CONSTRUCTING ENSEMBLES OF ASR SYSTEMS USING RANDOMIZED DECISION TREES

Olivier Siohan Bhuvana Ramabhadran Brian Kingsbury

IBM T.J. Watson Research Center
1101 Kitchawan Rd., Rte 134/PO BOX 218
Yorktown Heights, NY 10598, USA
{siohan,bhuvana,bedk}@us.ibm.com

ABSTRACT

Building multiple automatic speech recognition (ASR) systems and combining their outputs using voting techniques such as ROVER is an effective technique for lowering the overall word error rate. A successful system combination approach requires the construction of multiple systems with complementary errors, or the combination will not outperform any of the individual systems. In general, this is achieved empirically, for example by building systems on different input features. In this paper, we present a systematic approach for building multiple ASR systems in which the decision tree state-tying procedure that is used to specify context-dependent acoustic models is randomized. Experiments carried out on two large vocabulary recognition tasks, MALACH and DARPA EARS, illustrate the effectiveness of the approach.

1. INTRODUCTION

In the past several years the Machine Learning community has been extensively studying and advocating the use of an ensemble of classifiers as an alternative approach to designing a single strong classifier, both for practical and theoretical reasons [1]. An ensemble of classifiers is a set of classifiers whose decisions are combined, typically through some form of voting, to classify new examples. Obviously, the ensemble will only be more accurate than its individual components if these individual classifiers disagree with one another [2].

Some of the most successful techniques for constructing classifier ensembles manipulate the training data to build classifiers on different subsets of the data. For example, Breiman's bagging approach builds several classifiers on different, randomly selected subsets of the training data [3]. Another technique, called Adaboost, sequentially builds an ensemble of classifiers, with each individual classifier being trained on a re-weighted version of the training data, placing increasing weight on the training examples that were misclassified by the previous classifiers [4]. Such sub-sampling techniques have been extensively studied in the machine learning literature and are very effective when used with "unstable" learning algorithms: classifiers that vary significantly for small changes in the training data, such as neural networks or decision trees.

In the speech recognition community, combining the output of several ASR systems has been extremely popular within the context of DARPA evaluations, since Jon Fiscus, from NIST, introduced the ROVER procedure [5]. ROVER combines the recognition results of several ASR systems and derives a single recognition hypothesis through some form of majority voting. Such a system combination procedure has been shown to be very effective in bringing the word error rate down. A key challenge is to design multiple ASR systems that exhibit different error patterns, so that a majority voting procedure can be successfully applied.

One could think of designing multiple systems through a procedure that directly minimizes the correlation between their respective errors, as in [6]. This is, however, too complex an approach to be applied on the Hidden Markov Model (HMM) classifiers used in modern ASR systems. Instead, there have been several attempts to design HMM-based ASR systems following ideas related to boosting. For example, on small tasks, such as digit recognition, Adaboost was used to build a hybrid Neural-Network/HMM-based classifier that led to better word error rates [7]. More recently, in [8], Adaboost was reformulated for HMM-based classifiers and applied to digit recognition. For large vocabulary recognition, Zweig suggested "boosting" the Gaussian mixtures by applying Adaboost to improve frame classification [9]. Unfortunately, the procedure did not lead to a significant improvement of the word error rate, and up to now the use of boosting has been rather unsuccessful in designing large vocabulary continuous speech recognition (LVCSR) systems.

In this paper we follow another approach advocated in the Machine Learning community, based on building multiple classifiers by introducing randomness in the classifier learning process [10]. In the next section we describe the proposed approach. Then, in Section 3, we apply that technique on two large vocabulary recognition tasks and illustrate its effectiveness. The first task is the NSF-funded MALACH project [11] involving recognition of testimonies of Holocaust survivors, where we built multiple LVCSR systems on about 65 hours of training data. The second one is the DARPA EARS project, where multiple LVCSR systems were trained on 2100 hours of training data. Section 4 concludes the paper.

2. PRINCIPLE

Building multiple classifiers by randomizing the learning procedure is an indirect way of building classifiers that hopefully will have uncorrelated errors. Such techniques are especially effective with *unstable* classifiers where small changes caused by randomization lead to major changes in the classifier itself.

Randomness can be introduced at different levels of the learning procedure. For example, in bagging, random subsets of the training data are selected and individual classifiers are trained on each of these subsets [3]. In [12], random subspaces of the feature space are selected and classifiers are trained on these subspaces. In [13], decision trees are grown by randomly selecting the split from among the top-N best splits. An idea common to all these papers is that an ensemble of trees is grown, and then voting is used to do classification. Such a learning procedure has been unified under the "Random Forest" name [14], and detailed theoretical error analyses have been carried out illustrating the effectiveness of such approaches.

Continuous density HMM-based classifiers, which are commonly used in speech recognition, are rather stable classifiers. Because of this, a simple replication of the bagging procedure in which

multiple systems are trained on randomly selected subsets of the training data and then combined usually does not lead to improved performance. However, modern speech recognition systems do include an unstable learner, namely the decision trees that are typically used to tie context-dependent acoustic units [15]. As the number of possible context-dependent units is large, the decision tree state-tying provides a data-driven way to cluster "similar" acoustic contexts, so that the Gaussian mixtures in the corresponding HMM states can be reliably estimated. The decision tree state-tying procedure used in speech recognition typically selects splits that maximize the likelihood of the data [16], using a procedure similar to CART [17]. This is the optimization criterion that is used in our experiments to grow decision trees.

In this paper, we suggest growing the decision trees for state-tying *by randomly selecting the split at each node, from the top-N best splits*. This contrasts with the baseline approach, which selects the best split. In essence, the proposed approach is a direct application of the randomized decision tree procedure introduced by Dietterich [13] for the purpose of clustering the context-dependent units, and is therefore dubbed "Randomized decision tree state-tying". ASR systems built on different sets of randomized decision trees will model different clusters of context-dependent units. This may appear as a rather ad-hoc and indirect way of building multiple systems, but the performance of the approach will be illustrated in Section 3.

We implemented this approach by modifying the procedure that selects the candidate split for each node to randomly choose among the top-N best splits instead of always selecting the best split as in the baseline tree-growing strategy. The randomized decision tree state-tying algorithm is controlled by two additional parameters compared to the standard procedure: N, which specifies the top-N best splits, and a seed that controls the random number generator. In all our experiments, all other parameters used to control the growth of the tree were left unchanged.

Multiple ASR systems can then be systematically built by selecting different seeds for the randomized decision tree state-tying procedure. Note that the procedure that is commonly followed to build an entire ASR system is unchanged, except for the decision tree part, where the random decision trees are used instead of the baseline decision trees. Recognition experiments are carried out by running all the different ASR systems, and combining their recognition hypotheses using ROVER.

3. EXPERIMENTS AND RESULTS

3.1. Databases and Setup

Experiments were carried out on two large vocabulary tasks. The first is the MALACH project, an NSF-funded research program related to the development of multilingual access to large audio archives [11]. The archive of interest is a large collection of testimonies from survivors, liberators, rescuers and witnesses of the Nazi Holocaust, assembled by the Shoah Visual History Foundation. Our experiments were conducted on the English subset of the MALACH corpus, consisting of 65 hours of training data, and 2 hours of test data. The setup was similar to the one described in the English acoustic modeling section in [11]. For the baseline system, a set of speaker adaptively trained (SAT) [18] triphone models were trained using speaker-dependent fMLLR transforms, leading to a total of 3.2K tied-states and 74K Gaussian mixtures. Then, multiple randomized-tree systems were built on the same SAT features used to build the baseline (i.e., the SAT transform were not re-estimated), selecting random splits either from the top-5 or top-10 best splits. Note that the size of the random tree systems was typically very similar to the size of the baseline system, and that the random tree based systems were not tuned in any way but were instead built systematically without any supervision.

The second task is the DARPA EARS project. We used 2100 hours of training data provided for the 2004 EARS evaluation.[1] A baseline quinphone-context SAT system was built using an early version of the system described in [19], for a total of 7.5K tied-states and 242K Gaussian mixtures. Multiple recognizers were systematically built using randomized trees on the SAT features used to build the baseline, using either the top-10 or top-20 best splits. Again, the size of the random tree based systems were similar to the baseline. Experiments were run on various Switchboard evaluation test sets.

3.2. MALACH Results

We first built a set of 10 random systems, and rovered them together in the order they were built, without including the baseline system. In all our ROVER experiments, an additional "empty" system with empty transcriptions was included last in the list of systems to combine and ROVER was invoked with the "maxconfa" option. The results are presented in Figure 1 in terms of word error rates (WER), first for the baseline system (WER=45.6%), then for each of the individual random systems (WER ranging from 45.9% to 46.7%), and last for the ROVERed systems, using 2 to all available random systems. Remarkably, rovering 3 of the random systems together outperformed the carefully designed baseline system, illustrating that the systems built on random trees were making different errors. When 5 or more of the random systems are combined with ROVER, the result is significantly better than the baseline, where we define significance as the probability of improvement over the baseline exceeding 99%, measured using an utterance-wise bootstrap estimation procedure [20] with 10,000 replications.

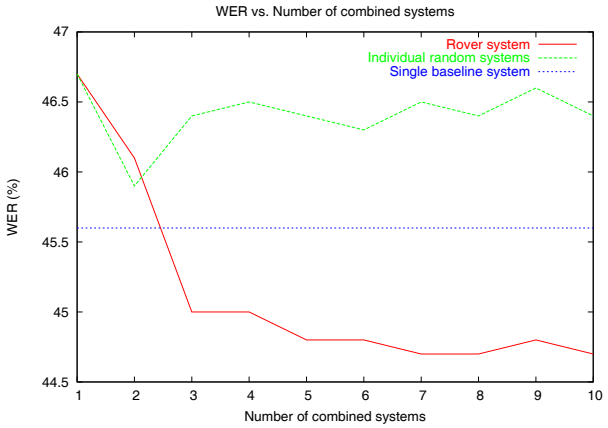


Fig. 1. WER for baseline system, individual random-tree based systems (top-5 best splits), ROVERed random-tree based systems.

Next, in Figure 2, the same systems were rovered, this time including the baseline system. When the baseline system is combined with 2 or more randomized systems, the combination significantly outperforms the baseline alone.

We then built a new set of 20 random systems, selecting the random split from the top-10 best splits instead of the top-5 as in the previous experiments. Results are given in Figure 3 where we plotted both the previous results based on N=5 and the new results based

[1] A total of 2300 hours of data were provided for the evaluation. 2100 hours is what remained following resegmentation and normalization of the data.

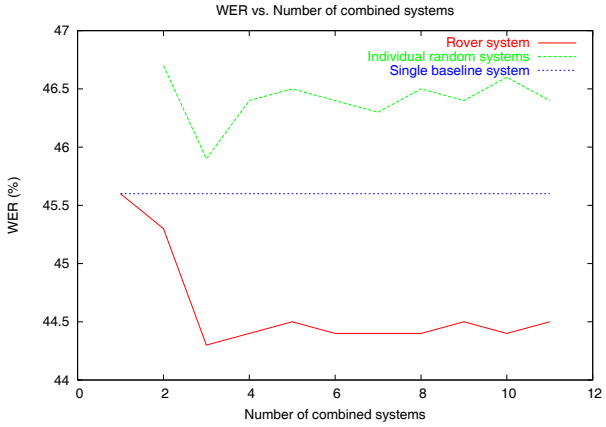


Fig. 2. WER for baseline system, individual random-tree based systems (top-5 best splits), ROVERed baseline and random-tree based systems.

on N=10. Clearly, using N=10 leads to individual random systems that are worse than with N=5. However, rovering the top-10 based random systems with the baseline clearly outperforms rovering the top-5 based random systems. This departs from the current practice of building multiple systems on different feature sets, where each individual system is highly tuned to the best performance. Our results illustrate that the nature of the errors made by the individual systems to be combined matters more than the absolute performance of these individual systems. It is also worth noting that after combining 21 systems together (baseline + 20 random), the performance saturates to a plateau, even though all of the individual systems perform significantly worse than the baseline. As before, a combination of the baseline and 2 or more of the randomized systems significantly outperforms the baseline alone.

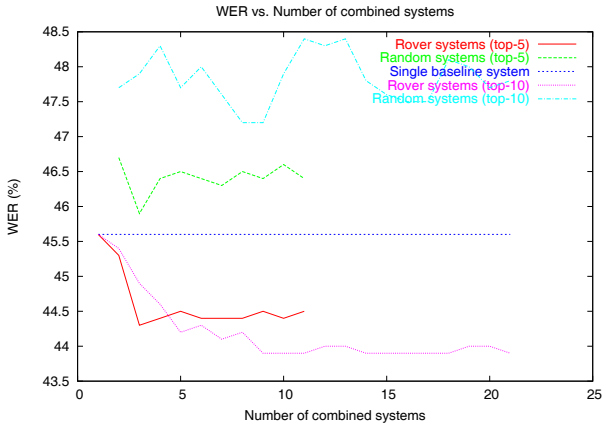


Fig. 3. WER for baseline system, individual random-tree based systems (top-5 and top-10 best splits), ROVERed baseline and random-tree based systems (top-5 and top-10).

3.3. EARS Results

Recognition experiments were carried out on the CallHome'98, Switchboard'00, RT'02 and RT'03 evaluation sets. In addition to our baseline system, we first built a set of 4 random tree systems by randomly selecting the split from the top-10 best splits (N=10). With such a setting, the performance of each random system was quite close to the baseline, so in accordance to what was learned on the MALACH test set, we built an additional set of 5 random systems using N=20. These 9 systems were then rovered with the baseline, and systems were ordered by increasing word error rates when running rover (while in the MALACH setup, no specific ordering to the systems was used when running ROVER). Experimental results are given in Figure 4 for all evaluation sets. Across all evaluation sets, between 0.7% and 1.1% absolute reduction in WER is obtained. Also, across all tests we observe that the combination of the baseline system and 2 or more randomized systems significantly outperforms the baseline alone.

4. DISCUSSION AND CONCLUSION

A typical burden when attempting to build several ASR systems that will be combined with ROVER is that there are few systematic ways to build a large number of systems that are different enough to be successfully combined. People usually resort to building a small number of carefully crafted ASR systems on different features sets, such as MFCC or PLP, or to exchanging their carefully crafted systems with systems developed by a different group of researchers. The randomized decision tree state tying procedure attempts to overcome some of these limitations, and greatly simplifies the design of multiple systems. Versus the usual approach, the randomized decision tree procedure licenses the production of an arbitrary number of systems built on the same input features, and the individual randomized systems can be built systematically, without any additional tuning. While there is no guarantee that the systems built on different randomized decision trees will make complementary errors, the effectiveness of the approach has been illustrated in practice on two large vocabulary tasks.

5. REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[2] L. K. Hansen and P. Salamon, "Neural network ensemble," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993–1001, 1990.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

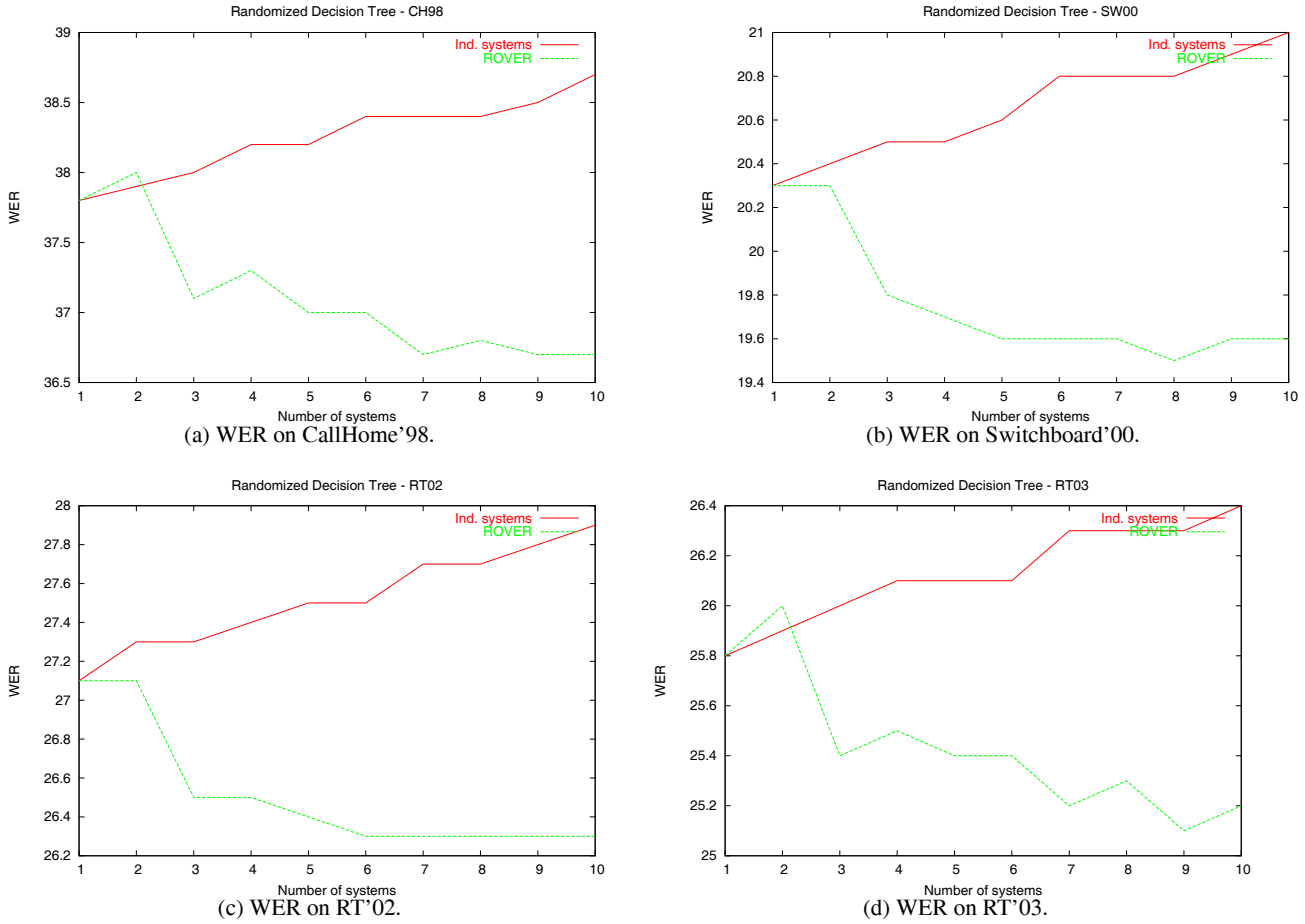
[4] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. of the Thirteenth Int. Conf. on Machine Learning*, 1996.

[5] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER)," in *Proc. IEEE ASRU Workshop*, Santa Barbara, 1997, pp. 347–352.

[6] P. Niyogi, J.-B. Pierrot, and O. Siohan, "Multiple classifiers by constrained minimization," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[7] H. Schwenk, "Using boosting to improve a hybrid HMM/neural network speech recognizer," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999, vol. 2, pp. 1009–1012.

[8] C. Dimitrakakis and S. Bengio, "Boosting HMMs with an application to speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Montreal, Canada, 2004.



(a) WER on CallHome'98.

(b) WER on Switchboard'00.

(c) WER on RT'02.

(d) WER on RT'03.

Fig. 4. WER for baseline system, individual random systems (sorted by increasing WER), and ROVERed systems (including baseline).

[9] G. Zweig and M. Padmanabhan, "Boosting gaussian mixtures in an LVCSR system," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000, vol. 3, pp. 1527–1530.

[10] L. Breiman, "Random forests – random features," Tech. Rep. 567, Statistics Department, U. C. Berkeley, Sept. 1999, ftp://ftp.stat.berkeley.edu/pub/users/breiman.

[11] W. Byrne, D. Doermann, M. Franz, S. Gustman, J. Hajič, D. Oard, M. Picheny, J. Psutka, B. Ramabhadran, D. Soergel, T. Ward, and W.-J. Zhu, "Automatic recognition of spontaneous speech for access to multilingual oral history archives," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 420–435, 2004.

[12] T. K. Ho, "The random subspace method for constructiong decision forests," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[13] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–158, 1998.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, and M.A. Picheny, "Decision trees for phonological rules in continuous speech," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Toronto, Canada, May 1991, pp. 185–188.

[16] W. Reichl and W. Chou, "Robust decision tree state tying for continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 555–566, Sept. 2000.

[17] L. Breiman, J. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[18] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker adaptive training," in *Proc. Int. Conf. on Spoken Language Processing*, Philadelphia, USA, 1996, pp. 1137–1140.

[19] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig, "The IBM 2004 conversational telephony system for rich transcription in EARS," Submitted to ICASSP'05.

[20] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Montreal, Canada, 2004, pp. 409–412.