Smit, Peter; Gangireddy, Siva; Enarvi, Seppo; Virpioja, Sami; Kurimo, Mikko

Aalto system for the 2017 Arabic multi-genre broadcast challenge

Published in:
Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop on

DOI:
10.1109/ASRU.2017.8268955

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Smit, P., Gangireddy, S., Enarvi, S., Virpioja, S., & Kurimo, M. (2018). Aalto system for the 2017 Arabic multi-genre broadcast challenge. In *Automatic Speech Recognition and Understanding (ASRU), IEEE Workshop on* (pp. 338-345). IEEE. https://doi.org/10.1109/ASRU.2017.8268955

AALTO SYSTEM FOR THE 2017 ARABIC MULTI-GENRE BROADCAST CHALLENGE

Peter Smit, Siva Reddy Gangireddy, Seppo Enarvi, Sami Virpioja, Mikko Kurimo

Department of Signal Processing and Acoustics, Aalto University, Finland

ABSTRACT

We describe the speech recognition systems we have created for MGB-3, the 3rd Multi Genre Broadcast challenge, which this year consisted of a task of building a system for transcribing Egyptian Dialect Arabic speech, using a big audio corpus of primarily Modern Standard Arabic speech and only a small amount (5 hours) of Egyptian adaptation data. Our system, which was a combination of different acoustic models, language models and lexical units, achieved a Multi-Reference Word Error Rate of 29.25%, which was the lowest in the competition. Also on the old MGB-2 task, which was run again to indicate progress, we achieved the lowest error rate: 13.2%.

The result is a combination of the application of state-of-the-art speech recognition methods such as simple dialect adaptation for a Time-Delay Neural Network (TDNN) acoustic model (-27% errors compared to the baseline), Recurrent Neural Network Language Model (RNNLM) rescoring (an additional -5%), and system combination with Minimum Bayes Risk (MBR) decoding (yet another -10%). We also explored the use of morph and character language models, which was particularly beneficial in providing a rich pool of systems for the MBR decoding.

***Index Terms*—** speech recognition, dialect adaptation, subwords, neural network language models, system combination

1. INTRODUCTION

The Arabic track of the 3rd Multi-Genre Broadcast (MGB-3) challenge [1] consists of both a speech-to-text transcription task of Egyptian Dialect Speech and an Arabic Dialect Identification task.

Our team at Aalto University has recently got good results in subword language models in Finnish and Estonian ASR [2]. Even though we have no Arabic speakers and no previous experience in Arabic, we decided to take the challenge and benchmark our methods in Arabic ASR. We participated in the speech-to-text transcription task to recognize Arabic Dialect Speech (Egyptian) using only those language and audio resources provided by the challenge organizers. In total 1200 hours of Arabic transcribed broadcast data, which was not identified by dialect, is provided for the training of acoustic models. For dialect adaptation 5 hours of transcribed data is provided. For language modeling only a single corpus of 110M words is provided which is sourced from the Aljazeera Arabic website. In addition to this year's task of recognizing dialectal speech, also the MGB-2 task [3] is re-run for progress evaluation. The Aalto submission was jointly optimized on both of these two tasks.

The following sections describe both our baseline system for MGB-3 and MGB-2, as well as all individual improvements done. As the MGB-3 task is to recognize Egyptian, which does not have a canonical transcription method, multiple transcriptions per utterance were provided. Scoring is done using Multi-Reference Word Error Rate (MRWER) [4]. The MGB-2 task did have a canonical transcription and is hence using the normal Word Error Rate (WER) for reporting. All speech recognition results in this paper, except for Section 7, report error rates for the development sets of MGB-3 and MGB-2.

2. BASELINE SYSTEM

As a baseline to improve upon we first developed a system similar to the baseline published by the challenge organizers. Using the Kaldi toolkit [5] we build an acoustic model which is a purely sequence trained Time-Delay Neural Network (TDNN) [6, 7] with i-vector based speaker adaptation [8]. This is the same as done by the challenge organizers, and similar to the published systems of the MGB-2 challenge [9, 10, 11, 12], except for the Deep Neural Network (DNN) architecture as explained below.

The baseline development was started by training first a monophone Gaussian Mixture Model (GMM) from the 10,000 shortest utterances in the corpus that were marked to have the highest confidence of having a correct transcription (Word Match Error Rate, WMER = 0). After that, three tri-phone models were built in succession, first a regular GMM model, then a GMM model on top of features transformed with Linear Discriminant Analysis (LDA) and at last a Speaker Adaptive Trained (SAT) GMM model [13]. All models were trained with the complete set of utterances that had WMER = 0. For each model the alignments generated by the previous model

This work was financially supported by the Academy of Finland under the grant number 251170 and TELLme-project in Tekes Challenge Finland programme. Computational resources were provided by the Aalto Science-IT project.

were used for initialization.

After training a regular GMM model, the Kaldi ‘clean and segment’ scripts were used to automatically fix and clean the transcriptions of the full training dataset, including the utterances that had lower confidences in the training data. This resulted in 1,022 hours of data of which 875 hours were marked as speech. Using the ‘cleaned’ training data, a new SAT model was trained which was used for providing alignments for i-vector and DNN training.

The training of the TDNN models included volume perturbation and three-way speed perturbation of the training data [14] and the training of an LDA-based i-vector extractor. The i-vectors were extracted for at most two utterances at a time to provide training variability. Because the task did not include speaker information, in decoding the i-vectors were extracted per utterance.

From the final GMM model, alignment-lattices (which contain multiple alignments per utterance) were generated. Together with the i-vectors, the alignments and high-dimensional MFCC features were joined in neural network training examples with the amount of context applicable for the used network.

For the training of the TDNN’s the lattice-free MMI criterion was used which does forward-backward decoding using a phone language model to calculate the criterion and use it as a loss function. For our baseline system we used a regular TDNN with nine layers and ten million parameters. [6, 7]

For the language model we trained an n-gram model on the provided text corpus with the VariKN toolkit [15]. This toolkit is specifically made for modeling higher order n-grams which is useful in the experiments done in the following sections. No limit was set on the maximum n-gram order, but the pruning parameters were tuned to result in a model with approximately eight million n-gram contexts. In the resulting model more than seven million of the contexts were of order three or lower. The MGB-organizers’ baseline language models were trigrams trained with SRILM toolkit, thus probably roughly the same size, but pruned differently.

We experimented with both a grapheme-based lexicon and the phoneme-based lexicon. In our experiments the grapheme-based lexicon was superior and better suited to the planned experiments, so only grapheme-based results (where the acoustic modeling units are the graphemes in the surface form of the words instead of phonological units) are reported.

The results for both the Aalto and MGB-organizers’ baseline are shown in Table 1. Aalto’s baseline outperforms the organizers’ one, most likely because of different choices made in the training of the acoustic model (e.g. different amount of parameters / layers) and the language model.

3. ACOUSTIC MODELING IMPROVEMENTS

Because the recurrent architectures that capture longer term dependencies have shown remarkable improvements for the DNN acoustic models [16], we decided to first experiment with different DNN architectures. Both a unidirectional Long Short-Term Memory (LSTM) and Bidirectional LSTM (BLSTM) in combination with regular TDNN layers were trained. The TDNN-LSTM model had 3 recurrent layers interleaved with 7 regular TDNN layers, looking back in total 50 frames (~1.5 seconds), predicting the label with a delay of 5 frames. In total the TDNN-LSTM architecture had 37 million parameters.

Table 1. Results for Aalto’s baseline compared to the results published in [1].

System	MGB-3	MGB-2
Baseline (Aalto)	51.66	21.64
Baseline (MGB)	58.0	22.6

For the BLSTM architecture three forward and three backward layers were used that were preceded by three regular TDNN layers. Both forward and backward it utilized 45 frames (~1.4 seconds) and it had in total 48M parameters.

The number of layers and parameters for the acoustic model were not very carefully tuned, most were kept to their defaults or taken from successful Kaldi recipes. During training, a new recognition pass on the development set was done in every 100 iterations and the training was stopped after the WER on the MGB-2 development set stopped improving. For the BLSTM model this occurred at iteration 1300, which corresponds to two completed epochs.

3.1. Adaptation

Up to this point the available six hours of MGB-3 Egyptian dialect data had not been utilized at all. To utilize this data and adapt the models to the Egyptian dialect data a simple adaptation scheme was used.

First, the data was aligned with the same GMM model that was used for the alignment of the TDNN training data. Because all utterances had four transcriptions, all variants were handled as their own utterance. This meant that the audio data was replicated four times.

The resulting data was also speed and volume perturbated and the same i-vector extractor was used to generate accompanying i-vectors. After that, TDNN examples were generated in the same manner and with the same parameters as done for the original data.

To adapt the model, the training was continued from the earlier best iteration with the same parameters as in the last iteration of the normal training, but only using the Egyptian adaptation data. Different learning rates were experimented with and it was concluded that keeping the learning rate constant at the same value as in the last normal training iteration gave the best results. After every iteration, the development set was recognized and the training was stopped after the MR-WER on the MGB-3 development set stopped improving. For

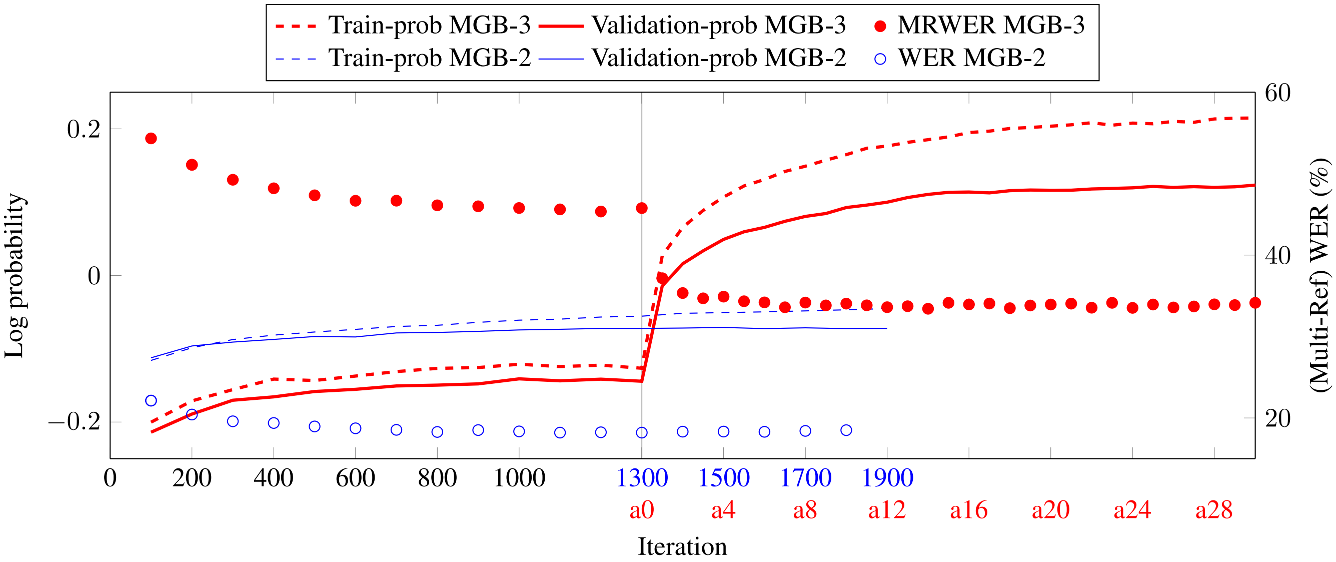

Fig. 1. Log probabilities and development word error rates in different stages of training. The single markers are error rates, lines are log-probabilities. Red is MGB-3, blue is MGB-2. The iterations prefixed with 'a' are adaptation iterations.

Table 2. Results for acoustic modeling improvements.

Acoustic model	**MGB-3**	**MGB-2**
Baseline (TDNN)	51.66	21.64
TDNN-LSTM	46.88	19.43
TDNN-BLSTM	45.77	18.20
ADAP-TDNN-LSTM	33.94	–
ADAP-TDNN-BLSTM	33.52	–

the adaptation of the BLSTM model this occurred after 24 iterations, which corresponds to 1.5 epochs. Note that one epoch already had 12 variants of each audio sample, the product of four transcription variants and three speed perturbation variants.

Besides adapting the complete model, we also tried just adapting one or multiple layers. This however, did not have as strong effect as the adaptation of the complete model.

Our methods are similar to the ones used for speaker adaptation for DNN's as done in [17, 18]. However, our method was simpler as no extra regularization is used. Perhaps with a better regularization method the model could be adapted even more accurately.

3.2. Analyses

Figure 1 presents the evolution of both the log-probability of the training and validation sets during training, as well as the evolution of the WER. It clearly shows that the adaptation has a great impact on both the log-probabilities and the MRWER of the MGB-3 dataset.

Table 2 summarizes improvement obtained with the acoustic modeling improvements. The use of a TDNN-BLSTM architecture over a normal TDNN improved the MGB-2 error rates with 16% relative. On the MGB-3 dataset a 35% relative improvement was obtained by using a TDNN-BLSTM architecture and adapting the final model with a small amount of Egyptian dialect adaptation data. All further development experiments use ADAP-TDNN-BLSTM for MGB-3 and TDNN-BLSTM for MGB-2.

4. IN-DOMAIN N-GRAM LANGUAGE MODELS

Our baseline system does not utilize the in-domain data that is available in the training transcripts for MGB-2 and adaptation transcripts for MGB-3. To use this data, we trained separate language models on the in-domain data and used linear interpolation to create our target language models. For the MGB-3 data we trained the model on the concatenation of all transcription variants.

For both MGB-3 and MGB-2 we created a linear interpolated model of the background data and the applicable in-domain data. The interpolation weight was optimized on the perplexity of the development transcripts. Also the perplexity calculation of MGB-3 was done using the union of all transcription variants in the development set.

Also a smaller variant of mixed language model was created for the first-pass decoding. The size was controlled with the growing and pruning parameters of the VariKN toolkit and the target size for the smaller model was approximately 6-8 million n-gram contexts. The larger n-gram model was used for rescoring.

Perplexities and error rates are shown in Table 3. Because

Table 3. Results for n-gram language modeling improvements. Lexicon size and oov rate are shown in italics.

Language	MGB-3		MGB-2	
model	MRWER	ppl	WER	ppl
	11k / 21.14% oov		*177k / 3.96% oov*	
In-Domain	47.07	968	19.42	2.053
	1.3M / 3.99% oov		*1.3M / 1.68% oov*	
Background	33.52	12,643	18.20	3,633
Interpolated	33.50	3,265	17.66	2,129
Rescored	32.57	2,834	17.25	1,883

of the small size of the in-domain data there are no improvements in only using that data. Interpolating however does have a positive effect, as does rescoring with a bigger language model.

Different interpolation weights were also tested for the different units as described in Section 5, but the same interpolation weights were optimal for all units so a fixed interpolation weight was kept also for the other language modeling units.

5. LANGUAGE MODELING UNITS

5.1. Subword modeling in ASR

Although the words might seem the most logical basic unit for lexicon modeling, it can be beneficial to consider different lexical units such as subwords or even characters. Previous Arabic ASR systems have utilized grammatical or statistical morphemes as units with varying success [19, 20, 21, 22, 23].

In agglutinative languages like Finnish and Estonian subword units, such as morphs, are typically better than word-based models, because the subword units cover almost unlimited vocabulary of the language. Although Arabic is not an agglutinative language, it has a general structure that is suitable for surface form segmentation into subwords.

In our recent work [2] we developed efficient modeling of subword units in the weighted finite state transducer framework as is used in Kaldi. We continue similar experiments here with different subword units and even character units. In [24] there is a more detailed analysis on the use of character models.

We trained subword models with four different vocabulary sizes using Morfessor [25, 26], as well as character models. The vocabulary sizes and parameters used as language modeling units are shown in Table 4. As previous research did not indicate clearly about the ideal marking of subword units for Arabic (in order to be able to reconstruct them), we trained all models in 4 different variations; boundary tag (<w>), left-marked (+m), right-marked (m+) and left+right-marked (+m+) [2].

For all models we trained variable order n-grams with the VariKN toolkit [15]. This toolkit trains a language model for variable order n-grams by growing and pruning. As the toolkit is geared towards subword models, it allows for including arbitrary long contexts which are useful in both subword and character models to span over the same original context. Naturally, the smaller the vocabulary, the higher order the resulting n-gram models are.

Table 4. Various language modeling units used in this work.

Unit	Size	Parameters
word	1.5M	Words
sub-71k	71.2k	Morfessor, $\alpha = 0.05$, tokens
sub-51k	50.7k	Morfessor, $\alpha = 1$, types
sub-17k	17.2k	Morfessor, $\alpha = 0.005$, tokens
sub-1k	1.2k	Morfessor, $\alpha = 0.05$, types
char	39	Characters

5.2. Results

Table 5 shows rescored n-gram results for all subword segmentations and boundary marking styles. For MGB-3 there is no direct improvement in using smaller units than words, although most results are close. In general the <w> style marking seems to perform best.

For MGB-2 there is a clear benefit in using subword models, and even one of the character models outperforms the word model. The optimal marker is slightly different, with 'm+' performing best for the larger vocabularies and <w> for the smaller ones.

6. RNNLM

Most recently RNNs and their variants (LSTMs) have shown significant improvements over traditional n-grams [27, 28, 29, 30, 31]. In addition to the n-grams, in this work we also explored RNN based language models (RNNLMs) for lattice rescoring in the second pass of recognition. We used ThenanoLM toolkit [32] to train and test the RNNLMs. In this work we trained long short-term memory (LSTM) based LMs on characters, subwords (statistical morphs) and words. We used the same LSTM architecture for both character and subword models. A different architecture was used for models based on words. Our LSTM architecture consists of a projection layer, hidden layer, highway layer and an output layer. The projection layer projects the words into a continuous space. The output layer is a softmax layer which computes the probability of the current word given its context. The hidden and highway layers use the *tanh* activation function. From our experiments we found out that a highway layer, followed by a LSTM hidden layer performs better than an architecture which has multiple LSTM layers. The size of the projection layer in character and subword models is 200 and in word models 300. The hidden and highway layers of character and

Table 5. Speech recognition results for different lexical units and unit markers.

Unit		MGB-3	MGB-2
word		**32.57**	17.25
sub-71k	+m+	32.99	17.09
	+m	32.97	17.27
	m+	33.09	16.94
	<w>	32.77	17.15
sub-51k	+m+	33.42	17.13
	+m	33.45	17.30
	m+	33.34	16.93
	<w>	32.86	17.07
sub-17k	+m+	33.16	16.93
	+m	33.11	17.16
	m+	32.75	**16.79**
	<w>	32.76	17.01
sub-1k	+m+	33.26	17.33
	+m	33.54	17.47
	m+	33.30	17.32
	<w>	32.81	17.13
char	+m+	33.65	17.38
	+m	34.38	17.61
	m+	33.92	17.59
	<w>	33.68	17.22

subword models consist of 1000 neurons, whereas the hidden and highway layers of word models consist of 1500 neurons.

The size of the output layer of character, subword and word models is dependent on the size of the vocabulary. Since the vocabulary size is huge for word and subword models we use classes to reduce the computational complexity in the output layer. The words and subwords were grouped into classes using the exchange word clustering algorithm [33, 34]. Since the words and subwords were grouped into classes the probability of the current word is equal to the product of the word probability given its class and the word's class probability given the context. For all the word and subword experiments, except `sub-1k`, we used 2000 classes in the output layer. The character and `sub-1k` experiments did not use classes.

All LSTM models were trained by back-propagation using the adaptive gradient (Adagrad) algorithm. Adagrad is a modified version of stochastic gradient decent (SGD) with a per-parameter learning rate. The parameters of the model were updated after processing a mini-batch of training examples. We used a mini-batch size 64 for character and subword models, and 32 for word models. In each mini-batch we used a sequence length of 100 for character models, 50 for subword models and 25 for word models. For all the experiments we used an initial learning rate of 0.1. During training we used a dropout of 0.2 for the hidden layer. We trained the models for a maximum of 15 iterations. Since the maximum time limit of a single job on our computing cluster is 15 days, we trained the models for a maximum of 15 days. Except few, all the models reported took less than 15 days to converge. Training times of each model are given in Table 6.

For training the language models MGB-3 challenge provided us a cleaned and normalized dataset of 131M tokens (includes sentence start and end markers) created from data crawled from Aljazeera.net. In addition, 8.3M tokens of in-domain data was available for MGB-2 (MGB-2 training data transcripts) and 170K tokens of Egyptian data for MGB-3 (MGB-3 adaptation data transcripts). We tried to train LSTMs on the in-domain data as well, but we did not succeed in obtaining any gains over the LSTMs trained on the background data.

6.1. Results

In Table 6 we report the WERs of MGB-3 and MGB-2 development sets using LSTM word, subword and character LMs. First we report the error rates of rescoring the lattices with the LSTM LMs (**pure**). Since the n-grams and LSTM LMs are complimentary to each other we also report the WERs after interpolating the lattices of the n-gram and LSTM LMs (**int.**). From the Table 6 we observe that on the average, the word, subword and character LMs improve the n-gram baselines by 3.9% and 5.4% relative, on MGB-3 and MGB-2 development sets respectively. For both MGB-3 and MGB-2 the best segmentations are subword models. For MGB-2 the character model also outperforms the word model, for MGB-3 this is not the case. Overall, the differences between the word and character models are very small, leading us to believe that the character models are on-par with the word models. When we compare the different subword markers we see the same pattern as we saw for MGB-2 in Section 5.2, the m+ markings seem to perform best for the larger vocabularies and the <w> marker for the smaller ones.

7. FINAL SYSTEMS

For our final system we have used results from multiple acoustic models and language models with different units. To take advantage of the information contained in the lattices of different systems we used MBR as a system combination technique [35]. Lattices from different systems were combined and MBR-decoded to give the final transcriptions.

Normally, lattices with different units are not combinable. Therefore, a finite state transducer (FST) is created that maps all subword sequences in a lattice to words. When care is taken that the word vocabularies are the same, the systems can be combined in a regular manner.

As the primary submission for MGB-3 we took the RNN-interpolated systems created for all different language modeling units and both the ADAP-TDNN-BLSTM and ADAP-

Table 6. The WERs of MGB-3 and MGB-2 development sets using LSTM word, subword and character models. We also report the WERs after linear interpolation with the n-gram scores (**int.**). The interpolation coefficient was optimized for the WER on development data.

Unit		**Time**	**MGB-3**		**MGB-2**	
		(hours)	**pure**	**int.**	**pure**	**int.**
word		360	32.34	**31.34**	17.13	**16.33**
sub-71k	+m+	238	32.53	31.81	16.76	16.08
	+m	222	32.57	31.78	16.83	16.20
	m+	216	32.37	31.53	16.51	15.96
	<w>	160	32.96	31.89	17.20	16.34
sub-51k	+m+	360	32.73	31.78	16.85	16.16
	+m	232	32.67	31.93	16.78	16.16
	m+	201	32.10	31.39	16.43	15.90
	<w>	256	32.29	**31.29**	16.60	16.03
sub-17k	+m+	182	33.01	32.03	16.73	16.00
	+m	165	33.05	32.14	16.79	16.21
	m+	170	32.85	31.85	16.56	**15.88**
	<w>	236	32.35	31.33	16.87	16.19
sub-1k	+m+	360	33.09	32.19	16.97	16.43
	+m	266	33.12	32.33	17.09	16.60
	m+	331	33.12	32.31	16.83	16.32
	<w>	355	32.47	31.71	16.56	16.12
char	+m+	157	33.49	32.34	17.46	16.72
	+m	360	33.55	32.66	17.43	16.78
	m+	360	33.65	32.44	17.37	16.67
	<w>	188	32.77	**31.81**	16.93	**16.27**

TDNN-LSTM acoustic models. Unfortunately, at the challenge submission time a small number of subword systems had not finished training and only 36 out of 42 systems were used. For the secondary task, the MGB-2 challenge re-run, we made a similar combination of the successful individual system as we did for MGB-3. For contrastive submissions we took for both MGB-3 and MGB-2 the best performing character model, the best subword model and the word model.

7.1. Final results

Table 7 shows the results on both the development sets as well as the official evaluation results for our selected systems. For MGB-3 the evaluation results are close to the development results. For MGB-2 however, the evaluation results are much better, as a more careful text normalization had been applied by the challenge organizers. Both tasks show the subword models outperform word and character models. The word and character models are close to each other in performance with a slight edge for word models.

Besides MRWER, the organizers also published for the MGB-3 evaluation set the WER for each reference transcription and reported the average. For our primary system this was 37.5%, also the best compared to any of the other participants.

Table 7. Development results and official MGB evaluation results of all Aalto submissions.

System	**MGB-3**		**MGB-2**	
	Dev.	**Eval.**	**Dev.**	**Eval.**
Primary	28.19	**29.25**	14.79	**13.2**
Contr. 1 (char)	31.81	31.32	16.27	14.4
Contr. 2 (sub-17k)	31.33	30.52	15.88	14.0
Contr. 3 (word)	31.34	31.23	16.33	14.3

Table 8. Itemized improvements on dev set (relative to previous step).

Stage	**MGB-3**		**MGB-2**	
	MRWER	**rel.**	**WER**	**rel.**
Baseline (Aalto)	51.66	–	21.64	–
BLSTM	45.77	11.4	18.20	15.9
Adapted BLSTM	33.52	26.7		
n-gram interp.	32.57	2.8	17.25	5.2
Subwords	32.75	-0.06	16.76	2.8
RNN interp.	31.29	4.5	15.88	5.3
MBR-decoding	28.19	9.9	14.79	6.8

8. CONCLUSIONS

We have successfully build the best performing systems for the 2017 Arabic MGB challenge. We have precisely documented our development process and reported intermediate results on all steps. Table 8 shows a summary of the improvements as well as their relative improvements to the previous steps. The most relative gain was made by improving and adapting the acoustic model, but also the rescoring with LSTM LMs and the combination of systems that used different lexical units contributed to the success of the system.

In future, both the acoustic model adaptation and the usage of in-domain data in LSTM LM rescoring could be improved. Also, we plan to further optimize and explore different LSTM LM architectures for character and subword models.

9. ACKNOWLEDGMENTS

We would like to thank the authors of the excellent toolkits we have used in building this system, specifically the authors of the Kaldi Toolkit and the current and former members of Aalto University's ASR group that have authored the VariKN, Morfessor, and TheanoLM toolkit. We also thank the organizers of the challenge for their effort.

10. REFERENCES

[1] Ahmed Ali, Stephan Vogel, and Steve Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *ASRU 2017 – IEEE Workshop on Automatic Speech Recognition & Understanding*, December 2017.

[2] Peter Smit, Sami Virpioja, and Mikko Kurimo, "Improved subword modeling for WFST-based speech recognition," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017.

[3] Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 279–284.

[4] Ahmed Ali, Walid Magdy, Peter Bell, and Steve Renals, "Multi-reference WER for evaluating ASR for languages with no orthographic rules," in *ASRU 2015 – IEEE Workshop on Automatic Speech Recognition & Understanding*, 2015, pp. 576–580.

[5] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *ASRU 2011 – IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011.

[6] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015, pp. 3214–3218.

[7] 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 *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, San Francisco, September 2016, pp. 2751–2755.

[8] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU 2013 – IEEE Workshop on Automatic Speech Recognition & Understanding*, 2013, pp. 55–59.

[9] Tuka AlHanai, Wei-Ning Hsu, and James Glass, "Development of the MIT ASR system for the 2016 Arabic Multi-genre Broadcast Challenge," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 299–304.

[10] Sameer Khurana and Ahmed Ali, "QCRI advanced transcription system (QATS) for the Arabic Multi-Dialect Broadcast media recognition: MGB-2 challenge," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 292–298.

[11] Natalia Tomashenko, Kévin Vythelingum, Anthony Rousseau, and Yannick Estève, "LIUM ASR systems for the 2016 Multi-Genre Broadcast Arabic challenge," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 285–291.

[12] Xu-Kui Yang, Dan Qu, Wen-Lin Zhang, and Wei-Qiang Zhang, "The NDSC transcription system for the 2016 multi-genre broadcast challenge," in *SLT 2016 – IEEE Spoken Language Technology Workshop*, December 2016, pp. 273–278.

[13] Tasos Anastasakos, John McDonough, Richard Schwartz, and John Makhoul, "A compact model for speaker-adaptive training," in *ICSLP 1996 – Fourth International Conference on Spoken Language*, Oct 1996, vol. 2, pp. 1137–1140 vol.2.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015, pp. 3586–3589.

[15] Vesa Siivola, Teemu Hirsimäki, and Sami Virpioja, "On growing and pruning Kneser-Ney smoothed n-gram models," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 5, pp. 1617–1624, 2007.

[16] Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP 2013 – IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6645–6649.

[17] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *SLT 2012 – IEEE Spoken Language Technology Workshop*, December 2012, pp. 366–369.

[18] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP 2013 – IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7893–7897.

[19] Ghinwa Choueiter, Daniel Povey, Stanley F. Chen, and Geoffrey Zweig, "Morpheme-based language modeling for Arabic LVCSR," in *ICASSP 2006 – IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, pp. 1053–1056.

[20] Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 1, pp. 3:1–3:29, Dec. 2007.

[21] Katrin Kirchhoff, Dimitra Vergyri, Jeff Bilmes, Kevin Duh, and Andreas Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Computer Speech & Language*, vol. 20, no. 4, pp. 589 – 608, 2006.

[22] Ruhi Sarikaya, Mohamed Afify, and Yuqing Gao, "Joint Morphological-Lexical Language Modeling (JMLLM) for Arabic," in *ICASSP 2007 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, vol. 4, pp. 181–184.

[23] Amr El-Desoky Mousa, Hong-Kwang Jeff Kuo, Lidia Mangu, and Hagen Soltau, "Morpheme-based feature-rich language models using deep neural networks for lvcsr of egyptian arabic," in *ICASSP 2013 – IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8435–8439.

[24] Peter Smit, Siva Reddy Gangireddy, Seppo Enarvi, Sami Virpioja, and Mikko Kurimo, "Character-based units for unlimited vocabulary continuous speech recognition," in *ASRU 2017 – IEEE Workshop on Automatic Speech Recognition & Understanding*, December 2017.

[25] Mathias Creutz and Krista Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL 2002 Workshop on Morphological and Phonological Learning*, Stroudsburg, PA, USA, 2002, vol. 6 of *MPL '02*, pp. 21–30, Association for Computational Linguistics.

[26] Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo, "Morfessor 2.0: Python implementation and extensions for Morfessor Baseline," Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University, Helsinki, Finland, 2013.

[27] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[28] Tomas Mikolov, "Statistical language models based on neural networks," *PhD Thesis*, 2012.

[29] Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Extensions of recurrent neural network language model," in *ICASSP 2011 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 5528–5531.

[30] Tomas Mikolov, Martin Karafit, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010 – 11th Annual Conference of the International Speech Communication Association*, Makuhari, Japan, September 2010, pp. 1045–1048.

[31] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney, "LSTM neural networks for language modeling," in *INTERSPEECH 2012 – 13th Annual Conference of the International Speech Communication Association*, Portland, OR, USA, September 2012, pp. 194–197.

[32] Seppo Enarvi and Mikko Kurimo, "TheanoLM an extensible toolkit for neural network language modeling," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, San Francisco, September 2016, pp. 3052–3056.

[33] Sven Martin, Jörg Liermann, and Hermann Ney, "Algorithms for bigram and trigram word clustering," *Speech communication*, vol. 24, no. 1, pp. 19–37, 1998.

[34] Rami Botros, Kazuki Irie, Martin Sundermeyer, and Hermann Ney, "On efficient training of word classes and their application to recurrent neural network language models.," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, Dresden, Germany, September 2015, pp. 1443–1447.

[35] Haihua Xu, Daniel Povey, Lidia Mangu, and Jie Zhu, "An improved consensus-like method for Minimum Bayes Risk decoding and lattice combination," in *ICASSP 2010 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4938–4941.