JHU KALDI SYSTEM FOR ARABIC MGB-3 ASR CHALLENGE USING DIARIZATION, AUDIO-TRANSCRIPT ALIGNMENT AND TRANSFER LEARNING

Vimal Manohar, Daniel Povey, Sanjeev Khudanpur

Center for Language and Speech Processing,
Human Language Technology Center Of Excellence,
Johns Hopkins University, Baltimore MD

{vimal.manohar91,dpovey}@gmail.com,khudanpur@jhu.edu

ABSTRACT

This paper describes the JHU team's Kaldi system submission to the Arabic MGB-3: The Arabic speech recognition in the Wild Challenge for ASRU-2017. We use a weights transfer approach to adapt a neural network trained on the out-of-domain MGB-2 multi-dialect Arabic TV broadcast corpus to the MGB-3 Egyptian YouTube video corpus. The neural network has a TDNN-LSTM architecture and is trained using lattice-free maximum mutual information (LF-MMI) objective followed by sMBR discriminative training. For supervision, we fuse transcripts from 4 independent transcribers into confusion network training graphs. We also describe our own approach for speaker diarization and audio-transcript alignment. We use this to prepare lightly supervised transcriptions for training the seed system used for adaptation to MGB-3. Our primary submission to the challenge gives a multi-reference WER of 32.78% on the MGB-3 test set.

***Index Terms*—** Multi-genre broadcast, Automatic speech recognition, Lightly-supervised training, LF-MMI, Segmentation

1. INTRODUCTION

The Arabic Multi-Genre Broadcast MGB-3 challenge [1] is an extension to the MGB-2 challenge [2]. While the MGB-2 challenge was on a multi-dialect Arabic corpus with recordings from Aljazeera Arabic TV channel spanning over 10 years, the MGB-3 challenge focuses on dialectal Arabic from Egyptian multi-genre YouTube videos. The MGB-3 challenge has a total of 80 YouTube programs from seven different genre. The first 12 minutes of each was segmented and transcribed manually. This 16 hours of data was split into 5 hour adaptation set, 5 hour development set and 6 hour evaluation set. Unlike the MGB-2 challenge that had 1200 hours of audio data, albeit with only lightly-supervised transcriptions, the MGB-3 challenge has only 5 hours of in-domain audio data (the adaptation set) for training. The tiny amount of in-domain data along with mismatch in audio conditions (TV broadcast vs YouTube) makes this a transfer learning challenge.

An additional challenge is that the MGB-3 data was transcribed independently by 4 different annotators. This is to get around the fact that the dialect does not have clearly defined orthography i.e. different people write the same word in slightly different forms. The evaluation was also performed using a multi-reference word error rate (MR-WER) [3] metric instead of the standard WER.

The JHU team did not participate in the MGB-2 challenge. So our first work in the MGB-3 challenge was to develop a seed system trained on the MGB-2 data. We used the baseline lightly-supervised transcripts [2] provided by the challenge organizers to train seed systems. The seed systems were used in our own Kaldi [4] implementation of diarization and segmentation to retrieve best-matching transcripts for audio segments. We recovered a total of 982 hours of Arabic TV data. We used it to train a TDNN-LSTM neural network [5] with LF-MMI [6] objective followed by sMBR [7] training.

The second part of the challenge was to adapt the system to the 5 hours MGB-3 adaptation data, for which we used transfer learning using LF-MMI objective [8]. We tried both using the 4 independent transcriptions separately as well as fusing them into a confusion network to create training graphs. Confusion network has been proven to be effective in dealing with uncertainties in various ASR applications [9]. The primary system submission was a lattice-level minimum Bayes risk (MBR) system combination [10] of the sMBR and non-sMBR system. This system had an MR-WER of 33.41% on the MGB-3 dev set and 32.78% on the test set.

This paper is organized as follows. Section 2 describes our approach of segmenting the 1200 hours MGB-2 training data and retrieving the matching transcriptions for the same. Section 3 describes the seed system trained on the segmented data. Section 4.1 describes the adaptation to MGB-3 data. In sections 5 and 6, we present the results and conclusions, and discuss some post-evaluation findings.

2. SEGMENTATION AND TRANSCRIPT RETRIEVAL

While we used the lightly-supervised transcripts [2] provided by the challenge organizers for a seed system, we used our own implementation in Kaldi of diarization and audio-transcript alignment[1] to obtain transcripts for audio segments of the MGB-2 corpus. This involves two stages: first one is a diarization involving only the audio, and the second one is a segmentation of the recording along with retrieval of transcripts for those segments. The diarization is required to create segments that have a single speaker, so that we can use online i-vector based speaker adaptation for neural networks [11, 12]. The diarization is based on i-vectors [13] and probabilistic linear discriminant analysis (PLDA) [14] using a pipeline similar to [15], but adapted to broadcast scenario by incorporating some approaches from conventional diarization systems like [16]. This is described in section 2.1. The transcript alignment approach we used has similarities to [17, 18, 2] and uses decoding with a biased language model (LM) trained on the raw transcript. The raw recordings are segmented and transcripts are obtained as the best-matching sub-sequence of words in the raw transcription, followed by some cleanup. This procedure is described in section 2.2

2.1. Speaker diarization

The objective in this section is to create initial segments that contain only speech from a single speaker, so that we can use i-vector based speaker adaptation of neural networks [11]. Since, we use online i-vector extraction, the actual identity of the speaker is not very important. However, we do want clusters (speakers) containing sufficient number of frames to reliably estimate i-vectors, and hence must avoid over-creating clusters.

We use a speaker diarization approach based on i-vectors [13] and PLDA [14] similar to the work in [15]. Prior to doing diarization, we use a neural network based speech activity detection to remove silence regions and work only on segments of speech.

2.1.1. Training

We use an i-vector extractor trained on 1200 hours of MGB-2 data with segments and speaker metadata provided by the challenge organizers[2]. The universal background model (UBM) comprises of 2048 gaussians trained on 20 dimensional MFCCs with delta and delta-delta features. The total variability matrix has a dimension of 400. We scaled down the i-vector estimation statistics by 0.3 to account for the correlation between adjacent frames, and only accumulated statistics on the high energy frames, using an energy-based VAD.

To train the PLDA model, we use random chunks of utterances with durations between 3 and 20 seconds from each utterance and keep at least three chunks per speaker. We assume the mean-normalized, whitened and length-normalized [19] i-vectors follow a gaussian PLDA model with full-covariance residue i.e. no explicit eigenchannel space. We transform the i-vector offset around mean $\mathbf{m}$ using a matrix $\mathring{A}$ as in [14] to a new space where the within-class covariance is unit and between-class covariance is a diagonal matrix Λ. Thus, the i-vector $\mathbf{w}$ has a generative model:

$$\mathbf{w} = \mathbf{m} + \mathbf{A}\mathbf{u}, \tag{1}$$

$$\mathbf{u} \sim \mathcal{N}(\cdot|\mathbf{v}, \mathbf{I}), \tag{2}$$

$$\mathbf{v} \sim \mathcal{N}(\cdot|0, \Lambda). \tag{3}$$

The parameters of the PLDA model $\{\mathbf{m}, \mathbf{A}, \Lambda\}$ are trained using expectation-maximization algorithm for 10 iterations. At the end of this stage, we have an i-vector extractor consisting of the UBM and the total variability matrix, and a trained PLDA model consisting of the i-vector transformation matrix and diagonal between-class covariance.

2.1.2. Clustering

The clustering process is similar to the one in [15] and consists of the three stages – temporal segmentation, i-vector extraction and agglomerative hierarchical clustering (AHC). But we had to make minor modifications inspired by conventional diarization systems [16] for the system to work efficiently in the broadcast recording scenario. This approach was tuned to work well on the English broadcast news evaluation sets [20] from '96 to '99.

The raw recordings of the MGB-2 corpus are about 20-60 minutes long. This made it inefficient to uniformly segment the recording into 1.5s overlapping chunks as in [15]. Instead, we first do a change point detection [21] using full-covariance gaussians estimated on adjacent 1.5s long windows. We measure the generalized likelihood ratio (GLR) between gaussians estimated on the individual windows and a gaussian estimated on the combined 3 second window. We designate a change point at isolated local maxima of these GLR distance values. We follow this up with a linear clustering step that merges adjacent segments based on full-covariance gaussian Bayesian information criterion (ΔBIC) [16]. We extract i-vectors for each cluster at this stage. These are mean centered, whitened, length-normalized [19] and transformed using a recording-dependent PCA [15]. We retain the top dimensions in the PCA space so as to keep 0.9 of the total energy; this is usually around 200-250.

The initial clusters are merged using AHC using a distance metric of PLDA log-likelihood ratio (LLR) before and

[1] https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/cleanup/segment_long_utterances.sh

[2] It was filtered with a threshold of 80% word matching error rate.

after merging:

$$d(S_1, S_2) = \log \frac{P(\mathbf{w}_{1,2\ldots n_1}^{(S_1)}|S_1;\Lambda)P(\mathbf{w}_{1,2\ldots n_2}^{(S_2)}|S_2;\Lambda)}{P(\mathbf{w}_{1,2\ldots n_1}^{(S_1)}\mathbf{w}_{1,2\ldots n_2}^{(S_2)}|S_1 \cup S_2;\Lambda)}, \quad (4)$$

where $\mathbf{w}_{1,2\ldots n_1}^{(S_1)}$ are transformed i-vectors corresponding to the n_1 segments in cluster S_1, $\mathbf{w}_{1,2\ldots n_2}^{(S_2)}$ are transformed i-vectors corresponding to the n_2 segments in cluster S_2, Λ is the diagonal between-class covariance matrix. The likelihoods in (4) computed using the equation 6 in [14]. An alternative distance metric for the one in (4) is the average pairwise PLDA LLR, which was used in [15]. However, in our experiments in English broadcast news, we found the actual PLDA log-likelihood ratio was more robust. We, however, believe that using average pairwise PLDA LLR objective would have worked equally well.

The merging of clusters using the AHC procedure is stopped when the best distance is larger than a particular threshold that is usually obtained by a score calibration procedure. We could not use the approach in [15] for automatic calibration because the number of segments corresponding to the different speakers far out-numbers the segments corresponding to the same speaker, and we could not fit two separate gaussians on the PLDA LLR scores of the initial clusters (at the beginning of AHC). We use the following automatic calibration approach for determining the stopping threshold, which was also tested on the English broadcast news. We fit a gaussian to the pairwise PLDA LLR scores of the initial clusters. We take as the threshold the value at two standard deviations away from the mean of gaussian. We point out that the diarization is only for getting good initial segments that have the same or similar speakers and sufficient number of frames for i-vector estimation – the actual identity of the speakers is not very crucial since we use online i-vectors for speaker adaptation [12]. So we did not do experiments to tune the distance metrics and calibration thresholds.

2.2. Transcript retrieval

In this stage, we retrieve transcripts for segments of raw audio obtained from section 2.1. We follow the baseline recipe [22] to prepare a 250 hour subset of MGB-2 corpus with lightly-supervised transcripts. This is used to train a TDNN-LSTM network with 3 LSTMP [23] layers with LF-MMI objective. The system also uses i-vectors for speaker adaptation [11] using speaker information obtained from section 2.1. The reader is directed to [5] for details about the structure of the network and [6] for training details.

For each recording of the MGB-2 training set, the diarized segments obtained from section 2.1 are uniformly segmented if they are longer than 30 seconds. These segments are decoded using the seed acoustic model and a 4-gram unmodified Kneser-Ney interpolated LM [24, 25] trained on the raw transcript of that recording. The best path transcript is obtained for each segment. This transcript is aligned with the raw transcript of the recording using Smith-Waterman alignment [26] to select the best matching sub-sequence of words. For efficiency, we assume a linear cost for insertions and deletions; hence the algorithm is same as Levenshtein alignment, but forgives errors at the edges. Since the raw transcript of the recording can be really long with many (say 10000 words), we do not try to align each segment with the whole raw transcript, but align it with only some parts (we refer to these as "documents") of the raw transcript. This process is described in the section 2.2.1. The alignment information containing the best matching sub-sequence of words is converted into transcript. This process is described in section 2.2.2.

2.2.1. Document retrieval

We split the raw transcript into documents of around 1000 words. For each segment, we retrieve the documents that best match the hypothesized best path transcript of the segment. The document retrieval is done using term-frequency inverse-document-frequency (TF-IDF) similarity score [27] using n-gram terms with $n \in \{1, 2\}$. The IDF statistics are computed over all the documents from all the raw training transcripts. For the source document's TF-IDF value, we use

$$\text{tfidf}(t, d) = f(t, d) \log \frac{N}{n_t} \quad (5)$$

and for the query segment's TF-IDF value, we use

$$\text{tfidf}(t, q) = \left(0.5 + 0.5\frac{f(t, q)}{\max_t f(t, q)}\right) \log \frac{N}{n_t}, \quad (6)$$

where $f(t, d)$ is the raw count of the term-document pair (t, d), n_t is the number of source documents containing the term t and N is the total number of source documents.

Along with the retrieved best document, we also include about 200 words from the adjacent document on either side so that we don't lose any transcription that is at the edge of the best document. The sequence of words in the retrieved documents form the reference for the Smith-Waterman alignment.

2.2.2. Obtaining transcripts

We perform Smith-Waterman alignment between the reference sequence of words retrieved from the process in section 2.2.1 and the ASR hypothesized transcript. The reference part of Smith-Waterman alignment is retained as the "correct" transcript. Since the reference was aligned with the ASR hypothesis, we also have the timing for each reference word. We can use this to create segments by retaining mostly the correctly recognized words. We throw away segments for which the WMER between the reference and hypothesis is more than 50% or more than half the segment contains non-scored words (like Laugh, Cough etc.) or silence.

3. MGB-2 SEED SYSTEM

This section describes the seed system for MGB-3 challenge that is trained on the out-of-domain Arabic TV recordings from the MGB-2 corpus. After the first stage of segmentation and transcript retrieval described in section 2, we recovered 982 hours of audio with matched transcripts. This data is used to train a TDNN-LSTM network with LF-MMI objective [6] using the suggested Gale Arabic Kaldi recipe[3]. We use 100 dimensional i-vector for speaker adaptation of network [11, 12]. We additional add dropout on the LSTMP layers with a dropout proportion that peaks to 0.2 at 50% training and is 0 at the beginning and end of the training [5]. This is followed by sMBR training for 1 epoch.

For decoding, we used a 3-gram modified Kneser-Ney interpolated LM trained on the lightly-supervised MGB-2 transcripts to generate lattices. These were rescored with a 4-gram modified Kneser-Ney interpolated LM trained on all the MGB-2 transcripts and the 110 million word LM corpus collected from Aljazeera.net website as provided by the challenge organizers [2]. We did not do RNN-LM rescoring at this time and leave it for the future work.

The results are in table 1. The first row shows results with the lightly-supervised transcripts provided by the challenge organizers. The second row shows results with the transcripts retrieved by our approach of diarization and audio-transcript alignment described in section 2. This gives a comparable, but slightly better WER performance (17.6% vs 17.8% on MGB-2 dev); this demonstrates the utility of our diarization and audio-transcript alignment tools for this scenario. We further improved the acoustic model using dropout on LSTMP layers [5], which improves WER by 0.4%. Discriminative training using sMBR gives additional 1% improvement. This is our primary submission for the MGB-2 progress evaluation that gave 16% WER on the eval set.

The last column in table 1 shows the MR-WER results on decoding the MGB-3 data directly using the system i.e. without any adaptation to the MGB-3 data. We see that the MR-WER is poor in all cases for the unadapted network. We choose our primary submission (the last row) for adaptation. Our approaches for adaptation are described in the following sections.

Table 1. MGB-2 results

System	MGB-2		MGB-3
	dev	eval	dev
Baseline transcripts	17.8	-	49.20
Our transcripts	17.6	-	48.47
+ Dropout	17.2	-	47.42
+ sMBR	16.2	16.0	47.32

[3] https://github.com/kaldi-asr/kaldi/blob/f1d7891c5ea55884baceb4645754aff74fc3e0d3/egs/gale_arabic/s5b/local/chain/tuning/run_tdnn_lstm_1a.sh

4. ADAPTATION TO MGB-3 DATA

This section describes adaptation of the MGB-2 seed neural network to the Egyptian Arabic adaptation data of the MGB-3 corpus. We did not do any LM adaptation as we got no improvement on the dev data by interpolating n-gram counts. We leave this for future work.

4.1. Transfer learning

The MGB-2 seed system from section 3 is used to create the numerator supervision lattices to adapt the neural network. Since each segment in the adaptation data is transcribed by 4 independent transcribers, we create 4x copies of the utterances, one for each. We also perturb the speed to create 3x copies at speeds 0.9, 1.0 and 1.1, and perturb the volume randomly by a factor between 0.8 and 1.2. For LF-MMI training, we need to train a phone LM to create the denominator finite state transducer (FST) [6]. Since the amount of transcription in MGB-3 adaptation set is very small, we estimate the phone LM by combining the counts from the MGB-2 transcripts and the MGB-3 transcripts from all four transcribers. We use the same context-dependency tree of the seed system for the MGB-3 system.

We use a weight transfer approach suggested in [8]. We also tried the multi-task learning approach suggested in the same work, but initial experiments did not show better results than weight transfer. Since the suggested multi-task approach required training simultaneously on all of the MGB-2 data and the MGB-3 data, it was very difficult to do it within the time frame of the challenge and so we did not go forward with it. The affine component before the output is retrained, while the rest of the network is updated with a smaller (by a 0.1 factor) learning rate. The network is trained with LF-MMI objective for 0.5 epochs[4], followed by sMBR training for 1 epoch with the same learning rate for all layers. To generate the denominator lattice for sMBR, the decoding graph is created using a unigram word LM estimated with 0.01 weight on the counts from the MGB-2 acoustic transcripts and 1.0 weight on the counts from each transcriber of the MGB-3 adaptation data.

Since the amount of data in the adaptation set is very small, the network gets over-trained if a high learning rate is used or trained for many epochs. We found after the challenge period that it is better to just update the whole neural network using the MGB-3 data without reinitializing the final affine component as suggested in [8]. This suggests that even out-of-domain data is very critical for the training deep neural networks.

[4] Effectively like 6 epochs because we create 3x speed perturbation and 4x transcribers

5. RESULTS AND DISCUSSION

Table 2 shows results of transfer learning to MGB-3 adaptation data for the contrastive system submissions and the primary submission. For comparison, the unadapted neural network results are shown in the first two rows. These are the same as the last two rows in table 1. The first adapted system is just the LF-MMI trained system. This gives an MR-WER of 35.97%, which is a 11.35% absolute improvement over the seed system (47.32%). Adding sMBR training gives a 0.82% absolute improvement over this. The primary system submission, which is an MBR lattice combination of the two systems gives a 1.74% absolute gain in MR-WER.

Table 2. MR-WER (%) results on the submitted MGB-3 systems

	System	dev	test
Unadapted	without sMBR	47.42	-
	with sMBR	47.32	-
Adapted	LF-MMI	35.97	-
	LF-MMI + sMBR	35.15	-
	Primary	33.41	32.78

5.1. Post-evaluation period results

5.1.1. Transcript combination

The MGB-3 adaptation data has transcripts from 4 independent transcribers. We combined the transcripts into a confusion network using algorithm 1 and used these to creating supervision lattices for LF-MMI training. For estimating the phone LM for LF-MMI training, we combine the counts from the best path phone sequence in these lattices with the counts from the MGB-2 phone sequences. The results are in the column "re-init" of table 3. The first row shows results with using separate transcripts from the 4 transcribers independently. This is same as the LF-MMI result in table 2. Using confusion network (row 2) improves the result by 0.7% absolute to 35.27%.

5.1.2. Transferring all layers

After the challenge period, we found that it was better to not re-initialize the final layer of the neural network during transfer learning as done in section 4.1. So we just transfer all the layers trained on the MGB-2 dataset and train on MGB-3 data for 2 epochs. The learning rate for the final layer was set to 10 times that of the other layers. This gave a big improvement of almost 1.5-2% absolute as shown in the column "no re-init" of table 3.

Algorithm 1 Procedure to create confusion network fusing transcripts $\mathcal{T}$ from multiple transcribers

Input: Set of transcripts $\mathcal{T}$

procedure CREATECONFUSIONNETWORK($\mathcal{T}$)
  $N \leftarrow |\mathcal{T}|$
  Choose one of the N transcripts arbitrarily as the primary transcript p.
  for $t \in \mathcal{T} \setminus \{p\}$ **do**
    Do Levenshtein alignment of p and t to get a list of pairs $[(p_1, t_1), (p_2, t_2) \dots]$. We will refer to t as the secondary transcript. Insertions and deletions are represented with p_i and t_i respectively being ϵ.
  end for
  Thus, we have $\mathcal{A}$ = Set of $N-2$ alignments.
  Pick the longest alignment as the "base" alignment A_0 and add it to the confusion network so that each segment i of the confusion network has two arcs that correspond to p_i and t_i.
  for $A \in \mathcal{A} \setminus \{A_0\}$ **do**
    Align the entries of A with those of A_0 based on the word of the primary transcript. When there are ϵs as the primary word on A_0, we allow a sequence $t_a, t_{a+1} \dots$ from A to get aligned with the primary word entry in A.
  end for
end procedure

Output: Confusion network graph with N arcs on each segment. But one or more of the arcs could be ϵ.

Table 3. MR-WER (%) results on MGB-3 dev set with and without confusion networks

Supervision	re-init	no re-init
Separate transcripts	35.97	34.09
Confusion network	35.27	33.65

5.1.3. Improved phone LM for LF-MMI training

A 4-gram phone LM is used to create the denominator FST for LF-MMI training. In the previous experiments, we used a combination of phone n-gram counts from both MGB-2 and MGB-3 corpora (Results repeated in column "1:10" or table 4). This is because the MGB-3 corpus has too less data to estimate a good LM. But we only really need the LM to represent the MGB-3 transcripts. So we tried to give a higher weight by a factor 10 to the MGB-3 transcripts compared to the MGB-2 transcripts during phone LM estimation for LF-MMI denominator FST. From column "1:10" of table 4, we see that the results improved by 0.4% to 33.22% when using confusion networks. However, we could not get a similar improvement when using the individual transcripts separately; the performance degraded by 0.22% to 34.31%. More experiments are needed to find the right weights on the two sources, and how important it is to have the phone LM match the training transcripts. For comparison, we also show results using only the

counts from the targets data in the column "0:1". With confusion network supervision, this results in a much worse WER of 36.87%.

Adding sMBR on top of confusion network trained network without reinitializing the final affine component and using weights of 1:10 for source and target counts during phone LM estimation, gives around 0.8% absolute improvement to 32.45%.

Table 4. MR-WER (%) results on MGB-3 dev set with different weights (source:target) on counts for phone LM

	Weights		
Supervision	0:1	1:1	1:10
Separate transcripts		34.09	34.31
Confusion network	36.87	33.65	33.22
+sMBR	-	-	32.45

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the details of JHU's Kaldi system for MGB-3 Arabic ASR Challenge. We used a weights transfer approach using LF-MMI objective to adapt a TDNN-LSTM from MGB-2 Arabic TV broadcast corpus to MGB-3 Egyptian YouTube video corpus. Our primary submission, which is an MBR lattice-combination of sMBR and non-sMBR systems gave a competitive MR-WER of 32.78% on the MGB-3 test set. We further improved the system using a confusion network-type fusion of transcripts from independent transcribers to account for orthographic differences. We also presented our Kaldi implementation of diarization of long broadcast recordings and retrieval of transcripts for audio segments. This approach is shown to give a performance competitive with the lightly-supervised transcripts provided by the challenge organizers.

As future work, we would like to incorporate confidences into the audio-transcript alignment approach. The language model should be improved using RNN-LM and we would investigate methods for adaptation of language model to the MGB-3 adaptation data. More experiments are needed to finalize the diarization and segmentation recipes. We need to test the applicablity of the proposed transfer learning approach to other datasets and investigate how it would be affected by data mismatch and the amount of data in source and target domains.

7. ACKNOWLEDGEMENTS

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128 and IARPA Contract No 2012-12050800010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, IARPA, DoD/ARL or the U.S. Government.

The authors thank David Snyder and Pegah Ghahrehmani for initial Kaldi recipes for diarization and transfer learning respectively.

8. REFERENCES

[1] Ahmed Ali, Stephen Vogel, and Steve Renals, "Speech Recognition Challenge in the Wild: Arabic MGB-3," in *Submitted to Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on*, 2017.

[2] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The mgb-2 challenge: Arabic multi-dialect broadcast media recognition," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 279–284.

[3] A. Ali, W. Magdy, P. Bell, and S. Renais, "Multi-reference WER for evaluating ASR for languages with no orthographic rules," in *Proc. Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, 2015, pp. 576–580.

[4] 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 *Proc. Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, 2011, number EPFL-CONF-192584.

[5] Gaofeng Cheng, Vijayaditya Peddinti, Daniel Povey, Vimal Manohar, Sanjeev Khudanpur, and Yonghong Yan, "An exploration of dropout with LSTMs," in *Proc. INTERSPEECH*, 2017.

[6] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI," in *Proc. INTERSPEECH*, 2016, pp. 2751–2755.

[7] Karel Veselỳ, Arnab Ghoshal, Lukás Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks.," in *Interspeech*, 2013, pp. 2345–2349.

[8] Pegah Ghahrehmani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "Investigation of Transfer Learning for LF-MMI Trained Neural Networks for ASR," in

Submitted to Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on, 2017.

[9] Lidia Mangu, Eric Brill, and Andreas Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[10] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802 – 828, 2011.

[11] Martin Karafiat, Lukas Burget, Pavel Matejka, Ondrej Glembek, and Jan Cernocky, "iVector-based discriminative adaptation for automatic speech recognition," in *Proc. Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. Dec. 2011, pp. 152–157, IEEE.

[12] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015, pp. 3214–3218.

[13] 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.

[14] Sergey Ioffe, *Probabilistic Linear Discriminant Analysis*, pp. 531–542, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[15] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 413–417.

[16] Sylvain Meignier and Teva Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010, vol. 2010.

[17] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 5206–5210.

[19] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, 2011, vol. 2011, pp. 249–252.

[20] David Graff, "An overview of broadcast news corpora," *Speech Communication*, vol. 37, no. 1, pp. 15–26, 2002.

[21] Matthew A Siegler, Uday Jain, Bhiksha Raj, and Richard M Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA speech recognition workshop*, 1997, vol. 1997.

[22] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "ArabicASRChallenge2016," `https://github.com/qcri/ArabicASRChallenge2016/commit/090897c27a57be3d39d9abb4f93fd135f329eb67`, 2016.

[23] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[24] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *1995 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1995, vol. 1, pp. 181–184.

[25] Stanley F Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. Association for Computational Linguistics, 34th annual meeting on*, 1996, pp. 310–318.

[26] Temple F Smith and Michael S Waterman, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[27] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, Nov 2016.