COMBINING SEARCH SPACES OF HETEROGENEOUS RECOGNIZERS FOR IMPROVED SPEECH RECOGNITON

Xiang Li, Rita Singh, Richard M. Stern

Department of Electrical and Computer Engineering and School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213, USA
{xiangl, rsingh, rms}@cs.cmu.edu

ABSTRACT

In speech recognition systems, information from multiple sources such as different feature streams or acoustic models can be combined in many different ways to yield better recognition performance. It is theoretically expected that the best performance is obtainable through the simultaneous use of all sources of information, in a system capable of using these in parallel. Such systems, however, are extremely complex and difficult to construct. In this paper we propose a simple alternative criterion for combination which can factorize the complex recognizer into several simple recognizers, each of which is based on a single source of information. We use this criterion in simple experiments which combine lattices from recognizers built with different feature streams. Experimental results obtained on five different corpora show that the proposed method is effective in improving recognition performance.

1. INTRODUCTION

The performance of an automatic speech recognition (ASR) system depends critically on the feature set that it uses. Even though there currently exist several different kinds of features which may generate good results under certain conditions (*e.g.* MFCC, PLP), none of them work perfectly for all conditions. Instead, they represent different subsets of information embedded in the speech signal to different levels of accuracy. It stands to reason that combining the information from these features properly would result in better recognition accuracy than the result obtained with any single feature set alone.

Generally speaking, there are two ways of combining information from different features together: We can either concatenate different feature vectors to form a larger feature vector and perform recognition based on this combined feature, or perform recognition directly based on the individual features and combine their outputs together. The latter method has several advantages. It permits parallel processing in training and recognition, as well as flexibility in adding new features and combining different systems of heterogeneous architectures without additional effort. For instance, such an approach permits us combine the outputs of a word-based HMM recognizer with a phone-based segmental recognizer, with no integration of the core systems required.

Combination of the outputs of heterogeneous recognizers has typically taken the form of combining their recognition hypotheses (*e.g.* ROVER [1], hypothesis combination [2], and discriminative model combination [3]). Although these methods effectively improve recognition accuracy, they only combine the single best hypotheses from the recognizers. We know that the actual words in an utterance may not appear in the single-best hypothesis, although they may appear with high score in the search space of the recognizer. When the single best hypotheses of several recognizers are combined, the correct word may still not be found in any of these hypotheses. Any combination scheme based on single-best hypotheses would fail to hypothesize the correct words. On the other hand, it has been shown that appropriate processing of multiple hypotheses from the search space of even just a single recognizer can give us better performance than just picking up the single-best hypothesis [4].

Motivated by the considerations above, we present in this paper our recent work on combining recognition lattices from heterogeneous recognizers for improved overall recognition. Our choice of lattices as the representation of the search spaces of recognizers, rather than other multiple output formats such as *N*-best lists, is based on the fact that lattices represent the search space more accurately. Furthermore, combining lattices results in a much more expanded search space in the combined lattice, giving us a better chance of hypothesizing the correct words.

In the following section we present an alternate criterion for classification based on multiple information sources. In Sec. 3 we discuss the combination of lattices from multiple recognizers. In Sec. 4 we present our experimental results. This is followed by a discussion in Sec. 5.

2. LATTICES AND COMBINATION CRITERIA

Ideally, when using multiple sources of information for recognition, one would construct a composite search space and find the best hypothesis in that space using an appropriate objective criterion that incorporates all the available information. When the multiple sources of information are multiple features or acoustic models (AM), for example, the ideal way to combine them would be to generate a lattice of possible hypotheses using all features or models at all times, and to obtain the best path through this lattice. This, however, implies that the recognizer must be constructed in such a manner as to permit the use of multiple features or multiple acoustic models in parallel. Such recognizers must, of necessity, be complex. Furthermore, incorporation of additional sources of information could require a reconstruction of the recognizer.

With these considerations in mind, we propose an alternative mechanism for combining heterogeneous sources of information that does not require the construction of complex recognizers.

Instead, we combine the search spaces of the individual simple recognizers to effectively approximate the ideal complex recognizer. We do this by redefining the objective criterion used for recognition in a way that permits the factorization of the individual information sources in the complex recognizers into separate simple recognizers. We now describe with a simple example the modified objective criterion, which we refer to as the *max-max* criterion. In the next section we describe the combination of recognizer search spaces and the effect of the max-max criterion on obtaining hypotheses from these combined search spaces.

Consider a simple task where we must recognize a given signal as one of a set of words, $W_1, W_2 \ldots, W_J$. We are given two sources of information in the form of two different features of the signal, f_1 and f_2. The recognition task can be stated as estimating the word $\hat{W}$ such that

$$\hat{W} = W_i : i = \max_j\{\log P(f_1, f_2 | W_j) + \log P(W_j)\} \quad (1)$$

where $\log P(f_1, f_2 | W_j)$ is the combined acoustic evidence derived from f_1 and f_2. For the simple case where the two sources of evidence are two different features, this term can be computed by treating the two features as components of a single extended feature and estimating the joint probability of the two features. More generically, however, it is assumed that the two sources of information are independent of each other. This results in the following estimate for the combined acoustic evidence from the information sources:

$$\log P(f_1, f_2 | W_j) = \sum_k \log(P(f_k | W_j)) \quad (2)$$

In other words, the acoustic evidence for each word is the sum of the acoustic evidences derived from each of the information sources. The recognition problem gets restated as:

$$\hat{W} = W_i : i = \max_j\left\{\log P(W_j) + \sum_k \log(P(f_k | W_j))\right\} \quad (3)$$

Recognition can only be performed using the complex recognizer in Eq. (3), which considers all information sources jointly. We refer to the recognizer in Eq. (3) as a *max-add* classifier, since the acoustic evidence from all the information sources must be added to obtain the combined evidence.

In most practical situations, however, the joint acoustic evidence from multiple sources is dominated by the evidence from one, or a small number of the sources, the composition of which may vary from instance to instance. *i.e.*

$$\log P(f_1, f_2 | W_j) \approx \max(\log P(f_1 | W_j), \log P(f_2 | W_j)) \quad (4)$$

In this paper we use this observation to recast the recognizer in Eq. (1) as

$$\hat{W} = W_i : i = \max_j\{\log P(W_j) + \max(\log P(f_1 | W_j), \log P(f_2 | W_j))\} \quad (5)$$

which can be generalized to

$$\hat{W} = W_i : i = \max_j\{\max_k(\log P(f_k | W_j) + \log P(W_j))\} \quad (6)$$

We refer to the recognizer in Eq. (6) as a *max-max* classifier. The advantage with the max-max classifier is that the max operation is transitive. *i.e.*

$$\max_j\{\max_k(\log P(f_k | W_j) + \log P(W_j))\} = \max_k\{\max_j(\log P(f_k | W_j) + \log P(W_j))\} \quad (7)$$

We note that the term within the brackets on the right hand side of Eq. (7) is in fact the log probability of the hypothesis of a simple recognizer based only on the k^{th} information source. In other words, Eq. (7) represents the factorization of the max-max classifier into two simple classifiers (in our two-feature example). A max-max classifier can therefore be factored into several simple recognizers, and the final hypothesis is merely the output of the simple recognizer with highest log probability.

3. LATTICE COMBINATION

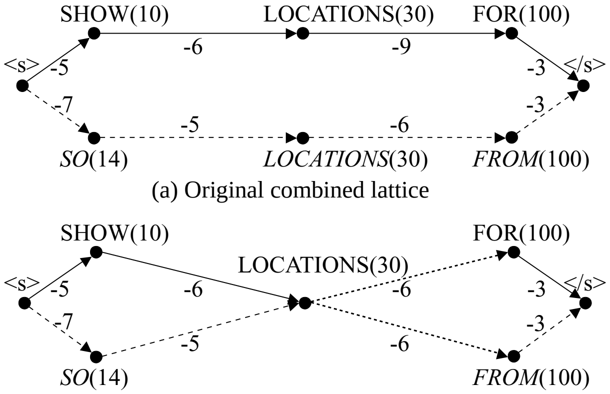
In more detailed speech recognition problems, the max-max classifier can be approximated by choosing the highest scoring hypotheses among the various simple recognizers. Better gains can be had, however, by applying the factorization piecewise to every word considered during recognition. For this, however, the search spaces of the various simple recognizers must be combined. Lattices are a particularly convenient representation of these search spaces, for such an exercise.

The lattice is a compact representation of all the highest scoring hypotheses considered by the recognizer. It is a directed acyclic graph in which nodes are associated with words and their starting and ending frames, and the edges represent the possible transition of words in the hypothesis. A single-best hypothesis is merely the best path through the graph. Lattices from multiple recognizers can be combined by merging them into larger graphs using various rules. Conventional combination of multiple information sources for recognition requires rescoring of the combined lattice using all sources jointly. However, since the edges in the component lattices are the best scoring local paths within the search space of the individual recognizers, it can be assumed that rescoring the combined lattice using only the native edge scores from the component lattices is consistent with the max-max criterion. This gives us the advantage of not having to recompute the acoustic scores associated with the edges. Finding the best hypothesis in the combined lattice reduces to a graph search problem.

We now describe the merging of lattices from multiple recognizers. We begin by merging the utterance-begin and utterance-end nodes of the component lattices to generate an initial large lattice. We then merge edges and nodes, and add new edges using some rules which we describe below. Note that these rules are specific to a recognizer based on triphonetic sub-word units.

3.1 Merging edges

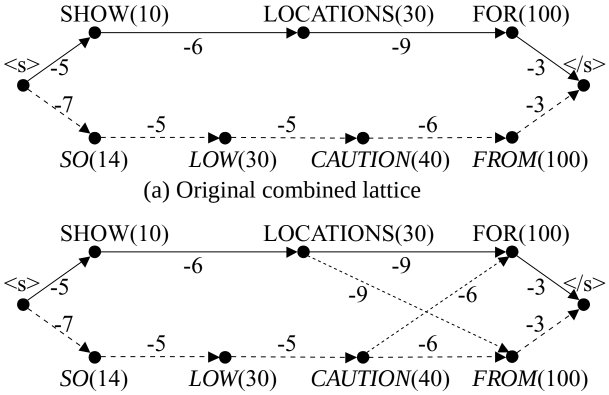
Acoustic scores are typically associated with edges in a lattice. Edges from two or more lattices being combined are merged if their outgoing nodes have the same word label, same beginning and ending frame, and the ending nodes of these edges have the same first phone. To merge these edges, we first merge their outgoing nodes together to a new node, and then update their acoustic scores. In following the max-max logic, the new score associated with the merged edge is the maximum score for that edge from the corresponding edges of the component lattices. Fig. 1 shows an example wherein edges in two lattices are merged.



(a) Original combined lattice

(b) Combined lattice after merging edge

Figure 1. Merging edges in two lattices. (a) is a simplified representation of two lattices. Edges in the two lattices are represented by solid and dashed lines respectively. The number next to each node is the starting frame of the word associated with the node. (b) Edges starting at the word "LOCATIONS" are merged, with acoustic scores updated using the max-max criterion.



(a) Original combined lattice

(b) Combined lattice after creating new edges

Figure 2. Creating new edges without AM recomputation. A new edge is built from the node "LOCATIONS" to the node *"FROM"* and is represented by the dotted line. Since the words "FOR" and *"FROM"*, begin at the same frame, the score of the new edge is the same as the existing one. Similarly, another edge is created from *"CAUTION"* to "FOR".

3.2 Creating new edges without AM recomputation

In this step, we build a new edge from Node A to Node B so long as there exists an edge from Node A to Node C whose word label has the same first phone as the word label in Node B, and the difference in the beginning frame of Node B and Node C lies below a chosen threshold (*e.g.* 30 or 40 ms). Since the edge from *A* to *C* tells us that Node A can end just before Node C, *A* can also end just before Node B so long as Node B and *C* have similar beginning times. The acoustic score associated the new edge is assigned as:

$$W_{A \to B} = \frac{D_{A \to B}}{D_{A \to C}} \cdot W_{A \to C} \qquad (8)$$

where $W_{I \to J}$ and $D_{I \to J}$ are the acoustic score and duration of the edge from Node I to Node *J*. Since the acoustic score of an edge is the product of the acoustic likelihoods of each frame in Eq. (8), the longer the duration of an edge, the less its acoustic score. Fig. 2 shows an example of creating new edges without AM recomputation:

3.3 Creating new edges with AM recomputation

Theoretically speaking, when we create new edges between different nodes from different lattices, the constraint of requiring the same first phone for the incoming node of the existing edge (*e.g.* "FOR" in Fig. 2) and the node into which we want to build a new edge (*e.g. "FROM"*) is too strict. In fact, we should be able to create new edges so long as the starting time of ending nodes of existing edges is the same as the starting time of the node into which we want to build a new edge. For example, if we replace the node "FOR(100)" in Fig. 2 with "OF(100)", we should still be able to build an edge from "LOCATIONS" to *"FROM"* even though the only existing edge from "LOCATIONS" is to "OF", since the starting time of "OF" is the same as the starting time of *"FROM"*.

The reason we impose "same phone" constraints is that if the first phone of Nodes B and C are different, we have no way of assigning a score to the new edge $A \to B$, since the acoustic models for A-B and A-C are different. Usually, in addition to lattices, we also have acoustic models and feature vectors of the corresponding underlying systems. In these situations, we can recompute the acoustic score of any edge that we want to create directly, rather than through Eq. (8). Creating new edges must be done with some caution. While new edges may compensate for those missing due to deletions during decoding, some edges may also be incorrect due to spurious insertions during decoding. The latter, if dominant, can degrade the accuracy of the combined systems.

With this in consideration, in creating a new edge from Node A to Node B we follow four steps: (a) Check whether there exists another edge from Node A to Node C, and whether the difference in the starting time of Node C and Node B lies within a threshold (*e.g.* 30-40 ms). (b) If there does exist such an edge from A to C, use the acoustic models and features of the system that generated Node A to compute the AM score of Node A, using as right context the first phone of Node B and using an ending time of one frame before the starting time of Node B. (c) If the first phone of Node C is in the same confusable phone set as the first phone of Node B, we assign the AM score computed in Step (b) to the new edge A → B. (d) If the first phone of Node B and Node C are in different confusable phone sets, we multiply the AM score generated from Step (b) by a weight less than 1 and assign the weighted AM score to the new edge A → B.

3.4 Score normalization

For the max-max classifier to be applicable, we need to consider parallel systems whose acoustic scores are within numerically comparable ranges. Thus acoustic score renormalization prior to combination is generally important. To achieve this, in each of the component recognizers, for each feature vector the scores of all HMM states which constitute the acoustic models are normalized with respect to the highest score for that vector. The normalization factor for any edge in a lattice is the sum of the maximum state scores for all the feature vectors in the duration of that edge.

WER (%)	Feat 1	Feat 2	ROVER	Hyp-Comb	Lat-Comb
RM	11.5	12.0	11.1	8.4	8.0
TI+D 5dB	25.5	26.6	26.1	25.6	24.7
TI+D 10dB	12.5	13.0	13.7	11.9	11.3
SPINE 1	35.1	36.2	35.4	34.2	33.2
SPINE 2	17.5	16.6	17.8	15.9	15.0

Table 1. Recognition accuracy of three combination schemes on five corpora. The lattice combination scheme uses edge merging and building of new edges without AM recomputation.

4. EXPERIMENTAL RESULTS

We tested the performance of the lattice combination on five corpora: the DARPA Resource Management (RM) corpus, the Telefónica (TI+D) Cellular Telephone corpus with artificially corrupted traffic noise at SNRs of 5 and 10 dB, the Speech In Noisy Environments 1 (SPINE1) corpus and the Speech In Noisy Environments 2 (SPINE2) corpus. The RM corpus consists of clean speech.The TI+D and SPINE databases are telephone bandwidth databases with added noise in Spanish and English, respectively.

All experiments were conducted using the CMU SPHINX-III speech recognition system. For each corpus, two different features were used to generate lattices. For the RM and TI+D 5 dB and TI+D 10 dB corpora, standard MFCC and PLP features were used. For the SPINE1 corpus, two versions of MFCCs with different DCT implementations were used. For the SPINE2 corpus, we first performed a Karhunen-Loeve transform to generate a 20-dimensional feature vector from 40 dimensional log-spectral vectors, and then performed linear discriminant analyses to generate two different 13-dimensional feature vectors. Each of these features was designed to discriminate amongst two different sets of subword-unit classes in each case. In all our experiments, lattice combination was compared to combination of the best recognition hypotheses using *ROVER* and conventional hypothesis combination. Lattices were combined using the steps described in this paper. The Viterbi algorithm was used to obtain hypotheses from the combined lattices. Table 1 shows word error rates (WERs) obtained in the combination experiments, where edge-scores were assigned without AM recomputation. Table 2 gives statistical significance measurements for the results reported in Table 1, using the matched-pair test [5].

For the RM, TI+D 5 dB and TI+D 10 dB corpora, we also tested the performance of building new edges with AM recomputation. Table 3 shows the recognition accuracy of lattice combination and statistical significance level (P) between the results of lattice combination and hypothesis combination.

RM	TI+D 5 dB	TI+D 10 dB	SPINE1	SPINE2
0.3	0.05	0.18	0.12	0.04

Table 2. Statistical significance level of the difference in recognition performances of hypothesis and lattice combination.

	RM	TI+D 5dB	TI+D 10dB
WER (%)	7.8	24.3	11.1
P	0.16	0.03	0.08

Table 3. Recognition accuracies and statistical significance (P) of the differences between hypothesis combination and lattice combination with edge merging and building of new edges with AM recomputation.

5. DISCUSSION AND CONCLUSIONS

Experimental results show that lattice combination improves the recognition accuracy consistently in all tested corpora. The relative improvement of lattice combination over hypothesis combination [2] ranges from 3 to 6 percent without AM recomputation and from 6 to 8 percent with AM recomputation. For the TI+D 5 dB corpus, lattice combination is the only combination scheme that reduces the WER.

The intrinsic similarity of the feature sets plays an important role in the combination. It can be seen from Tables 1 and 2 that the greatest improvement in WER and the most significant difference between hypothesis combination and lattice combination was achieved on the SPINE 2 corpus, for which the feature sets had been developed specifically to maximize the difference between different sub-word classes. This is expected, since we have used the max-max criterion which is designed to select the locally most prominently scoring words in its composite best path through the combined lattice. That at any given time one or the other feature would enhance the likelihood of a word was specifically ensured by the features which were designed specifically to complement each other in enhancing subsets of acoustic units.

6. ACKNOWLEDGEMENTS

This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

7. REFERENCES

[1] Fiscus, J.G., “A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER),” *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 347-354, 1997.

[2] Singh, R., Seltzer, M., Raj, B., and Stern, R.M, “Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination”, *Proc. ICASSP 2001*, Salt lake city, Vol 1, pp.273-276, 2001.

[3] Beyerlein, P., “Discriminative model combination”, *Proc. ICASSP 1998*, Vol. 1, pp 481-484, 1998.

[3] Mangu, L., Brill, E., Stolcke, A., “Finding Consensus Among Words: Lattice-Based Word Error Minimization”, *Proc. EUROSPEECH 1997*, Vol 1, pp. 495-498, 1997.

[4] Gillick, L., Cox, S.J., “Some statistical issues in the comparison of speech recognition algorithms”, *Proc. ICASSP 1989*, Vol 1, pp. 532-535, 1989.