TIME SHIFT INVARIANT SPEECH RECOGNITION

S. Basu, A. Ittycheriah & S. Maes

IBM T.J. Watson Research Center
PO Box 218, Yorktown Heights, NY, 10598, USA
sankar@watson.ibm.com abei@us.ibm.com smaes@us.ibm.com

1. ABSTRACT

When shifting by a few samples a speech signal, we have observed significant variations of the feature vectors produced by the acoustic front-end. Furthermore, these utterances when decoded with a continuous speech recognition system leads to dramatically different word error rates.

This paper analyzes the phenomena and illustrates the well known result that classical acoustic front end processors including spectrum and cepstra based techniques suffer from time-shift. After describing the effect of sample sized shifts on the spectral estimates of the signal, we propose several techniques which take advantage of shift variations to multiply the amount of training that speech utterances can provide. Eventually, we illustrate how it is possible to slightly modify the acoustic front-end to render the recognizer invariant to small shifts.

2. INTRODUCTION

2.1. Overview

Speech recognition systems present significant accuracy variations when decoding speech utterances. Numerous phenomena induce these variations: intra-speaker changes, inter-speaker differences, speaking style, dialect, accent, word rate, formality or casualty of speech, conversational (dialog) versus talk, prepared versus spontaneous, channel mismatch , microphone mismatch , background noise mismatch, reverberation etc. Traditionally, acoustic robustness against these effects is achieved either by training over a large set of utterances representative of all the variations that the decoder is expected to meet. Alternatively, a general purpose decoder can be adapted to specific conditions by adaptation. This adaptation can be done in supervised or unsupervised mode, in advance or in real time. Eventually, special acoustic features, robust to specific environments, can be developed to replace a classical front-end.

In any case, it is always important to reduce the impact of any acoustic variations on the recognizer accuracy. It is always useful to use as much data as possible to train a system and to learn the effect of these variations.

In this paper, we use IBM speech recognition engine technology, namely continuous large vocabulary speech recognition stack decoder with context-based decision tree lefemes modeled with continuous HMMs and trigram language models . This engine was used in the classical ARPA evaluations. The reader is invited to consult [3, 4, 5] for more details. It is also the base technology of IBM telephony and desktop ViaVoice products. The observations and algorithms that we present in this paper are valid over desktop, telephony and broadcast data.

Although the same behavior is observed with other set of FFT-derived acoustic features [6], for the purpose of this paper, we are using MFFC, with C0, delta and delta-delta. These features are computed over frames of 25 ms shifted from frame to frame by 10ms.

Nothing is more arbitrary that the initial synchronization of the analog acoustic waveform and the first frame of the front-end. However, it is expected that it should only slightly, if at all, impact the accuracy of the recognizer. However, we observe significant modification of the spectral estimates and MFCC produced by the front-end which in turn result into variations of up to 10% of the word error rate on the same database!

Such variations are appreciable enough to be studied more carefully. Obviously, the speech recognition engine should be more robust against such a small effect. However, with unmodified classical systems, the amplitude of the WER variations indicates that time shifts can provide a way to extract more non-redundant training or adaptation data out of a regular speech database.

Since each feature vector is derived from frames which are shifted by 10 ms, an additional time shift by T will perturb the acoustic feature vector by $T_1 = T \ mod \ 10ms$ plus a system delay of the quotient, T_2. Therefore, we limit our analysis to shifts, T_1 smaller than 10ms.

2.2. Approaches

Approaches can now be considered to reduce the effect of time shifts on the acoustic features and their impact on the decoded accuracy.

New sets of acoustic features are introduced to directly remove the time-shift variability. These features are obtained by multiplexing shifted versions of the original signal. Intuitively, it amounts to compute multiple sets of acoustic feature vectors and recombine them before feeding them to the recognition engine. It amounts to low-pass filtering the output of the spectral estimator (FFT), before cepstral transform. Because the extension is trivial, we will present a simple average of shifted MFCC. Although, it would be preferable to use this new front-end to train the recognizer, a conventional recognizer could be used with the new front-end introduced only during decoding. Such front-end modification increases the complexity and memory/CPU requirement of the engine at training, if performed, as well as at testing.

Since the first approach modifies the speech recognition engine at decoding time, and preferably also during training or adaptation, we propose a second method of altering the training data to build time-shift invariant acoustic models, with a conventional acoustic front end. Only the training phase is to modified with respect to a conventional system. At the difference of the first method, we do not need to introduce any change in the speech recognition engine and its complexity or

memory/CPU requirements.

Furthermore, the proposed approach increases dramatically the amount of available training data resulting from a conventional data collection and scripting. It results in improvement of the error rate variability on conventional recognition tasks, using a conventional engine with the shift-invariant acoustic models.

Results will be presented to illustrate the improvements of simple implementations of the two approaches.

Eventually, we should mention alternate features like the wavelet-based synchrosqueezed cepstra which are inherently closer to time-shift invariance [7].

2.3. Problem

In our recent speech recognition experiments it has been noticed that shifting speech signal by one time sample changes the accuracy of the recognizer by unexpected amounts. To quantify this phenomenon we considered a simple experiment of running the recognizer for two almost identical segments of speech signals — one obtained from the other by dropping the first sample. The test signal in our experiment reported here was the first sentence from the test set of the Wall-Street journal database [1]. Similar anomalies were noticed elsewhere as well. We recorded the respective detailed match [8] values as a measure recognition accuracy in each of the two cases. The system was trained on 35 hours of wall-street-journal standard data set available from DARPA [1]. The result of this experiment is tabulated in Table 1. It is clear that for certain lexemes, a significant deviation occurs. A correspondingly obvious deviation was also observed in the word recogntion accuracy.

Table 1: Table showing the % variation in detail match as a measure of recognition accuracy due to shift of unit time sample. Columns corresponding to DM(1) and DM(0) indicate shifted and unshifted values respectively.

Lexeme	DM(0)	DM(1)	%change
.(TRM)	6.92	6.93	0.26
RICHARD	3.97	4.18	5.36
SARAZEN	0.49	0.45	9.69
SIL	-1.30	-1.29	0.85
CHIEF	3.41	3.38	0.88
FINANCIAL	2.79	2.95	5.59
OFFICER	4.63	3.85	20.00
OF	0.02	0.09	238.00
NEWS	2.69	2.67	0.71
CORPORATION	8.53	8.53	0.00
SAID	1.71	1.68	2.08
THE	0.32	0.35	7.97
COMPANY	4.67	4.95	5.90
BELIEVES	3.02	2.99	1.17
A	0.17	0.08	53.09
DOWNGRADING	7.30	7.26	0.55
ISN'T	3.18	3.10	2.54
IN	0.89	0.96	7.59
ORDER	2.42	2.49	2.86
SIL	10.61	10.70	0.85

It is clear that this phenomenon must result from deviations in the cepstra or, equivalently, in the logarithms of mel-binned fft's in the front-end of the recognizer. In an attempt to explain this unwanted effect, we conjectured a number of different reasons. These are as follows: (1) The pre-emphasis filter that preceeds the computation of fft in the front-end. For all practical purposes, this approximates a differenttiator (2) unwieldly values values of $\log x$ for small x, because $\log x \to -\infty$ as $x \to 0$

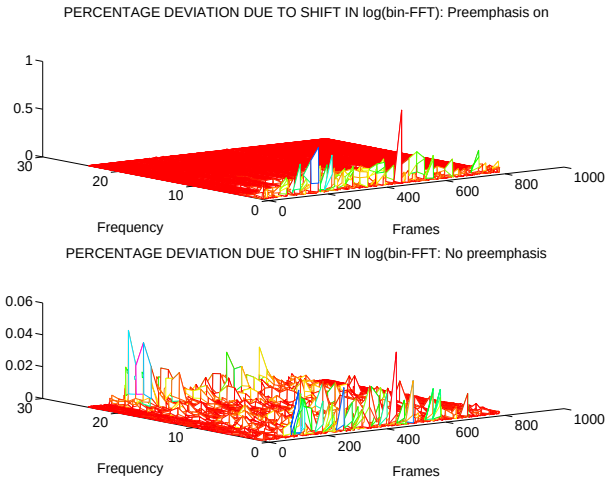

Figure 1: Effect of preemphasis filter on the logarithm of binned FFT. x-axis corresponds to frame number; y-axis corresponds to the 24 binned log FFT values

(3) The choice of Hamming windows vs. hanning windows was a potential reason, and lastly (4) the size of the 25 ms window (at a frame rate of 10ms) was a suspect as well. We note that there has been significant debate on the proper choice of the latter in the speech literature.

To examine the effect of pre-emphasis filter on the phenomenon of interest, we computed the logarithms of mel-binned fft's with the pre-emphasis filter turned on and then later with the pre-emphasis filter turned off. The *percentage* deviation measured as:

$$\left| \frac{(\log \mathrm{FFT})_0 - (\log \mathrm{FFT})_1}{(\log \mathrm{FFT})_0} \right|$$

is plotted in figure 1. To be specific, the plot corresponds to the first sentence of the wall-street-journal test set considered in Table 2.3. The dependence of the magnitude spectrum of windowed fourier transform of a impulse train on the position of the window placement is demonstrated in Figure 2 in the extreme (admittedly artificial) situation when the window size is nearly the same as the separation between the impulses. The 1st, 3rd and 5th subplot show the impulse train with the window placement, whereas 2nd, 4th and 6th subplot shows the correponding discrete Fourier transforms. Notice that the magnitude spectrum can vary from a flat spectrum to near sinusoid. This demonstrates the importance of choosing the proper window size when dealing with signals with rapid variation as in stops and plosives. As a remedy to the problems arising from the fact that $\log x \to -\infty$ as $x \to 0$, we use the following *regularized logarithm*. This proves to be an approximation to the logarithmic function for small values of x, while coinciding with $\log x$ for large x. Let

$$f(x) = \begin{cases} \left(\left(\frac{x}{\alpha}\right)^n - 1\right) + \log \alpha & \text{for} \quad \alpha > x \\ \log x & \text{for} \quad \alpha \leq x \end{cases} \qquad (1)$$

Then it is easy to see that $f(\alpha) = \log \alpha$ and as $n \to \infty$ we have for all values of x and α that $f(x) \to \log x$. The integer parameter n determines the rapidity at which the function $f(x)$ drops as x approaches 0, whereas α is the *knee*, to be appropriately chosen, beyond which $f(x)$ coincides with $\log x$. We choose α depending on the dynamic range of the mel-binned FFT. In particular, the values $n = 2$ or $n = 4$ appear to be appropriate for our experiments, and we choose $\alpha = x_{\max}/20$, where $x_{\max}$ is the largest value of the magnitude mel-binned frequency spectrum.

We shall explore the effect of using hanning vs. hamming

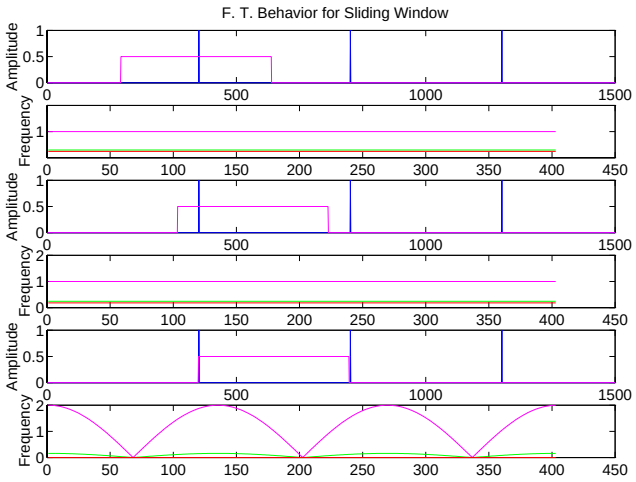

Figure 2: Effect of window size on the FFT

window by computing detail match for all of the 40 sentences of the wall-street -journal test set [1], and subsequently by plotting the histogram of the differences of the detail match values between the shifted and unshifted cases. Ideally, this histogram should be a delta function, and any departure from it should indicate the sensitivity of the recognizer due to shift in time samples. In other words, a histogram concentrated near the origin corresponds to less sentivity due to unit shifts of samples, whereas a more spread out histogram correponds to (less desirable) higher sentivity. Figure 3, 4 and 5 respectively show the histograms for using hamming window with pure logarithm, hanning window with pure logarithm and hanning window with the regularized logarithm (1) with $n = 2$ in equation (1). The pre-emphasis filter was turned off for all these experiments. Clearly, the progression from Figure (3) to Figure (4) and then

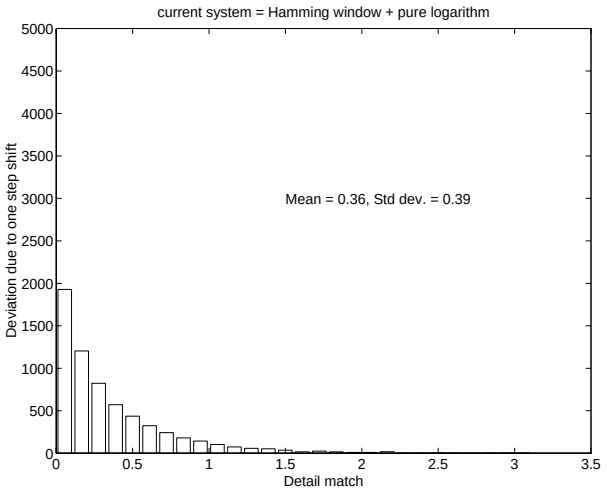

Figure 3: Histogram showing $|\mathrm{DM}_0 - \mathrm{DM}_1|$; hamming window and pure logarithm

to Figure (5) indicates histograms increasingly concentrated towards the origin, and thus improved robustness to unit shifts in time sample. These improvements are accordingly reflected in the word error rate in recognition.

It behooves us to explain this experimentally observed phenomenon. It might be (erroneously) argued that mel-binning avoids the problem of small values of sampled frequency spectra near the zero frequency (i.e., the problem arising due to divergence of logarithm of zero) by averging the spectra out over

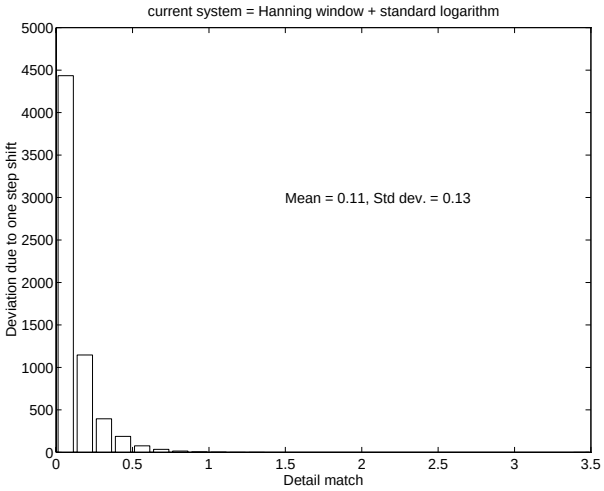

Figure 4: Histogram showing $|\mathrm{DM}_0 - \mathrm{DM}_1|$ with hanning window

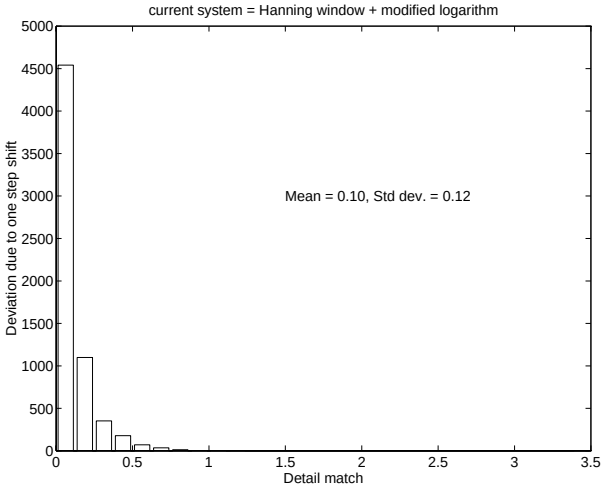

Figure 5: Histogram showing $|\mathrm{DM}_0 - \mathrm{DM}_1|$ with hanning and modified log window

the lowermost mel frequency bin. However, recall that the mel-bins are of increasing size in a logarithmic scale, thus, resulting in small low frequency mel-bins and large high frequency mel-bins. This makes the low frequency samples of mel-binned spectra look even smaller relative to the high frequency samples of mel-binned spectra, thus accentuating the problem arising from divergence of logarithm of small numbers. This necessitates the use of (1) instead of the pure logarithm.

The difference in behaviour from the use of hanning as opposed to hamming window may be attributed to the fact that hamming window is a raised cosine, whereas hanning is only a cosine window. A raised cosine is a cosine with a square pulse added to it. It is well known [2] that the latter has wider sidelobes in its spectrum. Another way of viewing its drawbacks is to consider sudden changes in signal values as in stops or plosives. Onset of such a sound at, say, the trailing edge of a window may result in significant differences from the use of hanning or hamming window, because the former, being a pure cosine, has the effect of zeroing out the large changes in signal values, and windowed signal is, thus, relatively less sensitive to sudden variations in signal amplitude near the window edge.

3. SHIFT INVARIANT METHODS

We have explored three methods to overcome the effect of this time-shift problem in terms of speech recognition. These methods are

Method 1 Averaging FFT's of shifted windows: Here the many shifted replica's are created from the input pcm stream and fft's computed on each. The magnitude spectrum is then averaged to provide a more robust estimate of the spectrum.

Method 2 Training on shifted data: Here the training data to the recognizer is shifted and models are built on many shifted replicas. This allows the front-end of the speech recognizer to remain the same during training and testing.

Method 3 Shift invariant features: Here the acoustic features are themselves chosen to be independent of time shift, for example the wavelet features described in a companion paper [7].

3.1. Method 1

The data we used for this experiment consisted of 2017 utterances of yellow pages categories collected over the telephone. The testing vocabulary contains 1857 words and 2473 phonetic baseforms for the words. The data was collected from a variety of speakers calling from across North America with a wide variety of handsets. Shifted replicas of this database with 1,2,3,4,5 ms shifts were created. The speech recognition system used in this experiment was trained on a corpus containing roughly 300K sentences from Macrophone, Phonebook and an internal database of digits and other command vocabularies.

Though many methods can be explored for creating averaged fft spectra of the same data, we chose to create shifted windows and average the resulting normally derived fft magnitude spectrum with that of the shifted data. It is important to note that the pre-emphasis and windowing of the data must also be done on the shifted window. The recognition results are presented in the following table.

Shift	Base	2xFFT	3xFFT
0	3.70	3.56	3.65
1	3.47	3.40	3.42
2	3.42	3.74	3.35
3	3.47	3.49	3.53
4	3.80	3.51	3.53
Avg	3.57	3.54	3.49
Var	0.028	0.013	0.013

Here 2xFFT refers to computing 2 ffts, one with the normally presented data and one with a shift of 2.5 ms. 3xFFT refers to using shifts of 1.8 and 3.6 ms. The variance clearly shows that the technique seems to be reducing the variance though because the error rate itself is quite small, the variance is also small.

3.2. Method 2

The training method offers the distinct advantage of cpu reduction during testing over the previous method. During the training phase, the speech data is shifted by 1,2,4,5 ms in this experiment thus producing 5x the amount of speech training material. For this test, a telephony names database was chosen with 7K utterances for training, and with 643 utterances for test. The vocabulary consists of 20K names.

Shift	Base	Base+Shifted
0	24.55	19.36
1	19.51	19.44
2	20.37	19.05
3	20.13	20.13
4	21.76	21.30
Avg	21.26	19.85
Var	4.05	0.81

Here the base+shifted system achieves on average an improvement of 6.6% over the base system. This data clearly shows the reduction in variance of the base+shifted system.

4. CONCLUSION

Time-shift effect on the acoustic features and the recognition accuracy was a striking discovery for us. Even if well established from a signal processing point of view, the impact of small shift on recognition accuracy is widely unknown or ignored among developers of speech recognition systems. We have shown that it is possible to reduce this effect with appropriate acoustic front-ends. On the other hand, this effect can be taken advantage of to increase the amount of exploitable information provided by a speech database used for training or adaptation of the speech recognition engine.

Acknowledgement:
The authors wish to acknowledge helpful discussions with Raimo Bakis.

5. REFERENCES

1. ARPA Wall Street Journal data, Available from Linguistcs Data Corporation.
2. J. R. Deller and J. G. Proakis and J. H. L. Hansen, Discrete-time processing of speech signals, Macmillan, New York, NY, 1993.
3. LR Bahl, P.V. De Souza, P.S. Gopalakrishnan, D. Nahamoo, M.A. Picheny, 'Robust methods for using context-dependent features and models in a continuous speech recognizer', Proc. ICASSP'94
4. P.S. Gopalakrishnan, L.R. Bahl, R. Mercer, 'A tree search strategy for large vocabulary continuous speech recognition', Proc. Icassp'95
5. R. Bakis, S. Chen, P.S. Gopalakrishnan, R. Gopinath, S. Maes, L. Polymenakos, 'Transcription of Broadcast news - system robustness issues and adaptation techniques', Proc. ICASSP 97.
6. LR Bahl, et al. 'Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal task', Proc. ICASSP'95.
7. S. Basu, S. Maes, 'Wavelet-based energy binning cepstral features for automatic speech recognition', Proc. ICSLP'98.
8. F. Jelinek, Statistical methods for speech recognition, MIT Press, 1997