DEVELOPING SPEECH RECOGNITION SYSTEMS FOR CORPUS INDEXING UNDER THE IARPA BABEL PROGRAM

Jia Cui[1]*, Xiaodong Cui*[1]*, Bhuvana Ramabhadran*[1]*, Janice Kim*[1]*, Brian Kingsbury*[1]*,*
Jonathan Mamou[2]*, Lidia Mangu*[1]*, Michael Picheny*[1]*, Tara N. Sainath*[1]*, Abhinav Sethy*[1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
[2]IBM Haifa Research Labs, Haifa 31905, Israel

ABSTRACT

Automatic speech recognition is a core component of many applications, including keyword search. In this paper we describe experiments on acoustic modeling, language modeling, and decoding for keyword search on a Cantonese conversational telephony corpus collected as part of the IARPA Babel program. We show that acoustic modeling techniques such as the bootstrapped-and-restructured model and deep neural network acoustic model significantly outperform a state-of-the-art baseline GMM/HMM model, in terms of both recognition performance and keyword search performance, with improvements of up to 11% relative character error rate reduction and 31% relative maximum term weighted value improvement. We show that while an interpolated Model M and neural network LM improve recognition performance, they do not improve keyword search results; however, the advanced LM does reduce the size of the keyword search index. Finally, we show that a simple form of automatically adapted keyword search performs 16% better than a pre-indexed search system, indicating that out-of-vocabulary search is still a challenge.

Index Terms— acoustic modeling, language modeling, bootstrap, deep learning, keyword search

1. INTRODUCTION

The IARPA Babel program is a research program to develop technologies that enable rapid deployment of spoken term detection systems for low-resource languages. In this work, we focus on speech recognition and keyword search experiments on Cantonese conversational telephony speech collected for the first period of the Babel program, although the techniques we describe are also being applied to Pashto, Tagalog, and Turkish. We explore the effects of different automatic speech recognition methods on Cantonese keyword search performance using two different query lists: a 389-term list used for system development that was created by the Babelon and RADICAL teams in the Babel program and a larger 1000-term list used in a dry run evaluation held in August 2012.

In Section 2 we describe four different acoustic models used in this work: a state-of-the-art baseline GMM/HMM model, a GMM/HMM model trained using the bootstrap-and-restructuring procedure, a deep neural network acoustic model, and a GMM model that uses deep neural network features. In Section 3 we describe three language models used in this work: a baseline tri-gram LM, a Model M LM, and a neural network LM. In Section 4 we describe the metric used to measure keyword search performance and explain our approach to audio indexing and keyword search, which is based on weighed finite state transducers. We present experimental results in Section 5 and draw conclusions in Section 6.

2. ACOUSTIC MODELING

2.1. Baseline GMM/HMM Model

The baseline GMM/HMM model is a discriminatively trained speaker adaptive model trained using IBM's standard procedures [1]. The feature space is derived from 13-dimensional PLP features. Acoustic context is taken into account by splicing 9 adjacent frames of mean-normalized PLP features and then projecting to a 40-dimensional feature space using linear discriminant analysis (LDA), followed by a global semi-tied covariance (STC). Vocal tract length normalization and speaker adaptive training (SAT) using a single feature-space maximum likelihood linear regression (FMLLR) transform are used to reduce speaker variability. Following SAT training, feature- and model-space discriminative training are carried out under the boosted maximum mutual information (BMMI) criterion. At test time, additional speaker adaptation is performed with multiple MLLR transforms. The baseline GMM/HMM model has 3,000 quinphone states and 200K Gaussians. Similar baselines have been used in previous evaluation systems [2, 3, 1], but in the previous work there were more resources available for acoustic and language model training than are provided in the Babel program.

2.2. Bootstrap and Restructuring Model

Bootstrap and restructuring (BSRS) is a statistical approach based on subset bagging to deal with limited training data [4]. In BSRS, the training utterances are first sampled without replacement into N subsets and an HMM is trained from each subset. As a sequence classifier, the discriminant function for the HMM λ^B is defined as

$$D_{\lambda^B}(\mathcal{O}, S) = \sum_{t=1}^{T} \log f_{s_t}^B(O_t) + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \log \pi_{s_1}$$

where $\mathcal{O}$ and S are feature and state sequences, respectively. $f_{s_t}^B(O_t)$ is the state observation probability density function (PDF), which usually assumes a GMM distribution. a_{ij} and π_i are state transition probabilities and initial probabilities which we assume to be the same for all HMMs. The discriminant functions from the randomized HMMs are then aggregated for a more reliable decision in sequence classification, which with some approximations has the form

$$\mathbf{E}_{\mathcal{L}^B}\{D_{\lambda^B}(\mathcal{O}, S)\} \approx \sum_{t=1}^{T} \log \mathbf{E}_{\mathcal{L}^B}\{f_{s_t}^B(O_t)\} + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \log \pi_{s_1}.$$

Aggregation produces an HMM whose state observation PDF is

$$f_s^A(O_t) = \mathbf{E}_{\mathcal{L}^B}\{f_s^B(O_t)\} = \sum_{i=1}^{N} \sum_{k \in \lambda_i^B(s)} c_k \mathcal{N}(O_t; \mu_k, \Sigma_k)$$

It is equivalent to an HMM with Gaussian components from all N individual randomized HMMs.

Although this aggregated HMM achieves better performance, it has a substantial number of parameters. It is therefore desirable to restructure the model to a smaller size. Restructuring is a process of Gaussian clustering followed by model refinement. The Gaussian clustering uses a greedy algorithm to merge pairs of Gaussians based on the entropy metric. After clustering, the clustered Gaussians are further refined by Monte Carlo based Kullback-Leibler minimization. In the end, feature- and model-space discriminative training with the BMMI objective is performed on the restructured HMM, just like the baseline model. The BSRS model has 5,000 quinphone states and 240K Gaussians.

2.3. Deep Neural Network Hybrid Model

The deep neural network (DNN) hybrid model uses the same speaker-adaptive (SA) feature pipeline as the GMM and BSRS models, and a set of quinphone context-dependent HMM state targets defined using standard state clustering procedures. The first stage of DNN training is a greedy, layer-wise discriminative pretraining step [5] using the cross-entropy criterion and backpropagation. Each layer is trained using one pass over the training set and keeping the weights in all preceding layers fixed. After pre-training, the weights for the softmax layer are randomly initialized, and then the entire network is trained using the cross-entropy objective function. This training process monitors performance on a held-out set to determine when to reduce the learning rate and when to terminate training [6]. Finally, the DNN model is trained with the state-level minimum Bayes risk criterion using a distributed implementation [7] of Hessian-free optimization [8], with progress monitored on the same held-out set as in the cross-entropy training. The DNN hybrid model uses 9 frames of input features (PLP+LDA+STC+FMLLR, with no pitch), contains five hidden layers with 2,048 hidden units per layer. For the hybrid model, the softmax layer has 3,000 quinphone context-dependent states, the same number of context-dependent states in the HMM model. This choice was based on prior work [9] showing that deep networks benefit from using a state alphabet as large as that used by a standard GMM/HMM system.

2.4. Deep Neural Network Features Model

In addition to the hybrid DNN model, we also built a DNN features model (DNN-fea) in which the neural network is used to compute acoustic features for a standard GMM acoustic model. The DNN architecture and training procedures are identical to those in Section 2.3, except that a smaller set of 512 context-dependent HMM state output targets is used. After training is done, we extract probabilistic DNN features from the input to the softmax layer of the DNN, using principal components analysis (PCA) to reduce the feature dimensionality from 512 to 40. We use the PCA-based approach instead of the autoencoder approach [10] because the performance of the two methods is very similar, and the PCA training is much faster. We use a smaller number of output targets because we have found in a set of unpublished experiments that the performance of probabilistic neural network features is better when the network has a relatively small number of context dependent HMM state targets, probably because the dimensionality reduction task is simpler. As in [11], nine frames of the PCA features are spliced then project to 40 dimensions using an LDA followed by a global STC transform, then a standard GMM/HMM acoustic model is built from these features. Finally, feature- and model-space discriminative training with the BMMI objective is performed, like for the baseline and BSRS models. The GMM acoustic model has the same number of states and Gaussians as the baseline system.

3. LANGUAGE MODELING

Our baseline LM is a word-based tri-gram LM with modified Kneser-Ney smoothing. We also explored using a neural network language model (NNLM) [12, 13] and Model M [14] language model. Model M is a maximum-entropy language model that uses a specific form of $L1 + L2$ regularization and word class features to achieve better generalization performance than standard n-gram language models. The NNLM achieves better smoothing by representing words in a continuous space, where the mapping into continuous space is learned such that words with similar properties are mapped to nearby locations.

4. BABEL KEYWORD SEARCH

KWS performance is measured as Term-Weighted Value [15], a function of the probability of missed detections and the probability of false alarms:

$$TWV(\theta) = 1 - [P_{Miss}(\theta) + \beta \cdot P_{FA}(\theta)]$$

where θ is the threshold used to determine a hit or a miss and $\beta = 999.9$ is a weight that accounts for the presumed prior probability of a term and the relative costs of misses and false alarms. We report keyword search performance in terms of the maximum term-weighted value (MTWV), which is an oracle metric corresponding to the best TWV for all values of the decision threshold, θ.

There are two variations of the KWS task: automatically adapted KWS (AA-KWS) and pre-indexed KWS (PI-KWS). The former allows system components (including the lexicon, acoustic models, and language models, and audio indexes) to be modified via automatic procedures after keywords are provided to developers; the latter requires that the system components and word indexes be frozen before keywords are provided to developers.

The keyword terms are split into two categories: in and out of vocabulary). The in-vocabulary (IV) terms are searched through a word index. A popular approach for handling the Out-of-Vocabulary (OOV) problem is to search sub-word lattices [16, 17, 18]. In this approach it is assumed that at query time an orthographic representation of the term can be converted to a sensible phonetic representation. This is typically done using grapheme to phoneme conversion algorithms which may not work accurately for all query terms. For Cantonese, which is an ideographic language, we used a rule based approach to generate pronunciations for OOV words. The OOV terms are searched for in a phonetic index derived from word lattices.

We use a two-pass variant of weighted finite state transducer indexing and search, where the lattice indexes (utterances) are identified in the first pass and the second pass loads the relevant lattices and extracts the time marks corresponding to the query. For more details on our indexing system please refer to [19].

5. EXPERIMENTAL RESULTS

5.1. Data and model Description

The Babel training data for each language includes both conversational and scripted telephony speech data collected using mobile and fixed telephone networks. The conversational data are free-speech conversations that approximately 10 minutes in duration, and are between two speakers, usually friends or family members, with a broad coverage of topics and vocabulary. Each of the two speakers is recorded on a separate channel which is stored in a separate signal file. Although the test data is limited to conversational data, the training package includes scripted data which are designed to achieve broad coverage of the selected language. Prompt sheets are used and are distributed to speakers. The prompts consist of text to be read (generating read speech) and questions or tasks to be answered (generating short spontaneous speech). The data collection attempts to cover a broad speaker population, and includes a variety of dialects and speaker ages, and is approximately gender-balanced. The scripted data transcripts are labeled by content (e.g., number, date, money, name or location). This is potentially useful information for word classing or other language modeling methods.

In the Babel Cantonese training set, there is 192 hours of training audio (156 hours for conversational and 36 hours for scripted data), but only 40–50% of the audio is speech. In addition, a significant portion of the audio data is labeled as non-lexical speech events (e.g., unintelligible speech, hesitations, mispronunciations, fragments, truncations and foreign words) or non-speech events (e.g., breath, cough, ring, laugh, or lip smack). The development data contains 20 hours of conversational data. Overall, the data poses a good challenge to acoustic modeling in terms of spontaneous speaking style, dialect diversity, speaker variability, environment and channel robustness, and sparse data.

5.2. Speech Recognition and Keyword Search Results

The IBM Attila toolkit [1] is used for all ASR training (both GMM and DNN models) and decoding. The toolkit provides two different decoders, a dynamic network decoder [3] and static WFST decoder. The baseline GMM/HMM and BSRS models use the static decoder, while the other models use the dynamic decoder. Combining lattices from different decoders has been shown to improve keyword search performance [20]. We report speech recognition performance in terms of character error rate (CER).

Table 1 shows that the BSRS model gives 0.6% absolute improvement over the baseline GMM/HMM, while the DNN hybrid model and DNN features model give 5.9% and 3.1% absolute CER improvement, respectively. These results are consistent with previous work. The bootstrap-and-restructuring approach was previously used for the DARPA Transtac project [4] to deal with data sparsity in the Dari and Pashto languages, and DNN-based acoustic models have been shown to outperform traditional GMM/HMM models on a variety of speech recognition benchmarks by a number of research groups [21]

5.3. Language Modeling

For Babel Cantonese language model training, only the acoustic transcripts are available, which poses a significant data sparsity challenge. The training transcripts include a total of 106K sentences and 992K words. The vocabulary contains 25K words. Scripted data is included in LM modeling because adding it improves KWS performance, even though it does not help recognition performance.

Model	CER
GMM/HMM	55.9
BSRS	55.3
DNN hybrid	50.0
DNN features	52.8

Table 1. Character error rates (CER) for four acoustic models.

Model	Rescored-CER (Baseline-CER)
GMM/HMM	54.4 (55.9)
BSRS	53.8 (55.3)
DNN	49.0 (50.0)

Table 2. CERs after lattice rescoring with interpolated NNLM and Model M

Given the small data set, a 4-gram LM has slightly worse perplexity on development data than a tri-gram LM (123.9 vs. 123.3), even with modified Kneser-Ney smoothing, so we use the tri-gram as our baseline. The out-of-vocabulary (OOV) rate on the development set is 8.4%.

A Model M LM with 150 automatically generated word classes lowers the perplexity on development data to 116. With simple processing, words in the scripted transcriptions are labeled as terms of address, numbers, dates, money, names and scripts. Integrating those labels in Model M can lower the perplexity to 117. Our NNLM is a word-based 4-gram model that uses a 30-dimensional embedding space for the words. It has 100 hidden units and predicts all words in the vocabulary. The NNLM alone doesn't provide very good perplexity (130), but when it is interpolated with Model M with 150 word classes, it can improve perplexity to 109. This interpolated LM can lower CER through either lattice re-scoring or decoding, both leading to similar improvement. Table 2 shows that the interpolated LM consistently yields 1% absolute improvement in CER over the baseline LM. Thus, we see that Model M and the NNLM both work even with sparse training data.

5.4. Lattice Analysis

The KWS system indexes lattices. Although some lattices have similar one-best path CER results, their contribution to the KWS performance can vary greatly. Table 3 shows lattice densities, keyword miss rates, and keyword search performance for different combinations of acoustic model and language model, with different levels of pruning. All results in this subsection are for pre-indexed keyword search. Scores (expected counts) for each query are normalized to sum to 1.0 [20] in order to improve KWS performance (as measured by MTWV). A detailed analysis of the effects of this normalization on Babel keyword search may be found in [22].

The keyword miss rate indicates how many of the 1000 keyword search terms from the August 2012 Babel dry run evaluation are not present the lattices. MTWV scores are evaluated on the Babel development test set with a list of 389 queries produced by the Babelon and RADICAL teams. Of the 389 development terms, 361 are searched for using in-vocabulary lexical search, while the remaining 28 are searched for using OOV phonetic search.

In Table 3, lattices generated by static decoding are marked (S), while lattices generated by dynamic decoding are marked (D). We experimented with both static decoding and dynamic decoding (Table 3, rows 3 and 4) when using DNNs. Even though both lattices have the same CER and similar densities, the lattices generated by

Model	lattice density	Miss Rate	MTWV
GMM/HMM(S)	678	0.176	0.335
BSRS(S)	691	0.171	0.359
DNN(S)	575	0.181	0.401
DNN_S(D)	415	0.198	0.427
DNN_M(D)	712	0.191	0.431
DNN_L(D)	2876	0.177	0.440
DNN_L(D) advLM	1224	0.183	0.441
DNN-fea_S(D)	611	0.193	0.384

Table 3. Comparing various lattices in terms of lattice density (arcs per second of audio), query miss rate on the dry run data, and keyword search performance on the development data.

the dynamic decoder lead to better KWS performance (0.4270 vs. 0.4012 MTWV). The dynamic decoder offers more fine-grained control over various pruning parameters, allowing us to tune it to produce more diverse lattices (that is, lattices with containing a larger number of word types at a given lattice density). Such diversity appears to improve keyword search performance.

For the same DNN model with the baseline LM, three lattice sets are generated with different sizes: small ($\_S$), medium ($\_M$), and large ($\_L$). Table 3 shows that deeper lattices provide additional improvements in MTWV; however, beyond a certain size, no further improvement is seen.

DNN advLM is generated by the interpolated NNLM and Model M. Even though its MTWV score is almost the same as the ones obtained with the best DNN and the baseline LM (Row DNN_L(D) in Table 3), when using the same lattice generation parameters, the lattice size of DNN advLM is much smaller than that obtained from the baseline LM (1224 vs. 2876). Thus, although the advanced LMs do not improve keyword search performance, they do reduce the size of the index significantly.

When compared to the baseline GMM/HMM model, the BSRS model improves MTWV from 0.3352 to 0.3594. The DNN model produces similar-sized lattices, but much better MTWV performance (0.3842 vs. 0.3352). The best performance comes from the DNN advLM, which is 0.4407, an improvement of 31% over the baseline system.

5.5. Automatic Adaptation

We implemented a simple version of automatically adapted keyword search and evaluated it on the Cantonese development data in terms of MTWV. In this automatically adapted system, all query terms (single and multiword) are added to the language model training data as individual utterances. Each query term is added once as a sentence. While this is a very simple method for adding the queries to the language model, it has the advantage of ensuring that n-grams from new multi-word queries appear in the language model. This extension of the language model ensures that the new queries are more likely to be included in the decoding lattices, especially when they really occur in the reference. Thus, all OOV words from the query terms were added to the lexicon. The audio to be searched is subsequently decoded using the new lexicon and LM. While this method for adding query terms to the LM is extremely simple, it has the advantage of ensuring that all the query terms are covered by high-order n-grams in the LM.

The results of this experiment are summarized below in Table 4. The acoustic model used for indexing is the DNN model mentioned in above sections. The lattices are generated using the dynamic decoder. We observe a 16% improvement in MTWV, which indicates that there is a significant advantage for in-vocabulary queries, even when they are added in a very simple manner.

Model	CER	lat-density	MTWV
PI-KWS	50	415	0.427
AA-KWS	49.8	417	0.495

Table 4. Comparison of pre-indexed (PI) and automatically adapted (AA) KWS with the DNN acoustic model.

6. CONCLUSIONS

In this paper, we present four different ASR systems using diverse acoustic models, and measure their performance on a Cantonese keyword search task. We observe that a deep neural network model can improve not only transcription accuracy, measured by character error rate, but also that it greatly improves keyword search performance. Rescoring lattices with an advanced LM that interpolates a Model M LM and neural network LM improves transcription accuracy, but not keyword search performance. This is because rescoring does not introduce any new words to the lattice; instead, it simply modifies the scores of existing words in the lattice. However, the rescoring does produce a smaller index with no loss in keyword search performance. Different lattice generation methods may produce the same CER but can yield very different keyword search results.

7. ACKNOWLEDGMENTS

We are grateful to Hong-Kwang Kuo, Ebru Arisoy and Hagen Soltau of IBM Research for sharing their rich experience in language modeling and system building. This effort uses the IARPA Babel Program Cantonese language collection release babel101b-v0.4c. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

8. REFERENCES

[1] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010.

[2] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1596 – 1608, 2006.

[3] H. Soltau and G. Saon, "Dynamic network decoding revisited," in *Proc. ASRU*, 2009.

[4] X. Cui, J. Xue, X. Chen, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, J. R. Hershey, and B. Zhou, "Hidden Markov

acoustic modeling with bootstrap and restructuring for low-resourced languages," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 20, no. 8, pp. 2252–2264, 2012.

[5] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011.

[6] H. Bourlard and N. Morgan, "A continuous speech recognition system embedding MLP into HMM," in *Advanced in Neural Information Processing Systems 2*, D. S. Touretzky, Ed., 1990, pp. 186–193.

[7] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012.

[8] J. Martens, "Deep learning via Hessian-free optimization," in *Proc. Intl. Conf. on Machine Learning (ICML)*, 2010.

[9] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novk, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011, pp. 30–35.

[10] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP*, 2012.

[11] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. ICASSP*, 2007.

[12] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Proc. Neural Information Processing Systems (NIPS)*, 2000.

[13] H. Kuo, E. Arisoy, A. Emami, and P. Vozila, "Large scale hierarchical neural network language models," in *Proc. Interspeech*, 2012.

[14] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, "Scaling shrinkage-based language models," in *Proceedings of ASRU*, 2009.

[15] J. G. Fiscus, J. G. Ajot, J. Garofalo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2007, pp. 51–57.

[16] Dogan Can, Erica Cooper, Abhinav Sethy, Chris White, Bhuvana Ramabhadran, and Murat Saraclar, "Effect of pronounciations on OOV queries in spoken term detection," *Proceedings of ICASSP*, 2009.

[17] Murat Saraclar and Richard W. Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, 2004.

[18] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *Proceedings of SIGIR*, 2007.

[19] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *ASRU*, 2009.

[20] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proc. ICASSP*, 2013. To appear.

[21] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[22] J. Mamou, J. Cui, X. Cui, M. J. F. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. ICASSP*, 2013. To appear.