The 2010 CMU GALE Speech-to-Text System

Florian Metze, Roger Hsiao, Qin Jin, Udhyakumar Nallasamy, and Tanja Schultz

Language Technologies Institute; Carnegie Mellon University; Pittsburgh, PA; USA,
{fmetze|wrhsiao|qjin|unallasa|tanja}@cs.cmu.edu

Abstract

This paper describes the latest Speech-to-Text system developed for the Global Autonomous Language Exploitation ("GALE") domain by Carnegie Mellon University (CMU). This systems uses discriminative training, bottle-neck features and other techniques that were not used in previous versions of our system, and is trained on 1150 hours of data from a variety of Arabic speech sources. In this paper, we show how different lexica, pre-processing, and system combination techniques can be used to improve the final output, and provide analysis of the improvements achieved by the individual techniques.

Index Terms: speech recognition, discriminative training, bottle-neck features

1. Introduction

This paper describes recent improvements on the CMU Speech-to-Text system for Modern Standard Arabic (MSA), which was developed as part of our efforts in DARPA's "Global Autonomous Language Exploitation" ("GALE"[1]) program, for the 2009 Speech-to-Text evaluation, within the "Rosetta" team.

In this paper, we focus on the improvements achieved by adding bottle-neck features [1] and model- as well as feature-space discriminative training [2] to our system, in order to create complementary systems for successful system combination.

1.1. The GALE Speech-to-Text Task

The goal of the GALE program is to develop and deploy the capability to absorb, analyze and interpret huge volumes of speech and text in multiple foreign languages, and make them available in English. Currently, efforts are centered on several variants of Arabic, and Mandarin.

There has been a lot of process on this task over the last couple of years, see e. g. [3, 4, 5, 6, 7]. This paper describes the progress of work at CMU since our initial efforts in 2006 [8], using the JRTk/ Ibis toolkit [9].

This paper reports numbers on the dev07, dev08, eval08, and dev09 data sets, which were also used in the official GALE evaluations, all of which contain about 3 h of audio data. For all experiments, system parameters were jointly tuned on the "dev" sets, unless indicated otherwise.

1.2. System Design

The present system is trained on approximately 1 150 h of training data, taken from the GALE P2 and P3 sets[2], using both a vowelized, and an un-vowelized dictionary. The un-vowelized system is trained on the Broadcast News (BN) data only, while the vowelized system is trained on the BN and BC (Broadcast Conversations) sets. The training data provides manual segmentation and speaker clusters.

We extract power spectral features using a FFT with a 10 ms frame-shift and a 16 ms Hamming window from the 16 kHz audio signal. We compute 13 Mel-Frequency Cepstral Coefficients (MFCC) per frame and perform cepstral mean subtraction and variance normalization on a per-cluster basis, followed by VTLN. To incorporate dynamic features, we concatenate 15 adjacent MFCC frames (± 7) and project the 195 dimensional features into a 42-dimensional space using a Linear Discriminant Analysis (LDA) transform. After LDA, we apply a globally pooled ML-trained covariance transformation matrix [10].

For the development of our Gaussian Mixture Model (GMM) based, context dependent acoustic models, we applied an entropy-based poly-phone decision tree clustering process using context questions of maximum width ± 2, resulting in quinphones. In addition, we included "word boundary" tags into the pronunciation dictionary, which can be used as questions in the decision tree. The system uses 6 000 quinphones with up to 64 Gaussians per state, assigned using merge and split training for Maximum Likelihood (ML) or subsequent discriminative training, with diagonal covariance matrices.

During decoding, we perform automatic speaker clustering of manually segmented audio. Segments are clustered into speaker-specific clusters using Bayesian Information Criterion (BIC), to enable adaptation and normalization [11].

The language model (LM) is trained from a variety of sources. The Arabic Gigaword corpus distributed by LDC is the major text resource for language modeling. In addition, we harvested transcripts from Al-Jazeera, Al-Akhbar, and Akhbar Elyom, as described in [8]. Acoustic transcripts from FBIS, TDT-4, GALE BN and BC up to 2008 were also used. The total number of words in the corpus amounted to $1.1 \cdot 10^9$. To improve coverage and specificity for both BN and BC data, we trained 11 different 4-gram language models and interpolated them using the SRILM toolkit [12]. Interpolation weights were selected based on a held-out data set selected from BN and BC sources. The final LM contains 692 M n-grams and a vocabulary of 737 k words. The Confusion Network Combination passes use an improved language model, which was trained on all transcriptions available to date, which however only resulted in an insignificant improvement in word error rate (WER).

Arabic is a phonetic language with a close correspondence between letters and sounds. One of the challenges however is that some vowels are omitted in the written form. These vowels carry grammatical disambiguation information, and may change the meaning of a word. Modeling the vowels in the pronunciation dictionary was found to give improvements, but we also retain an un-vowelized, grapheme-to-phoneme based system, as we find it to be beneficial in system combination. The un-vowelized pronunciation dictionary was generated using grapheme-to-phoneme rules. It contains 37 phones with 3

[1] http://www.darpa.mil/ipto/programs/gale/gale.asp
[2] Available from the Linguistic Data Consortium as LDC2008E38

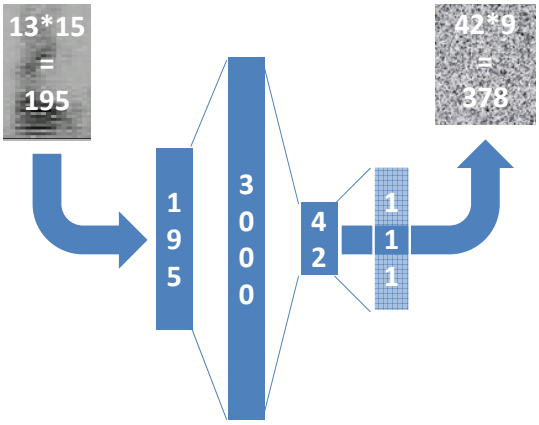

Figure 1: *The network architecture used in our experiments: the MLP input feature has a context window of 15 frames, on 13 MFCC coefficients. The MLP output is taken at the 42-dimensional bottle-neck layer, and 9 frames are stacked. The 111 nodes in the fourth layer are only used during training.*

special phones for silence, non-speech events and non-verbal effects such as hesitations. We preprocessed the text by mapping the 3 shapes of the grapheme for glottal stops to one shape at the beginning of the word since these are frequently miss-transcribed. For the vowelized system, we extended the Buckwalter-based [13] approach described in [8] and use a lexicon of about $2.5 \cdot 10^6$ entries.

The system uses three sets of acoustic models in four passes: (1) speaker independent decoding using the un-vowelized lexicon (UnVow SI), (2) speaker adapted decoding (using VTLN, CMLLR, and MLLR) using the un-vowelized lexicon (UnVow SA), and (3) speaker adapted decoding using the vowelized lexicon (Vow SA). After this pass, we adapt the UnVow SA models on the Vow SA hypotheses and re-decode (pass UnVow SA2), before final system combination.

2. New Techniques

Compared to our previous work, the present system incorporates two main additions. In this section, we will investigate these techniques individually, while the following section reports on their performance as part of the evaluation system.

2.1. Bottle-neck Features

Previous work argues that bottle-neck features, a variant of Tandem or MLP features [14], should be trained on a different input representation than the "conventional" system, for example wLP-TRAP [5, 15]. Improvements are achieved by concatenating and decorrelating the conventional and MLP features before model training. Our results however indicate that the bottle-neck process in itself creates complementary likelihood distributions, so that gains can also be achieved by combining a conventional system with a bottle-neck system using a context independent weighted sum in log-space, e. g. a "multi-stream" system. Compared to feature fusion as in most previous work, this late fusion approach allows for faster development and introduces additional parameters which can be used for optimization and tuning. We will therefore refer to single systems as "MFCC" and "MLP" variants, and use a multi-stream architecture to combine them.

Figure 1 shows the layout of our bottle-neck MLP architecture. Separate networks were trained for the SI (speaker independent: no VTLN, no CMLLR feature transform) and SA (speaker adapted: VTLN, CMLLR feature transform trained on the output of the MLP) cases, on their respective feature spaces. VTLN Warping factors for the SA systems were estimated using an ML-based approach [16], using MFCC models only.

During pre-processing for bottle-neck systems, the LDA transform is replaced by the first 3 layers of the Multi Layer Perceptron (MLP) using a 195–3000–42 feed-forward architecture, followed by stacking of 9 consecutive bottle-neck output frames. A 42-dimensional feature vector is again generated by LDA, followed by a covariance transform. The neural networks were trained using ICSI's QuickNet[3] software, on 500 h of data extracted from the training data using a modulo operator on the utterance list. The bottle-neck setup is shown in Figure 1.

UnVow SI on dev07	MFCC	MLP
WER (%)	20.1	19.6
RTF (median)	5.1	4.1
Average # of back-pointers	56 080	40 849
Average lattice density	59	54
Average neg. log. likelihood	51.8	45.9

Table 1: *Comparison of the MFCC and MLP ML-trained systems. The median per-utterance real time factor (RTF) is being reported, because measurements of total RTFs are unreliable on our cluster.*

Table 1 shows keys characteristics of the individual UnVow SI systems. The language model weights and beam settings for the MFCC and MLP systems were optimized separately, and the MLP system seems to perform better than the non-MLP system in all respects: all other parameters being similar, the MLP features can be decoded in less time and has a more compact search space for a given word accuracy, with better likelihood than the MFCC system.

For the UnVow SA system trained using ML, the MFCC system on its own reaches 16.6 % WER on dev07, the MLP system reaches 16.8 %, and a two-stream "MFCC+MLP" system reaches 15.9 %, using manually adjusted context independent stream weights. After adaptation however, the MLP stream no longer outperforms the MFCC stream.

2.2. Generalized Discriminative Feature Transform

Discriminative training was applied to the UnVow SA and Vow SA models and MLP and MFCC feature spaces, as shown in Table 2. We used boosted Maximum Mutual Information (bMMI) estimation [17] for model space Discriminative Training (DT), and generalized discriminative feature transformation (GDFT) [2] for feature space training. GDFT can be considered as a discriminative variant of the CMLLR algorithm.. The formulation of GDFT allows joint optimization of both HMM parameters and feature transforms which can significantly shorten the time for training. In our experiments, GDFT optimizes the feature transforms for the bMMI objective function.

Unlike the work conducted in [2], regularization is incorporated in the GDFT optimization problem. The resulting algorithm is named regularized GDFT (rGDFT). The primal prob-

[3] http://www.icsi.berkeley.edu/Speech/qn.html

lem of rGDFT is

$$G'(W) = \sum_i |Q_i(W) - C_i| + \frac{D}{2}||W - W^0||_F^2 \ ,$$

where $Q_i(W)$ is the negative log likelihood function of i-th utterance given a linear transform, W; C_i is the chosen target value that we want Q_i to achieve; W^0 is the backoff linear transform that we want W to backoff to; $||W - W^0||_F$ is the Frobenius norm between W and W^0 and D is a tunable parameter controlling the weight of this regularization term. When $D = 0$, rGDFT reduces to the original GDFT, and W^0 is chosen to be the identity matrix in our experiments.

GDFT has an update equation very similar to CMLLR [2]. With regularization, it only requires adding $D \times I$ to the G matrices and D times the row vectors of W^0 to the corresponding k vectors. This modification allows GDFT to incorporate more transforms, since the transforms without enough data will simply backoff to W^0. In our experiments, rGDFT adopts 2 048 transforms while the original GDFT can only support no more than a few hundred transforms. For the D parameter, we apply heuristics, i. e. $D = E \times \gamma_{den}$, where E is tuned from 1 to 2.

Overall, gains over ML are up to 10 % relative on the UNVOW SA systems, and about 5 % on the VOW SA systems, for fully trained systems. For lack of resources, the UNVOW SA MLP system has only been trained for one iteration without GDFT at this time, but shows improvements as well.

3. System Development

The techniques described above were integrated, and tested on the conditions of the 2009 GALE STT evaluation. Based on preliminary experiments, we decided to do an initial first pass using essentially an existing UNVOW SI system, then adapt a UNVOW SA system based on the un-vowelized lexicon on these hypotheses, and finally decode the data with a vowelized VOW SA system, adapted on the UNVOW SA hypotheses. This configuration, with appropriate cross adaptation, resulted in the best performance of the single best final system. MLP streams were added to the un-vowelized systems, for faster training and improved diversity. We improve individual systems and gain about 0.2 % when adapting the VOW SA system (cf. line "rGDFT+bMMI" in Table 2 and line "Vow SA" in Table 3).

3.1. Speaker Independent Pass

As the segmentation of the test data is given, the first pass UNVOW SI simply decodes the data without VTLN and CMLLR/

System	dev07	dev08	eval08	dev09
UNVOW SA MLP				
ML	16.7	19.7	16.1	23.5
1i bMMI	16.4	19.4	15.7	22.8
UNVOW SA MFCC				
ML	16.7	19.3	16.1	22.9
rGDFT + bMMI	15.0	17.7	15.2	22.0
VOW SA MFCC				
ML	14.3	15.9	13.9	N/A
rGDFT + bMMI	13.7	15.3	13.3	21.0

Table 2: *Summary of single system DT experiments (WER in %). These systems were adapted using hypotheses from a* UNVOW SI/ SA *single stream (MFCC) system, so that the numbers are slightly worse than the numbers reported in Table 3.*

System	dev07	dev08	eval08	dev09
UNVOW SI	18.1	20.9	17.2	24.7
UNVOW SA	14.8	17.2	14.3	21.2
VOW SA	13.5	15.2	13.6	20.6
UNVOW SA2	13.6	15.6	13.5	20.0
CNC of VOW SA & UNVOW SA/ SA2				
CNC	13.2	15.2	13.2	19.9
CNC2	12.9	14.9	12.8	19.5
Latent Semantic Analysis (LSA, on individual systems)				
UNVOW SA'	14.5	16.9	14.0	20.9
VOW SA'	13.0	15.0	13.2	20.2
UNVOW SA2'	13.4	15.5	13.2	19.6
CNC on LSA systems				
CNC'	12.6	14.5	12.4	18.7

Table 3: *Top part: Word Error Rates (in %) on GALE data for different passes, adapted sequentially. Then: Confusion Network Combination (CNC) between these systems and lattice rescoring using Latent Semantic Analysis (LSA), plus CNC of LSA lattices. All* UNVOW *systems are MFCC+MLP two-stream systems,* VOW SA *is MFCC only.*

MLLR adaptation, in order to generate a first hypothesis for subsequent unsupervised adaptation to the test data. The acoustic model of this two-stream "MFCC+MLP" system consists of an equally weighted log-linear interpolation of two acoustic scores computed by Gaussian Mixture Models (GMMs) trained as described in Sections 1.2 and 2.1. Both streams share the same context decision tree, trained on the non-MLP feature space with a context of ± 2 phones, and contains 6 000 leafs.

The MLP was trained on non-VTLN MFCC features from a 250 h subset of the GALE training data (selected using a modulo operation on utterances) for 8 epochs using QuickNet, and reached 52.8 % frame accuracy on the training data, and 51.4 % frame accuracy on the cross validation data, for which we randomly chose 13 h from the remaining GALE data. The MLP was trained on 111 context independent sub-phonetic states as targets. Training took 32 h on an 8 core Linux server.

On dev07, this two-stream system delivers a WER of 18.1 % (see Table 3) instead of 19.6 % and 20.1 % (see Table 1) for the single-stream MLP and MFCC systems. During adaptation, we compute scores for all needed codebooks and frames, and store them, instead of the adapted codebooks. This saves time, RAM, and disk space, because an array of codebooks can be evaluated very efficiently on modern multi-core processors.

3.2. Un-Vowelized Speaker Adapted Pass

The acoustic models for this UNVOW SA pass are adapted on hypotheses and confidences generated using UNVOW SI. The MLP was trained on a 500 h subset of the GALE training data, with the same 13 h test set. It achieved a frame accuracy of 53.3 % after 8 iterations of training (51.5 % on the cross validation data), which required 96 h of training.

The individual acoustic models are trained in a feature space that has been adapted to speakers using CMLLR, and we are using the rGDFT+bMMI acoustic models for the MFCC case, and bMMI acoustic models for the MLP case. Using ML models, the MLP stream reaches about the same performance as the MFCC stream (Table 2), and the optimized two-stream system numbers given in Table 3 are about 0.3-0.6 % better than the best single stream system. The MLP system was only trained

with a single iteration of bMMI due to training time constraints, so that the performance is not yet optimized. To increase the diversity within systems, we also trained the MFCC system with 8 000 states, instead of 6 000, however this did not improve the performance of the combined system.

For improved cross-adaptation, we also adapted these acoustic models using the hypotheses from the VOW SA pass (see below), and called this the UNVOW SA2 pass. This pass is 0.8-1.6 % better than UNVOW SA, and reaches roughly the same performance as the VOW SA pass.

3.3. Vowelized Speaker Adapted Pass

This pass VOW SA is adapted on UNVOW SA. Due to training time constraints, we did not train a separate MLP-based system for the vowelized condition, but used the MFCC system alone. This discriminatively trained single-stream system reaches the same performance as the two-stream discriminatively trained un-vowelized MFCC+MLP system UNVOW SA2, which was adapted on VOW SA, see Table 3.

3.4. Lattice Rescoring and System Combination

In a final step, we re-scored the lattices generated by our adapted systems using a Latent Semantic Analysis (LSA) [18] based language model. Also, we combined lattices from different passes before and after LSA using Confusion Network Combination (CNC). LSA typically improves the word error rate (WER) by about 0.3 % absolute. Combining the VOW SA system with UNVOW SA2 ("CNC2") instead of UNVOW SA ("CNC") improves the performance by about 0.3 %, even though UNVOW SA2 is about 1.2 % better than UNVOW SA. Combining the UNVOW SA' and VOW SA' LSA systems using CNC leads to the overall best system CNC'. At this point, a combination with the re-adapted system UNVOW SA2' does not improve the performance further.

4. Conclusion and Future Work

This paper presents recent work, mainly on core acoustic modeling techniques, applied to the GALE Arabic Speech-to-Text task. By adding discriminative training of acoustic models using a new approach which transforms both features and models in the same model update, and by adding a bottle-neck layer to the feature pre-processing, we were able to improve the word error rate of our Arabic STT system by more than 10% relative, compared to our 2008 system, which again presents a major improvement from previous own published work [8].

Absolute system performance could certainly be improved further, in particular on newer test data, by re-training acoustic and language models on all the available data, and be further optimizing settings. The MFCC+MLP setup performs well, also for system combination, however we were not yet able to fully explore the set-up for cross-adaptation of acoustic models, as in [19], and fully optimize the bottle-neck setup. Future work will investigate combinations of bottle-neck pre-processing and feature- and model-space discriminative training, particularly to improve the performance on low accuracy parts of the data, acoustically challenging recordings, and dialectal data.

5. Acknowledgements

This work was partly supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under contract HR0011-06-2-0001 ("GALE"). Any opinions, findings, conclusions and/or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of DARPA.

6. References

[1] F. Grézl and P. Fousek, "Optimizing bottle-neck features for LVCSR," in *Proc. ICASSP*. Las Vegas, NV; USA: IEEE, Apr. 2008.

[2] R. Hsiao and T. Schultz, "Generalized discriminative feature transformation for speech recognition," in *Proc. INTERSPEECH*. Brighton; UK: ISCA, Sep. 2009.

[3] G. Saon, H. Soltau, U. Chaudhari, S. Chu, B. Kingsbury, H.-K. Kuo, L. Mangu, and D. Povey, "The IBM 2008 GALE Arabic speech transcription system," in *Proc. ICASSP*. Dallas, TX; USA: IEEE, Apr. 2010.

[4] M. Tomalin, F. Diehl, M. Gales, J. Park, and P. Woodland, "Recent improvements to the Cambridge Arabic speech-to-text systems," in *Proc. ICASSP*. Dallas, TX; USA: IEEE, Apr. 2010.

[5] P. Fousek, L. Lamel, and J.-L. Gauvain, "Transcribing broadcast data using MLP features," in *Proc. InterSpeech 2008*. Brisbane; Australia: ISCA, Sep. 2008.

[6] D. Vergyri, A. Mandal, W. Wang, A. Stolcke, J. Zheng, M. Graciarena, D. Rybach, C. Gollan, R. Schlüter, K. Kirchhoff, A. Faria, and N. Morgan, "Development of the SRI/ Nightingale Arabic ASR system," in *Proc. INTERSPEECH*. Brisbane, Australia: ISCA, Sep. 2008.

[7] L. Nguyen, T. Ng, K. Nguyen, R. Zbib, and J. Makhoul, "Lexical and phonetic modeling for Arabic automatic speech recognition," in *Proc. INTERSPEECH*. Brighton, UK: ISCA, Sep. 2009.

[8] M. Noamany, T. Schaaf, and T. Schultz, "Advances in the CMU/InterACT Arabic GALE transcription system," in *Proc. NAACL/ HLT 2007; Companion Volume, Short Papers*. Rochester, NY; USA: ACL, April 2007, pp. 129–132.

[9] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A One-pass Decoder based on Polymorphic Linguistic Context Assignment," in *Proc. ASRU 2001*. Madonna di Campiglio, Italy: IEEE, Dec. 2001.

[10] M. J. F. Gales, "Semi-Tied Covariance Matrices for Hidden Markov Models," *IEEE Transactions on Speech and Audio Processing*, vol. Vol. 2, May 1999.

[11] Q. Jin and T. Schultz, "Speaker segmentation and clustering in meetings," in *Proc. ICSLP*. Jeju Island; Korea: ISCA, Oct. 2004.

[12] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. Intl. Conf. on Spoken Language Processing*. Denver, CO: ISCA, Sep. 2002.

[13] T. Buckwalter, "Issues in Arabic Orthography and Morphology Analysis," in *Proc. COLING*, Geneva; Switzerland, 2004.

[14] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, vol. 3. Istanbul; Turkey: IEEE, Apr. 2000.

[15] J. Park, F. Diehl, M. J. F. Gales, M. Tomalin, and P. C. Woodland, "Training and adapting MLP features for Arabic speech recognition," in *Proc. ICASSP 2009*. Taipei; Taiwan: IEEE, Apr. 2009.

[16] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *Proc. ICASSP 1997*. München; Bavaria: IEEE, Apr. 1997.

[17] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for Model and Feature-space Discriminative Training," in *Proc. ICASSP*. Las Vegas, NV; USA: IEEE, Apr. 2008.

[18] Y.-C. Tam and T. Schultz, "Correlated Bigram LSA for Unsupervised LM Adaptation," in *Proc. Neural Information Processing Systems, NIPS*, Vancouver, BC; Canada, Dec. 2008.

[19] C. Ma, H.-K. J. Kuo, H. Soltau, X. Cui, U. Chaudhari, L. Mangu, and C.-H. Lee, "A comparative study on system combination schemes for LVCSR," in *Proc ICASSP*. Dallas, TX; USA: IEEE, Mar. 2010.