QCRI ADVANCED TRANSCRIPTION SYSTEM (QATS) FOR THE ARABIC MULTI-DIALECT BROADCAST MEDIA RECOGNITION: MGB-2 CHALLENGE

Sameer Khurana, Ahmed Ali

Qatar Computing Research Institute, HBKU, Doha, Qatar

ABSTRACT

In this paper, we describe Qatar Computing Research Institute's (QCRI) speech transcription system for the 2016 Dialectal Arabic Multi-Genre Broadcast (MGB-2) challenge. MGB-2 is a controlled evaluation using 1,200 hours audio with lightly supervised transcription Our system which was a combination of three purely sequence trained recognition systems, achieved the lowest WER of 14.2% among the nine participating teams. Key features of our transcription system are: purely sequence trained acoustic models using the recently introduced Lattice free Maximum Mutual Information (LF-MMI) modeling framework; Language model rescoring using a four-gram and Recurrent Neural Network with MaxEnt connections (RNNME) language models; and system combination using Minimum Bayes Risk (MBR) decoding criterion. The whole system is built using kaldi speech recognition toolkit.

Index Terms— Kaldi, purely sequence trained acoustic models, Bi-directional LSTM, RNN LM, Arabic Speech Recognition, QATS

1. INTRODUCTION

The 2016 *Arabic MGB challenge* (MGB-2) is a continuation of *MGB challenge* introduced in 2015 for English language [1]. The challenge consisted of two tasks; Arabic speech transcription and speech-text alignment. The goal of the speech transcription task is to further the state-of-the-art in Arabic speech recognition, which is a challenging task considering the dialectal variation that is inherent in any Arabic speech database. In this paper, we present Qatar Computing Research Institute's Arabic speech transcription system that achieved the lowest Word Error Rate (WER) among the nine participating teams. The WER achieved by our system was 14.2%, which was 2 percentage points (absolute) better than the second best system, which achieved a WER of 16.2%.

Our main focus in this work was to investigate the performance of recently introduced, purely sequence trained acoustic modeling framework [2], trained using Lattice Free Maximum Mutual Information (LF-MMI) objective function. We trained three different acoustic model (AM) architectures using LF-MMI modeling framework; Time Delay Neural Network (TDNN) [3], Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN) [4] and Bi-directional LSTM (Section 4). Apart from acoustic modeling, we also construct N-gram and RNN language models (LMs), that are used in the decoding and language model rescoring stage (Section 3). Our final recognition system, is a combination of the three recognition systems that are LF-MMI trained and rescored using N-gram and RNN language models. (Section 5).

2. DATA DESCRIPTION

The MGB-2 Challenge used recorded programs from 10 years of Aljazeera Arabic TV channel with total of 1,200 hours worth of audio for Acoustic Modeling (AM). The original transcription has no timing information, and lightly supervised algorithms have been used to recover the timing information for each word. However, the human transcription was meant to be convenient for reader, and not necessary verbatim transcription. The quality of the transcription varies significantly, there have been two major challenges in the given transcription; a) conversational speech, which includes multiple dialects and overlapping talkers, which is the typical scenario for political debate and talk show programs, b) dubbed speech, this happens when the speech is not Arabic. Neither the overlap speech nor the dubbed speech is marked in the original transcription. The recordings are coming from TV programs with Modern Standard Arabic (MSA) dominating most of them. It was roughly estimated to be more than 70% of the speech is MSA, and the rest is spoken in different Dialectal Arabic (DA) namely as: Egyptian (EGY), Gulf (GLF), Levantine (LEV), and North African (NOR).
The recognized output was aligned with the original transcription to generate small speech segments on average between five and 30 seconds per segments suitable for building speech recognition system. For each segment, the Average Word Duration in seconds (AWD), Phoneme Matching Error Rate (PMER), and Word Matching Error Rate (WMER) are stored in the given meta-data to be potentially used for further filtering, and data selection.

2.1. Acoustic Modeling Data

Initially, we used about 250 hours of training data for AM experiments. This is the same amount of data that has been used to report the baseline system. This will be called sample data through the rest of the paper. The Sample data are those segments with WMER less than 80%, and limited to the first 500 programs. The full AM train data comes from all segments with MWER less than 80% which was summed up to more than 370K segmented across the 2214 programs creating more than 1200 hours speech segments. The development and evaluation are coming from diverse 17 hours each that have not been in the training data. The program title itself may have been seen, but not these particular episodes.

Type	Hours	Programs	#segments
SampleData	250h	500	83K
Training	1200h	2214	370K
Development	10h	17	5800
Evaluation	10h	17	5600

Table 1. *Data used for acoustic model training, development and evaluation*

2.2. Language Modeling Data

We have used the provided BuckWalter format for the transcription which doesn't have any punctuation or dicraization, however we didn't use any text normalization like normalizing Alef, yaa, and taa marbouta in the given text. LM experiments have been using the grapheme lexicon provided by the organizer for most of the experiment and for the final submission. The grapheme based lexicon has 1:1 word-to-grapheme mapping, which means the vocabulary size is the same as the lexicon size. The main motivation to use the grapheme is that Arabic is phonologically complex language [5], and we need a huge lexicon to reduce OOV, especially for Dialectal Arabic speech, more details about this can be found our previous study [6].

Type	Tokens	Vocab
In-domain	8M	200k
Background	130M	1M

Table 2. *In-domain data refers to the training transcripts and Background data refers to the extra Arabic language modeling text provided for the challenge*

3. LANGUAGE MODELS

3.1. N-Gram Language Models

We train two N-gram language models (LMs); A tri-gram LM (KN3) is trained using the spoken utterances transcripts, which we refer to as the in-domain data. This LM is used for decoding to generate decode lattices. These lattices are then rescored using a four-gram LM (KN4), which is trained on the in-domain and the extra language modeling text, which we refer to as the background data, provided by the challenge organizers for building better LMs. We use interpolated kneser-ney smoothing on both the LMs, which are built using the SRILM toolkit [7]. We limit the LM vocabulary to top 900k most frequent words in the text, which is same as the speech lexicon. Table 3 shows the perplexity on the dev set by using the tri-gram and four-gram LMs.

3.2. Recurrent Neural Network Language Model

We trained a Recurrent Neural Network Language Model with MaxEnt Connections (RNNME) using RNNLM-Toolkit [8]. RNNLM-Toolkit is arguably the first toolkit publicly released to construct RNN language models. As the training procedure in this toolkit is CPU based, it takes a considerable amount of time to train a LM and hence we go straight to building a RNNME LM, which has been shown to perform better than RNN LM without direct connections (MaxEnt) between the input and the ouput layer [8].

RNNME refers to a RNN architecture which along with recurrent connections, also has non-recurrent or direct connections between input and output layer. These direct connections are known as MaxEnt connections which derives its name from Maximum Entropy language model. This kind of RNN architecture provides a way to jointly train an N-gram LM and a RNN LM. RNNME has been shown to perform better than the conventional RNN LM. See [8] for details. Table 3 gives perplexity on the dev set using RNNME LM.

In this work, we train a Class Based RNNME LM, with hyperparameter settings as follows; **class dimensions:** 200, **input-layer-size:** 40k, which is also the language model vocabulary, which is restricted to the top 40k most frequent words, **hidden-dimension:** 300, **hidden-activation function:** sigmoid, **direct-connections:** 2000M, which are the number of weights used for direct connections between the input and the output layer, **n-gram order:** 3, which is referred to as the direct-order in the RNNLM-toolkit Fig 2 shows the RNNME architecture along with the hyperparameter settings used.

3.3. Summary

In this work, we train three LMs; Tri-gram, Four-gram and RNN with MaxEnt connections. We give the parameter settings of all the LMs and present the perplexity results on the

Model	Tokens	Vocab	LM-Vocab	PPL (Dev)
KN3	8M	200k	900k	640
KN4	130M	1M	900k	590
RNNME	130M	1M	40k	400

Table 3. *Perplexity on the dev set using LMs built for decoding and language model rescoring*

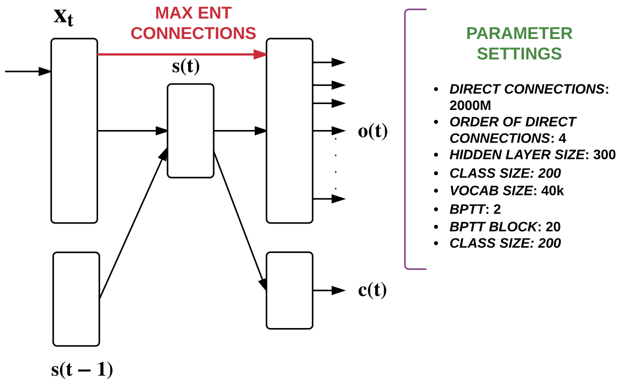


Fig. 1. RNNME LM architecture and the hyperparameter settings

dev set. The RNNME LM took 2 weeks to train and because of limited time available, we could only train one LM. A comprehensive hyperpaprameter tuning and experimentation with different RNN architectures is left as an extension to this paper.

4. NEURAL NETWORK ACOUSTIC MODELS

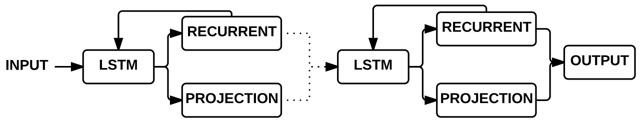
In this section, we give brief details about the acoustic models that we trained using the kaldi toolkit [9]. We discuss the architectures, the training objectives used for training these models, the hyperparameter settings and the input features used for developing these models.

4.1. Training Setups

We experimented with the following neural network architectures; feed forward deep neural network (FDNN), Long-short term memory RNN (LSTM), Bi-directional LSTM (BLSTM) and Time Delay Neural Network (TDNN). Below, we give brief details about the training setups used in our experiments.

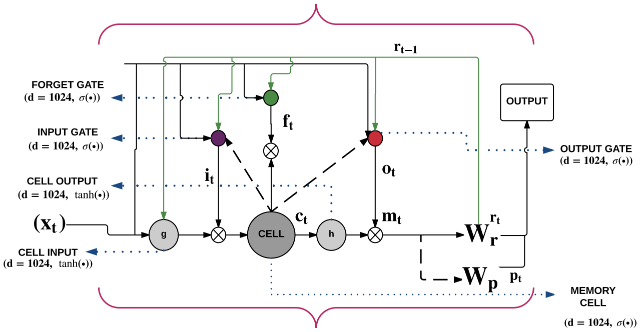
1. **FDNN**: FDNN is trained using adapted MFCC features as input. The model has 5 hidden layers, each layer having 2048 sigmoidal neurons. Input to the FDNN is 40 dimensional transformed Mel-Frequencey Ceptral Coefficient (MFCC) [10] feature vector (MFCC_A) which is extracted as follows; 9 frames of 13 dimensional MFCC feature vectors are spliced together, whitened (Mean Normalized) and reduced to 40 dimensional representation using LDA, followed by Maximum Likelihood Linear Transform (MLLT) [11] and speaker normalization technique known as feature-space Maximum Likelihood Linear Regression (fMLLR) or contrained maximum-likelihood linear regression (cMLLR) [12]. fMLLR transform is obtained from a baseline GMM-HMM system with speaker adaptive training (SAT). The output of the FDNN is a softmax layer, whose units correspond to triphone-states. A baseline GMM-HMM system provides frame vs HMM-state alignments that are used as training examples in a multi-class classification setting. The FDNN is trained to minimize the Cross Entropy loss function using Stochastic Gradient Descent (SGD). We use a learning rate of 0.008 for SGD for the first epoch and for later epochs, the learning rate is decided using ”new-bob” algorithm as explained in [13]. Training is performed in mini-batches; we use mini-batches of size 256.

2. **LSTM**: Over the past few years RNN based acoustic models have shown tremendous improvements in recognition performance by reducing the WER significantly. Long Short Term Memory (LSTM) is particular type of RNN architecture which is now widely used for AM. We use the LSTM architecture with recurrent and non-recurrent projection layers as used in [4, 14] and given by the block diagram below. LSTM AM is trained using concatenated 40 dimensional hi-resolution MFCC features (MFCC_hires) and 100 dimensional i-Vectors [15] for each frame. We denote these features as MFCC_B. We use 3 LSTM layers with a delay of -1,-2 and -3 at each layer for the Cross Entropy trained model, while for the purely sequence trained model the delay at each layer is chosen to be -3. An output label delay of 5 is also used. Purely sequence trained models are trained using a sequence objective, without the need of Cross Entopy training.



Sequence training of Neural Networks using Connectionist Temporal Classification (CTC) training objective have become quite popular in speech recognition [4, 14]. Recently, CTC inspired training framework for acoustic models was introduced in kaldi toolkit [2], where Lattice Free version of the Maximum Mutual Information training criterion (LF-MMI) is used to train the acoustic models. In this work, we train LSTM acoustic model with the Cross Entropy training objective and also the newly introduced LF-MMI modeling framework. For details about different training objectives, see [16].

A major component in an LSTM model is the memory block, that consists of input (i), output (o), forget (f) gates that control the flow of input information (g) from the previous hidden layer and the output information (h) to be passed onto the next layer. For more details see excellent explanation in [4]. The hyperparametrs of the LSTM memory block are best given by the following labeled diagram.



where, $\mathbf{W_r}$ and $\mathbf{W_p}$ refer to the recurrent and non-recurrent projection spaces respectively, which are of dimensions 256.

In this case, MFCCs, used as input features, are not adapted or whitened like in FDNN training. The reason for not doing mean normalization of MFCC features is to let the i-Vectors provide the speaker related mean offset information, as explained in [17] and [3]. For the same reason, the i-Vector extractor in trained on top of features that are not mean normalized so that the mean offset information can be encoded in the i-Vectors [3]. Once, the i-Vector extractor is trained, i-Vectors for the training and test data are extracted in an online fashion i.e. only prior frames to the current frame are used along with the prior utterances from the same speaker to extract the i-Vectors. The i-Vector extraction framework consist of a GMM-UBM trained on top of LDA+MLLT MFCCs and consists of 512 Gaussian components that makes use of 200k feature frames for UBM modeling. UBM stats are then modeled using a factor analysis model known as *total variability subspace model* [15] given by the following equation.

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

where, M is the utterance based mean super-vector, m is the UBM mean super-vector, u is the i-Vector and T is the *total variability subspace*. The parameters of the model are learned in an unsupervised manner. In our case, *variability subspace*, T, was chosen to be of 100 dimensions. For more details about i-Vector modeling framework, reader is referred to the excellent work in [15] and for more details about the input features, see [17].

3. **BLSTM:** Acoustic Model architecture for BLSTM is exactly similar to LSTM, except that the training occurs in both the directions; left to right and vice versa. We train BLSTM AM using only LF-MMI training objective [2].

4. **TDNN**: TDNN is trained using concatenated 40 dimensional hi-resolution MFCCs (MFCC_hires) with 100 dimensional i-Vectors for each speech frame; the same input features as used for (B)LSTM acoustic models. TDNNs require less training time than sequence models such as LSTMs, while attempting to capture the long-term temporal dependencies just like a sequence model. We use the same TDNN architecture as given in [3], except the input splicing indexes used are as given in [2]. The splicing indexes used are $-1, 0, 1$ $-1, 0, 1, 2$ $-3, 0, 3$ $-3, 0, 3$ $-3, 0, 3$ $-6, -3, 0$ 0 for the LF-MMI modeling framework i.e. input to the TDNN is 3 frames spliced together $(-1, 0, 1)$, the hidden layers see 3 frames of the previous layer, separated by three frames $(-3, 0, 3)$. Splicing indexes for the CE training of TDNN are $-2, -1, 0, 1, 2$ $-1, 2$ $-3, 3$ $-7, 2$ 0. More details about the architecture can be found in [3] and [2].

4.2. Summary

In this section, we give details about the training setups for the four AMs that we trained; FDNN, TDNN, LSTM and BLSTM. FDNN is trained using the Cross Entropy training criterion using adapted 40 dimensional MFCC features (MFCC_A) as input. TDNN and LSTM are trained using both the Cross Entropy and the recently introduced LF-MMI modeling framework (purely sequence trained), while BLSTM was trained using only LF-MMI training criterion. The input features used for these AMs are per frame concatenation of 40 dimensional unadapted and un-normalized hi-resolution MFCCs and 100 dimensional i-Vectors, which are extracted in an online fashion (MFCC_B).

5. SYSTEM DESCRIPTION

In this section, we give details about our overall final system that we submitted as part of the MGB-2 speech transcription challenge. We train six AMs, but three best AMs make it to the final system. The three AMs are LF-MMI trained TDNN, LSTM and BLSTM. AM training is followed by decoding using a tri-gram LM (KN3) that is built using the spoken transcripts text, that generates decode lattices. The decoding process is followed by four-gram LM rescoring of the full lattices, in which the alternate hypothesis paths in the decode lattices are rescored using a better LM, which is trained on the whole text (in-domain + background). RNNME LM is then used to perform N-best list rescoring of the rescored four-gram lattices. The three sets of final lattices corresponding to three AMs are then combined using Minimum Bayes Risk (MBR) decoding criterion to give the recognition output. See Fig 2 for the block diagram of our final system.

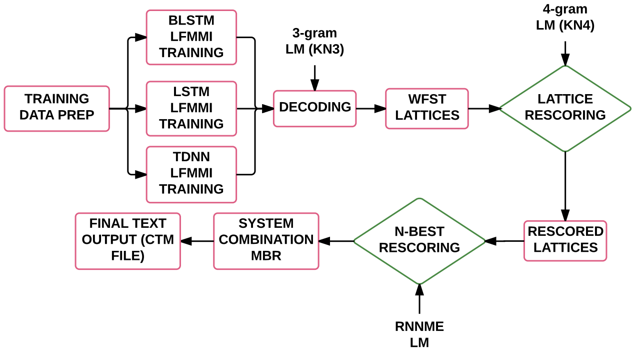


Fig. 2. *System Description of the final Arabic speech transcription system*

6. EXPERIMENTS AND RESULTS

6.1. GMM-HMM Baseline System

We train a GMM-HMM recognition system that provides frame vs HMM-state alignments that are used for training the neural network acoustic models. GMM-HMM system is built using whitened (Mean Normalized) spliced MFCC features that are transformed using LDA and MLLT, followed by Speaker Adaptive Training (SAT) [18]. We use kaldi to build the baseline system which is explained in [9]. The %WER using the baseline GMM-HMM system is 40.2 on the dev set.

6.2. Neural Network Based Recognition Systems

Here, we give performance of the neural network recognition systems, that are built as part of this work.

1. **Data Augmentation** We use the audio augmentation technique proposed in [19]. We perform audio speed perturbation with speed factors of $0.9, 1.0, 1.1$. This gives us three times the original speech utterances. The speed perturbed data is followed by volume perturbation with volume factors that are uniformly sampled from the interval $[\frac{1}{8}, 2.0]$. The same data augmentation approach was also used in [3].

2. **Decoding** Table 4 gives recognition performance on the dev set using the CE trained recognition systems, decoded using tri-gram LM (KN3). MFCC_A refers to the transformed features (MFCC+LDA+MLLT+fMLLR), while MFCC_B refers to concatenated hi-resolution MFCCs and i-Vectors. We use publicly available kaldi decoders [9], using a beam width of 15.0. Table 5 gives recognition performance on dev set of the LF-MMI trained recognition systems. In this case, augmented data is used for training.

Model	Feats	Criterion	AUG	%WER
FDNN	MFCC_A	CE	$\mathcal{N}$	31.8
TDNN	MFCC_B	CE	$\mathcal{N}$	27.3
LSTM	MFCC_B	CE	$\mathcal{N}$	23.6

Table 4. *Recognition results for the CE trained ASR systems. LM used for decoding is KN3. Data augmentation (AUG) is not used in this case*

Model	Feats	Criterion	AUG	%WER
TDNN	MFCC_B	LF-MMI	$\mathcal{Y}$	23.0
LSTM	MFCC_B	LF-MMI	$\mathcal{Y}$	20.9
BLSTM	MFCC_B	LF-MMI	$\mathcal{Y}$	19.3

Table 5. *Recognition results for the LF-MMI trained recognition systems. LM used for decoding is KN3. Data augmentation is used before training*

3. **Four-gram LM Rescoring:** The decode lattices obtained from LF-MMI trained recognition systems, from the previous step are rescored using the four-gram language model (KN4), which is built using all the language modeling text available. It assigns a new graph score to each alternated hypothesis path in the lattice by scoring it using the KN4 language model. Table 6 shows improvements in recognition results due to four-gram LM rescoring.

4. **RNNME LM Rescoring:** We rescore the KN4 rescored lattices obtained from the previous step, using an RNNME LM. Full lattice rescoring is inefficient using RNN LMs and hence, we extract the N-best hypotheses for each utterance and rescore the N-best list. In our case, N is 1000. We found out that the interpolation of

Model	Criterion	%WER(KN3)	%WER(KN4)
TDNN	LFMMI	23.0	21.5
LSTM	LFMMI	20.9	20.1
BLSTM	LFMMI	19.3	18.5

Table 6. *Recognition results after performing the four-gram LM (KN4) rescoring of the decode lattices*

the scores that RNNME LM assigns to the hypotheses with the score assigned by the KN4 language model gives us the best recognition performance. The interpolation parameters are 0.3 and 0.7 for the KN4 LM score and RNNME LM score respectively. These parameters are optimized on the dev set. Table 7 shows the results of N-best rescoring. Clear improvements in the recognition results can be seen after performing N-best list rescoring.

Model	Criterion	%WER(KN4)	%WER(KN4 +RNNME)
TDNN	LFMMI	21.5	20.5
LSTM	LFMMI	20.1	19.1
BLSTM	LFMMI	18.5	17.9

Table 7. *Recognition results after performing interpolated KN4 and RNNME LM rescoring. Interpolation parameters are 0.3 for KN4 and 0.7 for RNNME*

6.3. System Combination

Our best system is the combination of the LF-MMI trained recognition systems that are rescored using four-gram and RNNME language models i.e. we combine the three recognition systems mentioned in Table 7.

The three sets of output lattices are combined to form a union lattice, which is then used as an input to minimum bayes risk (MBR) decoding pipeline, to get the final recognition output on the evaluation and development set, which was submitted as our entry to the MGB-2 speech transcription challenge. **Our recognition output had the lowest %WER of 14.2% on the evaluation data and 16.7% on the development data, from among the nine-participating teams**.

6.4. Summary

In this section, we reported the recognition results obtained using the LF-MMI and CE trained recognition systems. Language model rescoring significantly helps the recognition performace, in particular interpolation of language model scores from the four-gram and RNNME LM gives significant improvements in the recognition performance. LF-MMI trained systems performs significantly better than CE trained systems. We acknowledge that the comparison between the CE trained TDNN, LSTM and BLSTM and LF-MMI trained systems is not fair, because of the fact that augmented data is used to train LF-MMI systems, while no data augmentation is used to train CE systems. Due to limited time, we could not perform a principled comparison between the two training criterion and plan to do so as an extension of this work. We also acknowledge that our recognition system is based on the publicly available recipes in the kaldi toolkit [9]. The LF-MMI systems are based on the excellent work in [2].

7. CONCLUSIONS AND FUTURE WORK

We participated in the MGB-2 Arabic speech transcription challenge with our recognition systems that are built using the recently introduced LF-MMI modeling framework [2] and achieve the lowest %WER of 14.2% on the evaluation data. Our final system is a combination of three LF-MMI trained models; TDNN, LSTM and BLSTM. The models are rescored using a four-gram and RNNME LM before combination. We acknowledge that due to lack of time, we were not able to make a detailed and principled comparison between the CE and LF-MMI trained systems. We also did not manage to train RNNLMs with different architectures and objective functions such as Noise Contrastive Estimation and Variance Regularization, implemented in the recently introduced CUED-RNNLM toolkit [20]. We are in the process of doing that analysis and hope to publish it as an extension to this work, soon.

8. REFERENCES

[1] Peter Bell, MJF Gales, Thomas Hain, Jonathan Kilgour, Pierre Lanchantin, Xunying Liu, Andrew McParland, Steve Renals, Oscar Saz, Mirjam Wester, et al., "The mgb challenge: Evaluating multi-genre broadcast media recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 687–693.

[2] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," *Submitted to Interspeech*, 2016.

[3] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," .

[4] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," 2014.

[5] Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj

Pooleery, Owen Rambow, and Ryan Roth, “Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic.,” in *LREC*, 2014, vol. 14, pp. 1094–1101.

[6] Ahmed Ali, Hamdy Mubarak, and Stephan Vogel, “Advances in dialectal arabic speech recognition: A study using twitter to improve egyptian asr,” in *International Workshop on Spoken Language Translation (IWSLT 2014)*, 2014, pp. http–workshop2014.

[7] Andreas Stolcke et al., “Srilm-an extensible language modeling toolkit.,” .

[8] Stefan Kombrink and Anoop Deoras, “RNNLM - Recurrent Neural Network Language Modeling Toolkit,” .

[9] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[10] Steven Davis and Paul Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[11] Ramesh A Gopinath, “Maximum likelihood modeling with gaussian distributions for classification,” in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. IEEE, 1998, vol. 2, pp. 661–664.

[12] Mark JF Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[13] Shakti P Rath, Daniel Povey, Karel Vesel, and Jan Honza Cernock, “Improved feature processing for Deep Neural Networks ,” pp. 1–5.

[14] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, “Fast and accurate recurrent neural network acoustic models for speech recognition,” *arXiv preprint arXiv:1507.06947*, 2015.

[15] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[16] Karel Veselỳ, Arnab Ghoshal, Lukás Burget, and Daniel Povey, “Sequence-discriminative training of deep neural networks.,” 2013.

[17] Daniel Povey, Xiaohui Zhang, Sanjeev Khudanpur, and Speech Processing, “PARALLEL TRAINING OF DNNS WITH NATURAL GRADIENT DESCENT,” pp. 1–28, 2015.

[18] Spyros Matsoukas, Rich Schwartz, Hubert Jin, and Long Nguyen, “Practical implementations of speaker-adaptive training,” in *DARPA Speech Recognition Workshop*. Citeseer, 1997.

[19] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *Proceedings of INTERSPEECH*, 2015.

[20] Xie Chen, Xunying Liu, Yanmin Qian, MJF Gales, and PC Woodland, “Cued-rnnlm??? an open-source toolkit for efficient training and evaluation of recurrent neural network language models,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6000–6004.