*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# Corpus Design using Convolutional Auto-Encoder Embeddings for Audio-Book Synthesis

*Meysam Shamsi, Damien Lolive, Nelly Barbot, Jonathan Chevelu*

Univ Rennes, CNRS, IRISA

meysam.shamsi@irisa.fr, damien.lolive@irisa.fr, nelly.barbot@irisa.fr, jonathan.chevelu@irisa.fr

## Abstract

In this study, we propose an approach for script selection in order to design TTS speech corpora. A Deep Convolutional Neural Network (DCNN) is used to project linguistic information to an embedding space. The embedded representation of the corpus is then fed to a selection process to extract a subset of utterances which offers a good linguistic coverage while tending to limit the linguistic unit repetition. We present two selection processes: a clustering approach based on utterance distance and another method that tends to reach a target distribution of linguistic events. We compare the synthetic signal quality of the proposed methods to state of art methods objectively and subjectively. The subjective and objective measures confirm the performance of the proposed methods in order to design speech corpora with better synthetic speech quality. The perceptual test shows that our TTS global cost can be used as an alternative to synthetic overall quality.

**Index Terms**: corpus design, deep neural networks, embedding space, clustering, text-to-speech synthesis

## 1. Introduction

Text-to-speech synthesis is widely used in industry nowadays. Nevertheless, some applications still require improvements for further developments, like audiobooks generation.

In practice, the synthetic speech quality is strongly affected by the quality of the corpus used to build the voice. Previous studies [1, 2, 3] have shown that a random selection is not efficient to design such speech corpus. This is true especially for unit selection-based speech synthesis but also statistical parametric and hybrid ones. A random or unbalanced corpus contains lots of phonological unit repetitions and does not guarantee a sufficient variety of units for the speech synthesis process.

Moreover, the corpus should be as small as possible in order to minimize the human cost of high quality recording and labeling checking stages. In the case of unit selection and hybrid approaches, a reduced corpus may also accelerate the synthesis process considering the size of the search space. In that case, removing redundant elements while adding critical ones to the corpus is important. A well-designed corpus combines parsimony and balanced unit coverage in order to gain a satisfactory level of richness with a minimal cost construction.

The conventional way to produce an audiobook is to record a speaker who reads the entire book. This process is costly and time consuming. The aim of this study is to design an automatic method to select the best recording script from the book. The selection is done at the utterance level to help the speaker to have a well-adapted intonation. The recorded signals form a voice corpus on which is based a text-to-speech synthesis system to vocalize the complementary part of the book.

In this paper, the proposed approach relies on convolutional neural networks [4] in order to learn a non linear transformation from textual and linguistic data into a new pertinent representations without manual feature extraction/selection. The derived utterance embedding enables to guide and compare some selection algorithms to extract a set of utterances offering a large linguistic richness. The first algorithm is to cover all clusters of the embedding space stemming from a K-means algorithm. The second one is based on a greedy strategy to design an utterance subset close to a target linguistic distribution. These selection approaches are compared to a standard set covering one as the covering of all diphonemes using a greedy strategy [5, 3]. Objective and subjective results show that these proposed methods are more efficient than the standard one. Moreover, a crucial asset of these embedding-based approaches is that it is not necessary to select features, it adapts automatically to the book to be generated.

In Section 2, the state of the art in the corpus design and evaluation is briefly reviewed. Section 3 explains the corpus design procedure in three steps. In Section 4, the objective and subjective results are presented. Finally, Section 5 discusses the results and future works.

## 2. Previous works

Several works on automatic TTS corpus design have been carried out since early 2000s (for instance [6, 7, 8] for some preliminary ones). The covering of linguistic units under a parsimony constraint is the main idea of script corpus design. In particular, the case of full covering can be formalized as a set-covering problem [8]. Two axes have been mainly studied: which units should be better to cover, and which algorithmic approach is the most adequate to best produce a solution according to the chosen optimization criteria? For instance, the following unit types have been considered: the rarest categories of allophones in priority [8], "sandwich" units [9], a balance between triphone, syllable and morpheme elements [10]. [1] focused on maximum variability of unit features in the selected subset instead of defining a discrete unit set to be covered.

The most commonly used algorithmic strategy is the greedy one. In [5], the combination of agglomerative and spitting greedy phases have been assessed and compared to a Lagrangian relaxation based algorithm to derive full multi-represented coverings of diphoneme and triphonemes. The Lagrangian relaxation approach provides a lower bound showing that greedy strategies build solutions close to optimal ones.

Regardless of set covering, some studies [11, 12] investigated the distribution of units in corpus. [11] suggested to design TTS corpus which minimizes the Kullback-Leibler Divergence (KLD) between its diphoneme and triphoneme distribution and a prior distribution. They then focused on usage



http://dx.doi.org/10.21437/Interspeech.2019-2190

frequency during synthesis and distribution of units in reduced speech corpus [13]. They assumed that the units which are more used in the synthesis process are more helpful to build a corpus. Although, the performance of this method depends on the performance of unit selection in TTS.

Recently [14] has introduced an extended entropy of phonetic and prosodic context. They have tried to maximize this entropy in order to design corpora. They showed that contextual information should be taken into account for corpus design.

Increasing the number of features and samples leads to an exponential growth of the covering size if no feature selection is done. Instead of introducing expert knowledge to select the features, we propose to use a model for that task. Deep neural networks and particularly deep auto-encoders could be used to do so. [15] introduced an approach to build *Paragraph Vectors*, also called *Doc2Vec*. Their model maps variable length pieces of text to a fixed-length vector. This method was introduced to work with words as units. The idea of mapping utterances into an embedding space with a fix number of dimensions helps to calculate a distance between utterances. Using a metric and a numerical representation of each utterance based on its linguistic content enables to select a set of distant utterances, in order to offer a larger linguistic covering. [16] presented a sequence-to-sequence model based on long short-term memories. However their proposition was used for translation task, the LSTM hidden states can be used as the embedding vector for utterances when the model has been trained as an auto-encoder.

Although the context and application of these studies were different, in both approaches, the linguistic information of each *piece of text* is embedded in a fixed-length vector. In our case, we map sentences to an embedding space in order to sum up their content into a meaningful fixed-length vector. Then, we try to find a tiling of the embedding space that could improve the speech synthesis quality compared to standard approaches. To do so, we propose to use an auto-encoder approach implemented using a deep convolutional neural network to compress linguistic information with latent features.

# 3. Methodology

The main idea of this paper is to derive a vector representation of the linguistic information in order to facilitate the selection of a subset of sentences offering a good linguistic variety from a text corpus. The overall process is the following: (1) information extraction from the text corpus, (2) projection of feature vectors into an embedding space, (3) utterance selection and (4) objective and subjective evaluations of the selected subset.

## 3.1. Information extraction

We define a linguistic feature vector for each phoneme in the utterance, extracted from a text utterance, and providing information about the phoneme, e.g., its identity, preceding and following neighbours, its position in the syllable/word/utterance it belongs to, etc. The linguistic features are automatically extracted [17]. The linguistic vector, of size 296, contains categorical and numerical features. 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 an 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 an utterance is then represented by the sequence of linguistic feature vectors associated to the phonemes that compose it.

## 3.2. Embedding space

From this initial representation of the linguistic content at phoneme and utterance level, using an embedding space enables to derive a continuous and compressed representation. Importantly, this approach avoids the injection of expert knowledge to drive the selection of the most important features, letting the model reveal what is of interest. To build up this embedding space, an auto-encoder based on a multi-layer Convolution Neural Network (CNN) has been implemented, as shown on Figure 1. To avoid overfitting, we use a dropout layer, with a 0.1 drop probability, after each layer in the encoder [18]. CNN layers are used with kernel size of 5 and the $\tanh$ activation function. The loss function is the Mean Squared Error (MSE).

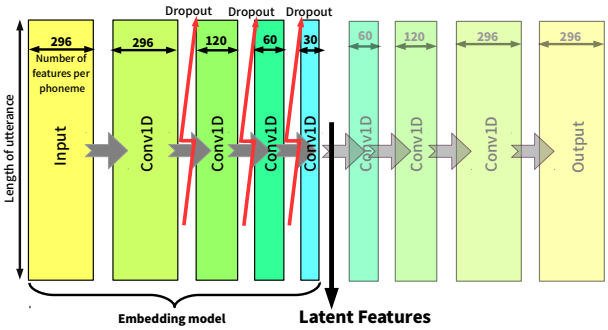


Figure 1: *CNN auto-encoder architecture.*

In order to train the CNN auto-encoder, three kinds of training sets have been tested:

1. Utterance (***Utt***): the training set is composed of utterances with variable length.
2. Breath Group (***BG***): the training set is composed of shorter samples, at the breath group level, in order to increase its size.
3. Sliding Window (***SlidWin***): a sliding window of size 100 phonemes is used to delimit the size of samples, which is about the average length of utterances. In order to get the next sample, the window is slided for 10 phonemes. The length of sequences in training phase is different from the length of sequences in prediction phase. While fixed-length sequences are used to train the network, it predicts variable-length sequences corresponding to utterances.

After training, the network is used to generate, for each input sequence of linguistic vectors at utterance level, a sequence of unit vectors in embedding space. Its is equal to the number of phonemes in the input utterance.

## 3.3. Utterance selection

The main idea behind utterance selection is to extract a set of utterances from a book that offers a representative linguistic coverage of its content while tending to limit the linguistic unit repetitions. In our case, the term unit stands for phonemes in context, based on the linguistic features used. Concretely, the main goal is to provide a large variety of options to the TTS system while minimizing the size of the corpus. We propose two methods for selecting utterances: the first one is based on a

clustering approach and the second tends to reach a target distribution of linguistic events.

#### *3.3.1. Clustering*

The clustering methods can group sequences based on the similarity of their attributes. By selecting one sequence per cluster, we assume that it represents the information of other sequences in the same cluster. In particular, one can consider that the most representative sequence is the closest one to the cluster center.

In order to compute the similarity measure between utterances with different lengths, we have built a numerical and fixed dimensional representation of utterances. Let us consider an utterance $u$ composed of $m$ phonemes. The $i^{\text{th}}$ phoneme of $u$ is represented by the embedded vector $p_i = (x_1^i, \ldots, x_N^i)$, where $N$ corresponds to the embedding dimension. Several aggregation operators could be used to take into account the contributions of phonemes in $u$, like the sum, and we have selected the average to avoid the utterance length-dependency: $u$ is then represented by $\hat{u} = (f_1, \ldots, f_N)$ where $f_j = \sum_{i=1}^{m} x_j^i / m$.

The clustering of the whole text corpus is made based on the K-Means algorithm using the Euclidean distance between utterances (vectors $\hat{u}$) as the similarity measure. As mentioned above, from each cluster, the closest utterance to the cluster center is selected. The length $l_i$ of this selected subset $i$ is given by the sum of the length of its utterances (number of phoneme instances). The reduction rate of this subset $\tau_i$ is defined by the ratio between $l_i$ and the length of the whole corpus $l_\Omega$. To achieve a target reduction rate $\tau^*$ of the whole corpus, the number of clusters $K^*$ is iteratively achieved: initially $K_0$ is set as $\lfloor \tau^* \times (\text{number of utterances in } \mathcal{F}) \rfloor$; resulting from step $i$, a selected subset is derived using $K_i$ clusters and $K_{i+1}$ is set to the $\lfloor K_i \times \tau^* \times l_\Omega / l_i \rfloor$.

#### *3.3.2. KLD minimization*

A greedy strategy to minimize the Kullback-Leibler Divergence (KLD) in the context of corpus design has been proposed in [11]. Although this method was based on the phonological unit distribution in the corpus and a target distribution, the idea can be transposed to continuous values in embedding space. In our case, the target distribution can be a general distribution of the whole book or particularly the features distribution of the utterances which are supposed to be synthesized.

Precisely, for each dimension of the embedded phoneme vectors, values are normalized to the range $[0, 1]$. An histogram $h$ is then computed for each dimension by binning the values into ten bins ($X = \{[0, 0.1), \ldots, [0.9, 1]\}$). Therefore, using histogram $h(f_j)$, we define the probability $P^j$ for each latent feature $f_j$. The KLD between two sets of utterances can be computed for each latent feature distribution as follows:

$$KLD(P_s^j \mid\mid P_t^j) \;=\; -\sum_{x \in X} P_s^j(x) \log \left( \frac{P_s^j(x)}{P_t^j(x)} \right),$$

where $P_s^j$ is the probability distribution of $f_j$ in the selected set of utterances and $P_t^j$ is the probability distribution of $f_j$ feature in the target set of utterances.

To achieve a target reduction rate, at each iteration, a greedy process selects the utterance which minimizes the average of KLDs (one KLD per feature) between the target distribution and the distribution computed from the new set of utterances, including the candidate utterance.

### 3.4. Evaluation

The reduced corpus which is resulted from the selection process should be evaluated. An ideal evaluation method would be to ask listeners to score the synthetic utterances in terms of quality, at each step of the selection. To alleviate the cost of subjective evaluation, we use some objective measures to evaluate the quality of synthetic utterances. We use the IRISA TTS system [19] for synthesis and its TTS costs to measure the quality of the synthetic signal. The concatenation, target, and global costs are investigated. The global cost is a linear combination of the concatenation and target costs. For each of them, the average value over the test set is calculated as well as the maximum value and the average of three maximum values of each utterance, to compare the achievements of the different systems. Using the two last measures is complementary since a global average value may hide local artefacts. Finally, we compare perceptually the best configurations with an AB test.

## 4. Experiments and results

### 4.1. Experimental setup

The initial corpus contains 3,339 utterances of a French expressive audio-book spoken by a male speaker. The overall length of the speech corpus is 10h44. More information on the annotation process can be found in [20]. The audio-book has been divided into two parts. A test set which is randomly selected as a continuous part with 334 utterances (10% of the whole corpus). The rest of the audio book is named the full corpus and is denoted $\mathcal{F}$ in the remainder. $\mathcal{F}$ is composed of 3,005 utterances and 362,126 phoneme instances. The objective is to extract from $\mathcal{F}$ a subset of a given size. The natural signal samples of this subset will be used to synthesise the utterances of the test set with maximum quality. To derive the embedded representation of utterances of $\mathcal{F}$, 90% of $\mathcal{F}$ are used for training the CNN models and 10% are used as a validation set to avoid overfitting. Table 1 shows the number of samples and their average length (number of phoneme instances) which are used for training with the different sample types.

Table 1: *Number and average length of samples*

| Sample type | Sample number | Avg length (in ph.) |
|---|---|---|
| Utt | 3,005 | 120.5 |
| BG | 10,287 | 35.2 |
| SlidWin | 36,203 | 100 |

In order to compare the performance of the selection methods and evaluate the impact of the selection size on the synthesised speech quality, several reduction sizes of $\mathcal{F}$ have been tested: 50%, 40%, 30%, 20%, and 10%.

The selection methods under comparison are the following:

- ***Random***: the baseline system is a random selection of utterances. To have representative results, each size has been built 10 times randomly and the average TTS costs of them are considered.
- ***SC***: this system is based on a set covering problem which is solved by a greedy strategy [5]. The best utterances are selected to cover $\eta$ times each linguistic feature. Starting from 1, $\eta$ is incremented until the target reduction rate is reached.

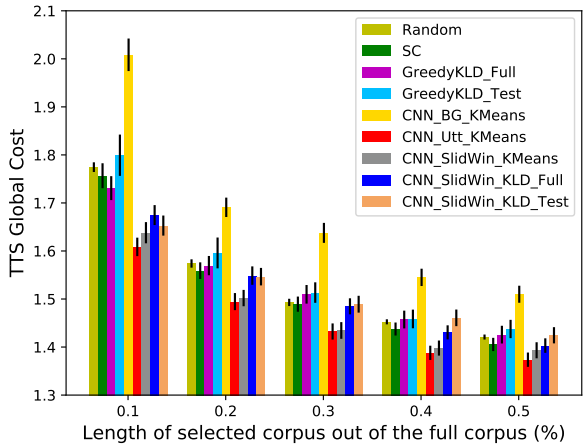


Figure 2: *TTS global cost*

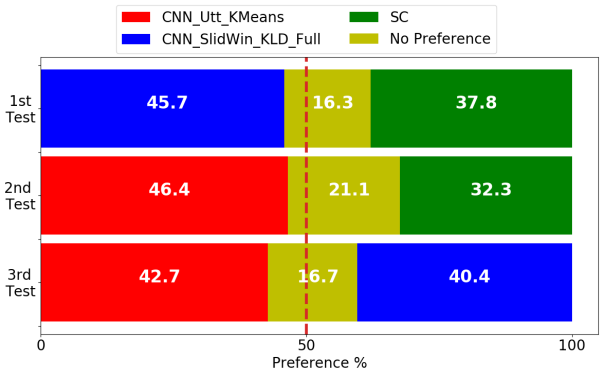


Figure 3: *Listening test results*

- ***GreedyKLD_Full/Test***: an agglomerative greedy algorithm is used to minimize the KLD between the diphoneme distribution of the selected corpus and the full (or test) corpus as done in [11].
- ***CNN_BG/Utt/SlidWin_KMeans***: as detailed in Sections 3.2 and 3.3.1, the selection strategy is KMeans algorithm which clusters the embedding space. This embedding is derived by a CNN auto-encoder trained on ***Utt***, ***BG*** or ***SlidWin*** samples.
- ***CNN_SlidWin_KLD_Full/Test***: they are some variants of ***GreedyKLD_Full/Test***. The considered distributions are those associated to the embedded vectorial representation as explained in Section 3.3.2.

### 4.2. Objective evaluation

We trained the CNN model with different embedding sizes ($N = 240, 120, 60, 30, 15$). It turns out that the embedding size $N = 30$ and ***SlidWin*** gives the best reconstruction error of CNN models: RMS is 0.00021 for ***SlidWin*** instead of 0.00067 for ***Utt***. But according to the TTS global cost, ***SlidWin_KLD*** and ***Utt_KMeans*** achieve lower costs.

For each selection method and different selection sizes, we synthesize the test set which enables us to gather a concatenation cost and a target cost for each utterance. Figure 2 displays the obtained TTS global costs. The other objective measures are not shown here since they indicate the same differences for the different methods/sizes.

Figure 2 shows that the CNN model using utterance as samples and K-Means as the selection method (***CNN_Utt_KMeans***) achieves the best performance for all the reduction rates. We have also compared the proposed CNN embedding to the Doc2Vec and LSTM models, as explained in section 2. Due to the lack of space detailed results are not reported here but overall, the proposed approach gives better results in this context. At last, considering the phonological richness of the selected subsets, the proposed methods diphone coverage is lower than the one of ***SC*** but significantly higher than the one of ***Random***.

### 4.3. Subjective evaluation

Based on objective measures, three methods have been chosen to be compared perceptually: ***SC***, ***CNN_Utt_KMeans*** and ***CNN_SlidWin_KLD_Full***. The utterances of the test section have been synthetically vocalized using 10% of $\mathcal{F}$ selected by each of these methods. Three AB preference tests have been conducted to compare the following pairs of systems:

1. ***CNN_SlidWin_KLD_Full*** and ***SC***, 19 listeners
2. ***CNN_Utt_KMeans*** and ***SC***, 17 listeners
3. ***CNN_Utt_KMeans*** and ***CNN_SlidWin_KLD_Full***, 13 listeners

Each test is composed of the 100 samples with the highest DTW on MCep features from test set [21]. The samples are shorter than 7 seconds. The listeners were asked to compare 30 pairs in terms of overall quality. Results are reported on Figure 3: taking into account all the answers, the averages given in this figure are based on 570 evaluations for the first test, respectively 510 and 393 for the second and third ones.

We can observe that listeners prefer the quality of synthetic signal from ***CNN_Utt_KMeans*** and ***CNN-SlidWin_KLD_Full*** rather than ***SC*** method. This means that the objective measures provide a good estimation of perceptual quality. Moreover, listeners does not have any significant preference between ***CNN_Utt_KMeans*** and ***CNN_SlidWin_KLD_Full***.

## 5. Conclusion

In this work, we have presented an end-to-end method for sentence selection. We have shown that a CNN auto-encoder can be used successfully to extract linguistic information in TTS corpus design. The K-Means clustering and the KLD methods work properly using embedded representations achieving better results than random, or even than the best methods in state of the art such as set covering. The subjective evaluation has confirmed this result showing a preference for the proposed approaches. Finally, it seems that the TTS global cost could be used as an alternative measure of overall quality.

As for future works, firstly, the embedded phoneme representations should be used for hybrid TTS as well. Secondly, in these experiments the model corresponds to a linguistic auto-encoder and it could be beneficial to use a general encoder-decoder from linguistic information to acoustic information for corpus design. Thirdly, in order to generate a whole audio-book as a mix of natural and synthetic signal, the selection method should take into account the utterances which are not selected as the target of synthesizing. First results obtained here tend to show that trying to minimize KLD between selected utterances and test set achieves better results than minimizing KLD with a global distribution.

# 6. References


[1] B. Bozkurt, O. Ozturk, and T. Dutoit, "Text design for TTS speech corpus building using a modified greedy selection," in 8th European Conference on Speech Communication and Technology, 2003, pp. 277–280.

[2] M. Isogai and H. Mizuno, "Speech database reduction method for corpus-based TTS system," in 11th Annual Conference of the International Speech Communication Association, 2010, pp. 158–161.

[3] J. Chevelu and D. Lolive, "Do not build your TTS training corpus randomly," in Signal Processing Conference (EUSIPCO), 2015 23rd European. IEEE, 2015, pp. 350–354.

[4] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series," The handbook of brain theory and neural networks, 1995.

[5] N. Barbot, O. Boëffard, J. Chevelu, and A. Delhay, "Large linguistic corpus reduction with SCP algorithms," Computational Linguistics, vol. 41, no. 3, pp. 355–383, 2015.

[6] J. Van Santen and A. Buchsbaum, "Methods for optimal text selection," in Proc. of the European Conference on Speech Communication and Technology (Eurospeech), 1997, pp. 553–556.

[7] H. Kawai, S. Yamamoto, N. Higuchi, and T. Shimizu, "A design method of speech corpus for text-to-speech synthesis taking account of prosody," in Proc. of the International Conference on Spoken Language Processing (ICSLP), vol. 3, 2000, pp. 420–425.

[8] H. François and O. Boëffard, "Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem." in INTERSPEECH, 2001, pp. 829–832.

[9] D. Cadic and C. d'Alessandro, "Towards optimal TTS corpora," in Proc. of the International Conference on Language Resources and Evaluation (LREC), 2010, pp. 99–104.

[10] M. Isogai, H. Mizuno, and K. Mano, "Recording script design for corpus-based TTS system based on coverage of various phonetic elements," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1. IEEE, 2005, pp. 301–304.

[11] A. Krul, G. Damnati, F. Yvon, and T. Moudenc, "Corpus design based on the kullback-leibler divergence for text-to-speech synthesis application," in 9th International Conference on Spoken Language Processing (ICSLP), 2006, pp. 2030–2033.

[12] Y. Shinohara, "A submodular optimization approach to sentence set selection," in International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE. IEEE, 2014, pp. 4112–4115.

[13] A. Krul, G. Damnati, F. Yvon, C. Boidin, and T. Moudenc, "Approaches for adaptive database reduction for text-to-speech synthesis." in INTERSPEECH, 2007, pp. 2881–2884.

[14] T. Nose, Y. Arao, T. Kobayashi, K. Sugiura, and Y. Shiga, "Sentence selection based on extended entropy using phonetic and prosodic contexts for statistical parametric speech synthesis," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 5, pp. 1107–1116, 2017.

[15] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in International Conference on Machine Learning, 2014, pp. 1188–1196.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, 2014, pp. 3104–3112.

[17] A. Perquin, G. Lecorvé, D. Lolive, and L. Amsaleg, "Phone-level embeddings for unit selection speech synthesis," in International Conference on Statistical Language and Speech Processing, LNCS/LNAI, P. G. Dutoit T., Martín-Vide C., Ed., vol. 11171. Springer, Cham, 2018, pp. 21–31.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] P. Alain, N. Barbot, J. Chevelu, G. Lecorvé, C. Simon, and M. Tahon, "The IRISA text-to-speech system for the blizzard challenge 2017," in Blizzard Challenge 2017 workshop, 2017.

[20] O. Boeffard, L. Charonnat, S. Le Maguer, D. Lolive, and G. Vidal, "Towards fully automatic annotation of audio books for tts." in LREC, 2012, pp. 975–980.

[21] J. Chevelu, D. Lolive, S. Le Maguer, and D. Guennec, "How to compare TTS systems: a new subjective evaluation methodology focused on differences," in Interspeech, 2015.