The IRISA Text-To-Speech System for the Blizzard Challenge 2016

Pierre Alain, Jonathan Chevelu, David Guennec, Gwénolé Lecorvé, Damien Lolive

The IRISA Text-To-Speech System for the Blizzard Challenge 2016

Pierre Alain, Jonathan Chevelu, David Guennec, Gwénolé Lecorvé, Damien Lolive

IRISA, University of Rennes 1 (ENSSAT), Lannion, France

pierre.alain@univ-rennes1.fr,jonathan.chevelu@irisa.fr

david.guennec@irisa.fr,gwenole.lecorve@irisa.fr,damien.lolive@irisa.fr

Abstract

This paper describes the implementation of the IRISA unit selection-based TTS system for our participation in the Blizzard Challenge 2016. We describe the process followed to build the voices from given data and the architecture of our system. The search is based on a A* algorithm with preselection filters used to reduce the search space. A penalty is introduced in the concatenation cost to block some concatenations based on their phonological class. Moreover, a fuzzy function is used to relax this penalty based on the concatenation quality with respect to the cost distribution.

Index Terms: speech synthesis, unit selection

1. Introduction

In recent years, research in text-to-speech synthesis essentially focused on two major approaches. The first one is the parametric approach, for which HTS [1] and DNN-based systems [2] are now dominating the academic research in recent years. This method offers advanced control on the signal and produces very intelligible speech but with a low naturalness. The second approach, unit selection, is a refinement of concatenative synthesis [3, 4, 5, 6, 7, 8, 9]. Speech synthesized with this method features high naturalness and its prosodic quality is unmatched by other methods, as it basically concatenates speech actually produced by a human being.

The challenge for 2016 is to build an expressive voice using children's audiobooks in English. The main difficulty with audiobooks, and in particular for children, is the change of characters and here the imitation of animals (*i.e.* roars) as well as other sounds (*i.e.* bell ringings) that may occur. For instance, in sample data provided, a signal is given to tell the child that he/she has to turn the page. Considering the expressivity of the voice and the different sounds and characters we can find in such books, the main challenge is phone segmentation and expressivity control.

In this paper we present the unit-selection based IRISA system for the Blizzard Challenge 2016. Basically, the system is based on preselection filters to reduce the acoustic unit space to explore and on an A* algorithm to find the best unit sequence. The objective function minimized by the algorithm is composed of a target cost and a join cost. The join cost relies mainly on acoustic features to evaluate the level of spectral resemblance between two voice stimuli, *on* and *around* the position of concatenation. For instance, distances based on MFCC coefficients and especially F0 are used [10, 11]. In particular, for the challenge, we have introduced a penalty on units whose concatenation is considered as risky. This follows the work of [12, 13] which showed that artefacts occur more often on some phonemes than others. For this purpose, we define a set of phoneme classes according to their "resistance" to concatenation. A phoneme is called resistent if the phones of its class are usually unlikely to produce artefacts when concatenated. This approach has been originally proposed in the context of recording script construction in [13] to favor the covering of what has been called "vocalic sandwiches".

The remainder of the paper is organized as follows. Section 2 describes the voice creation process from the given data. Section 3 details the TTS system. Section 4 presents the evaluation and results.

2. General voice creation process

This year, the challenge focuses on audiobook reading for children in English. The goal is then to build a voice based on the 5 hours of speech data provided as a set of wave files with the corresponding text. The very first step has been to clean the text and make sure that it was corresponding to the speech uttered.

Then to build the voice, we first phonetized the text thanks to a grapheme-to-phoneme converter (G2P) and then, using another tool, automatically segmented speech signals according to the resulting expected phonemes. As for the G2P tool, we used *eSpeak* [14]. Once phonetized, speech signals have been segmented using the language independent segmenter *MAUS* [15]. We have also used the *ROOTS* toolkit [16] to store all the necessary information and produce conversions from IPA (output from *eSpeak*) to the SAMPA alphabet (used by *MAUS*).

3. The IRISA system

3.1. General architecture

The IRISA TTS system [17], used for the experiments presented in this paper, relies on a unit selection approach with an optimal graph-search algorithm (here an A* algorithm). The optimization function is divided, as usually done, in two distinct parts; a target and a concatenation cost [4] as described below:

$$U^* = \underset{U}{argmin} \left(W_{tc} \sum_{n=1}^{card(U)} w_n C_t(u_n) + W_{cc} \sum_{n=2}^{card(U)} v_n C_c(u_{n-1}, u_n)\right) \quad (1)$$

where U^* is the best unit sequence according to the cost function and u_n the candidate unit trying to match the n^{th} target unit in the candidate sequence U. $C_t(u_n)$ is the target cost and $C_c(u_{n-1}, u_n)$ is the concatenation cost. W_{tc}, W_{cc}, w_n and v_n are weights for adjusting magnitude for the parameters. Sub-costs are weighted in order to compensate magnitudes of all sub-costs as in [18]. In practice, the weight for each sub-cost c

Table 1: List of features used in the target cost

Text related features:	
TEXT_DIALOG	
Phoneme position:	
LAST_OF_BREATHGROUP	
LAST_OF_WORD	LAST_OF_SENTENCE
IN_CODA	IN_ONSET
SYLLABLE_BEGIN	SYLLABLE_END
WORD_BEGIN	WORD_END
Phonological features:	
LONG	NASAL
LOW_STRESS	HIGH_STRESS
Syllable related features:	
HAS_CODA	
LAST_SYL_OF_SENTENCE	
LAST_SYL_OF_BREATHGROUP	
SYLLABLE_RISING	SYLLABLE_FALLING

is set to $1/\mu_c$, where μ_c is the mean sub-cost c for all units in the TTS corpus. The problem of tuning these weights is complex and no consensus on the method has emerged yet. [19] is a good review of the most common methods.

3.2. Join cost

The concatenation cost $C_c(u, v)$ between units u and v is composed of MFCCs (excluding Δ and $\Delta\Delta$ coefficients), amplitude, F0, syllable speech rate and syllable F0 level euclidean distances, as below:

$$\begin{aligned} C_c(u,v) &= C_{mfcc}(u,v) + C_{amp}(u,v) + C_{F0}(u,v) \\ &\quad + C_{rate}(u,v) + C_{lev}(u,v) + C_{dia}(u,v), \end{aligned}$$

where $C_{mfcc}(u, v)$, $C_{amp}(u, v)$, $C_{F0}(u, v)$, $C_{rate}(u, v)$, $C_{lev}(u, v)$ and $C_{dia}(u, v)$ are the sub-costs, resp., for MFCC, amplitude, F0, speech rate, F0 level and dialog section. The speech rate and the F0 mean level are computed on a syllable basis and are averaged on a window of ± 1 syllable around the current phoneme. The dialog sub-cost is a penalty that is added if the phonemes u and v are taken from inconsistent parts of the corpus with respect to the narrative style.

3.3. Target cost

For candidate units, we compute a numerical target cost as a weighted sum of the features given in table 1. The features used in the computation are nearly the same as the ones used for preselection as explained in section 3.4. The weights W_{tc} and W_{cc} used in (1) are arbitrarily set to balance the importance of the join cost compared to the target cost.

3.4. Preselection filters

When exploring new units in the graph, the algorithm accesses to the corpus via an ordered list of preselection filters, where the role of each filter is to reject speech units which do not respect a given specific property. Their purpose is twofold. First, it considerably prunes the graph explored by the unit selection algorithm, making the selection process faster. Second, it serves as a set of binary target cost functions relying on the assumption that if a unit does not respect the required set of features, it can't be used for selection. The preselection filters should therefore be seen as part of the cost for a unit. In our system, when no corpus unit (or not enough units) respects a given set of preselection filters, the set is temporarily relaxed (removing one by one the features that seem the less helpful) until units are found. This mechanism ensures finding a path in all cases under the assumption that the whole corpus contains at least one instance of the most basic units, *i.e.* diphonemes. In our case, the threshold number of units is set to 100.

In case a diphoneme is not present in the corpus, a fallback mechanism has been implemented. Precisely, the requested diphoneme is built artificially by concatenating two phonemes in context of a pause. As it does not take into account coarticulation effects, the result is not excellent but at least, it enables to produce speech. The set of preselection filters we use in this work is the following:

1. Unit label (mandatory).
2. Is the unit a pause (mandatory)?
3. Is the phone nasal ?
4. Is the phone long ?
5. Is the phone stressed (primary stress) ?
6. Is the phone stressed (secondary stress) ?
7. Is the phone in a dialog part ?
8. Is the phone in the last syllable of its sentence?
9. Is the phone in the last syllable of its breath group?
10. Is the phone in a syllable with a rising intonation ?
11. Is the phone in a syllable with a falling intonation ?
12. Is the current syllable in word end?

The two first filters, written as mandatory, cannot be relaxed as they represent the minimal information to retrieve units.

3.5. Fuzzy concatenation cost

Analysis of synthesized sentences containing artefacts shows that concatenation on some phonemes, especially vowels and semi-vowels, is more likely to engender artefacts than others (plosives and fricatives for example, especially unvoiced ones) [12]. Phonemes featuring voicing, high acoustic energy or important context dependency are generally subject to more distortions. Based on this ascertainment, [13, 20] proposed a corpus covering criterion where the objective is to get a maximum covering of "sandwich units". A sandwich unit is a sequence of phonemes where one or several syllabic nuclei are surrounded by two phonemes considered as robust to concatenation artefacts. Concerning unit selection concatenation costs, a few work can also be cited, for example [21, 22], but in these works, costs and penalties are not flexible enough. In unit selection, too many constraints generally means loss of quality (e.g. too many preselection filters is to prevent).

In the approach we introduced in [23, 24], we have defined a fuzzy concatenation cost taking into account three phonetic clusters:

V (vowel) : Vowels, on which concatenation is hardly acceptable.

A (acceptable) : Semi-vowels, liquids, nasals, voiced fricatives and schwa. These units are viewed as acceptable concatenation points, but still precarious.

R (resistant) : The remaining phonemes (unvoiced consonants, voiced plosives), where concatenation is definitely possible.

First, we give a fixed penalty to each phoneme class: 0 for phonemes in R, a penalty slightly higher than the highest value C_c observed in the corpus for all phonemes in A. Vowels (V) are given a huge penalty, big enough to prevent compensation by other costs in the candidate sequence. It corresponds to a penalization of candidate units based on the phonemes on which concatenation may be performed if choosing this unit. In this case, a new concatenation cost function C'_c is formulated as:

$$C'_c(u,v) = C_c(u,v) + K(u,v), \tag{2}$$

where $K(u,v) = p(v)$ is the penalty depending on the phoneme that begins the unit v as described before.

In order to relax this penalty when a concatenation between two candidate units is statistically among the best ones, we introduce a fuzzy weighting function, ranging from 0 to 1. It describes how much the unit belongs to one of the clusters defined earlier.

Assuming MFCC, Amplitude and F0 cost distributions follow normal distributions, we define two thresholds for each sub-cost. For instance, the two thresholds $T^1_{F_0}$ and $T^2_{F_0}$ for the F0 sub-cost may be defined as:

$$T^1_{F_0} = \mu_{C_{F_0}} - \sigma_{C_{F_0}} \tag{3}$$

$$T^2_{F_0} = \mu_{C_{F_0}} + \sigma_{C_{F_0}} \tag{4}$$

Formally, the fuzzy function is defined, for the F0 sub-cost:

$$f_{F_0}(u,v) = \begin{cases} 0 & \text{if } C_{F_0}(u,v) < T^1_{F_0}, \\ 1 & \text{if } C_{F_0}(u,v) > T^2_{F_0}, \\ 1.0 - \frac{(T^2_{F_0} - C_{F_0}(u,v))}{(T^2_{F_0} - T^1_{F_0})} & \text{otherwise.} \end{cases} \tag{5}$$

The choice for that tolerance interval is motivated by the observation of real cost distributions. To be complete, the choice of the thresholds should be differentiated depending on the type of sub-cost and optimized separately. In this paper, we used thresholds corresponding to 15% of the distribution for each sub-cost.

The penalty is then modified in the following way:

$$K(u,v) = (f_{mfcc}(u,v) + f_{amp}(u,v) + f_{F0}(u,v)) * p(v)$$

where $f_{mfcc}(u,v)$, $f_{amp}(u,v)$ and $f_{F0}(u,v)$ correspond to the fuzzy function of the form described previsouly respectively for MFCC, amplitude and F0. The value $p(v)$ is still the generic penalty value that depends on the phoneme class and which is not weighted.

With this fuzzy function, the main idea is to decrease the penalty when the unit has a concatenation sub-cost value which is statistically among the best ones. The sub-cost distributions are estimated from the voice corpus by computing concatenation sub-costs for F0, amplitude and MFCC using all the units in the corpus.

3.6. Break prediction

As we had no pause prediction module until now and as it is very important for audiobooks, specifically for children, we tried to introduce a simple prediction block in our system. We have chosen to formulate it as a classification task by assigning a label to each word telling if it is followed by a pause or not, and also the type of the pause. To simplify, we make 4 clusters:

- *No Break (NB)*: length less than 120ms;
- *Short Break (SB)*: length between 120ms and 250ms;
- *Medium Break (MB)*: length between 250ms and 550ms;
- *Long Break (LB)*: length greater than 550ms.

To predict those labels, we used a random forest classifier [25] with 100 estimators and 77 features as input which consisted of linguistic features, positional features and also *a narrative vs. dialog feature* with a window of ± 2. We used the Scikit toolkit [26] to learn the classifier.

Using the data provided, the overall accuracy score is 84%. The confusion matrix is given in table 2. As we can observe, NB class is very well predicted with nearly no confusion which is understandable as it is the dominating value in the data set. The most confused class is the SB one. Interestingly, the longer the break, the less confusion. Despite the unbalanced classes, the random forest is doing quite well.

Table 2: Normalized confusion matrix

	NB	SB	MD	LB
NB	98.1	0.6	1.0	0.2
SB	28.8	28.4	32.9	9.9
MB	14.7	19.2	46.1	20.1
LB	6.9	6.0	31.2	55.9

4. Evaluation and results

The evaluation assessed a number of criteria (overall impression, speech pauses, intonation, stress, emotion, pleasantness and listening effort) for book paragraphs as well as similarity to the original speaker, naturalness and intelligibility. The evaluation has been conducted for different groups of listeners: paid listeners, speech experts, and volunteers. In this section, we only give results including all participants. In every figure, results for all 17 systems are given. Among the system, we have the following : system A is natural speech, system B is the Festival benchmark system (standard unit selection), system C is the HTS benchmark and system D is a DNN benchmark. System H (in orange on all figures) is the system presented by IRISA.

4.1. Evaluation with paragraphs

Results are shown on figures from 1 to 7. For each criterion, our system achieves average results, except for the speech pauses evaluation (fig. 3) showing a low score. After the submission of speech samples for the evaluation, we have detected a bug in the break prediction module causing an inconsistency between results during the learning step and its use in the synthesis chain. Concerning the other criteria, these average results are likely to be explained by the quality of the phone segmentation. A manual checking revealed an alignment problem that couldn't be solved before the submission.

4.2. Similarity to original speaker and naturalness

The similarity of the speech produced, as shown on figure 8, is among the best ones with a mean score of 3.6 and the median value at 4. Similarly, naturalness is also quite good as shown on figure 9 with an average of 2.8 and a median of 3. Those results are coherent with the nature of our system. Unit selection uses natural stimuli and consequently preserves the original speaker's voice.

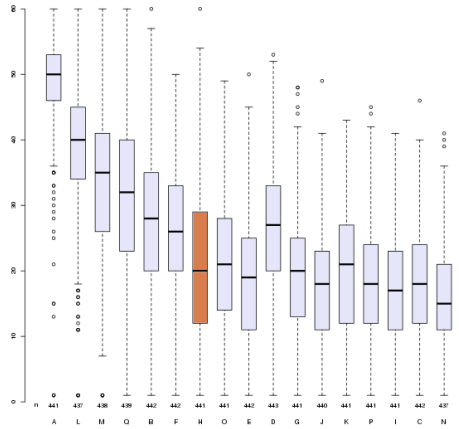

Figure 1: Mean Opinion Scores, overall impression evaluation, all listeners (from ”bad” to ”excellent”).

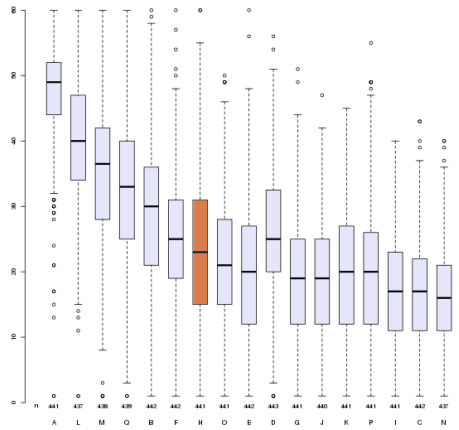

Figure 2: Mean Opinion Scores, pleasantness evaluation, all listeners (from ”very unpleasant” to ”very pleasant”).

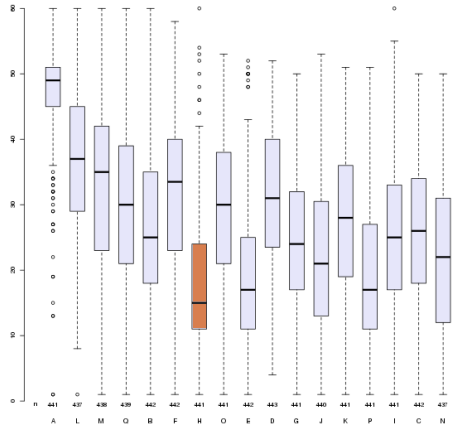

Figure 3: Mean Opinion Scores, speech pauses evaluation, all listeners (from ”speech pauses confusing/unpleasant” to ”speech pauses appropriate/pleasant”).

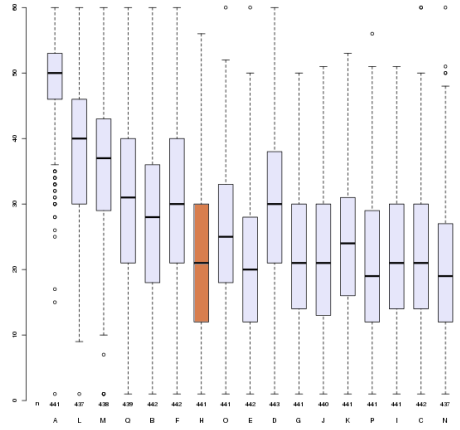

Figure 4: Mean Opinion Scores, intonation evaluation, all listeners (from ”melody did not fit the sentence type” to ”melody fitted the sentence type”).

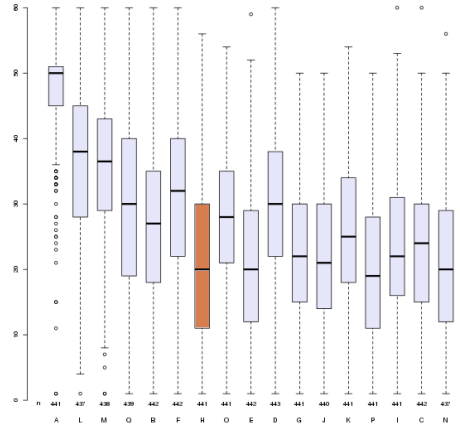

Figure 5: Mean Opinion Scores, stress evaluation, all listeners (from ”stress unnatural/confusing” to ”stress natural”).

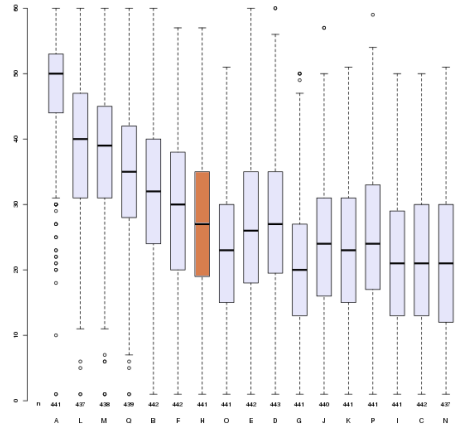

Figure 6: Mean Opinion Scores, emotion evaluation, all listeners (from ”no expression of emotions” to ”authentic expression of emotions”).

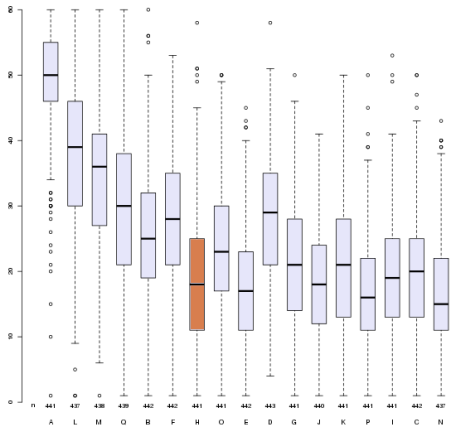

Figure 7: Mean Opinion Scores, listening effort evaluation, all listeners (from ”very exhausting” to ”very easy”).

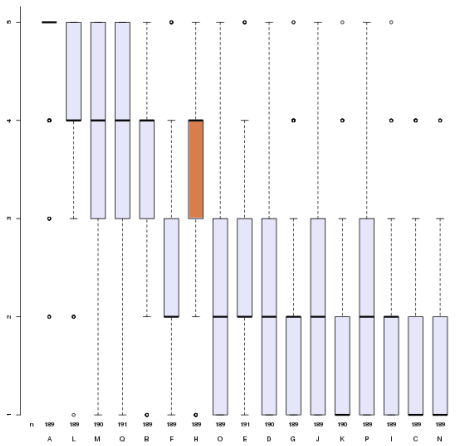

Figure 8: Mean Opinion Scores, similarity to the original speaker evaluation, all listeners.

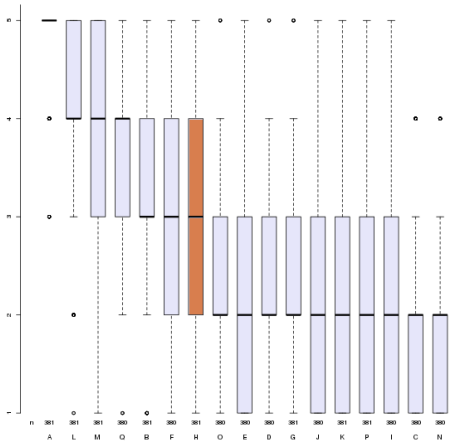

Figure 9: Mean Opinion Scores, naturalness evaluation, all listeners.

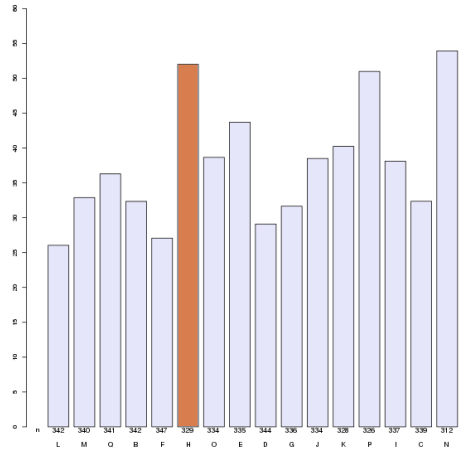

Figure 10: Word Error Rates, naturalness evaluation, all listeners.

4.3. Intelligibility

Concerning intelligibility, our system performed poorly compared to the majority of other systems with an average word error rate of 52%. Detailed results are given on figure10. The main explanation is the low quality of phone segmentation we obtained.

5. Conclusion

We described the unit-selection based IRISA system for the Blizzard challenge 2016. The unit selection method is based on a classic concatenation cost to which we add a fuzzy penalty that depends on phonological features. In order to improve the system, we added specific costs to deal with speech rate, melody and speaking style (narrative and dialog) consistency. Despite the improvements we've made, our system obtained average results. The main explanation is that the phone segmentation system we used performed badly on the given data. This was the cause of a drop in nearly all criteria. Concerning pause prediction, we have found a bug in the pause prediction module that will be fixed for next year edition.

6. Acknowledgements

This study has been partially funded thanks to the ANR (French National Research Agency) project SynPaFlex ANR-15-CE23-0015.

7. References

[1] J. Yamagishi, Z. Ling, and S. King, "Robustness of HMM-based speech synthesis," in *Proc. of Interspeech*, 2008, pp. 2–5.

[2] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7962–7966.

[3] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. of ICASSP*. Ieee, 1988, pp. 679–682.

[4] A. W. Black and P. Taylor, "CHATR: a generic speech synthesis system," in *Proc. of Coling*, vol. 2. Association for Computational Linguistics, 1994, pp. 983–986. [Online]. Available: http://dl.acm.org/citation.cfm?id=991307

[5] A. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. of ICASSP*, vol. 1. Ieee, 1996, pp. 373–376.

[6] P. Taylor, A. Black, and R. Caley, "The architecture of the Festival speech synthesis system," in *Proc. of the ESCA Workshop in Speech Synthesis*, 1998, pp. 147—-151.

[7] A. P. Breen and P. Jackson, "Non-uniform unit selection and the similarity metric within BTs Laureate TTS system," in *Proc. of the ESCA Workshop on Speech Synthesis*, 1998, pp. 373–376.

[8] R. A. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[9] H. Patil, T. Patel, N. Shah, H. Sailor, R. Krishnan, G. Kasthuri, T. Nagarajan, L. Christina, N. Kumar, V. Raghavendra, S. Kishore, S. Prasanna, N. Adiga, S. Singh, K. Anand, P. Kumar, B. Singh, S. Binil Kumar, T. Bhadran, T. Sajini, A. Saha, T. Basu, K. Rao, N. Narendra, A. Sao, R. Kumar, P. Talukdar, P. Acharyaa, S. Chandra, S. Lata, and H. Murthy, "A syllable-based framework for unit selection synthesis in 13 indian languages," in *Proc. O-COCOSDA*, 2013, pp. pp.1–8.

[10] Y. Stylianou and A. Syrdal, "Perceptual and objective detection of discontinuities in concatenative speech synthesis," *Proc. of ICASSP*, vol. 2, pp. 837–840, 2001.

[11] D. Tihelka, J. Matoušek, and Z. Hanzlíček, "Modelling F0 Dynamics in Unit Selection Based Speech Synthesis," in *Proc. of TSD*, vol. 1, no. Springer, 2014, pp. 457–464.

[12] J. Yi, "Natural-sounding speech synthesis using variable-length units," Ph.D. dissertation, 1998.

[13] D. Cadic, C. Boidin, and C. D'Alessandro, "Vocalic sandwich, a unit designed for unit selection TTS," in *Proc. of Interspeech*, no. 1, 2009, pp. 2079–2082.

[14] J. Duddington, "espeak text to speech," 2012.

[15] F. Schiel, "Automatic phonetic transcription of non-prompted speech," in *Proc. ICPhS*, 1999, p. pp. 607610.

[16] J. Chevelu, G. Lecorvé, and D. Lolive, "ROOTS: a toolkit for easy, fast and consistent processing of large sequential annotated data collections," in *Proc. of LREC*, 2014, pp. 619–626.

[17] D. Guennec and D. Lolive, "Unit Selection Cost Function Exploration Using an A* based Text-to-Speech System," in *Proc. of TSD*, 2014, pp. 432–440.

[18] C. Blouin, O. Rosec, P. Bagshaw, and C. D'Alessandro, "Concatenation cost calculation and optimisation for unit selection in TTS," in *IEEE Workshop on Speech Synthesis*, 2002, pp. 0–3.

[19] F. Alías, L. Formiga, and X. Llorá, "Efficient and reliable perceptual weight tuning for unit-selection text-to-speech synthesis based on active interactive genetic algorithms: A proof-of-concept," *Speech Communication*, vol. 53, no. 5, pp. 786–800, May 2011.

[20] D. Cadic and C. D'Alessandro, "High Quality TTS Voices Within One Day," in *Seventh ISCA Workshop on Speech Synthesis*, 2010.

[21] R. E. Donovan, "A new distance measure for costing spectral discontinuities in concatenative speech synthesizers," in *ITRW*, 2001.

[22] J. Yi, J. Yi, J. Glass, and J. Glass, "Natural-sounding speech synthesis using variable-length units," *Proc. of ICSLP*, 1998.

[23] P. Alain, J. Chevelu, D. Guennec, G. Lecorvé, and D. Lolive, "The IRISA Text-To-Speech System for the Blizzard Challenge 2015," in *Blizzard Challenge 2015 Workshop*, Berlin, Germany, Sep. 2015. [Online]. Available: https://hal.inria.fr/hal-01196168

[24] D. Guennec and D. Lolive, "On the suitability of vocalic sandwiches in a corpus-based tts engine," in *Interspeech*, 2016.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.