The IRISA Text-To-Speech System for the Blizzard Challenge 2015

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 2015. For our first participation, we chose to take part in the first task by building one voice for each of the 5 Indian languages out of the 6 requested (Bengali, Hindi, Malayalam, Tamil and Telugu). We describe the process followed to build the voices from given data and the architecture of our system. In particular, we introduce a penalty in the concatenation cost, inherited from the field of corpus covering, in order to block some concatenations based on their phonological class. Moreover, a fuzzy function is proposed to relax the penalty based on the concatenation quality with respect to the cost distribution. Considering that no language specific processing has been done and that we had never worked with Indian languages, results are very satisfying.

Index Terms: speech synthesis, unit selection

1. Introduction

In recent years, research in text-to-speech synthesis essentially focused on two major techniques. The parametric approach, for which HTS [1] is the main system, is the most recent and has been the framework used for many academic work in the recent years. This method offers advanced control on the signal and produces very intelligible speech but with a low naturalness. The historical approach, unit selection, is a refinement of concatenative synthesis [2, 3, 4, 5, 6, 7, 8]. Sound created 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.

This year, the challenge focuses on 6 different Indian languages. About 4 hours of speech data is available in each of three Indian languages (Hindi, Tamil and Telugu), and about 2 hours for the other three (Marathi, Bengali and Malayalam), all recorded by native professional speakers in high quality studio environments. One key issue for us is to verify that our system is able to synthesize speech as these languages are very different to English and French, and also to compare our choices to state of the art systems from the community.

In this paper we present the unit-selection based IRISA system for the Blizzard Challenge 2015. 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 cost function 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 [9, 10]. In particular, for the challenge, we have introduced a penalty on units whose concatenation is considered as risky. This follows the work of [11, 12] 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. This approach has been originally proposed in the context of recording script construction in [12] 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. Sections 4 presents the evaluation and results. Finally, section 5 gives a short comment on the pilot task.

2. General voice creation process

For each voice, a set of wave files with the corresponding text has been provided. To build a voice from these files, 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* [13]. It enabled us to successfully treat 4 languages: Bengali, Hindi, Malayalam, and Tamil. The two other languages, Marathi and Telugu, do not seem to be supported by the software. Nonetheless, a fallback solution has been set up for Telugu by using transliteration with IT3 scripts. Unfortunately, no solution has been found for Marathi and we thus did not submit any voice for this language. Once phonetized, speech signals have been segmented using the language independent segmenter *MAUS* [14]. We have also used the *ROOTS* toolkit [15] to store all the necessary information and to do 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 [16], used for the experiments presented in this paper, relies on a unit selection approach with an optimal graph-search algorithm (here A* algorithm). The optimization function is divided, as usually done, in two distinct parts; a target and a concatenation cost [3] as described below:

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

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 [17]. In practice, the weight for each sub-cost c 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. [18] is a good review of the most common methods. Our concatenation cost $C_c(u, v)$ between units u and v is composed of MFCCs (excluding Δ and $\Delta\Delta$ coefficients), amplitude and F0 euclidean distances, as below:

$$C_c(u, v) = C_{mfcc}(u, v) + C_{amp}(u, v) + C_{F0}(u, v), \quad (2)$$

where $C_{mfcc}(u, v)$, $C_{amp}(u, v)$ and $C_{F0}(u, v)$ are the three sub-costs for MFCC, amplitude and F0. When exploring new nodes 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 doesn't 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 node. In our system, when no corpus unit 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 case a diphoneme is not present in the corpus, a fallback mechanism has been implemented. Precisely, the requested diphone is built artificially by concatenating two phonemes in context of a pause. As it does not take into account co-articulation effects, the result is not excellent but it at least 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 the last syllable of its breath group?
8. Is the phone in the last syllable of its sentence?
9. 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.

3.2. 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) [11]. Phonemes featuring voicing, high acoustic energy or important context dependency are generally subject to more distortions. Based on this ascertainment, [12, 19] 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 [20, 21], 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 our approach, we have defined two distinct concatenation sub-costs 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.

A first method is to 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:

$$C'_c(u, v) = C_c(u, v) + K(u, v), \quad (3)$$

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

In a second time, 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 as shown on figure 1. It describes how much the unit belongs to one of the clusters defined earlier. For each sub-cost (F0, amplitude and MFCC), if its value is within the 15% lower costs observed in the voice corpus, the weight 0 is applied to the penalty (canceling it). If it is within the 15% biggest costs observed, the full penalty (weight = 1) is applied. Between these extrema, a linear function is used to determine the weight to apply to the penalty. The penalty is then modified in the following way:

$$\begin{aligned} K(u, v) = &(f_{mfcc}(u, v) + f_{amp}(u, v) + f_{F0}(u, v)) \\ &* p(v) \end{aligned}$$

where $f_{mfcc}(u, v)$, $f_{amp}(u, v)$ and $f_{F0}(u, v)$ correspond to the fuzzy function of the form described in figure 1 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 used for the challenge, 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 subcost distributions are estimated from the voice corpus by computing concatenation sub-costs for F0, amplitude and MFCC using all the units in the corpus. Then given a manually chosen threshold, the decision to remove or diminish the penalty for a unit not matching the constraint (no resistant phoneme at extremity) is taken. If the concatenation cost is above the

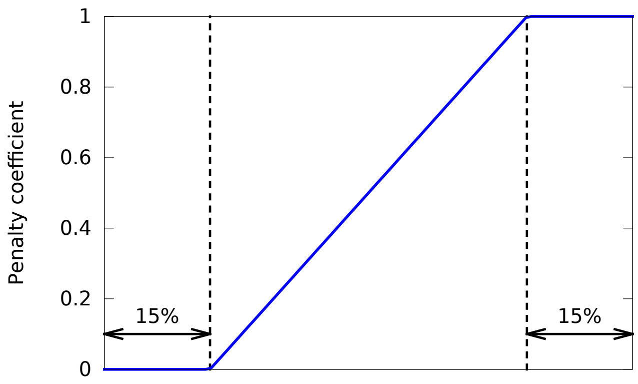

Figure 1: *Fuzzy function over the distribution of sub-costs. Weight 0 (resp. 1) is given to units that have a concatenation cost among the 15% lowest (resp. highest) costs. Between these thresholds, the weight increases linearly.*

highest threshold then the full penalty has to be applied as the unit considered is among worst possible units. Between the two thresholds, the penalty is augmented progressively as the concatenation cost increases.

4. Evaluation and results

The conducted perceptive evaluations involves two sub-corpora of test sentences for each language: a set of 50 ReaD sentences (RD) and a set of 50 Semantically Unpredictable Sentences (SUS). Three types of evaluations are then conducted with paid listeners: Mean Opinion Score (MOS) tests were performed to measure the *similarity to the speaker* and *naturalness* while *intelligibility* of the produced speech signals has been evaluated by means of manual transcription resulting in Word Error Rates (WERs). Details of the evaluation process are given in the summary paper. This section presents and comments results obtained for these evaluations by the IRISA system.

4.1. Similarity to the speaker

Similarity results are summarized in Table 1. Numbers in bold are those for which the IRISA system achieved the best results among all systems involved in the challenge. We can see that the overall results for similarity are rather good and among the very best ones compared to the other systems. This conclusion is not surprising since our system is based on unit selection and thus directly relies on natural speech from the speaker, as opposed to generating entirely new wave forms. For all languages, results are lower on SUS which can be explained by the fact that the words used may not be frequent or at least the fact that they appear jointly is not common. This may cause the use of rare diphones and therefore a fall in quality of concatenations. For Telugu, similarity on RD is very high (4.2) and not very far from similarity with natural speech (4.5) while the similarity of the second best system is much lower (3.1). The only difference for Telugu is that, as we had no phonetizer, we have directly used transliteration with IT3 scripts.

4.2. Naturalness

For naturalness, our system achieves average results (see Table 2). We have not used anything specific to Indian languages in our system. A difference exists in the numbers between RD and SUS with a fall in naturalness. Again, this is explained by the unit selection process which performs better when a large number of diphonemes is present in the voice. When using SUS, it is less the case due to words used in those sentences.

An interesting point is the drop in similarity and naturalness for Telugu with SUS. It is not present for other languages even if a decrease exists. As mentioned before, for Telugu, we have used transliteration instead of a phonetizer but it seems that it would make no difference. Then, an explanation may be that the speaker is a news reader with a high speaking rate and a rather expressive voice. The effect is strongly increased as the sentences are SUS.

As we only use data provided for the challenge and the size is quite limited, we may obtain better results with larger corpora.

4.3. Intelligibility

Finally, as summarized by the very high WERs in Table 3, the intelligibility of speech signals produced by our system seems to be low. Despite the globally bad numbers, these results are comparable to those of the other systems involved in the challenge. The only exception is the result for Bengali which is 100% (only two systems have this value) of transcription error. However, overall, some methodological issues may be pointed out to explain these high WERs and Some reasons given by the organizers of Blizzard challenge may explain such high WER:

- Native speakers are not used to typing Indian language scripts as there is no standard keyboard layout.
- For some testers, it is the first time they type in full sentences in Indian languages.
- Google transliteration APIs that are used require a space to be pressed before the ASCII character is changed to UTF8 script. The space is often missed by testers.
- WER computation is done using a binary match which gives a lot of errors (for instance considering the distinction between long vs. short vowels).

Table 1: *MOS results for similarity with the 5 languages. RD and SUS are acronyms for, resp., ReaD sentences (RD) and Semantically Unpredictable Sentences (SUS). Each time, results are presented in the following format: mean (std). N represents the number of votes.*

Language	**Similarity**		**N**	
	RD	SUS	RD	SUS
Bengali	2.9 (1.08)	2.7 (1.08)	48	48
Hindi	3.5 (1.11)	2.8 (1.02)	69	69
Malayalam	**3.0** (1.24)	**2.7** (0.90)	72	72
Tamil	**3.6** (1.11)	**3.2** (0.97)	70	70
Telugu	**4.2** (0.97)	2.9 (1.10)	70	70

5. Pilot task

This year, the pilot task for preparing next challenge is about synthesizing audiobooks in English targeted at children. 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.

Table 2: *MOS results for naturalness with the 5 languages. RD and SUS are acronyms for, resp., read sentences and Semantically Unpredictable Sentences. Each time, results are presented in the following format: mean (std). N represents the number of votes.*

Language	Naturalness		N	
	RD	SUS	RD	SUS
Bengali	2.3 (1.14)	2.1 (0.93)	192	144
Hindi	3.3 (1.10)	3.2 (1.09)	276	207
Malayalam	**3.2** (1.36)	**2.9** (0.87)	288	216
Tamil	3.4 (1.19)	3.0 (1.22)	280	210
Telugu	1.9 (1.04)	2.1 (0.86)	280	210

Table 3: *WER results for the 5 languages computed on SUS test sub-corpus (in %).*

Language	mean WER (std)
Bengali	100 (0)
Hindi	31 (21)
Malayalam	73 (18)
Tamil	50 (24)
Telugu	62 (19)

First attempts point out that we need a speech/non speech detector and also a set of labels describing precisely what type of non speech sound is occurring. Second, speaker segmentation or at least speech type detection seems necessary to deal with the character change problem.

6. Conclusion

We described the unit-selection based IRISA system for the Blizzard challenge 2015. The unit selection method is based on a classic concatenation cost to which we add a fuzzy penalty that depends on phonological features. No Indian languages dependent features have been used. This system has obtained average results that are satisfying for our first participation, especially addressing languages we had never synthesized before.

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

[3] 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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

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