Investigating the Relation Between Voice Corpus Design and Hybrid Synthesis Under Reduction Constraint

Meysam Shamsi(✉), Damien Lolive, Nelly Barbot, and Jonathan Chevelu

Univ Rennes, CNRS, IRISA, Lannion, France
{meysam.shamsi,damien.lolive,nelly.barbot,jonathan.chevelu}@irisa.fr

Abstract. Hybrid TTS systems generally try to optimise their cost function with the voice provided to generate the best signal. The voice is based on a speech corpus usually designed for a specific purpose. In this paper, we consider that the voice creation is realized through a corpus design step under reduction constraints. During this stage, a recording script is crafted to be optimal for the target TTS engine and its purpose. In this paper, we investigate the impact of sharing information between the corpus design step and the hybrid TTS optimisation step.

We start from a reduced voice optimized for a unit selection system using a CNN-based model. This baseline is compared to a hybrid TTS system that uses, as its target cost, a linguistic embedding built for the recording script design step. This approach is also compared to a standard hybrid TTS system trained only on the voice and so that does not have information about the corpus design process.

Objective measures and perceptual evaluations show how the integration of the corpus design embedding as target cost outperforms a classical hard-coded target cost. However, the feed-forward DNN acoustic model from the standard hybrid TTS system remains the best. This emphasizes the importance of acoustic information in the TTS target cost, which is not directly available before the voice recording.

Keywords: Hybrid speech synthesis · Corpus reduction · Linguistic and Phonological embeddings

1 Introduction

Nowadays, there are two main strategies for Text-To-Speech (TTS) synthesis. The first one is based on unit selection [1] and the second one is the Statistical Parametric Speech Synthesis (SPSS) [2,3]. The basic idea of unit selection-based TTS is to choose and concatenate a sequence of units from a natural speech corpus. The selected units should have linguistic features as close as possible to the target ones, associated to the text to vocalize, and the concatenations of consecutive unit signals should minimize differences in their joins. SPSS uses a vocoder and is known for the smoothness of its generated signals and

© Springer Nature Switzerland AG 2019
C. Martín-Vide et al. (Eds.): SLSP 2019, LNAI 11816, pp. 162–173, 2019.
https://doi.org/10.1007/978-3-030-31372-2_14

its flexibility. Conversely, unit selection based TTS systems provide more natural-sounding signals than SPSS [2,4].

The advantages and disadvantages of these TTS systems naturally led to the design of hybrid systems. The combination of both systems usually involves statistical models trained on the voice to predict parameters of an ideal generated speech and to guide a unit selection decoder that concatenates real signal segments extracted from the voice. Recent studies and the last Blizzard challenges have revealed good achievements of hybrid systems (see for instance [4–6]). In the last years, deep learning models such as Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) have been successfully used as acoustic models in hybrid systems, replacing HMMs, like in [7]. The main challenge in designing acoustic models is that the linguistic sequence does not have the same length as the acoustic sequence. For instance, in [8], a one to many approach is followed to deal with this problem. A LSTM-based auto-encoder is employed and permits to generate a sequence of acoustic frames representative of the input phoneme. As another example, in [9], each candidate phone unit is converted into a fix-length unit vector, called *Unit2Vec*, and DNNs are used as target and concatenation cost functions.

In order to manage the variable sequence length problem, a similar process has been applied in [10], a feed forward DNN for a one to one approach models phoneme frames, based on frame position, and the euclidean distance in the embedding space is used as the TTS target cost function. This approach also provides better results than an expert target cost.

In all cases, hybrid TTS systems are trained on a speech corpus independently of how it has been built. It may not lead to a significant difference when the voice is large enough, offering a good internal acoustic diversity. But, when the size of the voice is constrained in some ways, as in industrial applications which often need a high quality recorded voice, the adequacy between the voice and the TTS engine may impact the quality of the generated signals [11–13].

The cost, e.g. in terms of annotation time or recording time, to build a TTS voice for a professional usage is correlated to the length of the recording script. Hence, creating a voice under cost constraints requires to craft carefully the sentences to guarantee a good TTS quality in the end. To design such a script, a usual method is the selection of a subset of sentences as short and linguistically rich as possible from a large text corpus. This approach can be formalized as an optimisation problem in a discrete space [14]. The properties that the linguistic and phonological content of this subset has to achieve can stem from TTS engine needs or from the considered application independently or not of the TTS system. For instance, in [18,19], the phonological distribution in the script has to be close to a target one: natural, uniform or representative of a given domain. Conversely, the constraints and the nature of attributes to cover can be specific to the TTS engine, like in [17] where the phonological attributes used for the target cost function are covered, or in [20,21] where the internal descriptors of a SPSS system are considered, or, also in [19] where a pruning is done to remove units that are least used by the unit selection TTS system.

The resolution of this set covering problem for TTS corpus design has been widely studied in past studies [14–17].

Whereas the unit selection approach can support a small well-adapted voice corpus, the learning processes in hybrid systems are greedy in terms of voice data. Therefore, one may ask how to address and improve the use of hybrid TTS systems in a context of parsimonious voice building. In this paper, we investigate how the information from the voice creation process can be useful to help a hybrid TTS engine. To avoid disruption in the experiments, it focuses only on the inclusion of the information as the target cost of the hybrid TTS system. Using a unique voice, built from a simulated and controlled corpus design process, three variants of the same system are compared. The first one is based on an expert target cost function as in classical unit selection framework, whereas the target cost function of the second one is trained on linguistic, phonological and acoustic contents of the voice. This second approach illustrates a usual hybrid TTS system, as described in [10]. At last, the third approach uses a target cost function whose definition takes into account the voice creation process. The proposed method relies on the partition and covering of the embedding space used to design the recording script. Since this embedding is learnt before the recording phase, only linguistic and phonological features are required. Using objective and foremost perceptual evaluations, the experiments help to understand relations between corpus design and hybrid TTS.

This paper is organized as follows. First, Sect. 2 introduces the experimental framework. Especially, it explains how the corpus design is simulated and presents the resources used for training. Since all compared systems use the same voice, a voice creation process under size constraint is described in Sect. 3. This process is compared with the standard set-covering approach as a preliminary experiment. Section 4 details the different systems considered and especially differences between the hybrid ones. Evaluations and results are given in Sect. 5 before an overall discussion in Sect. 6.

2 Experimental Framework

In order to carry out the experiments presented in this paper and take into account the assumption of a recording phase, we have avoided the constraint of this recording work by reducing an already recorded and annotated corpus as in [11,25]. We have chosen an audio-book read by a professional speaker as initial corpus, thus limiting the bias inherent to the recording phase (speaker experience, recording conditions, etc.).

From this book, a randomly selected continuous part $\mathcal{T}$ has been taken away as a test set and the other part, denoted $\mathcal{F}$, is named the full corpus in the remainder.

The voice creation step is simulated by the selection of a sentence set $\mathcal{S}$ from $\mathcal{F}$, based on linguistic and phonological features only; the voice corresponds to the set of the signals associated to $\mathcal{S}$. The objective is to find the best set $\mathcal{S}$ to synthesize the entire book, and the voice quality is evaluated using the subset $\mathcal{T}$.

To illustrate the recording time constraint, $\mathcal{S}$ may be not longer than a given ratio of $\mathcal{F}$ in number of phoneme instances, and for the presented experiments, this ratio has been set to 10% of $\mathcal{F}$.

The initial corpus, i.e. the entire audio-book, contains 3,339 utterances of a French expressive audio-book spoken by a male speaker. The overall length of the speech corpus is 10 h44. More information on the annotation process can be found in [23]. $\mathcal{F}$ is composed of 3,005 utterances and 362,126 phoneme instances. The test set $\mathcal{T}$ contains the 334 other sentences from the initial corpus.

For all experiments, synthesis is done by the IRISA TTS unit selection system [22]. It can also be used as a hybrid TTS system like in [10].

3 Voice Construction and Preliminary Experiment

As explained in Sect. 1, several approaches can be used for corpus design under size constraint. The corpus design method used to build the voice that feeds all evaluated systems should be carefully selected so that the comparisons are fair. It needs to be usable in a hybrid TTS context and also leads to good performances with a unit-selection system. Among the methods optimized for a specific TTS engine (as in [17]) and others based on distributional information about the target domain (as in [19]), the latter seems preferable. This is particularly true since the corpus used here is in a consistent domain (a full audio-book as explained in 2). At last, distributional information can be well modeled by Neural Networks and can then be integrated into a hybrid TTS workflow. This section details the proposed corpus design method used to create the voice in further experiments. Moreover, in a preliminary experiment, this method will be compared to standard approaches to ensure its relevance.

The way to select sentences for the voice is accomplished as follows. Utterances from $\mathcal{F}$ are used to train an auto-encoder based on a multi-layer Convolution Neural Network (CNN) as illustrated in Fig. 1. The activation function is tanh and the loss function is the Mean Squared Error (MSE). The input vectors are composed of 296 components of categorical and numerical types automatically computed. The categorical attributes represent information about quinphonemes, syllables, articulatory features, and Part Of Speech for the current, previous and following words. These features are converted to a one-hot vector. The numerical features take into account information such as the phoneme position inside the word or utterance. These numerical features are normalized so that all the entries of the linguistic vector are in the range $[0, 1]$. The linguistic content of each input utterance is then represented by the sequence of linguistic feature vectors associated to the phonemes that compose it.

By taking the encoder part as the embedding model, each utterance of $\mathcal{F}$ is transformed into a sequence of vectors in the embedding space and its associated average vector is chosen to characterize this utterance. This results in a fixed-length vector whose size is the number of features (30) in the embedding space for each utterance. A K-Means algorithm is then applied to partition the set of average vectors represented $\mathcal{F}$. From each cluster, the utterance whose the

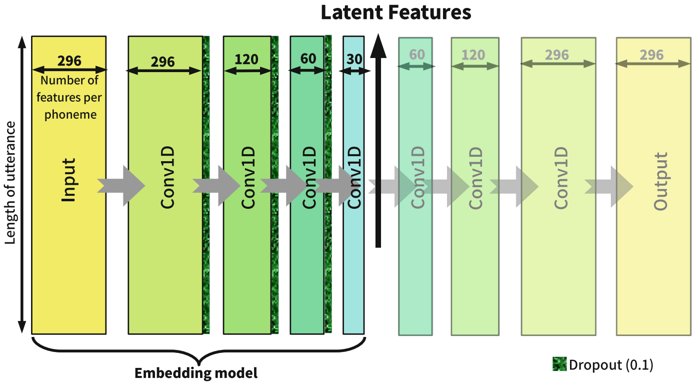

Fig. 1. Deep convolutional auto-encoder used to train linguistic and phonological embeddings.

Table 1. Objective evaluation of the proposed script design strategy using the TTS global cost.

Corpus design method	Average TTS global cost	95% confidence interval
Random	1.77	±0.01
Set covering Greedy	1.75	±0.02
CNN+KMeans	**1.60**	±0.02

average vector is the closest to the center is selected and add to $\mathcal{S}$. This subset $\mathcal{S}$ is thus built to represent the linguistic richness of F by covering all its clusters, with the length about 10% from that of $\mathcal{F}$. The natural speech signals associated to elements of $\mathcal{F}$ are used as the TTS voice corpus of the experiments described below.

In order to assess the achievements of this script design method and its derived voice, a second voice with an identical length is built using a classic set covering strategy [17]. For this, the features used are diphones with the same linguistic as for the CNN. The utterances of $\mathcal{T}$ are then vocalized using the two voices respectively but the same TTS system, namely the IRISA system based on an expert target cost function. Generated outputs are objectively evaluated using the TTS global cost (a linear combination of target and concatenation costs) and also compared using a perceptual assessment. Besides, as baseline, for each utterance of $\mathcal{T}$, the average TTS global cost stemming from the use of 10 randomly selected voices is added. As for the perceptual evaluation, it is conducted in the form of an AB test with 17 listeners. From the 334 samples of $\mathcal{T}$, 100 samples are evaluated at least 6 times. Results are summarized in Table 1 and Fig. 2.

Whereas the TTS global cost mean provided by the standard set covering is close to the one resulting from the random selection method, the CNN-Kmeans

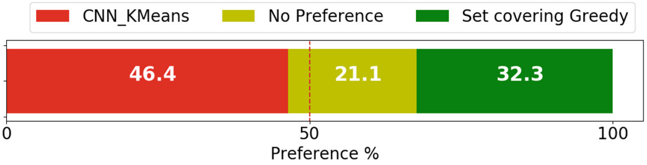

Fig. 2. Perceptual evaluation of the proposed script design strategy.

based corpus design method gives a significantly lower TTS global cost mean. This latter approach is also preferred during the listening test.

For the next experiments, we then keep this voice and the associated creation process based on the combination of CNN and KMeans algorithm. They will be used with different TTS engine configurations to investigate the relation between voice creation and hybrid synthesis.

4 TTS Systems Under Comparison

The objective of the paper is answering to this question: *Is it helpful to use the same phoneme representation in the corpus design step and in the TTS target cost?*

To do so, three methods for calculating the TTS target cost are compared. An expert target cost function which is a weighted sum of linguistic features is used as the baseline. The two other methods are based on embedded representations at phone level. The first one uses the same embedding for the corpus design step and the target cost function while the third one uses a specific embedding for the target cost function taking into account acoustics. The target cost is the euclidean distance in the embedding space between the candidate phone and the target one.

In the following, these three systems are described and then compared.

4.1 Expert-Based Target Cost (*Exp*)

In this method, the system used is a state of the art unit selection system. The target cost is defined as a weighted sum of linguistic features and has since been improved over the years [10]. The concatenation cost is the same as in [22], defined as a sum of euclidean distances on acoustic features between consecutive units.

4.2 Same Embedding for Corpus Design and TTS (*CNN*)

The second method replaces the expert target cost function by a cost function relying on the phoneme level embedding created during the corpus design step. Consequently, we propose here to use the same embedding model and phoneme representation for both corpus design and TTS target cost. The *CNN* auto-encoder described in Sect. 3 represents the linguistic information of phoneme by a vector of latent features. The TTS target cost is the euclidean distance in the embedding space between the candidate and target units. The *CNN* model is trained at the utterance level with $\mathcal{F}$ corpus and uses only linguistic information. One of the assets of this model is having contextual information of phonemes at the utterance level which could help a better representation in the embedding space.

4.3 Different Embeddings for Corpus Design and TTS (*MLP*)

The third method uses an embedding model specific to the target cost function using both linguistic and acoustic information. According to the proposition in [10], a feed-forward DNN is trained to predict the acoustic information at frame level for each input phoneme vector. The timing features are concatenated to embedding features in order to help prediction of the corresponding acoustic features. As in the previous system, the target cost function corresponds to the euclidean distance in the embedding space.

The learning data is the linguistic and acoustic information corresponding to phoneme/frame of the voice corpus $\mathcal{S}$. The timing features are the phoneme duration in seconds and the relative position of the corresponding frame inside the phoneme. The acoustic features consist of a 60 dimension Mel-Frequency Cepstral Coefficients (MFCC) vector, and the log of fundamental frequency F_0. The acoustic features are centered and reduced (unit variance). The frame length is 10 ms.

After training, the encoder part that transforms linguistic vector of phonemes into embedding space is detached and used as the embedding model.

4.4 Systems Differences

Table 2 summarizes and highlights the differences of the two embedding models described above and Fig. 3 displays the three approaches compared in this study.

Table 2. Embedding models comparison for both hybrid systems.

Method	CNN	MLP
Training data	Full corpus ($\mathcal{F}$)	Voice corpus ($\mathcal{S}$)
Input	Linguistic	Linguistic+Timing
Output	Linguistic	Acoustic
Training Level	Utterances (Sequence of phonemes)	Frames of signals

It is important to notice that the *MLP* model benefits from acoustics while the *CNN* model is only learnt with linguistic data. Also, both models learn, by

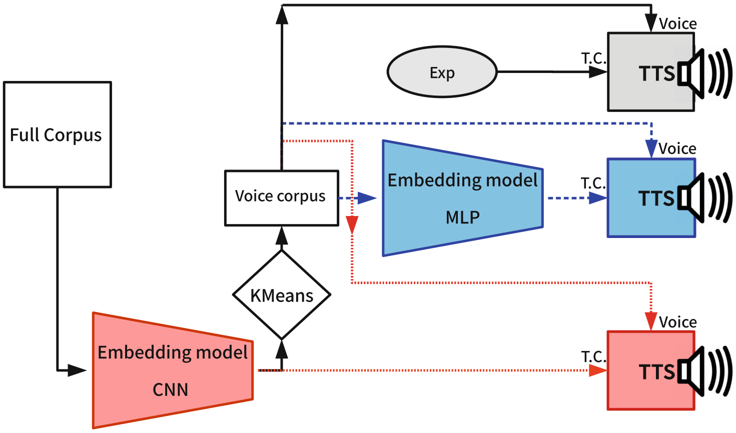

Fig. 3. TTS systems considered, namely Exp, MLP and CNN from top to bottom. The only difference come from the target cost (T.C.) computation.

construction, an embedding at the phoneme level, even if the *MLP* model is trained at the frame level (see [10]).

Besides, the *CNN* model is trained on the full corpus $\mathcal{F}$ and not only on the voice corpus $\mathcal{S}$ to maximize the quantity of data used for learning. The learning data is samples at the utterance level for the *CNN* model whereas the *MLP* one considers samples at the frame level. Hence, the *MLP* has much more data for training. It would not have been efficient to train the *CNN* model just with 300 samples from the $\mathcal{S}$ corpus.

Considering all this, we want to see if the consistency of embeddings between the corpus design step and the synthesis step helps to improve synthesis. However, the use of an acoustic model, with the *MLP* model, might not be completely fair. To be complete, further experiments are planned to try to inject acoustics in the corpus design step.

5 Experiments and Results

In the following subsections, we report the objective and perceptual evaluation results for the three methods.

5.1 Objective Evaluation

Since for the three methods, the target cost functions measure distances in three different (embedding or not) spaces, it is not possible to compare their outputs based on TTS costs. However, the same script is used as the test set and the *Concatenation rate* is then more appropriate to compare TTS performances. For each test utterance, this statistic is the number of concatenations in synthetic signal divided by the total number of possible concatenations. As for this measure, the lower is the better as it means more consecutive units from the same utterance. Less concatenation is assumed to result in higher quality. This measurement is computed for the test part ($\mathcal{T}$) and the rest of full corpus ($\mathcal{F} - \mathcal{S}$). It helps to find how methods can be generalized to other scripts than $\mathcal{F}$.

As shown in Table 3, the *CNN* method has better statistics than *Exp* method and *MLP* beats both for test part.

Table 3. Concatenation rate (%) results; confidence interval are calculated by using boot strap method with alpha = 0.05.

Measures/Methods	*Exp*	*CNN*	*MLP*
Rest of full corpus ($\mathcal{F} - \mathcal{S}$)	56.63 ± 0.16	**54.36 ± 0.16**	**54.34 ± 0.15**
Test part ($\mathcal{T}$)	56.64 ± 0.52	56.24 ± 0.51	**53.98 ± 0.50**

5.2 Perceptual Evaluation

In [10], the use of an acoustic model for the derivation of target cost has proved to be superior to an expert-based model. So two AB listening tests have been prepared to compare the synthetic quality of systems. The first one is between the *Exp* method and the *CNN* method and the other one is between the *CNN* and the *MLP* method. According to the protocol proposed for perceptual evaluation in [24], each AB test is composed of the 100 samples extracted from $\mathcal{T}$ with the highest DTW on MCep features. The samples are shorter than 7 s. The listeners have been asked to compare 40 pairs in terms of overall quality. The results are reported on Fig. 4.

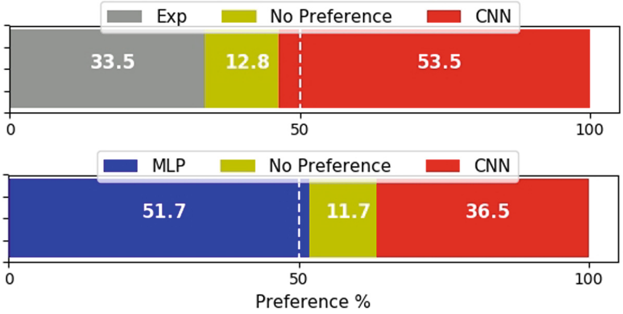

Fig. 4. Listening test results.

There are 14 listeners who have participated to the first test and 10 listeners as for the second test. Each pair of samples in the first test has been compared at least 5 times and in the second test at least 4 times. The result of the first test shows that the *CNN* based embedding as input of target cost can generate synthetic signals with significantly higher quality than the expert target cost. The second test indicates the preference of listeners for *MLP* model, which takes advantage of linguistic and acoustic information, rather than *CNN* model.

6 Conclusion

In this paper, we have investigated the relation between the corpus design process and a hybrid TTS. The TTS voice corpus has been selected based on an embedding model which uses the phonological information of the full corpus. This embedding model can be applied instead of the expert TTS cost or an acoustic model of phonemes. It has then be used to build a hybrid system by computing the target cost function as the euclidean distance between units in the embedding space.

In the first step, we have presented a phoneme embedding model which is basically the encoder part of a *CNN* auto-encoder. The transformation of utterances in the embedding space is followed by the KMeans algorithm to select a subset of full corpus in order to compose a voice corpus. Our preliminary experiment has shown that this method could achieve perceptually higher quality of synthetic signals than a voice designed by a classical set covering method.

The proposed *CNN* model has been applied to provide a phoneme embedding in hybrid TTS instead of an acoustic model (*MLP*) trained on the selected voice corpus. The perceptual test has shown that although the *CNN* model has better performances than expert-based target cost TTS, the *MLP* model has been preferred to the *CNN* model.

The *CNN* may be tuned or changed to improve performances. However, these results seem to emphasize the importance of acoustic information in any phone-embedding process for TTS tasks. The *CNN* model has been used for both corpus design and hybrid TTS, it is learnt on the full corpus, and takes into account more contextual information by the use of utterances as training samples (instead of frames). On the other side, the *MLP* model profits from acoustic information besides the linguistic one. Consequently, in future works, the use of an acoustic model as the embedding model for corpus reduction or corpus design should be investigated.

Acknowledgements. This study has been realized under the ANR (French National Research Agency) project SynPaFlex ANR-15-CE23-0015 and also funded by the Région Bretagne and the Conseil Départmental des Côtes d'armor.

References

1. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. ICASSP **1**, 373–376 (1996)
2. Zen, H., Tokuda, K., Black, A.: Statistical parametric speech synthesis. Speech Commun. **51**(11), 1039–1064 (2009)

3. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In: ICASSP, pp. 7962–7966 (2013)
4. King, S., Wihlborg, L., Guo, W.: The Blizzard Challenge 2017. In: Blizzard Challenge workshop (2017)
5. Fan, Y., Qian, Y., Xie, F., Soong, F.: TTS synthesis with bidirectional LSTM based recurrent neural networks, In: Interspeech, pp. 1964–1968 (2014)
6. King, S., Crumlish, J., Martin, A., Wihlborg, L.: The Blizzard Challenge 2018. In: Blizzard Challenge Workshop (2018)
7. Merritt, T., Clark, R., Wu, Z., Yamagishi, J., King, S.: Deep neural network-guided unit selection synthesis. In: ICASSP, pp. 5145–5149 (2016)
8. Wan, V., Agiomyrgiannakis, Y., Silen, H., Vit, J.: Google's next-generation real-time unit-selection synthesizer using sequence-to-sequence LSTM-based autoencoders. In: Interspeech, pp. 1143–1147 (2017)
9. Zhou, X., Ling, Z., Zhou, Z., Dai, L.: Learning and modeling unit embeddings for improving HMM-based unit selection speech synthesis. In: Interspeech, pp. 2509–2513 (2018)
10. Perquin, A., Lecorvé, G., Lolive, D., Amsaleg, L.: Phone-level embeddings for unit selection speech synthesis. In: Dutoit, T., Martín-Vide, C., Pironkov, G. (eds.) SLSP 2018. LNCS (LNAI), vol. 11171, pp. 21–31. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00810-9_3
11. Chevelu, J., Lolive, D.: Do not build your TTS training corpus randomly. In: EUSIPCO, pp. 350–354 (2015)
12. Szklanny, K., Koszuta, S.: Implementation and verification of speech database for unit selection speech synthesis. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1262–1267 (2017)
13. Nose, T., Arao, Y., Kobayashi, T., Sugiura, K., Shiga, Y., Ito, A.: Entropy-based sentence selection for speech synthesis using phonetic and prosodic contexts. In: Interspeech, pp. 3491–3495 (2015)
14. François, H., Boëffard, O.: Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem. In: Interspeech, pp. 829–832 (2001)
15. Cadic, D., D'Alessandro, C.: Towards optimal TTS corpora. In: LREC, pp. 99–104 (2010)
16. Isogai, M., Mizuno, H., Mano, K.: Recording script design for corpus-based TTS system based on coverage of various phonetic elements. In: ICASSP, pp. 301–304 (2005)
17. Barbot, N., Boëffard, O., Chevelu, J., Delhay, A.: Large linguistic corpus reduction with SCP algorithms. Computat. Linguist. **41**(3), 355–383 (2015)
18. Krul, A., Damnati, G., Yvon, F., Moudenc, T.: Corpus design based on the kullback-leibler divergence for text-to-speech synthesis application. In: ICSLP, pp. 2030–2033 (2006)
19. Krul, A., Damnati, G., Yvon, F., Boidin, C., Moudenc, T.: Approaches for adaptive database reduction for text-to-speech synthesis. In: Interspeech, pp. 2881–2884 (2007)
20. Cooper, E., Chang, A., Levitan, Y., Hirschberg, J.: Data selection and adaptation for naturalness in HMM-based speech synthesis. In: Interspeech, pp. 357–361 (2016)
21. Nose, T., Arao, Y., Kobayashi, T., Sugiura, K., Shiga, Y.: Sentence selection based on extended entropy using phonetic and prosodic contexts for statistical parametric speech synthesis. IEEE/ACM Trans. Audio, Speech, Lang. Process. **25**(5), 1107–1116 (2017)

22. Alain, P., Barbot, N., Chevelu, J., Lecorvé G., Simon, C., Tahon, M.: The IRISA text-to-speech system for the blizzard challenge 2017. In: Blizzard Challenge Workshop (2017)
23. Boeffard, O., Charonnat, L., Le Maguer, S., Lolive, D., Vidal, G.: Towards fully automatic annotation of audio books for TTS. In: LREC, pp. 975–980 (2012)
24. Chevelu, J., Lolive, D., Le Maguer, S., Guennec, D.: How to compare TTS systems: a new subjective evaluation methodology focused on differences. In: Interspeech (2015)
25. Lambert, T., Braunschweiler, N., Buchholz, S.: How (not) to select your voice corpus: random selection vs. phonologically balanced. In: SSW6, pp. 264–269 (2007)