker feature vector*

System	WER		
	Reports	Conversations	Overall
DNN	21.05%	34.71%	29.85%
DNN+i-vector	20.51%	34.38%	29.44%
DNN+fMLLR	20.51%	34.03%	29.22%
DNN+fMLLR+i-vector	19.55%	32.91%	28.16%
DNN+fMLLR+MPE	18.93%	30.27%	26.24%
DNN+fMLLR+ivec+MPE	17.99%	30.08%	25.78%

Table 5: *WER when using i-vector for speaker adaptation.*

These results are similar to those published in [19]. It's interesting to see that their approach still work even with a limited amount of training data. Based on these results, it will be interesting to incorporate the i-vector framework into the LrSBN scheme described in section 3.2.3.

4. Conclusion

This paper described the Arabic transcription system built for Al-Jazeera with a limited amount of data. The use of most recent speech recognition techniques have been used to train different systems. The best system presented is the hybrid DNN/HMM approach in which the DNN is sequentially trained using the MPE criterion for which a WER of 25.78% has been obtained. The input vector of the DNN was a combination of fMLLR adapted MFCC-based features combined to a i-vector extracted from the utterance.

A much more recordings of different type of shows and containing different dialects are available from Al-Jazeera. They have not been considered in this work because no transcriptions were available. A semi-supervised approach will be studied to take advantage of these data to improve the accuracy and the robustness of the transcription system. Another important step in the project is to build a speech recognition system that will work on different Arabic dialects by using the non-transcribed data combined with a dialect detection algorithm.

[2] A UBM is a large GMM trained on a large amount of data to represent general feature characteristics. This model plays the role of a prior on how all the sounds look like.

5. References

[1] L. Mangu, H. Kuo, S. Chu, B. Kingsbury, a. H. S. G. Saon, and F. Biadsy, “The ibm 2011 gale arabic speech transcription system,” in *2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2011, pp. 272–277.

[2] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, “The ibm 2006 gale arabic asr system,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, April 2007, pp. IV–349–IV–352.

[3] D. Rybach, S. Hahn, C. Gollan, R. Schluter, and H. Ney, “Advances in arabic broadcast news transcription at rwth,” in *IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, Dec 2007, pp. 449–454.

[4] R. Schlueter, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, J. Loeoef, C. Plahl, D. Rybach, and H. Ney, “Development of large vocabulary asr systems for mandarin and arabic,” in *2008 ITG Conference on Voice Communication (SprachKommunikation)*, Oct 2008, pp. 1–4.

[5] T. Buckwalter, “Arabic transliteration,” http://www.qamus.org/transliteration.htm.

[6] N. Habash, O. Rambow, and R. Roth, “Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization,” in *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, 2009, pp. 102–109.

[7] A. El Kholy, N. Habash, G. Leusch, E. Matusov, and H. Sawaf, “Language independent connectivity strength features for phrase pivot statistical machine translation,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 13, 2013.

[8] M. Carpuat and M. Diab, “Task-based evaluation of multiword expressions: a pilot study in statistical machine translation,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2010, pp. 242–245.

[9] B. Fadi, N. Habash, and J. Hirschberg, “Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*, 2009.

[10] 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 (ASRU)*, Dec. 2011.

[11] D. Povey and G. Saon, “Feature and model space feature adaptation with full covariance gaussian,” in *Proceedings of the 9th International Conference on Spoken Language Processing (ICSLP)*, March 2006, pp. 4330–4333.

[12] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, “Subspace gaussian mixture models for speech recognition,” in *Proceedings of 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, March 2010, pp. 4330–4333.

[13] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magagazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] G. E.Dahl, Y. Dong, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30 –42, january 2012.

[15] F. Seide, G. Li, X. Chien, and D. Yu, “Feature engineering in context- dependent deep neural networks for conversational speech transcription,” in *Proceedings of Automatic Speech Recognition and Understanding Workshop (ASRU))*, 2011.

[16] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *Proceedings of 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013.

[17] Y. Zhang, E. Chuangsuwanich, and J. Glass, “Extracting deep neural network bottleneck features using low-rank matrix factorization,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

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

[19] 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*, Dec 2013, pp. 55–59.