ACOUSTIC MODELING IMPROVEMENTS IN A SEGMENT-BASED SPEECH RECOGNIZER

N. Ström, L. Hetherington, T. J. Hazen, E. Sandness, J. Glass

MIT Laboratory for Computer Science
545 Technology Square
Cambridge, Massachusetts 02139 USA
{nikko,ilh,hazen,sandness,jrg}@sls.lcs.mit.edu

ABSTRACT

In this paper we report on some recent improvements on the acoustic modeling in a segment-based speech recognition system. Context-dependent segment models and improved pronunciation modeling are shown to reduce word error rates in a telephone-based, conversational system by over 18%, while the technique of Gaussian selection reduces overall computation by more than a factor of two.

1. INTRODUCTION

Since its deployment in 1997 [1], the number of calls made to the Jupiter telephone-based conversational weather information system has been steadily increasing. Currently, an average of about 200 calls, yielding 1000 new utterances, are made each day. This continuous influx of real data from a wide variety of users and channel conditions is an invaluable resource for our research in robust speech recognition and understanding. In this paper we report on recent refinements in acoustic modeling which have improved performance, and reduced computational requirements.

The SUMMIT segment-based speech recognition system [2] is capable of handling two rather different types of acoustic models: *segment* models, and *boundary* models. Segment models are intended to model hypothesized phonetic segments in a phonetic graph, and can be context-independent or context-dependent. The observation vector for these models is of fixed dimensionality, and is typically derived from spectral vectors spanning the segment. Thus, we only extract one segmental feature vector, and thus one likelihood, for a phone, regardless of its duration. This is in contrast to frame-based methods such as Hidden Markov Models (HMMs), which compute likelihoods at a fixed frame-rate.

In contrast to segment models, boundary models are intended to model transitions *between* phonetic units. The observation vector for these diphone models is also of fixed dimensionality, and is centered at hypothesized phonetic boundary locations, or *landmarks*. Since some landmarks will in fact be internal to a phone, both internal, and transition boundary models are computed, either in context-independent, or -dependent fashion.

Because the observation spaces of segment and boundary models differ significantly, they contribute different information to the search and ranking of hypotheses. It should be noted that different segmentation hypotheses typically contain different numbers of segments (although all segmentations contain the same number of landmarks). This is potentially a problem because it makes the comparison, and ultimately ranking, of different hypotheses more difficult. A solution is to normalize the segment probabilities. In this study the "anti-phone" [2] normalization method has been used, but we have experimented with other methods in the past as well [3].

In the early development of the Jupiter system, context-independent segment models were used. As more data became available context-dependent boundary models were added. The log probability model scores of the boundary and segment models were linearly combined to produce the total acoustic score for each hypothesis. Ideally, the combination of the models should provide more accurate results than either of the individual models. However, the boundary models, with their higher degree of context dependency, benefited more from the increasing training data than the segment models. Thus, as more training data became available, the use of the context-independent segment models actually began to degrade the overall recognition accuracy. For this reason, only boundary models were used in [4]. This study investigated context-dependent segment models to see if they could improve the boundary models.

In a real-time conversational system, improved (i.e., more detailed and therefore more computationally demanding) acoustic models are a mixed blessing. To keep the total amount of computation constant, the increased phonetic modeling accuracy must be accompanied by the adjustment of other parameters, such as the number of paths searched in the beam. In our case, the introduction of context-dependent segment models more than quadrupled the number of acoustic models in the system. To deal with the increased amount of computation we use a technique called Gaussian selection to greatly reduce the number of probability densities that are computed [5,6].

In comparison to standard HMM approaches, an idiosyncrasy of segment modeling is that it is not very forgiving of discrepancies between the pronunciation models of the system and user's pronunciations. While HMMs are capable of absorbing mismatched pronunciations within a few poorly scoring frames, it is not as easy to hide pronunciation mismatches using segments, which may span multiple frames. It is therefore necessary to accommodate, at the very least, the most common pronunciation variations for words in the vocabulary. In SUMMIT, a set of phonological rules is used to generate alternate pronunciations from the baseforms of the lexicon. In general these rules have a tendency to overgenerate, allowing many unlikely variants. Therefore, in this study we examine the effect of applying likelihoods to the arcs in the pronunciation network. ML estimation is used to find the likelihoods from forced transcriptions of the training data.

2. SYSTEM DESCRIPTION

2.1 Acoustic Features

The acoustic observation vectors for both segment and boundary models are based on the first 14 MFCCs. The features used for each landmark are identical to those used in [4]. The feature vectors are derived by computing averages and derivatives of MFCC frames over several fixed duration regions surrounding each landmark. This is similar to the feature extraction typically used in a standard HMM except that the width of the window over which the features are derived, 150ms, is relatively large. Another difference is that the landmark 'rate' is not constant like the frame rate of an HMM. The dimensionality is reduced to 50 using principal component analysis.

For segment modeling, segments are divided into three regions: initial 30%, middle 40%, and final 30%. In each of these regions, averages and derivatives of the MFCCs are computed. In addition, the logarithm of the segment duration is used. Not all averages and derivatives are used in all three regions; more emphasis is on the averages in the middle region, and on the derivatives in the beginning and end [7]. The resulting 40-dimensional vector is rotated using principal component analysis.

2.2 Probability Density Functions

Mixtures of Gaussian probability density functions with diagonal covariance matrices were used. To combat sparse data problems, the number of Gaussians per model is dependent on the number of training observations n and the size of the observation vector m. Thus, the number of mixture components is set to n/m, but not greater than a constant N. In these experiments, $N = 50$ for boundary models, and $N = 25$ for segment models. There is no sharing of Gaussians between models.

2.3 Maximum Likelihood Training

Both boundary and segment models were trained by iteratively alternating segment alignment and parameter estimation. This is sometimes called Viterbi training in contrast to embedded segmentation training schemes such as Baum-Welch. The initial segmentation was computed using a boundary-only recognizer. Boundary models were trained in the same manner as in [4], but we currently have about twice the amount of training data (50,000 utterances). Each context-dependent segment model was trained on all applicable segments. Thus some segments appear in the statistics for multiple models (e.g., the monophone 'ae' and the triphone 'k-ae+tcl').

2.4 Unit Selection

For boundary modeling, over 2200 different context-dependent diphone boundaries and 61 context-independent phone-internal boundaries were possible based on the lexicon. The entire group of transitions was semi-automatically collapsed into 715 classes for which boundary models were trained. The semi-automatic procedure utilized phonetic and acoustic properties of the phonemes, as well as their frequency in the training data, to derive the final set of classes.

The context-dependent segment models were selected by counting the occurrences of all triphones, left and right diphones, and monophones in the initial segmentation of the training utterances. Any such unit with a count greater than N was selected. Initial experiments indicated that $N = 250$ is a reasonable tradeoff between acoustic modeling accuracy and computation in this case. This gave 935 triphones, 1190 diphones, and the 61 monophones (i.e., a total of 2186 segment models). When we apply this model set to the pronunciation network of the 1957 words in the Jupiter lexicon, 71% of all arcs have triphone models, and the remaining 29% are backed off to diphones or monophones. Because of a small computational advantage in the search, the diphone with left context specified is always chosen when both types of diphones are trained but the triphone is not.

2.5 Finite-State Transducers and Search

SUMMIT now makes use of finite-state transducers (FSTs) to represent context, phonological, lexical, and language model constraints [4]. In particular, the first pass utilizes a *single* FST encompassing all these constraints. In this work, this FST is the minimized composition of a context-dependency transducer (rewrites labels to be context-dependent) [8], a lexicon with phonological rules applied, and a word-class bigram. Thus, we are using context-dependent models in the first pass. Boundary, segment, lexical, and bigram scores are synchronized at the phone level.

In a second pass, we incorporate a word-class trigram to compute N-best. (We have had difficulty using a trigram directly in the first pass due to FSTs becoming too large during determinization.) We do this by computing an intermediate word graph with the bigram constraint and then compose this with the inverse bigram composed with the trigram. This results in a word graph with trigram scores. We then compute the A* heuristic for every node of this graph, and finally compute the N-best list with an admissible phone-based A* search. All of these operations are performed after an utterance is complete, yet overall latency is typically well under 1s. Our previous A* heuristic was based on bigram scores, which could cause thrashing when bigram and trigram scores differed significantly, leading to larger latencies.

3. RECOGNITION ACCURACY

The test data consist of 2506 utterances randomly selected from the corpus, of which 1806 utterances were considered "in domain." Table 1 summarizes the error rates for these "in domain" data using a 1957 word lexicon, and class bi- and trigram language models [4]. As can be seen, the segment-only case is worse than the boundary-only condition, but linearly combining them yields a significant relative improvement of 13%. We also found improvements on the "out of domain" data, where the word error rate of the combined models was reduced from 60% to 48.8% compared with the boundary-only recognizer of [4].

Table 2 contains results obtained after ML estimated pronunciation likelihoods are applied to the arcs of the pronunciation network. The likelihoods do not significantly effect the results when using only boundary models. However, a significant reduction in error rate from 10.4% to 9.6% is obtained when using only segment models. This is expected because the segment

	Word Error Rate	Sentence Error Rate	Sub	Ins	Del
Boundaries	7.5	21.0	4.1	1.0	2.5
Segments	10.4	27.7	5.3	1.9	3.3
Combination	6.5	18.8	3.5	1.3	1.7

Table 1. Recognition results (in percent) for only boundary models, only segment models, and linear combination respectively. Since computational issues are covered in section 4, the numbers shown here are for very conservative beam pruning (i.e., the results are close to the case of no pruning).

	Word Error Rate	Sentence Error Rate	Sub	Ins	Del
Boundaries	7.6	20.7	4.1	1.0	2.5
Segments	9.6	25.7	4.6	1.3	3.7
Combination	6.1	17.4	3.1	1.2	1.8

Table 2. Recognition results (in percent) with pronunciation weighting for only boundary models, only segment models, and linear combination respectively. The effect of beam pruning is negligible here.

modeling approach is more sensitive to pronunciation variants than the boundary modeling approach. Like the HMM approach, boundary models, when used by themselves can absorb pronunciation mismatches within a few poorly scoring landmarks. However, because the landmark rate is considerably less then the frame rate of typical HMMs, accurate pronunciation models are still more important for boundary models than for HMMs. Finally, an error rate reduction from 6.5% to 6.1% is obtained in the combined segment and boundary system when the pronunciation likelihoods are used.

4. COMPUTATIONAL ISSUES

4.1 Computational Complexity of Boundary Modeling versus Segment Modeling

Although segments typically span multiple landmarks, the number of segments examined in the search may be significantly larger than the number of landmarks. The reason is that segments have two time attributes. Thus, the number of segments to consider is a quadratic function of the frame rate, while the number of landmarks is a linear function. Furthermore, in the current system, the boundary models utilize less context dependency than the segment models. In particular, the internal boundary models, which typically make up more than half of the landmarks of a hypothesis, are context-independent.

Figure 1 quantifies how much more computation the segment models require with equal number of Gaussian mixtures per model. Clearly, the boundary models only condition is the only one operating in real time. However, note that the accuracy degradation is rather small when, for segment models, the maximum number of Gaussians is reduced from N = 25 to 10 ('♦' versus '+'). Therefore, in the following experiments of this section we are using N =10 for segment models.

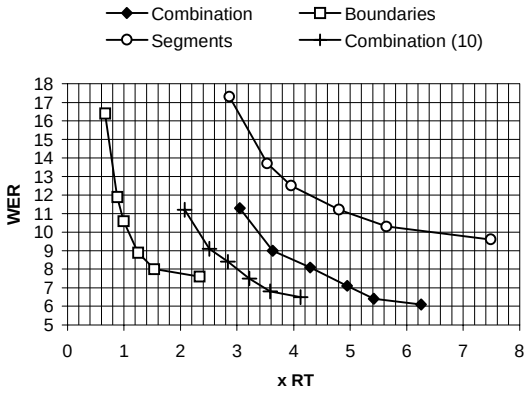

Figure 1. Word error rate versus computation time for boundary models only, segments models only and linear combination. Varying the aggressiveness of the beam pruning controls the computation time. The fourth series, '+', shows the case of 10 Gaussians per segment model, linearly combined with the boundary models. Timing performed on a 500MHz Pentium III.

4.2 Use of Gaussian Selection

Because well over half of the time is consumed by the calculation of Gaussian probability densities, we use Gaussian selection to reduce the number of Gaussians that are evaluated [5]. To evaluate a model, the feature vector is quantized using binary VQ, and the resulting codeword is used to look up the reduced list of Gaussians to evaluate, often resulting in a large reduction in overall computation.

We select the Gaussians for a particular model m and codeword k as follows: (1) Keep all Gaussians for which the distance from the Gaussian mean to the codeword center is below some threshold Θ. (2) Keep a Gaussian if its mean quantizes to k. (3) If a given model has no Gaussians associated with codeword k, select its closest Gaussian. Note that (2) means that every Gaussian will be selected for at least one codeword k, and (3) ensures that at least one Gaussian is evaluated for every model/codeword pair (m, k).

As for the distance criterion used in (1) above, we use the squared Euclidean distance normalized by the number of dimensions D. Thus, for (1), we select a Gaussian iff

$$\frac{1}{D}\sum_{i=1}^{D}(\mu_m(i)-c_k(i))^2 \le \Theta.$$

We have tried the more complex weighted distance criteria described in [6] (e.g., distance weighted by inverse average variance or weighted by the inverse of the geometric mean of average variance and codeword-specific variance), but these did not offer any advantages in our system. Perhaps this is due to the fact that we apply principal component analysis to our feature vectors to

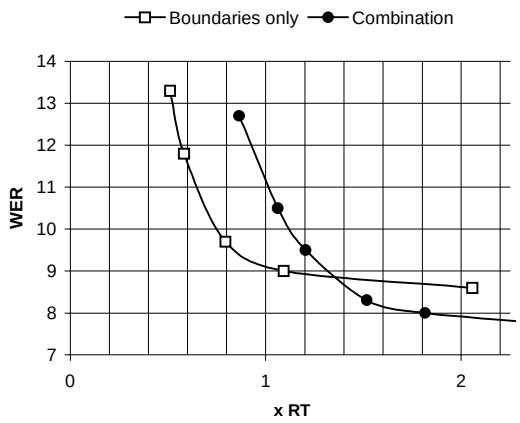

Figure 2. Word error rate versus computation for the boundary/segment model combination (with 10 mixtures per segment model) compared to the boundary models only system. Timing performed on a 500MHz Pentium III.

whiten the feature space prior to creating Gaussian mixtures. We also unsuccessfully tried to use Gaussian log probability in place of squared distance, thinking this was the most meaningful criterion, but without improvement. We further tried to set a limit on the size of the Gaussian lists as in [6], but were unable to improve upon the accuracy/speed tradeoff.

The number of codewords used is 512, but the accuracy and speed are not terribly sensitive to this setting in our system. We used $\Theta = 0.6$ as our selection threshold. Note that this is significantly smaller than that used in [6], where values between 1.0 and 1.9 were used.

Figure 2 shows the accuracy/speed curves for the boundary-only system and the combined boundary/segment system with Gaussian selection in use. Since the combined system, with triphones in the first pass evaluates considerably more Gaussians than the boundary-only system, it benefits more from the Gaussian selection. Note that the boundary-only and combined curves are closer than those of Figure 1. With Gaussian selection, the performance curves cross over at about 1.35 times real time. Below that point, the boundary-only system performs better, above that the combined system is preferable.

5. CONCLUSIONS AND FUTURE WORK

By making the segment models context-dependent, we have shown that the SUMMIT framework, combining boundary models and segment models, is scalable to a much larger training corpus than we have used in the past. Both the word error rate and the sentence error rate were significantly improved.

The increased computation for segment modeling was greatly reduced by the use of Gaussian selection. This allowed a speedup by a factor of about two with maintained recognition accuracy. Nevertheless, with current hardware, rather aggressive beam pruning thresholds are necessary to achieve real-time operation, and the recognition accuracy at this operating point is not better than that of system based on landmarks only. However, as can be seen in Figure 2, the slope of the RT/WER curve is rather steep at the current real-time operation point indicating that further reductions of computation may pay off significantly in terms of accuracy. In the near future we plan to add mixture tying to our system and hope to further reduce the number of Gaussians evaluated. This tying should also reduce the problem of sparse data when training context-dependent models.

There are many parameters of the segment modeling that were not investigated nor optimized in this study. For example, it is likely that using a different acoustic feature vector can further increase the recognition accuracy, because the current features were not selected to optimize performance of context-dependent models. The simple method of unit selection can also be refined. For example, the context dependency of landmark models is based on phonetic knowledge combined with statistics from the corpus. The same method could be applied to segment models, which would reduce problems due to sparse data and increase the coverage of phonetic contexts.

We have recently applied aggregation of multiple models with heterogeneous acoustic features and achieved significant accuracy improvements [9,10]. In the future we plan to combine this with the techniques reported on in this paper.

6. REFERENCES

1. V. Zue, S. Seneff, J. Glass, L. Hetherington, E. Hurley, H. Meng, C. Pao, J. Polifroni, R. Schloming, and P. Schmid, "From interface to content: translingual access and delivery of on-line information," in *Proc. Eurospeech 97*, pp. 2227-2230, Rhodes, Greece, Sep. 1997.

2. J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. ICSLP 96*, pp. 2277-2280, Philadelphia, PA, Oct. 1996.

3. J. Chang and J. Glass, "Segmentation and modeling in segment-based recognition," in *Proc. Eurospeech 97*, pp. 1199-1202, Rhodes, Greece, Sep. 1997.

4. J. Glass, T. Hazen, and L. Hetherington, "Real-time telephone-based speech recognition in the Jupiter domain," in *Proc. ICASSP 99*, pp. 61-64, Phoenix, AZ, Mar. 1999.

5. E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *Proc. ICASSP 93*, pp. 692-695, Minneapolis, MN, Apr. 1993.

6. K. Knill, M. Gales, and S. Young, "Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs," in *Proc. ICSLP 96*, pp. 470-473, Philadelphia, PA, Oct. 1996.

7. M. Muzumdar, *Automatic Acoustic Measurement Optimization for Segmental Speech Recognition*, M.Eng. Thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, MA, May 1996.

8. M. Riley, F. Pereira, and M. Mohri, "Transducer composition for context-dependent network expansion," in *Proc. Eurospeech 97*, pp. 1427-1430, Rhodes, Sep. 1997.

9. A. Halberstadt and J. Glass, "Heterogeneous Measurements and Multiple Classifiers for Speech Recognition," *in Proc. ICSLP 98*, pp. 995-998, Sydney, Australia, Dec. 1998.

10. T. Hazen, and A. Halberstadt, "Using aggregation to improve the performance of mixture Gaussian acoustic models," in *Proc. ICASSP 98*, pp. 653-656, Seattle, WA, May 1998