CMU Blizzard 2007: A Hybrid Acoustic Unit Selection System from Statistically Predicted Parameters

Alan W Black, Christina L. Bennett, Benjamin C. Blanchard, John Kominek,
Brian Langner, Kishore Prahallad and Arthur Toth

Language Technologies Institute,
Carnegie Mellon University, Pittsburgh, PA.
{awb,cbennett,bblancha,jkominek,blangner,skishore,atoth}@cs.cmu.edu

Abstract

This paper describes CMU's entry for the Blizzard Challenge 2007. Our eventual system consisted of a hybrid statistical parameter generation system whose output was used to do acoustic unit selection. After testing a number of varied systems, this system proved the best in our internal tests. This paper also explains some of the limitations we see in our techniques. The CMU system is identified as D in the result charts.

Index Terms: Speech Synthesis, Unit Selection, Statistical Parametric Synthesis, Hybrid Techniques.

1. Introduction

The purpose of the Blizzard Challenge is to compare and contrast different speech synthesis techniques and systems on a benchmarked database [8]. Since 2005, several universities and systems have participated in this challenge. This has led to congregation of several researchers on a common platform in Blizzard workshop to compare and contrast different synthesis techniques, with the goal to build naturally speaking synthesis systems.

In the previous Blizzard Challenges it was observed that the listeners preferred the intelligible, consistent speech produced by statistical parametric synthesizers as compared to natural but often inconsistent speech by unit selection techniques [11] [12]. Earlier Challenges were benchmarked on CMU ARCTIC databases which were typically of around one hour of speech recorded by native US speakers. One of the arguments for this was that unit selection synthesis techniques needed more than one hour of speech to produce natural and consistent speech. Thus the Blizzard Challenge 2007 was benchmarked on an 8 hour speech database recorded by a single speaker, which includes one hour of speech on ARCTIC utterances by the same speaker.

Given the larger speech databases, the teams (also referred to as sites) were asked to build the speech synthesis systems A, B, and C in the space of four weeks time, and then asked to synthesize a common set of sentences for perceptual evaluation. System A denotes the TTS system built from the whole of the database, B denotes the TTS system built from the ARCTIC subset, and C denotes the TTS system built from a site-defined subset. To avoid multiple submissions from a site, the sites were asked to submit their best system for A, B, and C to compare against those of other teams.

As a part of the Blizzard Challenge, we wanted to investigate techniques of generating a natural and consistent quality synthesis by a method of acoustic unit selection from statistically predicted parameters. We have built synthesis systems using the unit selection technique CLUNITS, the statistical parametric synthesis technique CLUSTERGEN, and also a hybrid technique of unit selection from statistically predicted parameters. An internal evaluation of the synthesis systems showed that the hybrid system produced consistent and natural speech and was perceived to be better than the CLUNITS and CLUSTERGEN systems. The hybrid system was submitted as our final system for A and B type comparisons. The remainder of this paper describes the details of the implementation and performance of CLUNITS, CLUSTERGEN, and hybrid systems on Blizzard datasets.

2. Development and Comparison of Synthesis Systems

For internal development and comparison purposes, we divided the Blizzard datasets into development data and held-out data. Every 10th utterance was held out to build a held-out data set. On the development data, three systems were built using unit selection, statistical parametric, and hybrid methods. During the build process each of these systems were tuned with different parameter settings. These systems were compared by doing perceptual listening tests amongst team members using AB comparison tests. The following sections describe the CLUNITS, CLUSTERGEN, and hybrid methods in detail.

2.1. CLUNITS

CLUNITS is a cluster-based unit selection technique [1]. It has been standard in the Festival distribution for many years. It has been especially useful in limited domain synthesis where the input sentences are similar to the database sentences.

The unit selection system relies on acoustic-based clusters of same-typed units using CART, allowing a model that indexes clusters of similar units with high level symbolic features such as phonetic, metrical and prosodic context. Using an offline clustering technique reduces the amount of computation that is necessary in the more standard Hunt and Black target/join cost [9], in that the desired cluster, which can be quickly indexed by asking a few CART questions, is less computationally expensive than calculating a full target cost for each candidate.

In CLUNITS, the units are clustered by type. By default this is phone name. In our Blizzard 2007 tests we additionally partitioned our type further, tagging vowels with stress value (0 or 1) and consonants with onset/coda. Additionally we tagged our units with a crude voiced/unvoiced/consonant/vowel four-way value derived from the previous unit in the databases. Thus, this was not a full diphone tagging, but a limited form of that. In previous databases, these additional tags have proven to be useful.

2.2. CLUSTERGEN

CLUSTERGEN is a statistical parametric synthesizer released as part of the Festival distribution [2]. It predicts frame-based MFCCs clustered using phonetic, metrical, and prosodic contexts. Unlike CLUNITS, the unit size is one frame (5ms by default), and the signal is partitioned at the HMM-state size level (3 states per phone). The clustering, done via CART, optimizes the standard deviation of the frames in the cluster. The frames are 24 coefficient MFCCs plus F0. CLUSTERGEN offers a number of options for clusters which can be single frames, trajectories, or trajectories with overlap and add. We used the simplest model for our Blizzard experiments. Synthesis is done by predicting the HMM-state durations, then predicting each frame with the appropriate CART tree. The track of MFCC plus F0 vectors is re-synthesized with the MLSA algorithm [3], as implemented in the HTS system and already implemented within Festival. We experimented with post filtering in the MLSA filter, which essentially performs liftering of MFCCs to emphasize the formant structure. As shown in Table 1, we did not find liftering of MFCC to improve the synthesis quality in CLUSTERGEN.

2.3. Parametric Trajectory Target Unit Selection

For this year's Blizzard Challenge, our entry is a hybrid unit-selection synthesizer. It differs from the CLUNITS implementations in that instead of performing pre-selection based on acoustic clustering and minimizing join costs at runtime, the target cost is minimized at runtime. The target cost is real-valued instead of 0/1, i.e. the clustering is “soft” not “hard.” Only minimal pre-selection is performed based on the phoneme string. Also, in contrast to unit selection schemes that employ prosodic target costs [4], our representation of the target is not symbolic, but numeric – it is a time sequence of real numbered vectors. The evaluation of target costs is computed at the frame level (with a 5 ms step size) and averaged over the duration of each unit. The frame-level representation is conventional: 25D Mel-scale cepstral vectors (MFCCs) augmented with F0, as is used in CLUSTERGEN.

Of course, the trajectory target used for unit selection has to come from somewhere. In the case of analysis-synthesis reconstruction, the target is simply the original waveform converted to parametric representation (pitch, power, Mel spectrum). In text-to-speech conversion, the target is generated by a predictive model capable of converting input text to a phoneme string, and from that to the parametric representation used for target matching. For this purpose we use the CLUSTERGEN statistical parametric system [2]. CLUSTERGEN is designed with an HMM-state (1/3 phoneme) as its underlying representational unit and generates Mel log spectral approximation vectors at 5 ms intervals. (In normal usage the parametric representation is converted to a wave file through inverse MLSA digital filters).

Thus, there is a separation of concerns in our architecture. The Predictive Modeling component is responsible for generating the trajectory target, while the Selection Synthesis component is responsible for selecting an optimal sequence of units to match the target. This two-way division is why our system is a parametric-selection hybrid. It differs from other implementations of hybrid synthesizers such as in [10] which use frame sized units. Our algorithm is broken down into four stages:

1. target generation;
2. candidate class selection,
   (includes unit merging and back off strategies);
3. unit selection through distance minimization;
4. waveform construction.

2.3.1. Target Generation

Given input text to synthesize, let $\mathbf{u} = (u_1,u_2,...,u_n)$ be the sequence of target phonemes, trivially predicted from a pronunciation dictionary or, failing that, letter-to-sound rules. Since CLUSTERGEN represents each phoneme as a sequence of three states, each target $u_i(t_0,t_1,t_2,t_3)$ is affiliated with four state boundary times defined. In this notation, for example, $u_{15} = ae(1.1, 1.15, 1.18, 1.21)$ – the fifteenth phoneme of some phone sequence is /ae/ and spans 1.1 to 1.21s, with two internal state boundaries at 1.15 and 1.18s. Since segments are contiguous, the end time of the previous phone is the same as the beginning time of the current phone. These times are constrained to be integral multiples of the frame step size.

Let $\mathbf{x} = (x_0,x_1,...,x_L)$ be the sequence of L feature vectors that form the prediction target, where the *m*th frame is $x_m=(F0,c_0,c_1,\ldots,c_{24})_{m.}$ The cepstral c_0 coefficient represents log power. In unvoiced segments, F_0 is set to zero. The number of melcep coefficients may reasonably be varied from 6 to 48, but we stayed with the default of 24. Details explaining the generation of x are described in [2]. The pair of vector sequences $(\mathbf{u},\mathbf{x})$ is the input to the Selection Synthesis component.

2.3.2. Candidate Class Selection

Theoretically, one could search the entire speech corpus for wave file segments that best match the target trajectory, independent of the phoneme labels. Besides being prohibitively computationally expensive, this doesn't work well. Among others, stop consonants are troublesome. The best match of a segment containing a /d/ for example, might be a /t/ or a /k/ – but made on the basis of the longer silence section overwhelming the short but perceptually relevant transient part (assuming anything less than a very sophisticated perceptual distortion measure). Consequently, the target phoneme label sequence $\mathbf{u}$ is used to restrict the unit search to the candidate class sequence $\mathbf{v}$.

In the simplest case of $\mathbf{v} = \mathbf{u}$ the search is phoneme-based. Phoneme concatenation suffers from severe discontinuity artifacts though, so this is not preferred. Instead the input phoneme stream is converted to a sequence of diphones, where each diphone spans from the middle of the left to the middle of the right target phoneme. The midpoint of the diphone is defined naturally as the boundary between the two overlapping target phonemes.

$$v_i(t_0,t_1,t_2)=\left[u_{i-1}(0.5(t_1+t_2)),\ u_i(t_0),u_i(0.5(t_1+t_2))\right]\ -\ (1)$$

It is possible that the selection corpus contains no examples of the candidate diphone type. If this is true – or if the number of candidates is below some small threshold count – the candidate is split into a pair of half-phones.

$$v_{i,left}(t_0,t_1)=\left[u_{i-1}(0.5(t_1+t_2)),\ u_i(t_0)\right]$$
$$v_{i,\ right}(t_1,t_2)=[u_i\left(t_0\right),\ u_i(0.5(t_1+t_2))]-(2)$$

Matching candidate units to diphones is generally superior to phonemes since perceptual discontinuities are reduced. We found empirically an exception, however: splitting very short phones tends to increase discontinuity. Therefore, we introduced the heuristic policy of absorbing short vowels into the surrounding context. The particular vowels subjected to this treatment are /aa, ax, ah, eh, ih, uh/.

For example, for the utterance "coin mint" with phoneme sequence **u** = /k oy n m ih n t/ suppose the n-m pair is rare, while m-ih-n triple is common. The input is converted to the diphone sequence sil-k k-oy oy-n n-m m-ih, ih-n, n-t, t-sil and then to **v** = (sil-k, k-oy, oy-n, n-, -m, m-ih-n, n-t, t-sil). In conjunction with the associated times and the feature vector **x**, this is the target specification provided to the next stage.

2.3.3. Unit Selection

Given the target (**v**,**x**) the total selection cost is defined conventionally as the sum of target and join costs,

$$\text{cost}=\sum_{i=0}^{N}c_{\text{target}}\left(x\left(v_i\right)\right)+\sum_{i=0}^{N-1}c_{\text{join}}\left(x\left(v_{i-1}\right),x\left(v_i\right)\right)-(3)$$

where minimizing this function is solved using the well-known Viterbi algorithm. In this work we wanted to see how well target costs alone could manage, and therefore deliberately set all join costs to zero. This neglect is partially compensated by frame padding. For a given unit $v_i(t_0,t_1)$, the corresponding feature vector is extended on either side by a padding factor of p=6 frames. Thus $x(v_i)$ spans from t_0-p to t_1+p.

If the target vector is $x(v_i)$ we call the set of candidate vectors of type v contained in the speech corpus $Y_v = \{y_{v,j}\}$, where the index j iterates through all the examples of that type. Since the diphone is the basic searchable unit, the speech catalog is indexed with diphone labels. Let k_0 be the first frame of a particular catalog unit. Let y_v^* be the minimum distance unit found in the corpus according to

$$y_v^{\ *}=\arg\min{}_{j,k}\left\|y(v_j)\left[k_0+k\right]-x\left(v_i\right)\right\|-(4)$$

where ||◦|| is a weighted L^2 norm between two vectors *x* an *y*. The index j ranges from 1 to |Y*v*|. The index k is a shift operator and ranges from -|*x*| to +|*x*|.

The purpose of the frame shift operator in (1) is twofold. First, it avoids linear interpolation of *y* to *x* if they are of unequal length – which is most of the time. Second, this local search insulates the result from errors in the catalog. It is not necessary for the catalog labels to have exact timing (i.e. to be hand corrected) for this algorithm to function properly, since it is not sensitive to small errors. The size of the search window varies depending on the length of the unit being matched.

The distance calculation in (4) allows for different weighting of components. The form we experimented with treats cepstral components 1 through N identically but has separate weights for F0 and power terms, as in (5).

$$d^2(\vec{x},\vec{y}) = \left[w_1\left(y_{f0}-x_{f0}\right)^2+w_4\left(y_{f0}{}^{\bullet}-x_{f0}{}^{\bullet}\right)^2\right]\delta(\text{voiced})$$
$$+\ w_2\left(y_{c0}-x_{c0}\right)^2+\ w_5\left(y_{c0}{}^{\bullet}-x_{c0}{}^{\bullet}\right)^2$$
$$+\ w_3\sum_{i=1}^{N}\left(y_{ci}-x_{ci}\right)^2+\ w_6\sum_{i=1}^{N}\left(y_{ci}{}^{\bullet}-x_{ci}{}^{\bullet}\right)^2-(5)$$

The dotted terms are first derivatives computed with a single step difference operator. When computing pitch differences, both vectors must be voiced for a particular frame to be included in the computation. This is indicated by the delta step function term, where δ=1 if both frames are voiced, and 0 otherwise.

We did not perform exhaustive tuning of the weighting terms. After some experimentation we set w_2=w_3, w_1=0.3w_2, and w_4 through w_6 to a relatively small value, e.g. 1/10 of the corresponding w_1 through w_3. As would be expected, increasing w_1 encourages pitch continuity at the expense of spectral mismatch. There is no direct control of the pitch period during waveform synthesis.

2.3.4. Waveform Construction

The selected units for an utterance are most commonly diphones, but may include half-phones (the back off condition), and dual-diphones (after absorption of short vowels). The non-uniform segments are overlapped 10 ms on either side of each boundary, and linearly blended together. Based on the experience of [5], we opted not to perform any signal processing, other than a global normalization of volume.

2.4. Perceptual Tests to Choose a Best System

From the held out data set, we selected around 100 sentences (20 sentences from ARCTIC, conversation style, news, semantically unrelated sentences, and Modified Rhyme Test). The synthesized sentences were randomized and subjected to AB listening tests. In informal listening experiments we observed that CLUNITS was better than CLUSTERGEN. Thus we had AB listening tests done for CLUSTERGEN with post filter on versus Off, and CLUNITS versus Hybrid. The mean opinion scores are summarized in Tables 1 and 2. The hybrid system evidently performed better than CLUNITS and the higher MOS scores of the hybrid system in Table 2 indicate this observation.

Table 1: Mean Opinion Scores (MOS) of AB listening tests with post filtering ON and OFF in CLUSTERGEN.

	Filter off	**Tie**	**Filter on**	**Pref. Ratio**
#test utts.	143	130	92	0.57 off
MOS	2.54		2.41	

Table 2: Mean Opinion Scores (MOS) of AB perceptual listening tests on CLUINTS and Hybrid synthesis systems.

	CLUNITS	Tie	Hybrid	Pref. Ratio
#test utts.	108	49	203	0.632 (hybrid)
MOS	2.98		3.38	

3. Performance on Blizzard Datasets

This year's Blizzard Challenge consisted of three tests similar to those of past Challenges and two new tests. The new tests were a 'similarity' Mean Opinion Score (MOS) test and a naturalness comparison test. Carry-over tests consisted of two MOS tests (news and conversation domains) and Semantically Unpredictable Sentences (SUS). Other than the 'similarity' test, individual results were not made available for the two MOS tests; instead, results are on the aggregate of the two. Likewise, results for the naturalness comparison have not yet been provided.

As mentioned previously, participants were asked to submit a system using the full training dataset (this year, called "set A"), as well as a system using only the ARCTIC subset of the data (now called "set B"). Additionally, a third system could be submitted using a site-defined subset (called "set C").

Results were provided in raw form as well as with box plots of median scores for the MOS tests. Since mean scores were used in previous years (as well as other differences in the construction of MOS scale), it is not clear that official results can be compared to previous years, despite use of similar training data sets. An argument was made in favor of using medians rather than means, and similarly, median absolute deviations rather than standard deviations, based on the assumption that MOS results are ordinals.

It is unclear whether they are in fact ordinals or simply a finite (limited precision) set of interval measurements without performing an analysis to verify this claim. In measurement theory [6], an interval variable is an ordinal variable over which addition and subtraction are defined. That is, if an interval variable, the difference between a MOS score of 3 and 4, is in some sense the same as the difference between 1 and 2.

Furthermore, since both median absolute deviations and standard deviations depend on the assumption that addition and subtraction are meaningful, they both rely on the underlying variable being an interval value, and thus, it is unclear why one should make for a better representation of results than another. Likewise, once relying upon an interval measurement for median absolute deviations, the superiority of the use of a median to a mean is unclear.

3.1. Set A and B Results

In the following tables, we have included results from each of the two datasets in the past two years. In the interest of comparing like items, we have chosen to include the mean scores over this year's MOS data. Additionally, we have computed scores across all listener types and merged the results of the news and conversation domains for the 2006 results. Note that these are not results that have been published elsewhere since they were constructed in this fashion solely for the purpose of comparison to this year's data.

Figure 1 shows the results for our system on the MOS tests for both years and both datasets. CMU's performance was substantially better in the conversation domain compared to the news domain last year; we are interested to learn whether this was similarly the case this year. Figure 2 shows a similar comparison of results for the SUS test, in terms of mean word error rate (WER). As can be seen in both figures, our system shows improvement in absolute results for 2007 when compared to similar tests and listener groups of 2006.

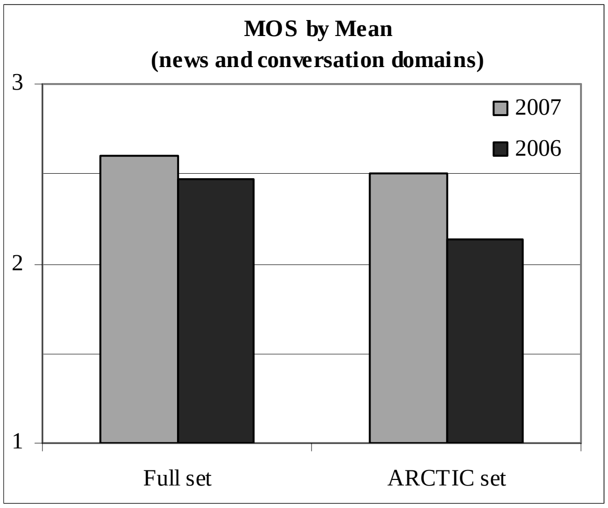

Figure 1. Comparison of results for the CMU system by mean MOS on the common domains in 2006 and 2007.

It should be noted that an additional listener group was added to this year's Challenge, namely UK English speaking undergraduates. This of course will have some effect on the overall results presented here, though an analysis of its effects has not been performed. Despite this additional category of listeners, the size of *n*, where *n* is the total number of listeners, seems to have been approximately the same in 2007 and 2006, indicating that some listener categories were smaller. Additionally, the data exclusion policy utilized in 2006 appears to have been stricter than that used this year.

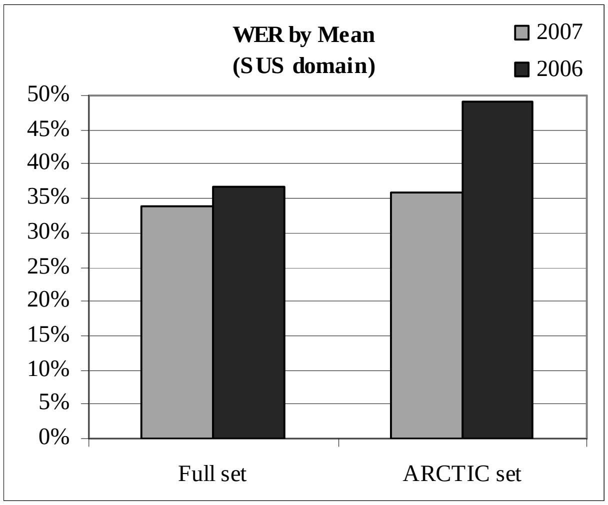

Figure 2. Comparison of results for the CMU system by mean WER on SUS in 2006 and 2007.

4. Lessons Learnt

We have made several observations regarding this evaluation, which will be included here, as well as some discussion of improvements for future tests.

4.1. Large versus Small Databases

CMU did not enter a “set C” system, wherein participants were asked to choose a subset of the full database that is no larger than the ARCTIC databases. In some sense, the ARCTIC databases could be called our selected subset.

It was discussed previously that it may be hard to find a subset of the databases that is better than the ARCTIC set as it was already selected from a much larger set of utterances and was designed to have good phonetic coverage (and be easy to read with minimal errors from the voice talent).

Based on the results, although there is no direct comparison between ARCTIC and institutional selected subsets, it seems that it is not clear that any other subsets were definitely better.

Looking at the CMU D scores for the full databases and the ARCTIC databases, it appears that our system did comparatively better in the ARCTIC case versus the full databases case when compared to performance differences of other systems. This suggests that our techniques are optimized for ARCTIC sized databases (and perhaps even ARCTIC itself given that it is our standard test set, though we have more recently been looking at much larger freely available datasets [7]).

4.2. Evaluation of Synthetic Speech

During the evaluation, some of the native speakers (and speech experts) found that it was fairly trivial to identify the natural speech examples within the evaluation. Given that even the highest quality synthesizers will not produce perfectly natural sounding speech, this is not surprising. However, this distinction could be made even without taking quality into account; the natural speech typically had several hundred milliseconds of leading silence in the waveforms, while the synthetic examples did not, allowing an alert participant to determine if a waveform was a natural example without even listening to it. Though a minor concern, we feel this should be addressed in future evaluations, either by trimming the leading silence from the natural examples or adding some amount to the start of the synthetic examples.

5. Conclusions

To synthesize natural and consistent speech, an attempt has been made to develop a hybrid system combining unit selection and statistical parametric synthesis. We found that during internal evaluation the MOS scores of the hybrid system were better than CLUNITS and CLUSTERGEN. The MOS scores also suggest that the hybrid system performed better than our previous system submitted for Blizzard Challenge 2006. However, we have also observed that the parameter settings of our system seem to have been more biased towards ARCTIC databases, and it would be interesting to look into techniques that would make use of larger multi-paragraph speech datasets using hybrid and statistical parametric synthesis techniques.

6. Acknowledgements

This work was in part supported by the US, National Science Foundation.

7. References

[1] Alan W Black and Paul Taylor, *Automatically clustering similar units for unit selection in speech synthesis* Proceedings of Eurospeech 97, vol2 pp 601-604, Rhodes, Greece.

[2] Alan W Black, *CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling*, Interspeech 2006 - ICSLP, Pittsburgh, PA.

[3] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. *Speech parameter generation algorithms for HMM-based speech synthesis. Proc. ICASSP*, pp. 1315–1318, Istanbul, Turkey, June 2000.

[4] R. Clark, K. Richmond, V. Strom, and S. King. Multisyn voice for the Blizzard Challenge 2006. In *Proc. Blizzard Challenge Workshop (Interspeech Satellite)*, Pittsburgh, USA, September 2006.

[5] John Kominek, Christina Bennett, Brian Langner, Arthur Toth, *The Blizzard Challenge 2005 CMU Entry: A Method for Improving Speech Synthesis Systems*, Interspeech 2005, Lisbon, Portugal.

[6] Stevens, S. S., *On the theory of scales of measurement,* Science, Vol. 103, pp 677-680, 1946.

[7] Kishore Prahallad, Arthur R Toth, and Alan W Black, *Automatic Building of Synthetic Voices from Large Multi-Paragraph Speech Databases*, Interspeech 2007, Antwerp, Belgium, 2007.

[8] Alan W Black and Keiichi Tokuda, *Blizzard Challenge -- 2005: Evaluating corpus-based speech synthesis on common datasets* Interspeech 2005, Lisbon, Portugal, 2007.

[9] Andrew Hunt and Alan W Black, “*Unit selection in a concatenative speech synthesis system using a large speech database*”, ICASSP96, Atlanta, Georgia, 1996.

[10] Ling, Zhen-Hua and Wang, Ren-Hua, *HMM-Based Unit Selection Using Frame Sized Speech Segments*, Interspeech 2006 – ICSLP, Pittsburgh, PA.

[11] Heiga Zen and Tomoki Toda, *An Overview of Nitech HMM-based Speech Synthesis System,* In Proc. Blizzard Challenge Workshop 2005 (Proceedings of Interspeech 2005 - Europseech), Lisbon, Portugal.

[12] Christina L. Bennett, *Large Scale Evaluation of Corpus-based Synthesizers: Results and Lessons from the Blizzard Challenge 2005*, in Proceedings of Interspeech 2005 - Eurospeech, Lisbon, Portugal, 2005.