IMPROVEMENTS ON SPEECH RECOGNITON FOR FAST TALKERS

M. Richardson[1]*, M. Hwang, A. Acero, and X.D. Huang*

Speech Technology Group
Microsoft Research
Redmond, Washington 98052, USA
http://research.microsoft.com/srg

ABSTRACT

The accuracy of a speech recognition (SR) system depends on many factors, such as the presence of background noise, mismatches in microphone and language models, variations in speaker, accent and even speaking rates. In addition to fast speakers, even normal speakers will tend to speak faster when using a speech recognition system in order to get higher throughput. Unfortunately, state-of-the-art SR systems perform significantly worse on fast speech. In this paper, we present our efforts in making our system more robust to fast speech. We propose cepstrum length normalization, applied to the incoming testing utterances, which results in a 13% word error rate reduction on an independent evaluation corpus. Moreover, this improvement is additive to the contribution of Maximum Likelihood Linear Regression (MLLR) adaptation. Together with MLLR, a 23% error rate reduction was achieved.

1. INTRODUCTION

The accuracy of a speech recognition (SR) system is severely affected when there are mismatches between the training and testing conditions. These mismatches can be on microphone, background noise, language mode, dialect, speaker and even speaking rates. There are many possible mismatches. For example, when the system makes mistakes or when the user tries to correct misrecognitions, the user tends to slow down or even begins to speak isolated words. When the system performs well, the user tends to speak faster to get higher throughput. Unfortunately, state-of-the-art SR systems perform significantly worse on either fast or slow speech. In [1], we collected isolated speech acoustic data and introduced duration modeling to handle slow speech. We will address faster speech problems here to further improve the robustness.

There have been attempts for improvements on the recognition of fast speech ([2], [3], [4]). It has been proven useful to set high transition probabilities on arcs leaving from one state to a later state in the hidden Markov model (HMM) [2]. Another option is create speaking-rate dependent acoustic models [3] by collecting a corpus of speech with normal, slow, and fast speaking rates, with a higher improvement obtained on the slow speech. Detecting the speaking rate has also attracted some attention [4] by analyzing the spectrum of speech with different speaking rates.

In this paper, we present our different approaches to making our system more robust to fast speech. In particular, we propose cepstrum length normalization (CLN), whose principle is to normalize the phone duration by stretching the length of the utterance in the cepstrum domain so that it matches the acoustic model trained on regular speech. More importantly, the stretched dynamic features (delta and delta delta cepstra) become less dynamic and resemble better those of speech with regular-speaking rate. It is the dynamics that are most affected by speech rate change as indicated in [3]. The length normalization can be applied to either the training or the incoming testing utterance. Both approaches yielded about a 10% word error rate reduction, although we will present the normalization on the test data only in this paper for its simplicity. Furthermore, we show that improvements made by CLN and Maximum Likelihood Linear Regression (MLLR, [5]) adaptation are additive. Together, they achieved more than a 20% error reduction on our test data. We also observed a similar but smaller improvement by using a shorter window shift in computing cepstra.

Section 2 will summarize our speech recognition system, Whisper [6], and describe the speech corpora used in this paper. Section 3 describes our speaking-rate determination algorithm. Section 4 presents the cepstrum length normalization algorithm in detail. Section 5 discusses various experimental results. Finally, conclusions and future work are outlined in the last section.

2. SYSTEM DESCRIPTION AND SPEECH CORPORA

This paper is based on the Microsoft speech recognition engine, Whisper [6], which uses continuous-density Hidden Markov Models (HMMs) with senonic decision trees. The SI system built here consists of 6000 gender-dependent context-dependent senones with 20 Gaussians each, with diagonal covariance matrices. The features used were 12 mel-frequency cepstrum coefficients (MFCCs), log energy and their first and second order differences in 10ms time frames. The standard 60,000-word lexicon and its trigram language model are used in a one-pass Viterbi beam search [7]. The speaker independent (SI) acoustic training corpus comes from the 284-speaker DARPA World Street Journal (WSJ) corpus 0 and corpus 1 (denoted as *SI-284*).

The development data is our in-house collected data, consisting of 3 male and 4 female speakers, each uttered in a fast fashion about 50 WSJ sentences without out-of-vocabulary (OOV) words. The reason to exclude OOV words is to exclude the OOV effect on our speaking-rate study. This set, denoted as *dev-fast*, contains 5245 words, a total of 1426 seconds of speech and silence. Similarly, another 3 male and 4 female speakers were asked to utter various 50 WSJ-typed sentences

[1] Intern at Microsoft Research. Current address: Department of Computer Science, Box 352350, University of Washington, Seattle, WA 98195-2350

fast to form an evaluation set, denoted as *eval-fast*. *Eval-fast* contains 5273 words in 1427 seconds of speech and silence.

To show that our CLN algorithm does not degrade the recognition performance on speech of regular speaking rates, we asked 7 (4 male and 3 female) of the above 14 speakers to utter their individual sets of 50 sentences in their regular speeds. This set, denoted as *regular*, consists of 5254 words in 2052 seconds of speech and silence. In addition, the DARPA 1994 WSJ H1 development set (denoted as *h1dev94*) is also used as additional speech of regular speaking rates to verify the robustness of our CLN algorithm. *H1dev94* consists of 7417 words in 2919 seconds of speech.

3. SPEAKING RATE DETERMINATION

In this paper, speaking rates are defined by the length of an individual phone relative to its average duration in the *SI-284* training corpus. We feel that a simple measure of the number of phones per second is not informative enough, and that some knowledge or estimation of the phones that were uttered will improve the estimation of real speaking rates. For this reason, we first computed the gender dependent histogram of phone durations from the *SI-284* corpus. Figure 1 shows the histogram of phone /b/ in the female training data.

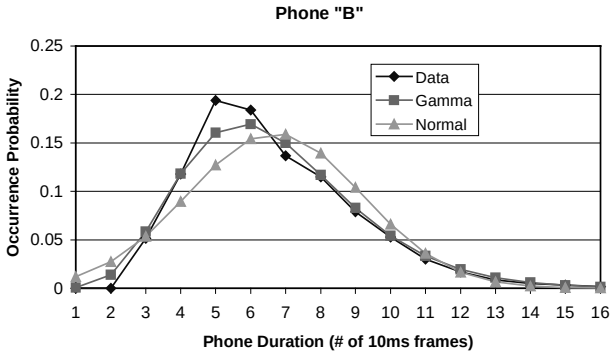

Figure 1. Duration distribution of phone /b/ in the WSJ SI-284 female training corpus. Also shown are the gamma and normal distribution approximations to the data.

From the figure, we can see that for phone duration, a Gamma distribution assumption is a closer fit to the real distribution than is a Gaussian distribution. Note also that the minimum duration for any phone is 30 ms because the phonetic HMM topology we used is the 3-state Bakis topology [8] without any skipping arcs.

In order to estimate the speaking rate of a testing utterance, a first pass recognition is run on the utterance, and phone segmentation information is recorded. By comparing the phone duration with those from the SI statistics, we can stretch the testing utterance either phone-by-phone or sentence by sentence.

3.1 Phone-by-phone Length Stretching

To stretch on a *phone-by-phone* basis, each phone segment is adjusted in length to the peak of the Gamma distribution of the phone in the SI training corpus:

$$\Gamma(x,\alpha,\beta) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)}$$

Let $\Gamma(x,\alpha_i,\beta_i)$ be the distribution for phone *i*, then since its mean is $\mu_i = \alpha_i / \beta_i$ and the variance is α_i / β_i^2, we can use the training data to approximate the true mean and variance in order to estimate parameter α_i and β_i. The peak of the Gamma distribution occurs at $peak_i$, and we thus define the length-stretching factor (ρ_i) for a phone segment with length l_i as:

$$peak_i = \frac{\alpha_i - 1}{\beta_i} \qquad \rho_i = \frac{peak_i}{l_i} \tag{1}$$

Notice for fast speech, $\rho_i > 1$, and for slow speech $\rho_i < 1$. Our first attempt showed that this phone-by-phone duration adjustment did not yield any recognition improvement if the correct phone sequence is unknown. This is due to the fact that a wrong phone identification and segmentation might adjust that segment of speech in the wrong direction. Therefore, although great improvement was achieved when the correct phone sequence is known, we decided not to use this approach.

3.2 Sentence-by-Sentence Length Stretching

To stretch on a *sentence-by-sentence* basis, there are many methods to compute a single *normalization factor* ρ to apply to the entire utterance. One method is to find ρ which maximizes the joint probability of the utterance with respect to the SI phone duration probability distributions. That is,

$$\tilde{\rho} = \arg\max_{\rho} \left\{ P(\rho l_1 | \Gamma_1) \cdot P(\rho l_2 | \Gamma_2) \cdots P(\rho l_n | \Gamma_n) \right\}$$

or alternatively

$$\rho = \frac{\sum_{i=1}^{n} \alpha_i}{\sum_{i=1}^{n} \beta_i l_i}$$

where *n* is the number of phone segments in an utterance. Again preliminary experiments showed this approach failed to make improvements, perhaps due to the same reason as in Section 3.1. Given that HMMs treat each frame of speech equally important, perhaps the influence of a phone segment should be proportional to its duration.

Because of the above reasons, the final technique we adopted, denoted as *AveragePeak*, for determining the best sentence-based normalization factor ρ is to simply average all the *phone-by-phone* peak factors:

$$\rho = \frac{1}{n} \sum_{i=1}^{n} \rho_i \tag{2}$$

where ρ_i is defined by Formula (1). This smoothing/averaging effect compensates for the mistakes in the phone sequence estimation and indeed provides us a stable improvement on fast speech and no degradation on regular-speed speech. Therefore, this paper only presents results from using *AveragePeak* speaking rate determination.

In addition, we also tried two other simple, intuitive stretching factors, defined as

$$\rho = \frac{\sum_{i=1}^{n} \mu_i}{\sum_{i=1}^{n} l_i} \quad \text{and} \quad \rho = \frac{\sum_{i=1}^{n} peak_i}{\sum_{i=1}^{n} l_i}$$

Again preliminary results showed that either way yielded similar but insignificantly less improvement as Formula (2).

Note that one of the big differences between our approach and the one in [3] is that our speaking rate (ρ) is a continuous variable while [3] always classified speech as one of three categories: slow, normal and fast.

4. THE CLN ALGORITHM

To compensate for the unexpected short duration and dramatic changes in the dynamic acoustic features in fast speech, we propose to lengthen and smooth the cepstrum. We tried three different ways to change the length of a speech segment from l to l' frames:

1. Inserting/dropping frames uniformly in the speech segment.
2. Repeating/deleting only those frames that represent the steady state of each phone segment. This was done by searching those frames that had the minimum distortion with respect to their neighbors in a phone segment.
3. Creating new frames by interpolating neighboring frames. This was done with approximated band-limited interpolation.

We found all three approaches gave similar improvement on the development data, with the last one best and most stable. Therefore we only reported the experimental results based on the interpolation of cepstrum frames. The interpolation was an approximation, since we didn't use the sinc function; rather we used a Lanczos filter, one of many FIR filters:

$$f(x) = \mathrm{sinc}(x) \cdot \mathrm{sinc}(x/3)$$

The choice of the filter was arbitrary. Similar results were obtained by using a Mitchell filter. Linear interpolation resulted in about relatively 2% less improvement than what was obtained with band-limited interpolation.

5. EXPERIMENTAL RESULTS

5.1 CLN on the Test Data of Fast Speech

In this experiment, CLN was applied only to the test data. The algorithm first estimated the phone segments in the testing utterance by running the decoder. It then used the hypothesized phone segments to find the sentence-based normalization factor, ρ. Table 1 shows 16.5% error rate reduction by interpolating the cepstrum frames on *dev-fast* test data. The normalization factor ρ was determined by *AveragePeak* as defined by formula (2). The normalization factors of the utterances in *dev-fast* varied between 0.92 and 1.47.

Training data \ test data	Original	Interpolation
Original	16.64%	13.90%

Table 1. Word error rates on dev-fast with and without MFCC interpolation.

5.2 CLN on the Test Data of Normal Speech

The cepstrum length normalization algorithm does not know the speed of an incoming speech in advance. The same algorithm is applied to every testing utterance. Therefore, it is crucial to prove that the algorithm does not degrade on the recognition of speech with regular speaking rates. This was verified in Table 2 on the *regular* and *h1dev94* data sets. Again, *AveragePeak* was used to determine the normalization factor and MFCC interpolation was used to stretch the MFCC frames. ρ ranged between 0.70 and 1.17 in *regular* and 0.77-1.32 in *h1dev94*. Compared with Table 1, this table also demonstrated that recognition on fast speech is usually worse (almost twice) than recognition on normal-speed speech.

5.3 Evaluation and MLLR

After the above exercises on the development data, we applied the *AveragePeak* rate determination and MFCC interpolation scheme to the evaluation data, *eval-fast*, with the SI models trained on the un-stretched MFCCs. As the first two results shown in Table 3, more than 10% error rate reduction was achieved again.

Regular		*H1dev94*	
Original MFCCs	MFCC Interpolation	Original MFCCs	MFCC Interpolation
8.36%	8.20%	8.71%	8.78%

Table 2: Word error rates on the *regular* and *h1dev94* data sets.

Original MFCC	MFCC Interpolation	MLLR on Gaussian Means	MFCC Interpolation + MLLR
18.34%	15.91%	16.03%	14.03%

Table 3: Word error rates on the *eval-fast* set. Combining MFCC interpolation and MLLR speaker adaptation yielded 23.5 % error rate reduction.

One might think speaker adaptation techniques such as MLLR could contribute the same improvement. Therefore, we ran unsupervised batch-mode MLLR on *eval-fast*. That is, assume the speaker boundary was known. All 50 sentences of a speaker were recognized first with the original MFCCs. Then the 50 hypotheses together with the original MFCCs were used to adapt the Gaussian means with 10 fixed phone classes. With the adapted Gaussian means, recognition was re-run on all 50 utterances with the original MFCCs. We can see that MLLR alone made a similar amount of improvement as the MFCC interpolation scheme alone.

To prove that the improvement from MFCC interpolation was not overridden by MLLR adaptation, these two techniques were combined by running MFCC interpolation first. The procedure is illustrated in Figure 2. The last result in Table 3 shows that these two improvements are actually additive. All together, the improvement was more than 20%.

As illustrated in Figure 2, the MFCC interpolation and MLLR adaptation can be repeated multiple times to achieve more improvement.

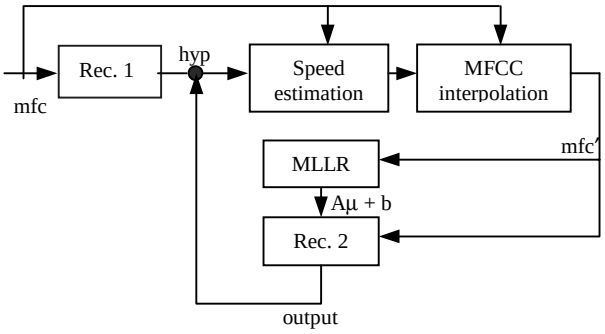

Figure 2: Combination of MFCC interpolation and MLLR adaptation.

5.4 Shrinking Hamming Window Shift

Since we believe the largest improvement on the recognition of fast speech is in the better match with the SI phone duration and dynamic features trained from regular speaking rate speech, another interesting experiment is to use a smaller window shift in generating the cepstrum. To do this, the speed factor ρ is first computed as formula (2) outlines with the default window shift (in our case 10ms). Then a new window shift is computed as inversely proportional to ρ,

$$s' = \frac{s}{\rho}$$

With the new window shift, new MFCCs are calculated for the given utterance and the 2nd phase recognition with the new MFCCs is run to get the final output. Notice we never modified the SI model. This approach yielded a similar but slightly smaller improvement than the MFCC interpolation scheme, with 14.38% error rate on *dev-fast*.

6. CONCLUSIONS AND FUTURE WORK

Our experiments showed that normalizing the duration and smoothing the dynamic features is important when building a speaking-rate robust acoustic model. In particular, by interpolating MFCC frames for fast speech, we observed a 13% error rate reduction. Moreover, the improvement is additive with that made by MLLR, resulting in a 23% error reduction all together.

One of the limitations our algorithm has is the assumption of a uniform speaking rate across the utterance. Though practically true for short sentences, this may not be valid for long sentences. One way to address this limitation is to collect a corpus of fast speech exclusively. Then, classify the utterances in the fast speech corpus into a few categories of different speeds (e.g., 10% faster than regular, 20%, 30%+, etc.) using the *AveragePeak* speed determination and a given regular speaking-rate training corpus such as *SI-284*. Next an acoustic model for each different speed category could be trained using that subset of data with un-stretched MFCCs. At decoding, all these models are loaded and evaluated at each time frame, with the un-stretched testing MFCCs. The maximum accumulated score determines which speed of model each frame chooses. This approach also eliminates the stretching of MFCCs completely and thus requires only one pass of recognition.

Another limitation is the necessity of running the decoder twice, with the first time to determine the speaking rate only. A rapid estimation of the speaking rate (such as the one used in [3] if a finite number of speaking rates is used) is valuable to avoid the first-pass decoding. Moreover, context-dependent senone duration modeling instead of context-independent phone duration modeling might provide more insights for speaking-rate determination.

ACKNOWLEDGMENTS

We would like to acknowledge the Packard foundation for supporting Richardson's presenting this paper at Eurospeech, and other coworkers at Microsoft for their assistance, ideas, and time.

REFERENCES

[1] Alleva F., Huang X., Hwang M., and Jiang L. "Can continuous speech recognizer handle isolated speech?" *Proceedings of the Eurospeech Conference* 1997.

[2] Mirghafori N., Fosler E., and Morgan N. "Fast speakers in large vocabulary continuous speech recognition: analysis & antidotes". *Proceedings of the Eurospeech Conference*, Madrid, pp. 491-494. Sep, 1995

[3] Martinez F., Tapias D. and Alvarez J. "Towards Speech Rate Independence in Large Vocabulary Continuous Speech Recognition". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle. pp. 725-728. May 1998.

[4] Morgan N. and Fosler-Lussier E. "Combining Multiple Estimators of Speaking Rate". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle. pp. 729-732, May 1998.

[5] Leggetter C. J. and Woodland P. C. "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs". *Computer Speech and Language*, 9:171-186, 1995.

[6] Huang X., Acero A., Alleva F., Hwang M., Jiang L., and Mahajan M. "From Sphinx-II to Whisper – Making Speech Recognition Usable", *Speech and Speaker Recognition – Advanced Topics*, Kluwer Academic Publishers, pp. 481-508, 1996.

[7] Alleva F., Huang X., and Hwang M. "Improvements on the pronunciation prefix tree search organization". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, pp. 133-136. May 1996.

[8] Bakis R. "Continuous Speech Recognition via Centisecond Acoustic States"., 91st *Meeting of the Acoustical Society of America*, April 1976.