HIGH-ACCURACY PHONE RECOGNITION BY COMBINING HIGH-PERFORMANCE LATTICE GENERATION AND KNOWLEDGE BASED RESCORING

Sabato Marco Siniscalchi[1], *Petr Schwarz*[2], *and Chin-Hui Lee*[1]

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology
[2]Speech@FIT group, Brno University of Technology
{marco, chl}@ece.gatech.edu, {schwarzp}@fit.vutbr.cz

ABSTRACT

This study is a result of a collaboration project between two groups, one from Brno University of Technology and the other from Georgia Institute of Technology (GT). Recently the Brno recognizer is known to outperform many state-of-the-art systems on phone recognition, while the GT knowledge-based lattice rescoring module has been shown to improve system performance on a number of speech recognition tasks. We believe a combination of the two system results in high-accuracy phone recognition. To integrate the two very different modules, we modify Brno's phone recognizer into a phone lattice hypothesizer to produce high-quality phone lattices, and feed them directly into the knowledge-based module to rescore the lattices. We test the combined system on the TIMIT continuous phone recognition task without retraining the individual subsystems, and we observe that the phone error rate was effectively reduced to 19.78% from 24.41% produced by the Brno phone recognizer. To the best of the authors' knowledge this result represents the lowest ever error rate reported on the TIMIT continuous phone recognition task.

Index Terms— Knowledge based system, speech recognition, hidden Markov models, neural networks.

1. INTRODUCTION

Recently, system combination has been shown to be a promising technique to improve the accuracy of conventional automatic speech recognition (ASR) systems. In this area of research, the main idea is to generate a confusion network by multiple string alignment. Then a voting scheme is performed in order to find the best hypotheses [1]. Usually the achieved improvement depends upon whether the individual systems have similar performance and are complementary in the errors they produce. Moreover, combining several systems together is not always a straightforward operation because the systems may be originally incompatible. In [2] we propose knowledge-based lattice rescoring as a way to overcome these difficulties.

Lattice rescoring is a well-known technique to improve ASR system performance by integrating multiple sources of knowledge. It is typically accomplished with multi-stage decoding. In particular, an ASR decoder first generates a collection of competing hypotheses. It is then followed by a rescoring algorithm to re-rank these hypotheses by incorporating additional information not used in the decoding process. More detail can be found in [3] [4] [5]. We have evaluated our knowledge-based lattice rescoring algorithm in several speech recognition applications, and shown that it outperforms conventional speech recognizers without rescoring in all of these cases. The success of our approach relies on the design of a bank of speech attribute detectors which capture articulatory information, such as manner and place of articulation. Nonetheless, the rescoring performance is often limited by the quality of the lattice. This quality is mainly associated with the goodness of the lattice segmentation which can be defined in terms of: (1) the number of word errors in the *N*-top competing lists embedded in the lattice itself, and (2) the precision of the detected word boundaries.

We believe that a better lattice will further enhance the system performance through knowledge-based rescoring. Thus we look for a way to improve the lattice quality to verify our conjecture. We found from recent studies [6] [7] that the system from Brno University of Technology seems to achieve higher performance phone recognition than most state-of-the-art hidden Markov model (HMM) based systems. Therefore, we wondered if this system could also yield high-performance lattices so that we could produce even higher phone accuracy to break today's performance limits by combining the Brno phone recognition subsystem and our knowledge-based rescoring module. Some interface and lattice generation issues need to be addressed first in order to combine the two seemingly incompatible subsystems into a cohesive system.

We then evaluate this combined system on the TIMIT continuous phone recognition task [8]. With this system combination we observed that the phone error rate (PER) was effectively reduced to 21.49% from 24.41% obtained with the original Brno phone recognizer [7]. The error rate can be further reduced to 19.78% if more emphasis is placed on the knowledge scores. We believe this result represents the lowest PER reported in the literature on the TIMIT phone recognition task.

The rest of the paper is organized as follows. The Brno system and the GT knowledge-based rescoring module are described in detail in the next section. In Section 3 the experimental setup is described, and the results are presented. Finally, we draw our conclusions in Section 4.

2. THE OVERALL PHONE RECOGNITION SYSTEM

As explained earlier the overall system consists of two main parts from two different groups: (1) Brno phone recognizer, and (2) GT knowledge module. The former is a well crafted fusion of artificial neural networks (ANNs) and hidden Markov Models (HMMs). The latter is the combination of a bank of speech attribute detectors and an ANN, and it represents the core of the knowledge-based rescoring procedure. Both these modules will be presented in the following. More details can be found in [7] and [2], respectively.

2.1. Brno Phone Recognizer

The Brno recognizer [9], shown in Figure 1, is a hybrid system with two key components: (1) a non-conventional front-end module based

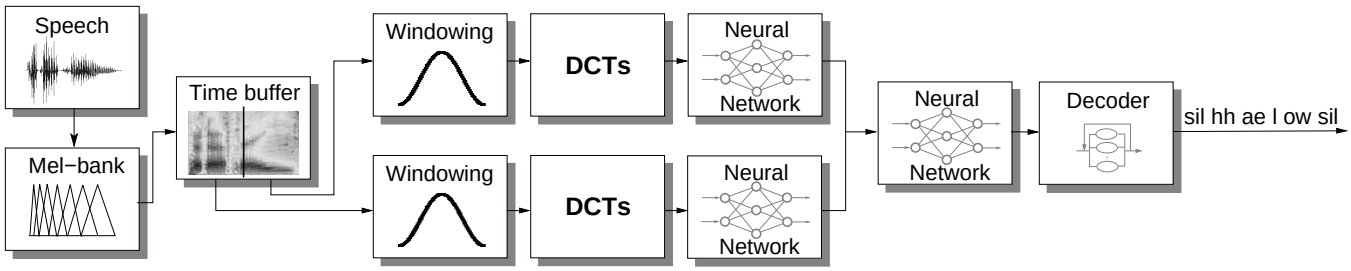

Fig. 1. Phoneme recognizer with split temporal context and three neural nets.

on a hierarchical structure of ANNs, and (2) a Viterbi decoder. The main assumption behind the Brno architecture is that information about phones is spread out in a long temporal context (more than 300 ms). In this case, input vectors for the ANN classifier are huge in size and there are too many patterns for each phone. The ANN needs many parameters which are difficult to estimate, and a lot of training data is necessary. Therefore, a hierarchical structure of ANNs was used [10] [7]. The structure incorporates intuitively obvious knowledge. The most pertinent information for classification of a phone is placed close to the center of the phone segment. Parts further away from the center are considered less important. Neighboring phones affect the actual phone too (co-articulation effect). Features describing different edges of a phone (or different contexts) are more independent than features describing the central part of the phone. The front-end generates phone-state posteriors as follows. First, mel filter bank energies are obtained in a conventional way. Temporal evolutions of critical band spectral densities are taken around each frame. A context of 31 frames (310 ms) around the current frame is used. This context is split into 2 halves: left and right contexts. The underlying idea is to divide the trajectory of a phone in the feature space in two blocks with the assumption that both parts of a phone can be processed independently. Each of these contexts is then processed by a separated ANN which yields phone-state posteriors. We refer to these ANNs as the lower nets. Usually, windowing and discrete cosine transform (DCT) are performed to the input of these lower nets to reduce the dimensionality. The outputs of the two lower nets are then merged by another ANN upper net trained to produce phone-state posteriors, as well.

2.2. Phone Lattice Hypothesizer

The Brno recognizer does not generate a lattice directly. Instead the HVite decoder [11] is used. Since HVite takes only a Gaussian Mixture Model in each state, it does not support direct decoding from posteriors produced by the ANN module in the Brno phone recognizer. To circumvent this problem, we convert the posteriors in likelihoods using the Bayes' formula

$$P(o_t|ph_{j,t}) = P(ph_{j,t}|o_t)/(P(ph_{j,t}) * P(o_t)); \qquad (1)$$

where o_t is the acoustic observation vector at time t, $ph_{j,t}$ is the j-th phone at time t, $P(ph_{j,t})$ is the prior probability of the phone j-th phone at time t, $P(o_t|ph_{j,t})$ is the likelihood of the observation vector o_t at time t given $ph_{j,t}$, $P(ph_{j,t}|o_t)$ is the posterior probability of j-th phone $ph_{j,t}$ at time t given the observation o_t, and $P(o_t)$ is the probability of the observation vector o_t which we will drop from this point on because it does not affect our decoding results. We assume that the prior probabilities of all phones are equal for decoding purposes. Then HVite in HTK can use them to generate the desired lattices. In order to avoid that an additive constant could bias the posteriors, the parameter GCONST in the HTK model is set equal to zero.

2.3. Rescoring Knowledge Module

The knowledge module [2] has two main blocks: (1) a bank of 15 manner and place of articulation detectors, and (2) an ANN. The bank of detectors is implemented with HMMs, and it maps a segment of speech into one of the 15 broad classes, namely fricative, vowel, stop, nasal, semi-vowel, low, mid, high, labial, coronal, dental, velar, glottal, retroflex, and silence. Based on our previous work in lattice rescoring, log-likelihood ratio (LLR) at a frame level is taken as the measure of goodness-to-fit between the input and the output of each detector. A feed-forward ANN is trained to produce phone scores for each set of LLR score. The j-th ANN output can be thought of as an estimate of the posterior probability of Ph_j at time t, $p_t(Ph_j|LLR_i(o_t))$ ($j = 1, ..., P$; where P is the total number of phones). The scores at the phone level are then used in the rescoring phase. In particular, the rescoring is done on an arc by arc basis, and it is a weighted sum between the log-likelihood score and the knowledge-based scores. Each arc in a lattice correspond to a phone in a string hypotheses. If we denote the rescored log-likelihood value as S_n for the given arc, the rescoring formula is

$$S_n = w_{kb}\, PS_n + w_l\, L_n \qquad (2)$$

where L_n is the log-likelihood of the *n*-th arc; PS_n is a linear combination of $PS_{n,m}$ for each arc, with $PS_{n,m}$ being a non linear transformation of the score of the *m*-th frame for the *n*-th arc; w_{kn}, and w_l are the weights of the log-likelihood score and the knowledge-based score, respectively.

3. THE PHONEME RECOGNITION TASK

Designing a high-performance phone recognition system with two very different subsystem developed by various researchers over a long period of time is a challenging exercise by itself. In the following we show that through a series of experiments we manage to achieve a very low phone error rate of 19.78% without retraining either the subsystems.

3.1. Experimental Setup

Databases: The TIMIT corpus was chosen for all experiments. The *SA* part of the TIMIT database was not used. The database was divided into three parts: training (412 speakers), cross-validation (CV – 50 speakers), and test (168 speakers) sets. The training and CV subsets are specified in the original TIMIT training set.

Phone set: The phone set consists of 39 phones. It is very similar to the CMU/MIT phone set [12], but closures were merged with burst instead of with silence (bcl b $\rightarrow$ b).

Brno evaluation criteria: Brno ANNs were trained on the training part of the database. The increment in classification error on the cross-validation part during training was used as a stopping criterion to avoid over-training. There is one ad hoc parameter in the system, the word (phone) insertion penalty, which has to be set. This constant was tuned to minimize phoneme error rate on the cross-validation set. The number of neurons in hidden layer of neural networks was increased until the saturation of phoneme error rate (PER) was observed. The obtained number of hidden layer nodes was approximately 500. All experiments reported in this paper use this number of hidden layer nodes unless stated otherwise.

Brno recognizer training: All Brno ANNs were trained using the classical back-propagation algorithm with cross-entropy error function [13]. Several iterations of training of the whole system followed by realignment of labels were done. For multi-state systems, the algorithm started with a uniform segmentation of phone into states. Then, the networks were trained, state posteriors were generated and these posteriors were used in the classical Baum-Welch algorithm to produce new labels. The algorithm creates hard labels – one label per frame. The label corresponds to a state with the highest state occupation probability. These new labels are used in the following iteration of ANN training.

The knowledge module: The bank of detectors and the follow-up ANN are trained on the training subset. The competing model of each HMM-based detector is trained on all data that do not correspond to the target model, for instance, all of the "non-nasal" sounds are used to estimate the parameters of the "non-nasal" HMM. Moreover the LLR scores at a frame level for both the target and the competing model were computed on the same state sequence. Each HMM has 3 states with 32 Gaussian mixture components per state. The ANN is a feed-forward multi-layer perceptron with one hidden layer of 100 hidden nodes. Increase of classification error on the CV development subset was used as a stopping criterion for ANN training.

3.2. Phone Recognition Results

A word (phone) lattice generated by a set of HMMs is usually implemented as a direct, acyclic, weighed graph *G(N, A)*. *N* is the number of nodes, and *A* the number of arcs. The timing information is embedded into the nodes; whereas the arcs carry the symbol along with the score information. As we mentioned, the effectiveness of our knowledge-based rescoring technique depends upon the quality of the lattices. To further investigate this conjecture, we compared the phone lattices of three phoneme recognizers. Toward this end, we designed three phoneme recognizers. The first system was the Brno recognizer. The second system was a conventional context-independent recognizer with 3 states per HMM and 16 gaussian mixtures per state. We refer to this system as GMM/CI-HMM. The third system was a conventional context-dependent recognizer, with 3 states per model, 8 gaussian mixture per state. We termed this system as GMM/CD-HMM. The latter two recognizers were designed by HTK, and the number of gaussian mixtures in the hidden states was increased until the saturation of PER on the CV development subset was observed. The feature vector has 39 components: 12 cepstral coefficients plus energy and their first and second time derivatives. Table 1 lists the performances of these three recognizers in terms of PER percentage.

Recognizer	GMM/CI-HMM	GMM/CD-HMM	Brno
PER (in %)	37.27	32.54	24.42

Table 1. *Benchmark comparison*

With a PER of 24.42% the Brno system yielded the best performance. The relative improvement was about 34.5% over the GMM/CI-HMM recognizer, and of about 25% over the GMM/CD-HMM recognizer. In the second case, the improvement became more significant if we consider that the Brno recognizer uses context-independent phoneme models; whereas, the GMM/CD-HMM is based on context-dependent phone models.

In order to compare the quality of the lattices generated by these recognizers we used an indirect method. We considered all the lattices associated with the GMM/CD-HMM, and the Brno recognizer. For each lattice, we kept its segmentation, but drop the acoustic scores of each arc. Then a new acoustic score is computed by force-aligning each arc of the lattice with the set of HMM models provided with the GMM/CI-HMM recognizer. Finally, the 1-best list was found with the HVite routine modules. Since the acoustic scores were computed by using the same set of CI models, and the language model was a fixed 0-gram for all the systems, the performance of each recognizer depends only upon the quality of the lattices. Therefore, the recognizer with the best performance in terms of PER is the one that produces the highest quality lattices. Table 2 shows the results of this experiment. Brno recognizer again yields the lowest PER, and the highest-quality lattices.

Recognizer	PER (in %)
GMM/CI-HMM	37.27
GMM/CD-HMM	39.23
Brno	35.41

Table 2. *Lattice quality.*

We now present knowledge based rescoring over the Brno lattices. In the following, the PER of this system is referred as baselines. In the first experiment, the language model was set to be a 0-gram, and in the second experiment a bigram model was used. In both cases we did not perform tuning of the weights. Thus w_{kb}, and w_l are set to be equal. In Table 3 the results are given. As expected, the knowledge-based rescoring lowers the PER in both cases. In particular, we reduce the PER from 24.41% to 21.49% using a 0-gram phone model, and from 23.84% to 20.96% with a phone bigram model. In average we observed an improvement of about 12.5% over the baseline.

PER (in%)	0-gram	bi-gram
Brno recognizer	24.41	23.84
Rescoring ($w_{ps} = w_l$)	21.49	20.96

Table 3. *Rescoring performance in terms of PER.*

In [7] it was shown that the Brno recognizer can achieve a PER of 21.48% on TIMIT. Although this result equals the rescoring performance reported in Table 3 for the 0-gram case, it was obtained by using a more involved system configuration . In the scheme shown in Figure 1, the number of ANNs was increased from 3 to 5 (four

ANNs for the lower net and one ANN for the upper net). The number of hidden nodes in each ANNs was increased from 500 to 800 for each ANN. The dimension of the training set was augmented by adding the CV development subset to the training subset. The number of training epochs was also increased. Finally, a bigram phone model was used.

In order to study the effectiveness of the knowledge-based scores on the overall phone recognition accuracy, we incrementally placed more and more emphasis, while keeping w_l fixed. It can be seen in Figure 2 that the PER consistently decreases for the first three steps, and then begins to increase. This pattern was also observed on the CV development subset. The best performance was achieved after three lattice rescoring steps for both the 0-gram and bigram phone models. For the case of using phone bigram models the PER was 19.78%. To the best of our knowledge, this result represents the lowest accuracy phone error rate on the TIMIT corpus.

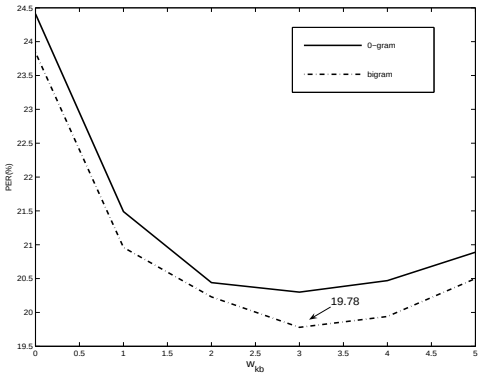

Fig. 2. *PER for different values of* w_{kb}.

4. SUMMARY AND DISCUSSION

In this paper we have combined the Brno phone recognizer and the GT knowledge scoring module into a cohesive architecture in order to design a high-accuracy phone recognizer. After having addressed some compatibility issues, several studies have been presented to assess the quality of the Brno lattices and the benefit of the knowledge-based lattice rescoring technique. We reported a PER as low as 19.78% on the TIMIT database.

We have also combined the acoustic scores of the GMM/CI-HMM and the Brno scores together. We performed this combination over the hypotheses space provided with the Brno lattices. Although we observed slight improvements, they were not significant. We have also observed one interesting fact that the knowledge-based score we adopted has a high discriminative power. This is demonstrated by replacing the GMM/CI-HMM acoustic likelihood with the knowledge-based score while keeping the Brno lattice and setting w_l equal to zero. We obtained a PER of 27.55% which is much better than all the results listed in Table 2. It turns out that only 15 broad classes are used in deriving knowledge scores. We believe this can be further explored to improve system performance in future studies.

We believe a high-quality phone lattice can be used for many new applications, including detection-based speech recognition currently being studied in the NSF-ASAT project [14]. In the future we intend to investigate other means of system combination to improve the quality of lattice generation and knowledge rescoring.

5. ACKNOWLEDGMENT

This study was partially supported by an IBM Faculty award, and we wish to gratefully acknowledge this support. The authors would like to thank Prof. Jen-Tzung Chien from National Cheng Kung University, Tainan, Taiwan for stimulating discussions.

6. REFERENCES

[1] J.G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)," in *Proc. of IEEE ASRU Workshop*, pp. 347-352, 1997.

[2] S. M. Siniscalchi, J. Li, and C.-H. Lee, "A Study on Lattice Rescoring with Knowledge Scores for Automatic Speech Recognition," in *Proc. of InterSpeech'06*, pp. 517-520, September 2006.

[3] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 11(1), pp. 43-72, January 1977.

[4] V. Goel, and W. Byrne, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, pp.373-400, October 2000.

[5] F. Wessel, R. Schter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 288-298, March 2001.

[6] P. Matějaka, Petr Schwarz, and Jan Černocký, "Brno University of Technology System for NIST 2005 Language Recognition Evaluation," in *Proc. of 2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, June 2006.

[7] P. Petr Schwarz, Matějaka, and Jan Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. of ICASSP'06*, pp. 325-328, May 2006.

[8] J.S., Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L.Dahlgren, "DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," U.S. Dept. of Commerce, NIST, Gaithersburg, MD, February 1993.

[9] P. Schwarz, P. Matějka, L. Burget, and O. Glembek, "Phoneme recognizer Based on Long Temporal Context," March 2006, [Online] available from: *http://www.fit.vutbr.cz/speech/index.php?id=phnrec*, [accessed September 29, 2006].

[10] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in Proc. of TSD2004, pp. 465-472, September 2004.

[11] S. Young et al., *The HTK Book*, Cambridge University Engineering Department, 2005.

[12] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37(11), pp. 1641-1648, November 1989.

[13] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, second ed. New York: Wiley, 2001.

[14] C.-H.Lee, "From knowledge-ignorant to knowledge-rich modeling: a new speech research paradigm for next generation automatic speech recognition", in *Proc. of ICSLP'04*, September 2004