# The INESC-ID Blizzard Entry: Unsupervised Voice Building and Synthesis

Luís C. Oliveira, Sérgio Paulo, Luís Figueira and Carlos Mendes

L$^2$F - Spoken Language Systems Lab
INESC-ID/IST, Lisbon, Portugal
{lco,spaulo,luisf,cmdm}@l2f.inesc-id.pt

## Abstract

This paper describes the INESC-ID participation in the Blizzard Challenge 2008, which consisted in building the two *English* voices. We have been developing a new *European Portuguese* TTS system, called *DIXI*, for the last two years. This year, the system was already stable enough to be used in the challenge, after a partial adaptation to support synthesis in *English*.

The major motivation for our participation in this year's edition of the challenge was to evaluate to what extent our unsupervised and less resource-demanding voice building methods, very successfully applied in limited domain applications, can be used in open domain synthesis.

**Index Terms**: speech synthesis, cluster unit selection, automatic voice building methods.

## 1. Background

Speech synthesis has been an active research topic in our lab for the last twenty years. Accordingly, in the early nineties, the first version of a fully functional text-to-speech system for *European Portuguese* was made available [1]. It consisted in a synthesis-by-rule formant synthesizer, using Klatt's model [2]. In mid-nineties, concatenative synthesis started being explored in the lab [3]. Then, the *Festival* speech synthesis system [4] came out and was adopted for the development of our corpus-based voices. In addition to research into generation of natural prosody [5] and emotive speech [6, 7], our synthesis group has been focused on:

- Minimizing the demands on linguistic resources for building natural sounding voices;
- Automating voice building by unsupervised methods capable of addressing speaker variability problems.[1]

Our corpus-based voices are based on the *cluster unit selection* approach [8] and are built by a modified version of the *Festival* clunits module and other Festival-embedded tools.

### 1.1. Tecnovoz project

Despite the great deal of success enjoyed by the *Festival* in our research projects, the *Tecnovoz* project[2] showed us that we were in dire need for a more robust system, capable of exploiting all computational capabilities of the nowadays machines.

The Tecnovoz project was a join effort to disseminate the use of spoken language technologies in different domains of application. The project consortium included 4 research centers and 9 companies specialized in a wide range of areas like banking, health systems, fleet management, access control, media, alternative and augmentative communication, computer desktop applications, etc. To meet the goals of the project a set of 13 demonstrators were developed based on 9 technology modules. Two of these modules were related with speech output: one module for limited domain speech synthesis and another for synthesis with unrestricted input.

After deciding in favor of using the same system to meet both requirements, limited and unlimited domains, the *DIXI* system was designed in order to accomplish the fast generation of speech with a high degree of naturalness.

### 1.2. The *DIXI* system

*DIXI* is intended to be a generic TTS, capable of supporting most of the western languages. However, it is still not fully functional in languages other than *European Portuguese*. In order to minimize the needed effort to support new languages, the language- and domain-specific knowledge sources were kept apart from the system's implementation. Also, machine learning techniques were used to train models for some components responsible for the linguistic analysis of the input text. The models – frequently encoded in the form of *Classification and Regression Trees* (CART) [9] – are loaded the same way no matter what domain or language the system is dealing with.

The system's operation mode (unlimited or limited domain) is defined by the currently selected voice, enabling the user to switch from a limited domain to a general purpose voice, and vice-versa, with a single engine. *DIXI* currently runs on *Windows* and *Linux*. The synthesis engine can be accessed, in both operating systems, by means of an *API* provided by a set of *Dynamic Linked Libraries* and *Shared Objects*, respectively.

*DIXI* was designed following a modular architecture, as depicted in Fig. 1, so that the speech generation can be sped up by exploiting the multi-threading capabilities of the host machines.[3] According to the figure, the input text is firstly processed by the *Text Splitter* module. This module splits the input text into several chunks to be processed independently by the following system modules. This approach allows the streaming synthesis problem to be addressed more efficiently. Then, we have the well known *Text Normalization*, *Part-of-Speech Tagging*, and *Prosodic Phrasing* modules. The *Grapheme to Phoneme* module generates an isolated pronunciation[4] for each word of the sentence. The *Post-lexical Analysis* module computes a set of transformations on the word phonemes to account for their production in connected speech. Finally, the *Waveform Generation* module takes the utterance description built so far and search the inventory for the best matching unit sequence to produce the signal.

[1] e.g. speaker-specific word pronunciations.
[2] http://www.tecnovoz.pt
[3] Each module runs a distinct thread.
[4] The pronunciation of the word when uttered in isolation.

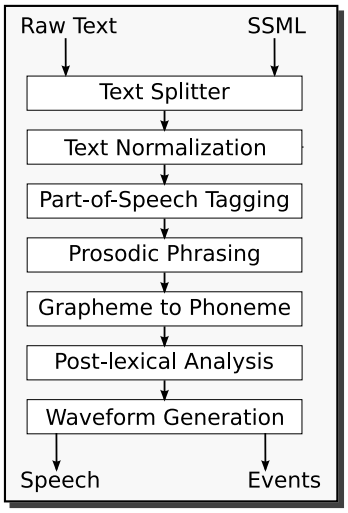


Figure 1: Overview of the *DIXI* architecture, where *SSML* stands means SSML-formatted input text.

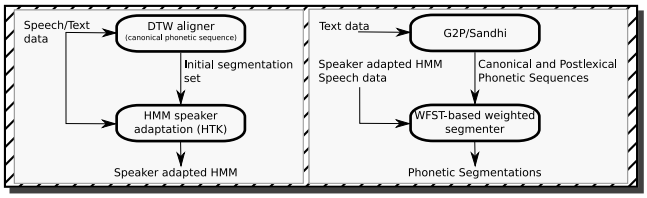


Figure 2: Phonetic segmentation steps: acoustic model training (on the left) and database labeling (on the right).

| *MRPA* | *X-SAMPA* |
|---|---|
| m | { m ; m! } |
| n | { n ; n! } |
| l | { l ; l! ; lw } |
| i | { i ; iy } |
| u | { uw ; u } |

Table 1: *MRPA* to *X-SAMPA* phone mappings, in the case of non-one-to-one correspondences.

### 1.3. External language resources

The external data we used to build the *Blizzard* voices consisted of:

- The *British English ogirab_diphone* voice of the *Festival-OGI* [10] to generate the synthetic reference for the DTW-based alignment, and to estimate the phoneme duration for the synthesis procedure;
- The UNISYN lexicon [11] mainly as a source of alternative pronunciations for the phonetic segmentation;
- The *Festival* utterance structures provided by the organization.

### 1.4. Paper organization

This paper is organized as follows. In section 2, we describe the procedure we used to build corpora to support the two synthetic voices. Section 3, contains the methodologies applied to build the voices A and voice B, whereas section 4 is intended to describe the way the speech samples were generated. Finally, in section 5, we show the results we obtained and draw some conclusions.

## 2. Corpus Building

The corpus building procedure described here is a fully automated process that uses the speech recordings along with the respective text prompts to build *Festival* utterance descriptions containing the text-predicted representations synchronized with the descriptions derived from the speech signal, such as phonetic segmentation, prosodic annotation and fundamental frequency contour. While this corpus building procedure cannot compute any prosodic annotation of the signal, if such annotations are available, they can be automatically integrated in the utterance descriptions, by using the *temporal overlap* criterion and further processing to set associations among elements of distinct linguistic levels.

### 2.1. Phonetic segmentation

The *Blizzard* corpus was phonetically segmented using the method comprised of two steps, as depicted in Fig. 2. In the first step, a set of acoustic models were built for the speaker, whereas the second step consists in using the models to segment the database with a phonetic segmentation tool allowing multiple acoustic realizations.

#### *2.1.1. Acoustic models*

At the beginning, the *Festival-OGI ogirab_diphone* voice was used to align the speech signals with the respective phonetic sequences using a DTW-based approach [12]. The phonetic sequences were predicted by the *Festival* front-end processing modules associated to the reference voice. Because of that, the phone labels arisen from this procedure belong to the *MRPA* phone set (for further details refer to [4]) and had to be subsequently mapped onto the *X-SAMPA* set used in both the *UNISYN* lexicon and the *Festival* utterance descriptions. After a short analysis of the pronunciations of some words in the two phone sets, we could draw a mapping from *MRPA* to *X-SAMPA*. While we could find a one-to-one mapping for most phones, some *MRPA* phones were mapped to more than one symbol in *X-SAMPA*. Those phone mappings are presented in Table 1.

After computing the DTW-based alignments, we used the HTK *HInit* program [13] to train context-independent acoustic models for the speaker, still using the *MRPA* phones. The resulting models were renamed according the mappings found earlier. In the case of the mapping shown in Table 1, the *MRPA* models were cloned and renamed to the respective *X-SAMPA* symbols. For example, the models of the *X-SAMPA* phones *l*, *l!*, and *lw* are the same as the *MRPA* *l*. Then, in order to refine to newly created acoustic models, the database was re-segmented using the phonetic sequences provided by *Festival* utterance structures distributed with the corpus, and another model training process took place.

#### *2.1.2. Database segmentation*

At last, the database was segmented using a tool based on *Weighted Finite-state Transducers* (WFST) [14]. This tool was first described in [15], and builds a pronunciation graph for each utterance by concatenating the respective word graphs, with an optional silence in the middle. The pronunciation graph is encoded in the form of a *WFST*, and is finally composed with an-

other *WFST* comprising a set of optional post-lexical phonological rules.

In order to build a pronunciation graph for each word of a sentence, we start by searching for it in the *UNISYN* lexicon. Both the *reduced* and *full* pronunciation forms were taken into account at this stage, because we could, by no means, know in advance the way the speaker produced that word. The *Festival* utterance structure were also searched in order to get further word pronunciations. All these pronunciation forms were then compiled in a *WFST*, and used as described above.

### 2.2. Building utterance descriptions

While the *Festival* data structures distributed with the corpus can eventually describe the spoken utterances to a large extent, realistic representations cannot be achieved without combining those descriptions with the annotations/segmentations derived from the speech signal.

We built more realistic utterance descriptions by combining the representations encoded in the *Festival* utterance structures with the phonetic segmentation results, making use of a combination method described in [16]. In addition to the phone labels and timings, F0 values were also computed from the recordings and appended to the newly created utterance structures. Accordingly, F0 values were assigned to the respective phonetic segments base on the *temporal inclusion* criterion. Temporal inclusion would also be used to assign intonation events to the respective syllables, if we had annotations at that level, but that was not the case. Thus, in the absence of any prosodic annotation, word break indexes were derived from the combination of phonetic phenomena - like duration stretching, silences - with punctuation marks at the word level. In a subsequent step, the utterances were re-phrased according to the newly created word break indexes.

### 2.3. Voice A and voice B corpora

Obeying to the *Blizzard 2008* rules, all the corpus building routines had to be run twice, in order to make the creation of *voice B* completely independent from the data supporting *voice A*.

## 3. Voice Building

Selecting the synthesis units to be used within a corpus-based TTS strongly depends on the target application. Accordingly, while words can be an obvious unit for a limited domain application, phonetic segments or diphones can be more suitable for open domain synthesis, since, unlike words, they make up finite sets, entirely known in advance.

In order to address both limited and open domain synthesis with a common framework, we decided to always use phone-sized units. The basename of such units consists in the concatenation of the phone name according to the word's isolated pronunciation (phone *name*) with the most likely phone realization in connected speech (phone *plex_name*) as shown in Fig. 3. The isolated word pronunciation results from the lexicon lookup. Whenever the pronunciation lexicon contains a *reduced* form, the *unit names* are set according to the *full* form, whereas the units' *plex_name* corresponds to the *reduced* form.

### 3.1. Unit tag generation

The degree of success enjoyed by a unit selection voice depends to a large extent upon the quality of its phonetic segmentations. Therefore, the detection of phonetic labeling errors constitute a

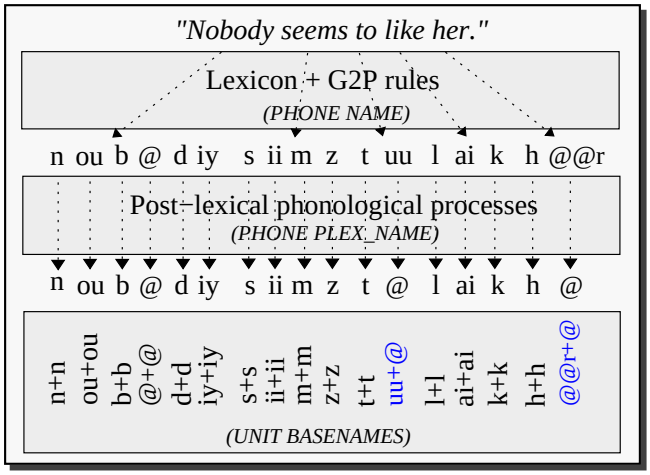


Figure 3: Toy example unit basenames for the sentence "Nobody seems to like her."

| *Tag* | *Assignment Rule* |
|---|---|
| $ERR$ | $dur(x) - \overline{dur(x)} \geq 5 \cdot \sigma_{dur_x}$ |
| $WRN1$ | $5 \cdot \sigma_{dur_x} > dur(x) - \overline{dur_x} \geq 3 \cdot \sigma_{dur_x}$ |
| $OK$ | $dur(x) - \overline{dur_x} < 3 \cdot \sigma_{dur_x} \wedge dur(x) \geq 20ms$ |
| $WRN2$ | $dur(x) < 20ms$ |

Table 2: Duration-based tag assignment rules for the corpus units.

| *Tag* | *Unit usage* |
|---|---|
| *ERR* | This unit is too large: *remove this unit from the inventory* |
| *WRN1* | This unit is larger than usual, so, unless it belongs to a sentence-final word, it can only by used while accompanied by its neighbors: *do not cut here* |
| *WRN2* | This unit is too short, thus it is strongly affected by the neighboring units: *do not cut here* |
| *WRN3* | This unit was not produced as expected, thus some problems may arise here if the unit is not used together with the neighboring units: *do not cut here* |
| *OK* | Nothing strange happened: *concatenations permitted here* |

Table 3: Description of the subsequent usage of the tagged units.

major concern for the voice developer, as a single mislabeled segment can ruin the naturalness of the whole sentence. Several confidence measures for phonetic segmentation have been proposed [17, 18]. However, the duration-based methods have prevailed as they are more robust to speaker variations [19].

The first step of the voice building consisted in a search for durational outliers within the unit catalogue. Accordingly, mean values and standard deviations were computed for each phone type.[5] Then, the algorithm went through the corpus' utterance structures to tag all the units according to rules in Table 2.

Form previous studies [20] we concluded that we should map the speaker-specific pronunciations onto those generated by the grapheme-to-phone rules of the TTS in order to increase the average number of consecutive units retrieved by the unit selection algorithm. It is thus extremely useful to detect where those pronunciations do not match. Therefore, the phonetic sequence generated by automatic phonetic segmentation tools were aligned with the sequence of *plex_name*'s. Whenever a mismatch occurred, a "WRN3" tag was assigned to the respective catalog unit, in order to allow for its use only if its neighboring units were used, too, in other words, to prevent the TTS from cutting the waveform somewhere inside that unit. Table 3 describes the way the distinctly tagged units were used later on in the waveform generation module.

### 3.2. Multi-level approach

The *multi-level cluster unit selection* method we use requires the voice building procedure to run in as many times as the number of levels (5 levels in the current case). A voice is built according to the level-specific naming conventions. The unit catalog comprising the time boundaries and other features of the units are shared among all levels, as the physical units are kept unchanged. Thus, only one file called *catalogue.alias*, with a mapping from the level-specific logical names to the physical ones, and a cluster tree collection, gathered in a single file, are built in each iteration.

The *catalogue.alias* file contains the logical unit names in the following format:

```
unit_type physical_unit

l+l_london 14453 142 534
l+l_l.ah.n.d  353 14453
l+l_pau-l-ah 3453 334 86 14453
l+l_l-ah 543 2345 14453
l+l 67 345 3 78 567 14453
```

The content of the line "l+l_london 14453 142 534" must be interpreted as:

- the logical unit *l+l_london_1* points to the physical unit *14453*
- the logical unit *l+l_london_2* points to the physical unit *142*
- the logical unit *l+l_london_3* points to the physical unit *534*

### 3.3. Physical and logical unit name convention

Spectral distance measures are still not reliable enough to accurately predict the occurrence of concatenation problems. Thus, we must control the selection of candidate units, in order not to rely too heavily on the spectral distance measure capability for eliminating unsuitable units. Therefore, although the physical unit is always a phone, its logical names are not always the same. Each catalog unit has a single universal id, but several logical names, which are created using one of five distinct methods:

- Word level name: the unit basename is concatenated with the word the phone is in;
- Triphone level name: the unit basename is concatenated with the triphone name centered in that phone;
- Syllable level name: the unit basename is concatenated with its syllable name;
- Diphone level name: the unit basename is concatenated with the name of the next phone;
- Backoff level name: the unit basename is not concatenated with anything else.

### 3.4. Blizzard voices

Following the *Blizzard 2008* rules, all the above-described voice building routines were carried out independently in the building of *voice A* and *voice B*.

## 4. Wave Generation

### 4.1. *DIXI* waveform generation module

The *DIXI* speech signal generation is based on the *cluster unit selection* approach, as mentioned above.

#### *4.1.1. Candidate unit selection*

The synthesis procedure goes through the same steps as the default *cluster unit selection* module of *Festival*, except the building of the candidate unit set, and the waveform generation, described later on in this paper.

Given a target segment $seg$, the candidate unit list is built as follows:
*name[0]←"Word";*
*name[1]←"Triphone";*
*name[2]←"Syllable";*
*name[3]←"Diphone";*
*name[4]←"Backoff";*
*level←0;*
*n_good_cands←0;*
*candidates←* $\emptyset$*;*
*WHILE((n_good_cands<THRESH)AND(level<5))*
*DO*
*n_good_cands+=search_for_candidates(*$seg$*,name[level],candidates);*
*level++;*
*DONE*

At the beginning, we start by looking for word-level logical units, if we find more than a predefined number (*THRESH*) of units without any *WRN** tags, the unit search stops. Otherwise, we repeat the search for each one of the remaining levels until either *THRESH* non-problematic units[6] are found, or the *Backoff* level is reached.

Another difference from the *Festival cluster unit selection* synthesis is that the target cost is not the acoustical distance of the candidate unit to the cluster centroid. Rather, it is duration-based and is computed according to (1), where *d* is the predicted

[5] Unstress vowels are distinguished from their stressed counterparts

[6] Tagged as *OK*.

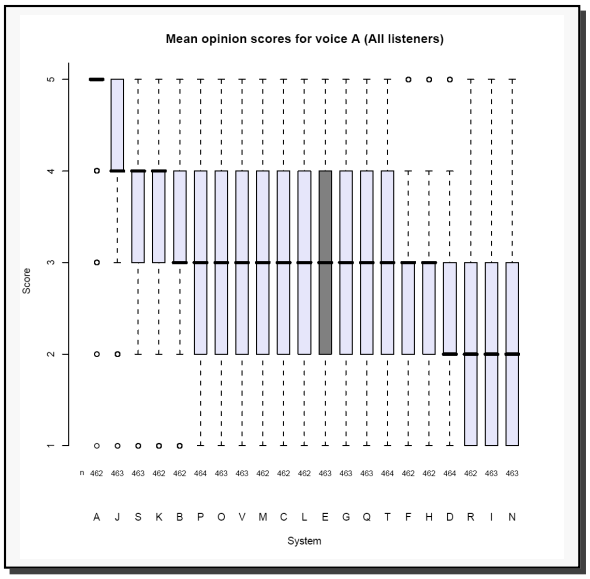


Figure 4: Mean opinion scores for systems' *voice A*.

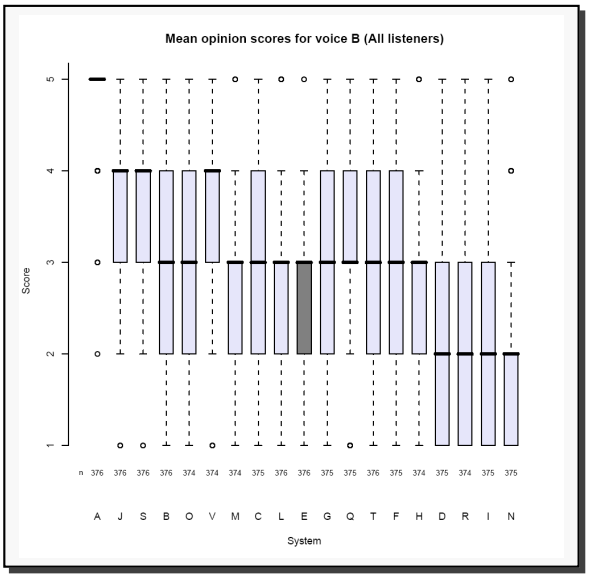


Figure 5: Mean opinion scores for systems' *voice B*.

duration of the phonetic segment, and $dur(u_i)$ is the duration of the candidate unit $u_i$.

$$C_t(u_i) = \frac{dur(u_i)}{d} + \frac{d}{dur(u_i)} \quad (1)$$

#### *4.1.2. Audio generation*

After the selection of the best candidate unit sequence, the speech signal is generated making use of cross-fading techniques to smooth the fundamental frequency (F0) around the concatenation points. The F0 smoothing algorithm builds on three major steps. Firstly, it builds a list of the units retrieved by the Viterbi decoder. Next, it loads the units' acoustic coefficients (containing the F0 values, pitch synchronous). The following step consists in removing the short units – whose coefficients contain a single frame – from the unit list. In the following step, contiguous units are merged in order to give rise to fewer and larger blocks. Then, every block is split into three parts, two transition regions at the borders, where F0 smoothing takes place, and a central block where F0 values are kept unchanged. The following step consists in generating pitchmarks according to the desired (smoothed) F0 values. Finally, the signal is generated with a *PSOLA*-based method [21].

### 4.2. Synthesizing the *Blizzard* sentences

The *Festival* utterance structures were converted into the *DIXI's* internal format in order that they could be used to replace our text analysis modules, that are still not fully implemented for the *English* language. The phone durations were predicted using the same duration models as the *ogirab_diphone* voice.

## 5. Discussion

The results obtained for *voice A* are partially depicted in Fig. 4, whereas Fig. 5 shows part of the results for *voice B*. Our participant letter is *E*. In spite of the very encouraging results we obtained, specially if we take into account that no voice tuning took place, as the first voice version was already the final version, there are still much work to do, both in terms of naturalness and intelligibility. In fact, observing Figs. 4 and 5, we notice our system is still lacking robustness to the absence of some linguistic contexts, since our system's quality was more dramatically hit by the database reduction than several other systems.

There can can be multiple explanations for degradation of the *voice B* results comparing with *voice A*. We can by no means draw final conclusions without a deeper study of the degree of success enjoyed by each one of the voice building stages. Here, we can only speculate on the reasons for these results. First, we are using the speaker prosody arisen from the selected speech chunks. Thus, whenever we have to synthesize sentences containing poorly represented contexts, smoothing the transitions – that is actually the only signal processing we apply – may be not enough to avoid generating awkward prosody. Explicit prosodic models are needed in order to draw suitable fundamental frequency contours both to help the unit selection and to generate the acoustic signal. Moreover, the lack of linguistic context dramatically affected the intelligibility of system, since as Fig. 6 shows, while *word error rate* (WER) for *voice A* was slightly higher than the average, the WER of *voice B* reached the third highest value among all the systems. Second, it is likely that the acoustic models derived from the *Arctic* subset are not as robust as those derived from the whole speech inventory. Thus, *voice B* overall quality could also have been affected by less quality segmentations. Third, the pitchmark detection tool we used – *make_pm_wave* script provided by the festvox package – can also have had a negative impact on the *voice B* quality, as more dramatic F0 modification had to take place in order to smooth more discrepant unit F0 values. Since all signal processing we apply is pitch synchronous, badly detected pitchmarks can ruin the naturalness of the signal.

## 6. Acknowledgements

This work was funded by PRIME National Project TECNOVOZ number 03/165. INESC-ID is funded by Fundação para a Ciência e Tecnologia (FCT).

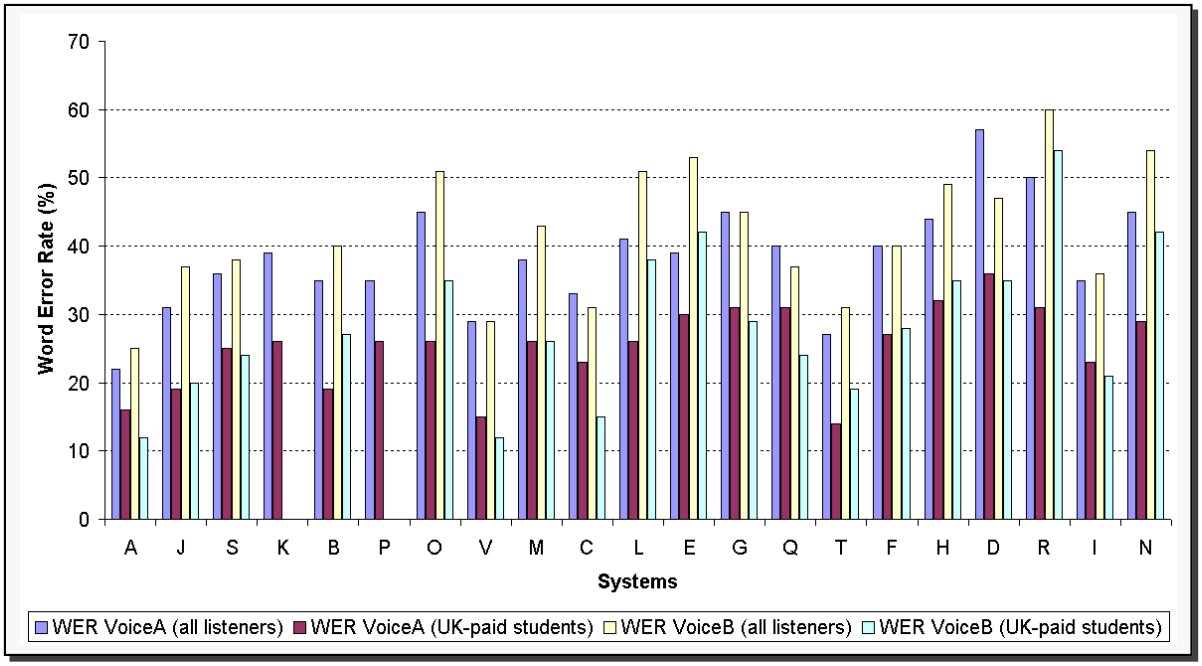


Figure 6: Word error rate for systems' voices A and B."

## 7. References


[1] L. C. Oliveira, M. C. Viana, and I. M. Trancoso, "Dixi - portuguese text-to-speech system," in *EUROSPEECH 1991*, 1991.

[2] D. H. Klatt, "Software for a cascade / parallel formant synthesizer," *Journal of the Acoustical Society of America*, vol. 67, no. 3, 1980.

[3] P. Carvalho, L. C. Oliveira, I. Trancoso, and M. do Céu Viana, "Concatenative speech synthesis for european portuguese," in *Third ESCA/COCOSDA International Workshop on Speech Synthesis*, 1998.

[4] A. W. Black, P. Taylor, and R. Caley, *The Festival Speech Synthesis - System documentation*, 2002.

[5] M. C. Viana, L. C. Oliveira, and A. I. Mata, "Prosodic phrasing: Machine and human evaluation," *International Journal of Speech Technology*, vol. 6, no. 1, 2003.

[6] J. Cabral and L. C. Oliveira, "Emovoice: a system to generate emotions in speech," in *Interspeech 2006*, 2006.

[7] C. Weiss, L. C. Oliveira, S. Paulo, C. Mendes, L. Figueira, M. Vala, P. Sequeira, A. Paiva, T. Vogt, and E. Andre, "Ecircus: Building voices for autonomous speaking agents," in *6th Speech Synthesis Workshop*, 2007.

[8] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Eurospeech 97*, 1997.

[9] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman and Hall, 1984.

[10] "http://cslu.cse.ogi.edu/tts/download/."

[11] S. Fitt, "Documentation and user guide to unisyn lexicon and post-lexical rules," Centre for Speech Technology Research, University of Edinburgh, Tech. Rep., 2000.

[12] S. Paulo and L. C. Oliveira, "DTW-based phonetic alignment using multiple acoustic features," in *Interspeech 2003*, 2003.

[13] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.2.1)*, 2002, available at http://htk.eng.cam.ac.uk/docs/docs.shtml.

[14] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, no. 2, 1997.

[15] S. Paulo and L. C. Oliveira, "Generation of word alternative pronunciations using weighted finite state transducers," in *Interspeech 2005*, 2005.

[16] S. Paulo and L. Oliveira, "MuLAS: A framework for automatically building multi-tier corpora," in *Interspeech 2007*, 2007.

[17] S. Nefti and O. B. T. Moudenc, "Confidence measures for phonetic segmentation of continuous speech," in *Eurospeech 2003*, 2003.

[18] S. Paulo and L. C. Oliveira, "Automatic phonetic alignment and its confidence measures," in *4th International Conference EsTAL 2004*, 2004.

[19] J. Kominek and A. Black, "Impact of durational outlier removal from unit selection catalogs," in *5th ISCA Speech Synthesis Workshop*, 2004.

[20] S. Paulo and L. C. Oliveira, "Reducing the corpus-based TTS signal degradation due to speaker's word pronunciations," in *Interspeech 2005*, 2005.

[21] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text to speech synthesis using diphones," *Speech Communications*, vol. 9, 1990.