LIUM ASR SYSTEMS FOR THE 2016 MULTI-GENRE BROADCAST ARABIC CHALLENGE

Natalia Tomashenko[1,2], *Kévin Vythelingum*[3,1], *Anthony Rousseau*[1], *Yannick Estève*[1]

[1] LIUM, University of Le Mans, France
[2] ITMO University, Saint-Petersburg, Russia
[3] Voxygen, Plemeur-Bodou, France

firstname.lastname@univ-lemans.fr

ABSTRACT

This paper describes the automatic speech recognition (ASR) systems developed by LIUM in the framework of the 2016 Multi-Genre Broadcast (MGB-2) Challenge in Arabic language. LIUM participated in the first of the two proposed tasks, namely the speech-to-text transcription of Aljazeera recordings. We present the approaches and specificities found in our systems, as well as our results in the evaluation campaign: the primary LIUM ASR system reached the second position. The main specificities come from the use of GMM-derived (GMMD) features for training a DNN, combined with the use of time-delay neural networks for acoustic models, the use of two different approaches in order to automatically phonetize Arabic words, and finally, the training data selection strategy for acoustic and language models.

Index Terms— deep neural networks (DNN), time-delay neural networks (TDNN), automatic speech recognition (ASR), Arabic ASR, confusion networks combination, broadcast transcription

1. INTRODUCTION

The 2016 Multi-Genre Broadcast (MGB) Challenge is a controlled evaluation of speech recognition and lightly supervised alignment using Aljazeera Arabic TV channel recordings over a span of 10 years [1]. LIUM participated in the speech recognition task of the challenge. In this evaluation campaign, data is restricted to what the organizers provide [2], both for acoustic and language modeling. For acoustic modeling LIUM developed four different systems, while the same language models set was used for every system. Different choices were made for each acoustic model, with the expectation of producing complementary models for the systems' output merging step. Every system uses the Kaldi ASR toolkit [3], and several experimental models and features were evaluated.

The remainder of this paper is structured as follows: in Section 2.2, we describe the data used for the transcription task and how the selection and alignment were performed. In Section 3, we describe the developed systems by detailing the acoustic and language modeling, along with the system merging strategy. Then in Section 4, we present both the official and new results from the challenge, before concluding in Section 5.

2. DATA SELECTION AND ALIGNMENT FOR ACOUSTIC MODELING

Since the MGB Challenge is an evaluation campaign fully restricted to the data provided by the organizers, in this section we will describe this data both for acoustic and language modeling as well as the strategy used for data selection and alignment.

2.1. Provided data

2.1.1. Acoustic modeling data

The acoustic modeling data provided by the organizers consists of approximately 1128 hours of Arabic broadcast speech, obtained from more than 2 000 broadcast shows on Aljazeera Arabic TV channel over a span of 10 years, from 2005 until 2015. The corresponding time-aligned transcriptions are also provided, but these transcriptions need an alignment process since they are an output from a lightly supervised alignment based on Aljazeera closed-captions, with varying quality of manual transcription. Table 1 presents some statistics for the original training, development and test sets.

Set	Shows	Duration, h	Utterances	Words
TRAIN	2 214	1 128.0	376 011	7 815 566
DEV1	16	8.5	4 940	57 647
DEV2	17	10.0	5 842	74 580
TEST	17	10.1	N/A	N/A

Table 1. Statistics for MGB-2016 training, development and test sets.

2.1.2. Language modeling data

The language modeling data provided, according to the organizers, consists of more than 110 million words from the Aljazeera.net website, collected between 2004 and 2011. Indeed, the provided normalized and transliterated language modeling data contains exactly 4 717 873 sentences for 121 277 408 words. The automatic transcriptions of Arabic broadcast speech were also available for LM training.

2.2. Data alignment and selection

For aligning and selecting data for training the final acoustic models (AM), we first trained a first LIUM boostrap ASR system on all the available data with a SAT GMM HMM model, which was then used as a basis to train a DNN HMM model. The Kaldi speech recognition toolkit [3] was used for this purpose. This system is based on perceptual linear prediction (PLP) features, with feature space maximum likelihood linear regression (fMLLR) speaker adaptation. The DNN system was trained using the frame-level cross entropy criterion and the senone alignment generated from the baseline SAT GMM system. For training this DNN, 40-dimensional PLP features were spliced across 10 neighboring frames, resulting in 440-dimensional (40×11) feature vectors. The DNN model was trained with the following topology: a 440-dimensional input layer; six 2048-dimensional hidden layers (HLs), and a 8827-dimensional output layer.

Two 2-gram and 4-gram language models (LMs) were trained using all the provided data, with the KENLM toolkit [4]. They were integrated in all the ASR systems used during the data selection process presented in this section. These LMs were only used for this task. The LIUM boostrap ASR system reached a word error rate (WER) of 27.9% by decoding with this 2-gram LM, followed by a lattice rescoring with the 4-gram LM on DEV1.

Then, using this boostrap system, we decoded with 2-gram LM every segment from the provided training data and compared the decoding results with the baseline original alignment from the organizers. Based on this comparison, we kept only those segments, for which all the words had an exact match (i.e. had the same sequence of words), and for which the word-level time-stamps from decoding segments were entirely located inside the corresponding original segments. These segments represent a total of 325.5 hours of speech for 150 201 utterances.

After that, based on these perfect segments, a second DNN with the same topology, was trained. The second DNN shows a small improvement (27.8% WER) in comparison with the first one.

Once again, we decoded all our unselected train set utterances, and this time instead of taking only the perfectly aligned segments (only 6 109, corresponding to 16.2 additional hours of speech), we also added all the segments containing no more than two word mismatches (insertions, substitutions or deletions), for a total of 34 475 new utterances accounting for 89.7 additional hours of speech). Finally, since we have word-level timings in our decoding results, in order to exploit the remainder of the train set, we aligned sub-parts of these remaining segments, which allowed us to salvage additional 233 hours of speech from the misaligned data. At the end, we extracted more than 648.3 hours of well-aligned speech from the provided data, which accounts for about 67.5% of the original data. Table 2 summarizes the alignment and selection process. WER results in the table are given for rescoring with the 4-gram LM on DEV1 set. The final realigned training set, containing 648.3 hours, was used to train all our AMs.

Step	Duration, h	Utterances	Words	WER %
Boostrap ASR	1 128.0	376 011	7 815 566	27.9
1st pass	325.5	150 201	2 177 989	27.8
2nd pass	648.3	398 438	4 422 123	26.2

Table 2. Statistics from alignment and selection process, and word error rate on DEV1 data set.

3. LIUM SYSTEMS DESCRIPTION

In this section we describe the four different AMs which were trained, as well as the language models which we used. We also expose the two phonetization processes used in our systems, and the system merging strategy for the final combination.

3.1. Phonetization

Our four systems were trained by using two different phonetization types: (1) grapheme to phoneme and (2) real phonetization. All AMs except the TDNN_2 (Section 3.2.2) were trained using the first type of phonetization.

The first type of phonetization is a simple grapheme-based lexicon, *i.e.* a 1-to-1 grapheme to phoneme mapping, which makes no use of vowels in it.

The second one is a real phonetization, where words are represented with vowels, generated by processing the MADAMIRA toolkit [5] on all provided textual data. MADAMIRA generates all the vowelized forms for each word, ordered by confidence. In order to cope with ineluctable diacritization errors from the MADAMIRA processing, we chose to keep the two best hypothesis instead of the best one. Also, we considered as a lexicon entry a triplet *(grapheme word, first vowelation, second vowelation)*: by this way, we aim to reduce the number of phonetization variants related to a grapheme word. Since the vowelization depends on the lexical context, this approach allows us to control this number of phonetization variants per word by taking into account its lexical context. Following this approach, our objective was to reduce the number of errors introduced during the phoneme/audio signal alignment needed for acoustic model training.

Then, we applied the pronunciation rules described in [6] on top of the fully vowelized words to build the training pronunciation dictionary. As a result, the lexicon has 868K vowelized lexicon entries (*i.e.* triplets) and more than 2M pronunciations, with an average of 2.6 pronunciations per lexicon entry. In comparison, the phoneme-based lexicon provided by the organizers, described in [7], has 526K grapheme words and 2M pronunciations, with an average of 3.84 pronunciations for each grapheme word. As seen above, this lower number of pronunciation alternatives for each word in our lexicon aims to prevent the introduction of some unwanted noise in acoustic model training.

3.2. Acoustic modeling

The Kaldi speech recognition toolkit [3] was used for acoustic model training.

3.2.1. TDNN chain model with 1-to-1 word to grapheme lexicon (TDNN_1*)*

The first acoustic model is a newly proposed type of model in the Kaldi toolkit, so-called "chain" model [8], based on a subsampled time-delay neural network (TDNN) [9]. This kind of model is trained with a sequence-level objective function (the log probability of the correct phone sequence). It can be viewed as training with the maximum mutual information (MMI) criterion, implemented without lattices, by doing a full forward-backward on a decoding graph derived from a phone n-gram language model and using a three times smaller frame rate at the output of the neural network.

Training this model was done by using high-resolution PLP features (without dimensionality reduction, keeping the 40 cepstra) concatenated with 100-dimensional i-vectors for speaker adaptation, accounting for an input dimension of 140. We also, as in a standard Kaldi recipe, applied data augmentation techniques before performing the actual training, namely time-warping of the raw audio by factors of 0.9, 1.0 and 1.1, as well as frame-shifting by 0, 1 and 2 frames. Our resulting network contains around 13.6 million parameters, uses left and right contexts of 17 and 12 respectively, has an output dimension of 576 and rectified linear units (ReLU) [10] as activation functions. On top of this network, we also performed a sequence-discriminative training, using the state-level minimum Bayes risk (sMBR) criterion [11].

3.2.2. TDNN chain model with vowels and phonetization (TDNN_2*)*

This acoustic model is a TDNN chain model built in the same way as the one described above, to the exception that no sequence-discriminative training was performed. The model was trained on realigned vowelized training set, where words were replaced by the two best diacritization candidates provided by MADAMIRA. For example, the word *ktb* could be vowelized as *kataba/kutiba* in one context of the training set, and *kataba/kutubN* in another context. Thus, we replaced *ktb* in the text by $(ktb, kataba, kutiba)$ and $(ktb, kataba, kutubN)$ respectively. Thereby, we were able to use for training our phoneme-based lexicon described before, while pronunciations are mapped to the grapheme words in the decoding process.

3.2.3. DNN on fMLLR speaker-adapted BN features ($\mathrm{DNN}_{\mathrm{BN}}$*)*

For training the third DNN model, first an auxiliary DNN was trained for bottle-neck (BN) feature extraction. State tying for training this auxiliary DNN was taken the same as for the final DNN that was used in the data selection procedure (Section 2.2). The senone alignment for training the auxiliary DNN was also obtained by the same DNN from Section 2.2. The auxiliary DNN was trained using the frame-level cross entropy criterion. For training this DNN, 40-dimensional log-scale filterbank features concatenated with 3-dimensional pitch-features, were spliced across 11 neighboring frames, resulting in 473-dimensional (43×11) feature vectors. After that a discrete cosine transform (DCT) transform was applied and the dimension was reduced to 258. The DNN model for extraction 40-dimensional BN features was trained with the following topology: a 258-dimensional input layer; four hidden layers (HL), where the third HL was a BN layer with 40 neurons and other three HLs were 1500-dimensional; the output layer was 8827-dimensional.

On the obtained BN features we trained the GMM model, which is used to produce forced alignment, and then SAT-GMM model was trained on fMLLR-adapted BN features. For training the final DNN model, fMLLR-adapted BN features were spliced in time with the context of 13 frames: [-10,-5...5,10]. The final DNN had a 520-dimensional input layer; six 2048-dimensional HLs with logistic sigmoid activation function, and a 8467-dimensional softmax output layer, with units corresponding to the context-dependent states. The DNN parameters were initialized with stacked restricted Boltzmann machines (RBMs) by using layer by layer generative pretraining. The DNN was trained with an initial learning rate of 0.008 using the cross-entropy objective function. After that four epochs of sequence-discriminative training with per-utterance updates, optimizing sMBR criteria, were performed.

3.2.4. DNN on MAP speaker-adapted GMMD features ($\mathrm{DNN}_{\mathrm{G}}$*)*

For training this DNN acoustic model we used speaker-adapted GMM-derived (GMMD) features, proposed in [12]. GMMD-derived features were extracted using an auxiliary speaker-adapted monophone GMM. For a given acoustic feature vector, a new GMM-derived feature vector was obtained by calculating log-likelihoods across all the states of the auxiliary GMM on the given vector. Speaker adaptation of the DNN model was performed by maximum a posterior adaptation (MAP) of the auxiliary GMM. Basic features for the auxiliary GMM model were BN-features from Section 3.2.3.

The MAP-adapted GMMD 130-dimensional feature vectors were concatenated with unadapted 40-dimensional BN feature vectors (from Section 3.2.3), and spliced across 13 frames as before [-10,-5...5,10], resulting in 2210-dimensional feature vectors, for training a new DNN model, in a similar way as described in [13]. For $\mathrm{DNN_G}$ model the same alignment and state tying as for the model $\mathrm{DNN_{BN}}$ was used. Topology of $\mathrm{DNN_G}$ is similar to $\mathrm{DNN_{BN}}$, except for the dimension of the input layer. Further the DNN was trained with an initial in a similar way as described in Section 3.2.3 for $\mathrm{DNN_{BN}}$ model.

In the experimental results, for adaptation on DEV1 set, we used the best transcripts, obtained from other AMs.

3.3. Language modeling

All the single ASR systems presented in this paper used the same 2-gram and 4-gram LMs, except for the data selection process (*cf.* Section 2.2). These LMs were trained on the official data (*cf.* Section 2.1.2). Before, numbers written in digits were converted in letters and all words containing invalid Buckwalter characters were omitted.

In order to build the vocabulary, we trained two different 1-gram language models: one trained on the Aljazeera.net data, and the other one on the automatic transcriptions of Arabic broadcast speech which were used to train our acoustic models: this means that we did not use all the available automatic transcriptions, but only the ones filtered for AM training. A composite 1-gram model was computed through a linear interpolation of these two single 1-gram LMs. The weights for this interpolation were optimized on the transcriptions of the DEV1 corpus to minimize the perplexity value. Last, we considered as our vocabulary the 300K most probable words according to the 1-gram composite LM. By using this vocabulary, we built a 2-gram and a 4-gram language models in the same way we trained the composite 1-gram. For these LMs, modified Kneser-Ney discounting was applied, in addition to an interpolation with lower n-gram orders. The SRILM toolkit was used to train these language models [14]. Table 3 presents the perplexities values obtained by the different LMs estimated for the final ASR systems. Results for 3-gram LMs are also provided for a more complete information. Also, the weight λ of the Aljazeera.net-based n-gram LM in the linear interpolation with the n-gram LMs estimated on the filtered AM training corpus are also provided.

Order	Source	PPL
2g	Aljazeera.net	2377
2g	filtered AM training corpus	3055
2g	**composite (lin. interpol.)** λ(Aljazeera.net)=0.52	**1005**
3g	Aljazeera.net	2046
3g	filtered AM training corpus	2781
3g	composite (lin. interpol.) λ(Aljazeera.net)=0.53	847
4g	Aljazeera.net	2002
4g	filtered AM training corpus	2754
4g	**composite (lin. interpol.)** λ(Aljazeera.net)=0.53	**832**

Table 3. Perplexities of single and composite language models on DEV1 set. The ASR systems used composite 2-gram and 4-gram LMs.

3.4. System merging

We applied the fusion of the recognition results from different AMs, on the word lattice level, in the form of confusion network combination (CNC) [15]. In this type of fusion, before merging lattices from different AMs, for each edge, scores were replaced by their a posteriori probabilities. Posteriors were computed for each lattice independently. The optimal normalizing factors for each model were found independently on the development set. Then several lattices were merged into a single lattice and posteriors were weighted, with the optimal weights found on the development set. The resulting lattice was converted into the CN and the final result was obtained from this CN.

4. RESULTS

In order to be able to apply speaker adaptation for TEST set, we used an automatic speaker segmentation, obtained by using the LIUM diarization toolkit [16].

For experiments on DEV1 set, we used the speaker segmentation provided by the organizers. Table 4 summaries the recognition results in terms of WER on DEV1 corpus. Some of experiments presented here were realized after the MGB submission deadline. Since the reference files of the MGB test set has not been distributed to participants, we cannot present our results on the test corpus by using these ASR systems, except for the official LIUM submission.

Results are reported after rescoring word-lattices with the 4-gram LM described in Section 3.3. These word-lattices were obtained by using the 2-gram LM to decode the audio signal, whatever the acoustic models. Numbers in the columns 2—5 are weights of the recognition results (lattices) from corresponding AMs in CNC. Lines 1–4 represent results for single AMs, both for 1-best result from lattices (latt.), and for 1-best result from consensus hypotheses (cons.). Lines 5–10 show results for pairwise combination; and lines 11–14, fusion results for three models. Finally, line 15 demonstrates the WER=19.68%, obtained by fusion the results from all the four AMs.

In addition, we slightly improved this result by multiple decoding with ”frame shifting” strategy for TDNN chain AMs. More precisely, in a standard case, for TDNN AM, decoding was done with 3 frames sub-sampling. So, in order to use all the available information from the features, we shifted features by 0, 1 and 2 frames and combine the decoding results for the obtained features in the same manner, as for other models, using CNC.

The last line (17) in the table shows WER=20.76% for the system that was submitted for the evaluation. Systems #17 and #9, which were obtained by the combination of TDNN_1 and $\mathrm{DNN}_{\mathrm{BN}}$ single ASR systems, differ because of different optimizations, including the weighting used for the confusion network combination. It could be seen that the best result, obtained by combining more acoustic models, exceeds this WER by 1.15% absolute (or 5.5% relative).

On TEST set, as shown in Table 5, the submitted system, according to the official results, gives WER=23.0%, computed in a similar way as the WER presented in Table 4 (original, excluding overlap speech segments). This corresponds to the official retained WER on the test set of 16.7%, computed by the organizers after removing any punctuation or diacritization from the recognition hypotheses and the reference, and by using the official global mapping file (GLM). Even if the system #16 was not ready at the official submission time, we were able to get later its WER on the test corpus, thanks to the MGB organizers. Our last system outperforms the one we used for our official submission by obtaining a WER of 15.7% instead of 16.7% following the official metric (WER/GLM). This last ASR takes benefits from the complementarities provided by the different phonetization used in TDNN_2 and from the use of GMM-D features in $\mathrm{DNN}_{\mathrm{G}}$.

#	TDNN_1	TDNN_2	$\mathrm{DNN}_{\mathrm{BN}}$	$\mathrm{DNN}_{\mathrm{G}}$	WER, % latt.	WER, % cons.
1	-	1	-	-	23.66	23.25
2	-	-	1	-	22.67	22.56
3	-	-	-	1	22.05	21.89
4	1	-	-	-	21.69	21.37
5	-	-	0.525	0.475		21.02
6	-	0.529	0.471	-		20.65
7	0.62	-	-	0.380		20.52
8	-	0.550	-	0.450		20.47
9	0.598	-	0.402	-		20.42
10	0.562	0.438	-	-		20.36
11	0.432	-	0.298	0.271		20.11
12	-	0.423	0.264	0.313		19.99
13	0.303	0.435	-	0.262		19.87
14	0.369	0.332	0.298	-		19.86
15	0.307	0.326	0.183	0.183		19.68
16	#15 with frame shift for TDNN_1 & TDNN_2					**19.61**
primary LIUM submission: (officially second-ranked)						
17	0.630	-	0.370	-		20.76

Table 4. Recognition results on DEV1 set for different acoustic models and fusion.

System	Original WER,%	GLM WER,%
#17 (official submission)	23.0	16.7
#16 (best on dev, not submitted)	**22.1**	**15.7**

Table 5. Recognition results on TEST set.

5. CONCLUSION

In this paper, we describe the LIUM ASR system that has been ranked in second position in the 2016 Arabic MGB Challenge. In addition we report results and methods for further system improvement. The key features of our system are the training data selection approach, four DNN AMs of different types (DNN and TDNN), with various acoustic features (PLP, BN, GMMD), different techniques for speaker adaptation (i-vectors, fMLLR, MAP) and two types of phonetization. The final system, obtained through a confusion network combination of the four developed systems exceeds in performance one the DEV1 corpus the best single LIUM ASR system by approximately 8% of relative WER reduction, and the baseline MGB system provided by the organizers by almost 43%.

6. ACKNOWLEDGEMENT

The authors would like to thank the MGB organizers, and especially Yifan Zhang, to have made possible the late evaluation on the test data of our unofficial last ASR system.

7. REFERENCES

[1] 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 *Proceedings of SLT*, 2016.

[2] Hamdy Mubarak, “Data Description of the Arabic Multi-Genre-Broadcast Challenge,” 2016.

[3] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burge, 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. december 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[4] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn, “Scalable modified Kneser-Ney language model estimation,” in *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013, pp. 690–696.

[5] Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M. Roth, “MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of arabic,” in *Proc. of LREC*, 2014.

[6] Fadi Biadsy, Nizar Habash, and Julia Hirschberg, “Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules,” 2009, pp. 397–405, Association for Computational Linguistics, 00033.

[7] Ahmed Ali, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass, “A complete KALDI recipe for building Arabic speech recognition systems,” in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. 2014, pp. 525–529, IEEE, 00006.

[8] 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,” in *Proc. of Interspeech*, 2016.

[9] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” *Proc. of Interspeech*, pp. 2440–2444, 2015.

[10] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, “Rectifier nonlinearities improve neural network acoustic models,” *Proc. ICML*, vol. 30, no. 1, 2013.

[11] Karel Vesely, A. Ghoshal, L. Burget, and Daniel Povey, “Sequence-discriminative training of deep neural networks,” in *Proc. Interspeech*, Lyon, France, 2013.

[12] Natalia Tomashenko and Yuri Khokhlov, “GMM-derived features for effective unsupervised adaptation of deep neural network acoustic models,” in *Proc. Interspeech*, 2015, pp. 2882–2886.

[13] Natalia Tomashenko, Yuri Khokhlov, and Yannick Esteve, “On the use of Gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models,” in *Proc. Interspeech*, 2016.

[14] Andreas Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proceedings of Interspeech*, Septembre 2002, pp. 901–904.

[15] Gunnar Evermann and PC Woodland, “Posterior probability decoding, confidence estimation and system combination,” 2000.

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