Adapting Acoustic Models to New Domains and Conditions Using Untranscribed Data

Asela Gunawardana and Alex Acero

Microsoft Research,
One Microsoft Way,
Redmond, WA 98101, USA
{aselag|alexac}@microsoft.com

Abstract

This paper investigates the unsupervised adaptation of an acoustic model to a domain with mismatched acoustic conditions. We use techniques borrowed from the unsupervised training literature to adapt an acoustic model trained on the Wall Street Journal corpus to the Aurora-2 domain, which is composed of read digit strings over a simulated noisy telephone channel. We show that it is possible to use untranscribed in-domain data to get significant performance improvements, even when it is severely mismatched to the acoustic model training data.

1. Introduction

Progress in automatic speech recognition (ASR) research in the last few decades has resulted in the increasing use of small to medium vocabulary closed domain ASR systems in commercial applications. Ideally, a large amount of acoustic data from each target domain would be collected and transcribed, so that a system trained to match the target application can be built. However, as ASR systems become more widespread, this ideal scenario becomes prohibitively expensive. Thus, it is becoming necessary to build systems that can be deployed in various domains and conditions with minimal effort spent on tailoring the system for each of these domains or conditions. Systems need to be robust with respect to a variety of factors, such as language model mismatch, out-of-vocabulary words, pronunciation variation, channel variation, noise, and speaking style. All parts of the system need to be robust to these changes, or be able to adapt to these new conditions with minimal human effort.

Here, we restrict our attention to the problem of adapting the acoustic models to a new domain where the channel and noise characteristics are different from those seen during training. A deployed system can easily collect large amounts of adaptation data from the target domain. However, this is untranscribed acoustic data, and transcribing it would require a relatively large effort. Therefore, using untranscribed acoustic data for domain adaptation is a realistic problem of immediate interest. The scenario we envision is that a mismatched system would be deployed, and would automatically adapt to the target conditions, becoming better and better matched.

We simulate this scenario as follows. We train a "domain-independent" unadapted acoustic model using the DARPA Wall Street Journal pilot (WSJ0) corpus [1]. Two "deployment conditions" are provided by the Aurora-2 database [2]. The first condition is a simulated telephone channel with no added noise, while the second condition is a simulated telephone channel with various forms of recorded noise (such as babble, subway, restaurant, etc) artificially added to the signal at various SNRs. The database provides training and test data from both conditions. To simulate our scenario, we use the training set as adaptation data and ignore the training transcriptions.

Most approaches to robust acoustic modeling either attempt to explicitly model and account for sources of variation such as channel and noise [3], or use adaptation techniques such MLLR [4, 5], MAP [6], or SMAP [7] that do not explicitly model the sources of variation. While the latter class of approaches are less powerful than the former, and usually require more data, they do not require explicit modeling of, or even the knowledge of, the causes of model mismatch. Because we may not have foreknowledge of these factors, and are in a data-rich situation where we are attempting to minimize the effort required for system adaptation, we opt for the latter class of approaches. Since we have a large amount of adaptation data which is untranscribed we follow the approach taken by [8] and [9] for the unsupervised training of acoustic models.

The paper is organized as follows. In Section 2, we formally present our adaptation scheme as an EM algorithm and discuss the approximations made in its implementation. In Section 3, we describe the WSJ0 models that we use to simulate the "domain-independent" baseline acoustic models. We then describe the target domain and acoustic conditions provided by the Aurora-2 database in Section 4. Here, we will also describe some cheating experiments that give upper bounds for the performance of our adaptation scheme. We then present our results in Section 5 and conclude with some discussion in Section 6.

2. Unsupervised Acoustic Model Training

The problem of estimating acoustic models from untranscribed acoustic data can be viewed as a missing data problem where the EM algorithm [10] can be applied. That is, we have a model $q_{W_1^N, O_1^L;\, \theta}$ on a joint word sequence and acoustic observations sequence (W_1^N, O_1^L) and need to estimate the parameter θ model based on only an observation $\hat{o}_1^l$ of O_1^L. In actuality, our model is an HMM, so

$$q(w_1^n, s_1^l, o_1^l;\ \theta) = q(w_1^n)q(s_1^l|w_1^n)q(o_1^l|s_1^l;\ \theta)$$

where we have assumed for simplicity that only the emission densities are being reestimated.

Thus, instead of the usual hidden state sequence S_1^L we usually encounter in acoustic modeling, we have two hidden variables W_1^N and S_1^L. We can still solve this estimation problem

using the EM algorithm, maximizing the auxiliary function:

$$\theta^* = \arg\max_{\theta} \mathbf{E}\left[\log q(W_1^N, S_1^L, \hat{o}_1^{\hat{l}}|;\ \theta) \mid \hat{o}_1^{\hat{l}}; \theta'\right]$$
$$= \arg\max_{\theta} \sum_{w_1^n} \sum_{s_1^l} q(w_1^n, s_1^l|\hat{o}_1^{\hat{l}}; \theta') \log q(w_1^n, s_1^l, \hat{o}_1^{\hat{l}}|;\ \theta).$$

For simplicity, we use a Viterbi approximation for the first sum, maximizing

$$\max_{w_1^n} \sum_{s_1^l} q(w_1^n, s_1^l|\hat{o}_1^{\hat{l}};\ \theta') \log q(w_1^n, s_1^l, \hat{o}_1^{\hat{l}};\ \theta)$$

over θ.

To implement this reestimation scheme, we run our decoder on the training set, and then run Baum-Welch reestimation using the generated transcription. At each iteration of training, we rerun the decoding to generate a new transcription. This procedure involves a further approximation. The decoder finds the most likely joint word and state sequence, and then discards this state sequence. An exact implementation of the reestimation scheme above would require a more sophisticated decoder that searches for the most likely word sequence, irrespective of alignment.

3. The Baseline Acoustic Model

Our baseline acoustic model was trained on the SI-84 training set of the WSJ0 corpus, using only the data from the Sennheiser close-talking microphone channel. The front end band-limited and down-sampled the data to 8 kHz, and used 12 mel-frequency cepstral coefficients (MFCC) and the log energy with their first and second derivatives to give a 39-dimensional feature vector. Cepstral mean subtraction was done at the per-utterance level. The acoustic models were three state left-to-right cross-word triphone HMMs, with states clustered using decision trees. There were 4062 clustered triphone states, each with an 8 component Gaussian mixture emission density. A proprietary pronunciation dictionary was used, with approximately 1.2 pronunciations per word, and no pronunciation weights.

When using the 5k word closed vocabulary language model supplied with the WSJ0 corpus, these acoustic models give a word accuracy of 92.18%, on the ARPA November 1992 WSJ evaluation test set. When a similar acoustic model is trained on full bandwidth data, the corresponding accuracy is 93.74%. Thus the baseline considered here is comparable with that reported in [11], and lacks enhancements such as quinphones and word boundary dependent models.

4. The Target Domain

We use the Aurora-2 database [2] as our target domain. This corpus is a subset of the TI Digits corpus, down sampled to telephone bandwidth, with various distortions artificially added to the data. The corpus contains two sets of training data. The first, referred to as the "clean" set, consists of 8440 digit strings that have been filtered to simulate a telephone channel. The second, referred to as the "multi-condition" set, consists of the same data, split into twenty subsets, with four kinds of noise (such as train, babble, etc) added at five different SNRs each.

Three test sets are provided. The first (Set A) contains 4004 utterances. These utterances are split into four subsets of 1001 each, and each of the four noise types from the multi-condition training set is added to a subset. This is then repeated at different SNR levels. Test Set B uses the identical procedure, but different noise types that are not seen in training. Both Set A and Set B use the same simulated channel as the training data. Set C consists of two of the four subsets of test utterances using a different simulated channel. One subset uses noise seen in training while the other uses noise not seen in training.

We use this data to create two adaption scenarios. In the first, we use the clean training set as adaptation data, and a clean version of all three test sets as test data. In the second, we use the multi-condition training set as adaptation data, and the noisy versions of all three test sets as test data.

During test, and when decoding the training data, we restrict the recognizer to the eleven digit words ("one" through "nine" plus "oh" and "zero") found in the corpus. As mentioned in Section 1, we do not address the problems of vocabulary, language model, and pronunciation adaptation here. When the baseline acoustic models are restricted to cross-word triphones found using this vocabulary, there are 388 clustered states.

Two sets of experiments are performed. In the first, the adaptation is done on the clean adaptation data, and test results on a clean version of the test sets described above are reported. Although there is no added noise in this case, the acoustics still have the telephone channel filter applied, and in the case of Set C, this filter is mismatched with training. In the second set of experiments, adaptation is done on the multi-condition data, and test results on the noisy test sets are reported. In this second set of experiments, the noise in the training data exposes the acoustic models to the danger of being corrupted by a large amount of data with bad hypothesized transcriptions. To guard against this possibility, we sort the training data by estimated SNR, and then reestimate the models in four phases. In phase 1, we use only the cleanest (according to our estimate) 25% of the data, and add a further 25% of the training set in each phase, in order of estimated SNR, so that all the data is used in phase 4. Thus, initial phases only use utterances whose transcriptions we have more reason to trust. The SNR estimate used here is the ratio of the energy in the highest energy frame of the utterance to that in the lowest energy frame. Note that more elaborate confidence measures could be used for training data selection, and that our SNR scheme is not an option where domain mismatch may not be caused by noise.

We performed two sets of cheating experiments to set performance bounds on the task. In the first, we trained clean and multi-condition triphone acoustic models on the Aurora data, ignoring the WSJ0 models. We refer to these models as the "Aurora models." In this case, we optimized the model size to the task. The resulting models had 342 tied states for the clean models and 267 tied states for the multi-condition models, with 22 Gaussian components per state in both cases. In the second, we retrained the WSJ0 models on the clean and multi-condition training sets using the reference transcriptions. These are referred to as the "cheating models." Here, we do not change the model topology or complexity during training, since the intent is to find the loss we suffer for not transcribing the data. The multi-condition training was performed without the data selection criterion described above.

5. Results

Table 1 summarizes the results of the experiments described above. Comparing the Aurora and cheating results shows that using the model structure and complexity that was tuned to the WSJ task instead of the target task costs about 20% relative in error rate. It can be seen that in the clean data case, there is virtually no improvement to be had through transcribing the acous-

tic data. In the noisy case, training without transcriptions gives only about two thirds of the gain that would have been possible had we transcribed the data.

	clean	multi
WSJ Baseline	97.83%	75.21%
4 it.s	99.56%	85.36%
8 it.s	99.61%	87.20%
12 it.s	–	87.60%
16 it.s	–	87.96%
Cheat	99.63%	93.09%
Aurora	99.80%	94.22%

Table 1: Word accuracies of the initial WSJ baseline system, the system after four, eight, twelve, and sixteen iterations of unsupervised retraining, the system retrained with supervising using the true (cheating) transcriptions, and the Aurora-only system on the clean and multi-condition test sets. The "clean" column is the performance of the clean data adapted models on clean data, while the "multi" column is the average performance of the multi-condition adapted models over all the noisy test sets. The first row corresponds to the unadapted model on these two sets. Reestimation of the clean system was terminated after 8 iterations due to the diminishing gains.

Table 2 expands the second column of Table 1, breaking the results out by SNR. Here, it can be seen that in the lower noise conditions (15dB, 20dB, and clean), we get most of the gain we could potentially have got had we transcribed the data. However, at lower SNRs, the improvement obtained from unsupervised retraining is somewhat smaller, though still significant.

	Baseline	4 iter.	16 iter.	Cheat
clean	94.11%	98.05%	97.79%	98.74%
20 dB	93.31%	97.97%	97.98%	98.91%
15 dB	90.90%	97.24%	97.48%	98.58%
10 dB	82.80%	94.02%	95.20%	97.24%
5 dB	63.01%	80.62%	85.66%	92.44%
0 dB	46.04%	56.95%	63.47%	78.30%
-5 dB	39.79%	42.58%	46.78%	55.48%
Overall	75.21%	85.36%	87.96%	93.09%

Table 2: Word accuracies of the initial WSJ baseline system, the system after four and sixteen iterations of retraining, and the system retrained with the true (cheating) transcriptions, in the multi-condition case, broken down by SNR. Note that clean and -5 dB SNR cases are excluded in the overall results, as this is the standard practice in reporting results on this task.

Note that the Aurora results presented here are somewhat better than those published elsewhere because we do not restrict ourselves to the back-end specified in [2], as these restrictions do not make sense for our purposes. In particular we use cross-word models instead of whole word models, and use an order of magnitude more parameters. Also, the front end is non-causal, since it uses information from the whole utterance for gain normalization and cepstral mean subtraction.

Finally, Figure 1 shows how the multi-condition training set transcription improves with retraining and re-transcription. This trend holds on the clean training set as well, but it less dramatic. It can be seen that the transcription accuracy saturates after 4 or five iterations. We observed that the test set accuracy stops improving an iteration or two after that.

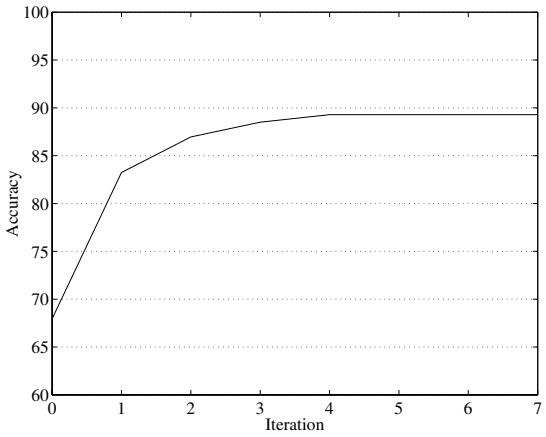

Figure 1: The accuracy of the automatically generated training transcription as a function of iteration number, for the multi-condition set. Notice that this saturates after about five iterations.

5.1. Data Selection in the Presence of Noise

To investigate the effect of data selection in the multi-condition case, we performed the adaptation in four phases as described in Section 4 above. Each phase consisted of four iterations of re-transcription and reestimation. Table 3 shows the performance of the system after each phase, while Table 4 shows the breakdown of the training data used in each phase. Notice that the introduction of data in order of increasing difficulty gives a significant improvement in performance over introducing all the data at once, as was done in Table 2. Given that sixteen iterations of retraining are to be performed, it is better to use only the cleaner utterances in the earlier iterations. Table 3 also shows that even the utterances with higher amounts of noise contribute to improving performance. That is, there is no gain from completely excluding the noisiest 25% of the data (rather than just delaying its introduction).

	Phase 1	Phase 2	Phase 3	Phase 4
clean	95.85%	95.92%	95.55%	95.28%
20 dB	96.89%	97.27%	97.16%	97.06%
15 dB	95.72%	96.45%	96.69%	96.78%
10 dB	91.55%	93.20%	94.14%	94.67%
5 dB	80.56%	82.60%	85.72%	87.69%
0 dB	60.20%	59.11%	66.29%	70.50%
-5 dB	45.42%	43.80%	47.01%	49.83%
Overall	84.89%	85.72%	88.0%	89.34%

Table 3: Word accuracies of the WSJ system when it is retrained in four phases, each phase including noisier data in addition to the data from the previous one. Notice that performance is superior in this ordered adaptation than in the "batch" adaptation shown in Tables 1 and 2.

6. Discussion

As mentioned in Section 1, this work is heavily inspired by the approach described in [9], where a seed acoustic model is trained on a small amount of transcribed data, and is then used to iteratively transcribe more data which is then used to reestimate the acoustic model. The problem we describe is even

	Phase 1	Phase 2	Phase 3	Phase 4
clean	1534	1667	1687	1688
20 dB	449	1243	1568	1688
15 dB	120	952	1429	1688
10 dB	7	367	1077	1688
5 dB	0	91	569	1688
Overall	2110	4220	6330	8440

Table 4: Amount of traning data used in each phase, broken down by SNR.

more conducive to this approach because the target task is fairly limited so that the accuracy of the automatically generated transcriptions is better. However, re-transcription of the training data between iterations is more important here, since the mismatch between training and target conditions means that initial transcriptions are of fairly low quality. The sequential nature of the approach in [9] also means that more and more data gets transcribed with better and better acoustic models, while in our batch solution, re-transcription is more important.

All adaptation experiments presented here were done using full model reestimation – no parameter tying methods (such as MLLR) or smoothing methods (such as MAP) were used. This is because, unlike in the case of speaker adaptation, we have sufficient data for unconstrained ML estimation. In tasks where this is not the case, the approach used here could be extended by using these more sophisticated adaptation techniques. However, since we do not require supervision, it should be possible to collect enough data for full unconstrained retraining in most cases.

The results of the previous section show that transcribed in-domain data was not necessary to adapt WSJ acoustic models to the Aurora-2 task, as long as noise level were not too high. Even when noise levels are higher, there is no penalty incurred by unsupervised adaptation, and in fact, significant improvements are still obtained. However, in these cases, the cost of transcribing at least some of the data may be justifiable. However, further work needs to be done on how to identify the most useful utterances for transcription. We also showed that in the noisy (high error-rate) case, the order in which data is introduced into the system makes a difference in performance, and that reestinmating with easier data first may be helpful. These results are likely to generalize to other tasks, as long as the target domain is fairly small, so that automatic transcriptions will still be fairly reliable.

7. Acknowledgments

The authors would like to thank Jasha Droppo and Mei-Yu Hwang for their assistance.

8. References

[1] D. Paul and J. Baker, "The design for the Wall Street Journal based CSR corpus," DARPA, Feb. 1992.

[2] H.-G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ITRW ASR*, (Paris), ISCA, 2000.

[3] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora2 databse," in *Eurospeech*, vol. 1, (Aarlborg, Denmark), pp. 217–220, ISCA, 2001.

[4] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Comp. Spch. & Lang.*, vol. 9, pp. 171–185, Apr. 1995.

[5] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Spch. & Aud. Proc.*, vol. 3, pp. 357–366, Sept. 1995.

[6] C.-H. Lee and J.-L. Gauvain, "Speaker adaptation based on MAP estimation of HMM parameters," in *ICASSP*, vol. II, pp. 558–561, IEEE, 1993.

[7] K. Shinoda and C.-H. Lee, "Structural MAP speaker adaptation using hierarchical priors," in *IEEE Wkshp. Spch. Recog. & Und.* (S. Furui, B.-H. Juang, and W. Chou, eds.), pp. 381–387, 1997.

[8] G. Zavaliagkos and T. Colthurst, "Utilizing untranscribed training datat to improve performance," in *Broad. Nws. Trans. & Und. Wkshp.*, DARPA, 1998.

[9] L. Lamel, J.-L. Gauvain, and G. Adda, "Unsupervised acoustic model training," in *ICASSP*, vol. 1, pp. 877–880, IEEE, 2002.

[10] A. P. Dempster, A. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data," *J. Roy. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[11] P. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using HTK," in *ICASSP*, vol. II, pp. 125–128, 1994.