The Cerevoice Blizzard Entry 2007: Are Small Database Errors Worse than Compression Artifacts?

Matthew P. Aylett[1], *J. Sebastian Andersson*[2], *Leonardo Badino*[2], *Christopher J. Pidcock*[1]

[1]CereProc Ltd, Edinburgh, UK
[2] CSTR, University of Edinburgh, UK
matthewa@cereproc.com

Abstract

In commercial systems the memory footprint of unit selection systems is often a key issue. This is especially true for PDAs and other embedded devices. In this years Blizzard entry CereProc®gave itself the criteria that the full database system entered would have a smaller memory footprint than either of the two smaller database entries. This was accomplished by applying speex speech compression to the full database entry. In turn a set of small database techniques used to improve the quality of small database systems in last years entry were extended. Finally, for all systems, two quality control methods were applied to the underlying database to improve the lexicon and transcription match to the underlying data.

Results suggest that mild audio quality artifacts introduced by lossy compression have almost as much impact on MOS perceived quality as concatenation errors introduced by sparse data in the smaller systems with bulked diphones.

Index Terms: speech synthesis, unit selection.

1. Introduction

CereVoice®is a unit selection speech synthesis SDK produced by CereProc Ltd., a company founded in late 2005 with a focus on creating characterful synthesis and massively increasing the efficiency of unit selection voice creation.

CereProc regards Blizzard as an important element of our development program allowing us to field test prototype systems under extensive and thorough evaluation criteria. This years entry focused on the issue of memory footprint and in particular two approaches to reducing memory footprint in unit selection systems:

1. Waveform compression using CELP coding.
2. Database reduction supported by bulking and script selection.

All three systems entered were prototype systems[1]. All three systems used the same unit selection paradigm as described in [1, 2, 3]. Differences between the systems were restricted to the data used to build them and the use of waveform compression to reduce footprint size. No manual intervention was carried out at any point in the synthesis process and all systems were operated completely automatically with the exception of adding out of vocabulary words present in the test sentences to our lexicon before synthesising the final results.

[1]For access to a demo of our current commercial offering please contact info@cereproc.com

The database was segmented using the CereProc Voice Building Kit. Two semi-automatic quality control techniques were applied to the data to ensure transcriptions used to segment the speech matched the underlying database which we termed *vocalic lexical matching* and *frequent diphone checking*.

The three systems that we entered took the following form:

Full database system (A) This system was built with the full ATR blizzard database. (78.7k words, 295.6k phones, 9.48 hours of audio, 7.36 hours of phonetic material with silence excluded). The main difference between this system and CereProc's commercial offering was the application of Speex [4] compression encoding to reduce the total size of the audio in the database from 1002Mb to 122Mb and the omission of pitch smoothing during concatenation.

Arctic database system (B) This system was built with the Arctic subset of the ATR database system. (12.8k words, 48.3k phones, 1.22 hours of audio, 1.16 hours of phonetic material with silence excluded). The bulking approach we applied to our small footprint system last year [5], was extended to increase the number of synthetic diphones added to the system. The system was identical to (A) except waveforms were stored uncompressed avoiding any possible compression artifacts.

Selected database system (C) A set of utterances were selected from the full ATR database so that the total audio time was less than or equal to the Arctic database size based on an utterance metric supplied by Blizzard organisers (11.8k words, 43.8k phones, 1.17 hours of audio, 0.76 hours minutes of phonetic material with silence excluded). In accordance with the rules this selection was carried out purely on the transcription of the utterances and was not allowed to make use of any audio information. As per system (B) extended bulking was applied to the selected database. However, in contrast to system (B), utterances were selected partly on the knowledge of what bulking heuristics could be applied during voice building.

2. Overview of the system

CereVoice is a faster-than-realtime diphone unit selection speech synthesis engine, available for academic and commercial use. The core CereVoice engine is an enhanced synthesis 'back end', written in C for portability to a variety of platforms. The engine does not fit the classical definition of a synthesis back end, as it includes lexicon lookup and letter-to-sound rule modules, see Fig. 1. An XML API defines the input to the engine. The API is based on the

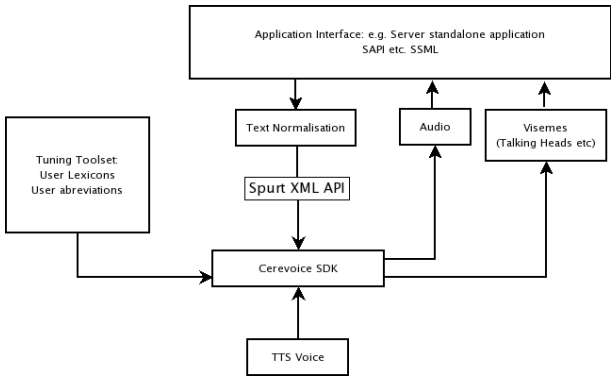

Figure 1: *Overview of the architecture of the CereVoice synthesis system. A key element in the architecture is the separation of text normalisation from the selection part of the system and the use of an XML API.*

principle of a 'spurt' of speech. A spurt is defined as a portion of speech between two pauses.

To simplify the creation of applications based on CereVoice, the core engine is wrapped in higher level languages such as Python using Swig. For example, a simple Python/Tk GUI was written to generate the test sentences for the Blizzard challenge.

The CereVoice engine is agnostic about the 'front end' used to generate spurt XML. CereProc use a modular Python system for text processing. Spurt generation is carried out using a greedy incremental text normaliser. Spurts are subsequently marked up by reduction and homograph taggers to inform the engine of the correct lexical variant dependent on the spurt context.

3. Processing the Blizzard Data

A preliminary check of the text files provided by Blizzard suggested that most of the text had been pre-normalised. However, there were still problems with data ambiguity in some areas. The word 'corp's' is read once as 'corp's' and at other times as 'corporation's'. Upper-case words were neither predictably spelt out as letters, or spoken as words. For example, 'GAP' and 'LAN' were read 'G A P' and 'L A N', with 'TOEIC' read as a word. A hand created list of the latter type of word was created, and all other upper case strings split into individual letters. The one exception to this rule was 'II', which was mapped to 'two'.

Two semi-automatic techniques were applied to improve the quality of the underlying database.

3.1. Vocalic Lexical Matching

A common source of variation in general American (GA) accents is caused by differences in the use and pronunciation of two low vowels /ɒ/ as in "lawn" and /ɑ/ as in "father", the correct transcription of short unstressed vowels /əɪʌ/, confusion between long /i/ as in "heat" and the short front vowel /ɪ/ as in "hit" and the long back vowel /ɑ/ and the short unrounded mid front vowel /æ/.

Although some of this variation is free, incorrectly transcribed vowels are a major source of error in unit selection systems, especially those based on large databases.

Resources were not available to check all the transcriptions of the database by hand. Instead a semi automatic approach was taken.

1. For each vowel in question the distribution of the F1/F2 formant values taken from the centre point of the vowel (as segmented by the CereProc Voice Building Kit), was plotted and visually examined. Where distinct clusters were clearly apparent the mean for the transcribed vowel was taken as the mean of the largest cluster. For the /ɒɑ/ comparison this was clearly the case while for the short vowels there was no clear clustering of data. For each vowel (or sub cluster) the standard deviation was calculated.
2. For each vowel pair that could be mis-transcribed, a list of words with these vowels were selected where the normalised mean of the vowel was closer to the mis-transcribed mean than its transcribed mean.
3. The proportion of each word that fulfilled this condition was calculated.
4. Words with the highest proportions for the mis-trancription vowel pair were entered into a listening test. If no incorrect transcriptions were detected for the first 20 instances, the vowel pair was abandoned as a source of mis-transcriptions.

Results varied extensively between vowel pairs. For example the /ɑ/ vs /æ/ distinction was 100% correct in determining incorrect transcriptions for "iraq", "hiragana" and "nagai". In contrast the /ɪ/ vs /ʌ/ distinction resulted in 0% found mistrancriptions (in this case most items were selected due to /l/ colouring.

In total the full listening process took 2 man hours and resulted in changes to 158 word transcriptions.

We are considering how this technique might be improved by using more complex representations of the acoustic distributions and, in particular, taking account of known F1/F2 disruption caused by phonetic context.

3.2. Frequent Diphone Checking

Our second semi-manual quality check was based on the observation that the extent individual diphones are used during synthesis varies greatly [6].

Half a million phrases were synthesised by our system resulting in the selection of approximately 20 million diphones. These diphones were then ordered by frequency. The top 313 frequent diphones were then entered into a listening test in order to ensure appropriate transcription and segmentation. These top frequency diphones accounted for 2.5% of all diphones generated.

The listening process took one man hour and resulted in a single correction of a mis-segmentation caused by an inaccurate transcription. However for our system, these top frequency diphones were correctly segmented and transcribed.

It is likely that common phones will generate good segmentation models, thus likely diphones may be better transcribed than unlikely ones. Given the failure to find serious errors with this approach we are unlikely to pursue it as a means of quality assurance.

4. Voice Bulking

Voice bulking is the creation of new units from existing non-contiguous, but probably well matched, sections of speech. The result is a new diphone created from two demiphones (half-diphones). This differed from approaches such as [7] in that the required diphones were synthesised offline and used to *bulk-up* the small database. See [5] for details on how bulking was implemented in the Blizzard 2006 CereProc entry.

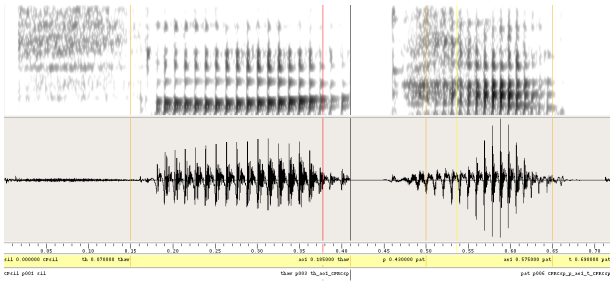

Figure 2: *Example of a bulked diphone constructed by concatenating the word 'thaw' to 'pat' offline. The resulting diphone is then added to the small database to reduce sparsity.*

The differences from last years bulking system were as follows:

1. In the 2006 entry bulking was restricted to cross word diphones, this restriction was lifted in the 2007 entry.
2. In the 2006 entry bulked diphones were only used if no diphones of that type were present in the database. In the 2007 entry this was extended to diphone left right contexts, stress and phrase final position.
3. Because of the possible explosion of added diphones, a manually chosen limit of number of bulked diphones was selected (in this case 3000). A greedy algorithm was used to order bulked diphones by the number of contexts they covered, and by how well matched the left and right half phones appeared to be.
4. The number of contexts was increased to include diphones with initial and final pauses as half phones.
5. For each context a heuristic for determining actual join location between desired half phones was implemented. This heuristic was dependent on which half was required. i.e. getting a fricative from a fricative-vowel used a different algorithm than getting the vowel from the same diphone. In general, bulked diphones had co-articulated material removed where possible.

4.1. The Algorithm in More Detail

We term the initial half phone of the diphone P1, and the final half phone P2. Fig. 2 shows an example of a bulked ɒ-p' diphone made from concatenating 'thaw' to 'pat'. The aim of the bulking is to construct a new diphone with these two half phones which reduces the sparsity of the database. The input to the process is an ascii dump of the entire unit selection database which is used to determine feature sparsity. Once bulked diphones have been selected in order to recude this sparsity, a second process uses the timings generated from this timing to generate the actual audio for the diphones based on the original speech audio.

An XML file is used to configure the system and consists of:

- A hierarchy of features associated with a score for how badly they contribute to sparsity. For example diphone identity is always top of this hierarchy but whether we then prefer to reduce sparsity for stress or left/right context can be configured. These features are used to generate a set of sparse feature matrices of increasing dimensionality. The requirement for a diphone is then calculated by the extent it fills these matrices.
- The features are marked as to whether they apply to P1 or P2 (*parity* - our term). For example it is the P2 which holds the stress of the second half of the diphone.
- Feature weight modifiers are specified. These modify the overall score of filling sparsity by specific feature values. For example we prefer diphones which cross a word boundary because they are easier to split. Thus we add a score if the word boundary feature is true. Thus if we have two possible candidates for a bulked diphone we will prefer the one crossing a word boundary.
- Bulking heuristics are then specified. These have an ID and a regular expression for choosing P1 and P2 given a set of feature values. For example we have a heuristic *pausep1*. This means we can construct diphones with P1 silence and P2 some other phone. In this heuristic we restrict the P2 phone to be a liquid with an initial P1 context which is an unvoiced fricative and with a word boundary. In effect the first part of the specification says what we need whereas the bulking heuristics limit the means we can use to satisfy this need.

Potential half phones are then listed for each bulking heuristic, a score is generated based on how well the diphone fills the sparse matrices and how well the half phones join together (based on energy and f0 matching). The top scoring bulked diphones across all heuristics are then used as candidates for the bulked database.

Once the new diphones have been selected the system then uses, for each bulking heuristic, an algorithm to extract the half phones from the original speech and concatenate them offline. This new audio is then added to the unit selection database.

We are in the early stages of evaluating the effect of bulking. An informal listening test of generated diphones suggests they rarely include discontinuities, mostly because the selection of candidates and their concatenation is conservative. However we have not yet carried out formal evaluation to determine how much this bulking technique improves the small domain voices.

5. Script Selection

Script selection was applied to the full ATR blizzard database to produce a sub-set of the data equal to or less than the size of the audio used in the Arctic portion of the database. The objective was to choose a sub-set of material which would give better coverage and reduce sparsity for the small voice (C). An advantage for our selection process was that it could interact with bulking. That is, diphones which would add to coverage and also add to the bulking potential were scored higher than ones that could not be bulked.

The chosen set contained 1342 distinct diphones based on the CereProc GA pronunciation of the database (compared to 1307 in Arctic). However the script selection also attempted to cover stress, and phrase break contexts as well. Taking these into account the chosen set contained 4819 distinct diphones as opposed to 3794 in Arctic. With potential bulking this set increased to 5636 with 36% of these potentially bulked diphones occurred in the full database.

6. Compression

Voice (A) stored its audio in compressed format using the public Speex library. The audio was compressed at a Speex quality level

of 6 with variable encoding switched on. The result was to reduce voice size (and memory footprint) by 88%. Artifacts from the compression are audible with a loss of sharpness and the faint addition of background noise.

7. Results

The effect of compression artifacts on the overall results for Voice (A) was much greater than we anticipated. The effects are more strongly noticeable when comparing the results for 'similarity to speaker' with 'mos score'. Only four systems showed a higher MOS score than similarity measure. The prototype CereVoice full voice, and systems J,M,N. Without being able to listen to these systems it is difficult to ascertain if general acoustic quality was also a problem for these systems. The two smaller voices with linear output both followed the normal pattern of having higher similarity scores than MOS scores. In addition they were both rated higher for similarity than the full voice despite an overall slightly lower MOS score.

Without carrying out direct comparison tests it is difficult to establish the the extent the Speex compression had an effect on the overall MOS score but we would estimate an effect of somewhere between -0.5 and -1.0.

For the smaller voices we are pleased that our Voice (C) did better than (B) suggesting the combination of bulking and script selection performed well. However again, it is hard to assess small database performance without knowing the make up of the systems which did better, arguably systems O and A.

8. Conclusion

The paper posed the question of whether unit selection small database artifacts were more problematic than compression artifacts. To a certain extent the finding support the results we have seen for parametric systems over the last couple of years. Although in these previous systems general audio quality was lower due to vocoder artifacts, the absence of critical errors caused by data sparsity had a more severe effect on small database results. Thus our full voice with compression did perform better than the small voices however the effect on voice similarity was marked.

This raises an important issue with regards to MOS testing. We really don't know what a five point mean opinion score response to the question 'Does this sound natural?' is testing. In a commercial environment the evaluation is much less thorough, more subjective and in many ways much more brutal. Listening to the Speex compression our subjective evaluation was that the quality deterioration was minimal and would not have a major impact on the results, this was almost certainly not the case. There is also anecdotal evidence that listeners adapt to these changes quite rapidly over a number of sentences. This is a problem for developers, where we can no longer 'hear' the problem, but perhaps an advantage for applications where listeners adapt and cease to be bothered by these artifacts. perhaps more useful than a MOS scale of naturalness would be a scale of how annoying synthetic speech is. After all that is what our customers are most concerned about.

9. References

[1] A.J. Hunt and A.W. Black, "Unit selection in concatanative speech synthesis using a large speech database," in *ICASSP*, 1996, vol. 1, pp. 192–252.

[2] Robert A.J. Clark, Korin Richmond, and Simon King, "Festival 2 - build your own general purpose unit selection speech synthesiser," in *5th ESCA Workshop in Speech Synthesis*, 2004, pp. 147–151.

[3] John Kominek, Christine L. Bennet, Brian Langer, and Arthur R. Toth, "The Blizzard challenge 2005 CMU entry - a method for improving speech synthesis systems," in *INTERSPEECH*, 2005, pp. 85–88.

[4] J.-M. Valin and C. Montgomery, "Improved noise weighting in celp coding of speech - applying the vorbis psychoacoustic model to speex," in *AES*, 2006.

[5] Matthew P. Aylett and Christopher J. Pidcock, "The cerevoice characterful speech synthesiser sdk," in *AISB*, 2007, pp. 174–8.

[6] Peter Rutten, Matthew Aylett, Justin Fackrell, and Paul Taylor, "A statistically motivated database pruning technique for unit selection synthesis," in *ICSLP*, 2002, pp. 125–8.

[7] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T next-gen TTS system," in *Joint Meeting of ASA, EAA, and DAGA*, 1999.