ISCA Archive
http://www.isca-speech.org/archive

ETRW on
Accessing Information in Spoken Audio
Cambridge, UK
April 19–20, 1999

RECOGNITION-COMPATIBLE SPEECH COMPRESSION FOR STORED SPEECH

Roger Tucker [1], Tony Robinson [2], James Christie [2], Carl Seymour [3]

[1] Hewlett Packard Laboratories, Bristol. *email: roger_tucker@hp.com*
[2] Cambridge University Engineering Dept: *email: ajr@eng.cam.ac.uk, jdmc2@eng.cam.ac.uk*
[3] now at Commerzbank Global Equities. *email:carl_seymour@CommerzbankZGE.com*

ABSTRACT

Two important components of a speech archiving system are the compression scheme and the search facility. We investigate two ways of providing these components. The first is to run the recogniser directly from the compressed speech – we show how even with a 2.4kbit/sec codec it is possible to produce good recognition results; but the search is slow. The second is to preprocess the speech and store the extra data in a compressed form along with the speech. In the case of an RNN-HMM hybrid system, the posterior probabilties provide a suitable intermediate data format. Vector quantizing these at just 625 bits/sec enables the search to run many times real-time and still maintain good recognition accuracy.

1. INTRODUCTION

There are a number of tools that are needed to make archiving reasonable quantities of speech (eg voicemail) viable. The two most important are the compression scheme and the search facility. There are numerous speech compression standards, any of which could be used, so this in itself is not an issue. As for the search, this can be achieved either by passing the speech through a large-vocabulary recogniser off-line, so that at search time only the transcription need be searched, or by searching the audio directly. The large-vocabulary recogniser approach is effective only if the word or phrase to search for was in the vocabulary of the recogniser, and the recogniser correctly transcribed it, and so is limited in its application. On the other hand, the direct search is potentially very slow.

The aims in choosing a compression scheme (i.e. speech codec) and search strategy should be to minimize storage and maximize search speed, whilst maintaining the intelligibility and subjective quality of the speech and the accuracy of the search. In this paper we consider two alternative approaches to achieving these aims. In the first approach, the search is performed on the compressed speech itself, which gives minimum storage. In the second approach, some processing of the speech takes place at record-time (or at least before performing the search), and the data produced is quantized and stored alongside the speech. This requires more storage but provides a faster search.

2. RECOGNITION FROM COMPRESSED SPEECH

The issues in recognising from compressed speech are:

Bandwidth. Speech compression standards are mostly telephone bandwidth, but recognition front-ends are wide-bandwidth (not an issue if the speech originated from the telephone). The reduction in bandwidth has a significant effect on the consonants of some speakers [1].

Speed. Decompression adds an extra step into the recognition process, slowing the search.

Performance. Even with good quality codecs like GSM @13kbit/sec, performance deteriorates slightly [2]. With much lower bit rate codecs, the performance is likely to deteriorate even more.

Our aim has been to resolve all these issues whilst maintaining a high level of compression. We use an enhanced 2.4kbit/sec LPC vocoder [3] as the baseline codec in all our experiments, which is the lowest quality and bit rate likely to be acceptable for speech archiving. The techniques and conclusions are applicable to speech compression schemes generally.

2.1 Wideband Extension

If the speech was recorded wideband (true for most broadcast material, but not for telephone-derived speech), we want to encode the 4-8kHz region as well as the standard 0-4kHz region in order to get the benefit for recognition of the extra information. The best way of doing this is to split the 8kHz band in two, encode the lower band with the chosen compression scheme, and then encode the upper band separately. Most of the important spectral information in the upper band can be encoded with a

second order (single pole) LPC analysis [3], thus only two coefficients and an energy value are needed.

Having encoded the upper band to aid the recogniser, we would also like to use the information to enhance the speech playback. In the case of the LPC vocoder this is a matter of combining the two bands into one LPC filter and then using a wideband excitation signal (see [3] for more details). For a waveform codec, synthesizing the upper band is more problematic without using a lot more bits to encode the waveform in the upper band as well. In [4] a VQ (vector quantizer) scheme operating at 0.025 bits/sample is used with some success. A better approach is explained in [5] where the upper band is synthesized purely with white noise, but attenuated when the speech is voiced.

2.2 Speed and Accuracy Improvement using a Direct Codec to Recogniser Interface

Almost all speech compression schemes encode spectral information. To speed up the recognition, we can transform that spectral information into the parameterisation of the recogniser's front-end without going through the decompression process.

This also means that the encoding of the spectral information is only done once. Since deriving the spectral information inevitably means using overlapping windows, doing it twice (once for the codec and once for the recogniser) introduces smoothing. With a vocoder the unspecified phase of the synthesized voicing signal introduces additional variability. A direct interface avoids all this.

In most codecs, including the 2.4kbit/sec codec we use, the spectral information is in the form of LPC coefficients. From these a power spectrum can be derived as follows:

$$P(\omega) = \frac{g^2}{\left|1 - \sum_{n=1}^{p} a(n) e^{-j\omega n}\right|^2}$$

where $a(n)$ and g are the LPC coefficients and gain respectively for a frame of speech and p is the LPC model order. This in effect takes a fourier transform of the impulse response of the LPC (all-zero) prediction filter, and then reciprocates the spectrum to derive the power spectrum of the LPC (all-pole) vocal tract filter.

For a wideband codec with split bands, this needs to be done for the lower *and* upper band and then the two spectra joined together. From this power spectrum the front-end parameters are computed in the normal way. We use MFCCs, which are derived directly from the power spectrum, but as most front-ends represent the data as a power spectrum at some stage in the processing the technique is fairly generic.

2.3 Performance Tests

In all our experiments we kept the bit-rate fixed at 2.4kbits/sec. This bit-rate is a good compromise between bit-rate and quality for the codec, so it should be possible to get good recognition performance without reverting to higher bit-rates. Given a fixed bit-rate, we were able to adjust the frame-rate, having a high frame-rate and coarse quantization or a lower frame-rate and more accurate quantization. As the codec uses interframe prediction of the LSP coefficients, the higher frame-rates do not increase the quantization noise proportionately.

The recogniser we used is the Abbot Large Vocabulary recogniser [6], which is a hybrid RNN-HMM system operating at a frame period of 16ms. As the issues we wanted to explore did not depend on the speech database, or even the particular recognition task, we used an experimental setup that already existed for other work. This setup used the Resource Management (RM) database, and achieved an error rate of 5.7% with the particular training and testing sets used.

First of all, we tested the LPC-MFCC interface without quantization, to find out if any information was being lost by constraining the front-end to fit an LPC model. The results are shown in Table 1.

Train Format	**Test Format**	**Frame Size**	**Error Rate**
Direct MFCC	Direct MFCC	16ms	5.7%
Direct MFCC	Unquantized LPC-MFCC	16ms	13%
Unquantized LPC-MFCC	Unquantized LPC-MFCC	16ms	5.5%

Table 1: Comparison of direct MFCC and LPC-MFCC front-ends

Clearly the LPC-derived MFCCs provide a different parameterisation to the direct MFCCs as the error rate is high when training and testing are not the same. But when they are the same, the two formats give almost exactly the same performance. This suggests that no important information is being lost by imposing an LPC model on the spectrum.

We then introduced quantization at 2.4kbit/sec, and tried both 16ms and 22.5ms frame sizes. The results are shown in Table 2.

Train Format	Test Format	Frame Size	Error Rate
Unquantized	Unquantized	16ms	5.5%
Unquantized	2.4kbit/sec	16ms	7.1%
2.4kbit/sec	2.4kbit/sec	16ms	7.6%
Unquantized	Unquantized	22.5ms	10.7%
Unquantized	2.4kbit/sec	22.5ms	11.2%
2.4kbit/sec	2.4kbit/sec	22.5ms	13.4%

Table 2: Performance of LPC-MFCC front-end

The larger frame-size (and therefore more accurate spectrum) reduces the degradation due to quantization, but at the expense of a very significant increase in the overall error rate. The frame-size of the spectral data must be kept low – it would seem that the recogniser is more sensitive than human listeners to the time-smoothing of the spectrum. The increase in error from 5.5% to 7.1% at the smaller frame-size seems an acceptable degradation in view of the very low bit rate of the codec.

The better performance when the recogniser is trained on unquantized LPCs may seem strange, as one would expect best performance when training and testing are matched. But the quantization process does not add noise in any acoustic sense, rather it introduces a random distortion to the spectrum. So for a limited amount of training data, the undistorted spectrum provides better models of the quantized spectrum than the quantized spectrum itself could.

2.4 Other Compression Schemes

Compression schemes operating at higher bit-rates are what is termed "waveform codecs" – unlike the LPC vocoder used in our experiments, they aim to reproduce the speech waveform, not just its spectral and voicing characteristics. However, they still use an LPC analysis, but the LPC coefficients are used only as predictors for the waveform coding, and do not have to be as accurate as they would be for a vocoder which depends completely on the LPC parameters for reproducing the speech. They are usually calculated at quite a high frame period – around 30ms.

But with modification any waveform codec using LPC analysis could be made to perform at least as well as the LPC vocoder used in these experiments by making these adjustments:

1) increase the frame-size of the LPC analysis and coefficient encoding to match that of the recogniser (10-16ms). Using inter-frame prediction prevents a significant rise in the bit rate.
2) ensure adequate quantization of LSPs. Because this is a small proportion of the overall bit rate in a waveform codec, it may be better to use more bits than we did with the LPC vocoder.
3) extend the codec to wideband using the generic split-band scheme in [5].
4) retrain recogniser on coded (but if possible unquantized) speech.

3. RECOGNITION FROM PARTIALLY PROCESSED SPEECH

A number of studies have been carried out recently into quantizing front-end parameters [2,7]. The motivation is for client-server speech recognition, but the same principles apply to the searching of stored speech. These show that at about 5kbit/sec, cepstral parameters can be quantized without any degredation in recognition performance. If the speech is compressed at a reasonably high rate, around 16kbit/sec, the extra storage needed for the front-end parameters is not all that significant.

In a hybrid RNN-HMM system, an alternative to encoding the front-end parameters is to quantize and store the posterior phone probabilities. This has the attraction of being much further on in the processing pipeline than the front-end computations, so enabling a much quicker search. On a dual-processor 500MHz PC with 512M RAM and running Linux 2.1, we can search for 10 keywords at 300 times real time and 100 keywords at 100 times real time. However, even with the 255-level encoding of the log probabilities used within Abbot, 22.5kbps are needed to store the 45 probabilities. We are interested in reducing this to around 600bits/sec, to allow speech and phone posteriors to both be stored inside 3kbits/sec in the baseline system. To get close to this target, vector quantization is needed. A 1024 entry codebook gives a data rate of just 625bits/sec.

The issue for the vector quantization process is the distance measure. Using raw probabilities, the greatest emphasis is given to the most probable phones. But the lower probability phones are often the correct ones, and these need to be encoded properly. Using log probabilities instead would work well except it would give the *very* small probability phones equal weight to all the others, even though they are rarely correct.

To solve this, we transformed the probabilities before computing distances using a *log(1+αx)* function, with α varying from 10 to 1000. This gives all the benefits of logs but prevents very small probabilities contributing to the distance.

As in the previous section, we used an experimental setup that already existed for other work, as the issue of quantization is unlikely to be affected by either the database or the recognition task. This time the WSJCAM0 read-speech database was used for training the vector quantizer, and the SQALE test set of 200 sentences used for testing. The baseline word error rate was 20.8%.

The results for the basic distance measures are shown in Table 3 and for the *log(1+ αx)* transformation in Table 4.

	% error
Baseline	20.8
Linear VQ	25.4
Log VQ	26.7

Table 3: Effect of VQ coding of posterior probabilities

α:	**% error**
10	23.8
30	23.6
100	23.1
300	23.7
1000	24.0

Table 4: VQ using *log(1+αx)* transform

With the *log(1+ α x)* transformation the results were better for all values of α. A value of $\alpha = 100$ gave the best performance, with just 11% increase in error rate.

The 600 bit/sec target is a very aggressive one, set so low because we don't want to increase the codec bit rate of 2.4kbits/sec by very much. If a higher bit-rate speech codec were being used, more bits could be allocated to the posterior probabilities.

4. SUMMARY

In this paper we have demonstrated two solutions to the problem of compressing speech in a way that allows audio searching. Firstly, we have shown that it is feasible to perform an audio search on the compressed speech itself even at bit rates as low as 2.4kbits/sec. The key enablers for this are:

1) Direct codec to recogniser interface by transforming the LPC coefficients to a power spectrum
2) Setting the LPC analysis frame-rate to match that of the recogniser
3) Wideband extension of the codec
4) Retraining the recogniser on coded, but if possible unquantized, speech.

It is straightforward to adapt any LPC-based compression scheme to give good recognition performance by incorporating these features.

As the speed of search is slow when recognising directly from the compressed speech, we consider as as an alternative storing the phone posterior probabilities of the RNN-HMM recognition system along with the speech, to avoid having to compute these at recognition time. So as not to increase the storage significantly, we have developed a very low bit-rate encoding based on vector quantization which uses 625 bits/sec. Even at such a low bit-rate, the error rate increases by just 11%. On a 500MHz PC, 100 words can be searched for at 100 times real time.

Further work will concentrate on eliminating the increase in error rate completely, and verifying the robustness of both approaches to acoustic background noise.

5. REFERENCES

[1] "Speech Enhancement for Bandlimited Speech", *D.A.Heide and G.S.Kang*, ICASSP98, Vol 1, pp 393-396

[2] "Quantization of cepstral parameters for speech recognition over the WWW", V.Digalakis, L.G.Neumeyer and M.Perakakis, ICASSP98, Vol 2, pp 989-992

[3] "A low-bit-rate speech coder using adaptive line spectral frequency prediction", *C.W.Seymour and A.J.Robinson*, Eurospeech97, pp 1319-1322

[4] "Wideband Speech Coding in 7.2 KB/S", *C.McElroy, B. Murray and A.D.Fagan,* ICASSP93, pp II-620 – II-623

[5] "Low Bit-Rate Frequency Extension Coding", *R.C.F.Tucker*, "Audio and music technology: the challenge of creative DSP", IEE Colloquium, 18 Nov 98, pp3/1-3/5

[6] "The Use of Recurrent Neural Networks in Continuous Speech Recognition", *A.J.Robinson, M.M.Hochberg and S.J.Renals*, ch. 19 in "Automatic Speech and Speaker Recognition", Kluwer 1995

[7] "Compression of acoustic features for speech recognition in network environments", G.N.Ramaswany and P.S.Gopalakrishnan, ICASSP98, Vol 2, pp 977-980