AUTOMATICALLY CLUSTERING SIMILAR UNITS FOR UNIT SELECTION IN SPEECH SYNTHESIS.

Alan W Black and Paul Taylor

Centre for Speech Technology Research, University of Edinburgh,
80, South Bridge, Edinburgh, U.K. EH1 1HN
http://www.cstr.ed.ac.uk
email: awb@cstr.ed.ac.uk, Paul.Taylor@ed.ac.uk

ABSTRACT

This paper describes a new method for synthesizing speech by concatenating sub-word units from a database of labelled speech. A large unit inventory is created by automatically clustering units of the same phone class based on their phonetic and prosodic context. The appropriate cluster is then selected for a target unit offering a small set of candidate units. An optimal path is found through the candidate units based on their distance from the cluster center and an acoustically based join cost. Details of the method and justification are presented. The results of experiments using two different databases are given, optimising various parameters within the system. Also a comparison with other existing selection based synthesis techniques is given showing the advantages this method has over existing ones. The method is implemented within a full text-to-speech system offering efficient natural sounding speech synthesis.

1. BACKGROUND

Speech synthesis by concatenation of sub-word units (e.g. diphones) has become basic technology. It produces reliable clear speech and is the basis for a number of commercial systems. However with simple diphones, although the speech is clear, it does not have the naturalness of real speech. In attempt to improve naturalness, a variety of techniques have been recently reported which expand the inventory of units used in concatenation from the basic diphone schema (e.g. [7] [5] [6]). There are a number of directions in which this has been done, both in changing the size of the units, the classification of the units themselves, and the number of occurrences of each unit.

A convenient term for these approaches is *selection based synthesis*. In general, there is a large database of speech with a variable number of units from a particular class. The goal of these algorithms is to *select* the best sequence of units from all the possibilities in the database, and concatenate them to produce the final speech.

The higher level (linguistic) components of the system produce a *target specification*, which is a sequence of *target units*, each of which is associated with a set of features. In the algorithm described here the database units are phones, but they can be diphones or other sized units. In the work of Sagisaka et al. [9], units are of variable length, giving rise to the term *non-uniform unit* synthesis. In that sense our units are *uniform*. The features include both phonetic and prosodic context, for instance the duration of the unit, or its position in a syllable. The selection algorithm has two jobs: (1) to find units in the database which best match this target specification and (2) to find units which join together smoothly.

2. CLUSTERING ALGORITHM

Our basic approach is to cluster units within a unit type (i.e. a particular phone) based on questions concerning prosodic and phonetic context. Specifically, these questions relate to information that can be produced by the linguistic component, e.g. is the unit phrase-final, or is the unit in a stressed syllable. Thus for each phone in the database a decision tree is constructed whose leaves are a list of database units that are best identified by the questions which lead to that leaf.

At synthesis time for each target in the target specification the appropriate decision tree is used to find the best cluster of candidate units. A search is then made to find the best path through the candidate units that takes into account the distance of a candidate unit from its cluster center and the cost of joining two adjacent units.

2.1. Clustering units

To cluster the units, we first define an acoustic measure to measure the distance between two units of the same phone type. Expanding on [7], we use an acoustic vector which comprises Mel frequency cepstrum coefficients, F_0, power, and delta cepstrum, F_0 and power. The acoustic distance between two units is simply the average distance for the vectors of all the frames in the units plus X% of the frames in the previous units, which helps ensure that close units will have similar preceding contexts. More formally, we use a weighted mahalanobis distance metric to define the acoustic distance

$Adist(U,V)$ between two units U and V of the same phoneme class as

$$\text{if } |V| > |U| \quad Adist(V,U)$$

$$\frac{WD*|U|}{|V|} * \sum_{i=1}^{|U|} \sum_{j=1}^{n} \frac{W_j.(abs(F_{ij}(U) - F_{(i*|V|/|U|)j}(V)))}{SD_j * n * |U|}$$

where $|U|$ is number of frames in U, $F_{xy}(U)$ is parameter y of frame x of unit U, SD_j is the standard deviation of parameter j, W_j is weight for parameter j. This measure gives the mean weighted distance between units with the shorter unit linear interpolated to the longer unit. WD is the duration penalty weighting the difference between the two units' lengths.

This acoustic measure is used to define the *impurity* of a cluster of units as the mean acoustic distance between all members. The object is to split clusters based on questions to produce a better classification of the units. A CART method [2] is used to build a decision tree whose questions best minimise the impurity of the sub-clusters at that point in the tree. A standard greedy algorithm is used for building the tree. This technique may not be globally optimal but a full global search would be prohibitively computationally expensive. A minimum cluster size is specified (typically between 10-20).

Although the available questions are the same for each phone type, the tree building algorithm will select only the questions that are significant in partitioning that particular type. The features used for CART questions include only those features that are available for target phones during synthesis. In our experiments these were: previous and following phonetic context (both phonetic identity and phonetic features), prosodic context (pitch and duration including that of previous and next units), stress, position in syllable, and position in phrase. Additional features were originally included, such as delta F_0 between a phone and its preceding phone, but they did not appear as significant and were removed. Different features are significant for different phones, for example we see that lexical stress is only used in the phones `schwa`, `i`, `a` and `n`, while a feature representing pitch is only rarely used in unvoiced consonants.

The CART building algorithm implicitly deals with sparseness of units in that it will only split a cluster if there are sufficient examples and significant difference to warrant it.

2.2. Joining units

To join consecutive candidate units from clusters selected by the decision trees, we use an optimal coupling [4] technique to measure the concatenation costs between two units. This technique offers two results: the cost of a join and a position for the join. Allowing the join point to move is particularly important when our units are phones: initial unit boundaries are on phone-phone boundaries which probably are the least stable part of the signal. Optimal coupling allows us to select more stable positions towards the center of the phone. In our implementation, if the previous phone in the database is of the same type as the selected phone we use a search region that extends 60% into the previous phone, otherwise the search region is defined to be the phone boundaries of the current phone.

Our actual measure of join cost is a frame based Euclidean distance. The frame information includes F_0, Mel frequency cepstrum coefficients, and power and their delta counterparts. Although this uses the same parameters as used in the acoustic measure used in clustering, now it is necessary to weight the F_0 parameter to deter discontinuity of local F_0 which can be particularly distracting in synthesized examples. Except for the delta features this measure is similar to that used in [7].

2.3. Selecting units

At synthesis time we have a stream of target segments that we wish to synthesize. For each target we use the CART for that unit type, and ask the questions to find the appropriate cluster which provides a set of candidate units. The function $Tdist(U)$ is defined as the distance of a unit U to its cluster center, and the function $Jdist(U_i, U_{i-1})$ as the join cost of the optimal coupling point between a candidate unit U_i and the previous candidate unit U_{i-1} it is to be joined to. We then use a Viterbi search to find the optimal path through the candidate units that minimizes the following expression:

$$\sum_{i=1}^{N} Tdist(U_i) + W * Jdist(U_i, U_{i-1})$$

W allows a weight to be set optimizing join cost over target cost. Given that clusters typically contain units that are very close, the join cost is usually the more important measure and hence is weighted accordingly.

2.4. Pruning

As distributing the *whole* database as part of a synthesis voice may be prohibitively large, especially if multiple voices are required, appropriate pruning of units can be done to reduce the size of the database. This has two effects. The first is to remove spurious atypical units which may have been caused by mislabelling or poor articulation in the original recording. The second is to remove those units which are so common that there is no significant distinction between candidates. Given this clustering algorithm it is easy (and worthwhile) to achieve the first by removing the units from a cluster that are furthest from its center. Results of some experiments on pruning are shown below.

The second type of pruning, removing overly common units, is a little harder as it requires looking at the distribution of the distances within clusters for a unit type to find what can be determined as, "close enough." Again this involves removal of those units furthest from the cluster center, though this is best done before the final splits in the tree, and only for the most common unit types.

As with all the measures and parameters there is a trade off between synthesis resources (size of database and time to select) verses quality, but it seems that pruning 20% of units makes no significant difference (and may even improve the results) while up to 50% may be removed without seriously degrading the quality. (Similar figures were also found in the work described in [7].)

3. EXPERIMENTS

Two databases have so far been tested with this technique, a male British English RP speaker consisting of 460 TIMIT phonetically balanced sentences (about 14,000 units) and a female American news reader from the Boston University FM Radio corpus [8] (about 37,000 units).

Testing the quality of speech synthesis is difficult. Initially we tried to score a model under some set of parameters by synthesizing a set of 50 sentences. The results were scored on a scale of 1-5 (excellent to incomprehensible). However the results were not consistent except when the quality widely differed. Therefore instead of using an *absolute* score we used a *relative* one, as it was found to be much easier and reliable to judge if an example was better, equal or worse than another than state its quality on some absolute scale.

In these tests we generated 20 sentences for a small set of models by varying some parameter (e.g. cluster size). The 20 sentences consisted of 10 "natural target" sentences (where the segments, duration and F_0 were derived directly from naturally spoken examples), and 10 examples of text to speech. None of the sentences in the test set were in the databases used to build the cluster models. Each set of 20 was played against each other set (in random order) and a score of better, worse or equal was recorded. A sample set was said to "win" if it had more better examples than another. A league table was kept recording the number of "wins" for each sample set thus giving an ordering on the sets.

In the following tests we varied cluster size, and F_0 weight in the acoustic cost, and the amount to prune final clusters. These full tests were only carried out on the male 460 sentence database.

For the cluster size we fixed the other parameters at what we thought were mid-values. The following table gives the number of "wins" of that sample set over the others.

	minimum cluster size				
	5	8	10	12	15
wins	1	0	4	3	2

Obviously we can see that when the cluster is too restrictive the quality decreases but at around 10 it is at its best and decreases as the cluster size gets bigger.

The importance of F_0 in the acoustic measure was tested by varying its weighting relative to the other parameters in the acoustic vector.

	F_0 acoustic weight			
	0.0	1.0	2.0	3.0
wins	1	3	2	0

This optimal value is lower than we expected but we believe this is because our listening test did not test against an original or actual desired F_0, thus no penalty was given to a "wrong" but acceptable F_0 contour, in a synthesized example.

The final test was to find the effect of pruning the clusters. In this case clusters of size 15 and 10 were tested, and pruning involved discarding a number of units from the clusters. In both cases discarding 1 or 2 made no perceptible difference in quality (though results actually differed in 2 units). In the size 10 cluster case, further pruning began to degrade quality. In the size 15 cluster case, quality only degraded after discarding more than 3 units. Overall the best quality was for the size 10 cluster and pruning 2 allows the database size to be reduced without affecting quality. The pruning was also tested on the f2b database with its much larger inventory. Best overall results with that database were found with pruning 3 and 4 from a cluster size of 20.

In these experiments no signal modification was done after selection, even though we believe that such processing (e.g. PSOLA) is necessary. We do not expect all prosodic forms to exist in the database and it is better to introduce a small amount of modification to the signal in return for fixing obvious discontinuities. However it is important for the selection algorithm to be sensitive to the prosodic variation required by the targets so that the selected units require only minimal modification. Ideally the selection scoring should take into account the cost of signal modification, and we intend to run similar tests on selections modified by signal processing.

4. DISCUSSION

This algorithm has a number of advantages over other selection based synthesis techniques. First the cluster method based on acoustic distances avoids the problem of estimating weights in a feature based target distance measure as described in [7], but still allows unit clusters to be sensitive to general prosodic and phonetic distinctions. It also neatly finesses the problem of variability in sparseness of units. The tree building algorithm only splits a cluster when there are a significant number and identifiable variation to make the split worthwhile. The second advantage over [7] is that no target cost measurement need be done at synthesis time as the tree effectively has pre-calculated the "target cost" (in this case simply the distance from the cluster center). This makes for more efficient synthesis as many distance measurements now need not be done.

Although this method removes the need to generate the target feature weights generated in [7] used in estimating acoustic distance there are still many other places in the model where parameters need to be estimated, particularly the acoustic cost and the continuity cost. Any frame based distance measure will not easily capture "discontinuity errors" perceived as bad joins between units. This probably makes it difficult to find automatic training methods to measure the quality of the synthesis produced.

Donovan and Woodland [5] use a similar clustering method, but the method described here differs in that instead of a single example being chosen from the cluster, all the members are used so that continuity costs may take part in the criteria for selection of the best units.

In [5], HMMs are used instead of a direct frame-based measure for acoustic distance. The advantage in using an HMM is that different states can be used for different parts of the unit. Our model is equivalent to a single state HMM and so may not capture transient information in the unit. We intend to investigate the use of HMMs as representations of units as this should lead to a better unit distance score.

Other selection algorithms use clustering, though not always in the way presented here. As stated, the cluster method presented here is most similar to [5]. Sagisaka et al. [9] also clusters units but only using phonetic information, they combine units forming longer, "non-uniform" units based on the distribution found in the database. Campbell and Black [3] also use similar phonetic based clustering but further cluster the units based on prosodic features, but still resorts to a weighted feature target distance for ultimate selection.

It is difficult to give realistic comparisons of the quality of this method over others. Unit selection techniques are renowned for both their extreme high quality examples and their extreme low quality ones, and minimising the bad examples is a major priority. This technique does not yet remove all low quality examples, but does try to minimise them. Most examples lie in the middle of the quality spectrum with mostly good selection but a few noticable errors which detract from the overall acceptability of the utterance. The best examples, however, are nearly indistinguishable from natural utterances.

This cluster method is fully implemented as a waveform synthesis component using the Festival Speech Synthesis System [1].

5. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of the UK Engineering and Physical Science Research Council (EPSRC grant GR/K54229 and EPSRC grant GR/L53250).

REFERENCES

[1] A. W. Black and P. Taylor. The Festival Speech Synthesis System: system documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK, January 1997. Avaliable at http://www.cstr.ed.ac.uk/projects/festival.html.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Pacific Grove, CA., 1984.

[3] N. Campbell and A. Black. Prosody and the selection of source units for concatenative synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in speech synthesis*, pages 279–282. Springer Verlag, 1996.

[4] A. Conkie and S. Isard. Optimal coupling of diphones. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in speech synthesis*, pages 293–305. Springer Verlag, 1996.

[5] R. Donovan and P. Woodland. Improvements in an HMM-based speech synthesiser. In *Eurospeech95*, volume 1, pages 573–576, Madrid, Spain, 1995.

[6] X. Huang, A. Acero, H. Hon, Y. Ju, J Liu, S. Meredith, and M. Plumpe. Recent improvements on microsoft's trainable text-to-speech synthesizer: Whistler. In *ICASSP-97*, volume II, pages 959–962, Munich, Germany, 1997.

[7] A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *ICASSP-96*, volume 1, pages 373–376, Atlanta, Georgia, 1996.

[8] M. Ostendorf, P. Price, and S. Shattuck-Hufnagel. The Boston University Radio News Corpus. Technical Report ECS-95-001, Electrical, Computer and Systems Engineering Department, Boston University, Boston, MA, 1995.

[9] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura. ATR – ν-TALK speech synthesis system. In *Proceedings of ICSLP 92*, volume 1, pages 483–486, 1992.