HMM Training Strategy for Incremental Speech Synthesis

Maël Pouget \(^1\), \(^2\), Thomas Hueber \(^1\), \(^2\), Gérard Bailly \(^1\), \(^2\), Timo Baumann \(^3\)

\(^1\) CNRS/GIPSA-Lab, Grenoble, France
\(^2\) Univ. Grenoble Alpes/GIPSA-Lab, Grenoble, France
\(^3\) Universität Hamburg, Informatics Department, Natural Language Systems, Hamburg, Germany

\(^1\), \(^2\) firstname.lastname@gipsa-lab.fr, \(^3\) baumann@informatik.uni-hamburg.de

Abstract

Incremental speech synthesis aims at delivering the synthetic voice while the sentence is still being typed. One of the main challenges is the online estimation of the target prosody from a partial knowledge of the sentence’s syntactic structure. In the context of HMM-based speech synthesis, this typically results in missing segmental and suprasegmental features, which describe the linguistic context of each phoneme. This study describes a voice training procedure which integrates explicitly a potential uncertainty on some contextual features. The proposed technique is compared to a baseline approach (previously published), which consists in substituting a missing contextual feature by a default value calculated on the training set. Both techniques were implemented in a HMM-based Text-To-Speech system for French, and compared using objective and perceptual measurements. Experimental results show that the proposed strategy outperforms the baseline technique for this language.

Index Terms: HMM-based speech synthesis, incremental, TTS, HTS, prosody

1. Introduction

Incremental Text-To-Speech (iTTS) systems aim at starting delivery of the synthetic voice before the full sentence context becomes available, e.g. while a user is still typing the text to vocalize. Contrary to a conventional TTS, the synthesis follows the text input, words after words (potentially with a delay of one word). This ‘synthesis-while-typing’ approach is illustrated in Figure 1. By reducing the latency between text input and speech output, iTTS should enhance the interactivity of communication. In particular, it should improve the user experience of people with communication disorders who use a TTS system in their daily life, as a substitute voice. Besides, iTTS could be chained with incremental speech recognition systems, in order to design highly responsive speech-to-speech conversion systems (for application in automatic translation, silent speech interface, real-time enhancement of pathological voice, etc.).

To our best knowledge, the concept of incremental speech synthesis was initially formulated in [1] in the context of dialogue systems. However, in the proposed proof-of-concept, the speech generation was delivered incrementally but was generated in a non-incremental way. In [2], Baumann & Schlangen proposed the first complete software architecture dedicated to incremental speech processing (including recognition, dialogue management and TTS modules). Another proof-of-concept based on the reactive HMM-based parameter generation system MAGE [3] was described in [4].

One of the main remaining challenges of iTTS systems is the online estimation of the target prosody from an incomplete sentence (and therefore an uncertain - incrementally unveiled - syntactic structure). In conventional TTS, target prosody is typically calculated from long-range contextual features [5], [6], extracted from the text by morphological and syntactic analyzers. Considering a current segment as reference (typically a phoneme), some of these features refer to its left context (i.e. the ‘past’); some of them refer to its right context (i.e. the ‘future’). These features can, for instance, be the part-of-speech tag (POS) of the next word, or the number of remaining words before the end of the current sentence. Indeed, such features related to the right context are usually not available in incremental processing. Therefore, strategies should be developed to deal with these ‘missing’ features and predict acceptable prosody from an ‘incomplete’ sentence. This is the general scope of the present study.

In [7], [8], Baumann first evaluated the impact of potentially missing features on the quality of the estimated prosody, in the context of HMM-based speech synthesis, for English and German languages. Then, the author proposed a strategy for predicting a ‘default’ value for these missing features (this strategy is therefore referred to as the ‘Default’ strategy here). This strategy exploits the decision trees that are classically used in HMM-based speech synthesis in the state clustering procedure. The goal of this present study is twofold. First, we evaluate this strategy [7] (briefly recalled in Section 2) for French language, which has different prosodic characteristics than English and German (for instance, French can be considered to have no lexical stress[9]). Second, we propose another approach for dealing with missing contextual features, also in the context of HMM-based speech synthesis (Section 3).

Contrary to [7], our approach does not aim at recovering the missing features at synthesis time. It rather consists in integrating a potential uncertainty on some features when building the synthetic voice (i.e. when training the HMM set). This strategy is here referred to as the ‘Joker’ strategy. The two strategies were implemented in an HMM-based TTS for French.
language, and compared using objective measurements and a perceptual listening test (Section 4).

![Diagram](image)

Figure 2. Procedure for recovering full context labels from incomplete labels and clustering tree exploration when building unseen labels with the default strategy.

2. Baseline strategy: ‘Default’

In most implementations of HMM-based TTS (such as [10] or [11]), each speech unit is a phone in context. The context is described by a set of segmental features such as the identity of the current and adjacent phonemes, and suprasegmental features, such as the POS of the current and adjacent words, the position of the word in the current breath group, etc. Since it is very difficult to build a training dataset covering all possible contexts, clustering techniques are used to group some HMM states and share their model parameters (similarly to ASR contexts, clustering techniques are used to group some HMM states and share their model parameters).

The most widely used technique is tree-based clustering [12]. Each node of the tree is associated with a context-related question, such as ‘R-SYLL-NB-PHON==3’ (“Are there 3 phonemes in the next syllable?”). The pertinence of the context-related questions and the structure of the tree are learnt automatically from the training dataset with respect to a specific criterion, such as the Minimum Description Length [13]. At synthesis time, the decision tree is extensively used when building the so-called ‘unseen models’ (i.e. corresponding to contexts with no acoustic observations in the training set).

In [7], Baumann proposed to exploit these decision trees to recover the missing contextual features at synthesis time. The procedure, which is illustrated in Figure 2, can be summarized as follows. First, a set of full-context HMMs (i.e. HMMs modeling each phoneme with information about its left and right contexts) is trained using a standard procedure (including tree-based clustering). Then, a ‘default answer’ is assigned to each contextual feature related to the right context (which might be unknown in incremental processing). This ‘default answer’ is calculated from the training set, by averaging the answers observed at each node of the decision tree associated with a question on the right context. For numerical features (e.g. the number of words before the end of the sentence), the default answer is the mean value calculated across the training dataset; for symbolic features (e.g. the POS tag of the next word), the default answer is the most common value observed in the training dataset.

This strategy gives encouraging results and works with pre-existing voices that were not trained with the incremental use-case in mind, but presents a major disadvantage. By imposing a default value to some contextual features, some branches of the clustering trees become totally unexplored when building the ‘unseen models’ (as shown by the red dashed zone in Figure 2).

Therefore, only a limited number of HMM states are used at synthesis time. In other words, the fine-grained modeling of the training dataset based on a rich set of contextual features is not exploited here. In the next section, we present another strategy to deal with missing contextual features in the context of HMM-based incremental synthesis.

3. Proposed strategy: ‘Joker’

In the proposed approach, the potential uncertainty on right-contextual features is handled during voice training rather than during the synthesis process, as in the ‘Default’ strategy. The proposed technique aims at considering a contextual feature that could potentially be missing as ‘relevant’ information that can be explicitly used when describing the linguistic context. Besides, when clustering the pool of HMM-states, we evaluate the need of tying model parameters among all contexts potentially sharing the same missing features. This training procedure results in a set of context-dependent HMMs that are likely to be slightly less accurate than full-context models (for instance, they are expected to deliver a neutral intonation for situations where the right context would trigger very different patterns). However, the ‘Joker’ strategy may lead to better perceptual results since there is no risk it uses an incorrect full-context model, as in the ‘Default’ strategy.

The ‘Joker’ has been implemented in the HTS framework as follows. First, the training corpus is labeled by introducing a so-called ‘Joker’ value (specified by the # character) to each contextual feature which cannot be determined when processing the text incrementally. This notably affects all the contextual features requiring information about the next word. As an example, let us consider the label associated with the phoneme in the last syllable of the current word, and the right-contextual feature ‘number of phonemes in the next syllable’ (usually denoted by the symbol ‘C’ in HTS). Since the value of this feature is unknown when processing the text incrementally, the ‘Joker’ tag is inserted in the label such as: ‘…”+p+…/C:#…”’ (other contextual features are omitted for clarity). Then, a set of context-dependent HMMs is trained using a standard procedure (similarly to a non-incremental system). The Joker tag (#) is simply considered as a possible value for some contextual features.

A tree-based clustering procedure is then applied to deal with data sparsity. However, contrary to a non-incremental system, we introduce questions about the possible known/unknown characteristic of each contextual feature. In the HTS format, this can be written as: “QS ‘R_nb_phone_in_next_syllable_is_unknown’ ‘/*C:#’” where C stands for the number of phonemes in the next syllable. At the end of the clustering procedure, the parameters of some models/states sharing a common missing feature are expected to be tied together. The rest of the training procedure, as well as the synthesis procedure are similar to a non-incremental system.
As shown in Figure 3, one of the main advantages of the ‘Joker’ strategy compared to the ‘Default’ strategy, is a better utilization of the decision tree when building the unseen models. Even if a question related to a missing contextual feature in used at a specific node of the tree, the label can continue to ‘go down’ to all sub-branches of this node. Therefore, in this case, all HMM states are reachable. This should result in more variety in the estimated trajectories, compared to the ‘Default’ strategy, as shown by the experimental results described in the next sections.

Figure 3. Usage of the decision tree for building unseen models using the ‘Joker’ strategy.

4. Experimental protocol and results

4.1. HMM-based TTS system for French language

The two strategies described in the previous sections were evaluated in the context of an HMM-based TTS system, developed in our group for French language. The specificities of this system are the following. The audio material used for training was extracted from an audiobook of the novel “Le tour du Monde en 80 jours” by Jules Verne (this corpus was also used in [14]). This corpus contains 3h17mn of speech data, after silence being removed. Phonetization as well as morphological features considered in this study are listed below (the features using the linguistic front-end COMPOST [15]. The contextual and syntactic analyses of the text transcriptions were achieved without the HNM-model of the spectrum (section 4.1) using the SPTK (5 emitting states for the acoustic models) were trained on this corpus using the HTS toolkit [10] and a standard procedure. Global variance optimization was not used in this study.

An initial segmentation of the audio recordings at phonetic level was obtained using a forced-alignment procedure and was then post-processed manually. In our system, the full-band spectral envelope is parameterized using a “Harmonic plus Noise Model” (HNM) [16] (and not mel-cepstral coefficients as in [17]), following the implementation detailed in [18] (p.82). Each acoustic observation (extracted each 5 ms) is a 93-dimensional vector composed of the fundamental frequency $f_0$, a (12th-order) LSF-modeling of the harmonic component of the spectral envelope (defined for voiced frames only), and a (16th-order) LSF-modeling of the residual spectrum, completed by first and second derivatives. A set of context-dependent HMMs (5 emitting states for the acoustic models) were trained on this corpus using the HTS toolkit [10] and a standard procedure. Global variance optimization was not used in this study.

Similarly to [7], we calculated the percentage of use of each right-contextual questions for clustering the training set, for spectrum-related parameters were related to the quinphone context. However, for pitch and duration, more questions related to current and next word were used for French than for German.

4.2. Objective evaluation

The two strategies considered in this study were first evaluated using a set of objective measurements. A subset of 165 test sentences was randomly selected from the corpus (and removed from the training set). These sentences were first synthesized using a non-incremental approach. The resulting acoustic feature vectors (i.e. spectrum, $f_0$ and duration) were considered as the ‘best possible result’. In other words, we assume that non-incremental processing will systematically lead to lower performance than non-incremental processing (a similar assumption was made in [7]). The 165 test sentences were then synthesized using both ‘Default’, and ‘Joker’ strategies. The accuracy of the estimated spectrum was evaluated by calculating a mel-cepstral distortion [19] in dB defined such as:

$$MCD(y_T, y^{NI}_T) = \frac{1}{F \cdot \ln(10)} \sum_{f=1}^{F} \sum_{d=0}^{D} 2\Sigma_{d=0}^{D} (y_T^d - y^{NI}_T)^2$$

where $y_T^d$ and $y^{NI}_T$ are respectively vectors of $D+1$ mel-cepstral coefficients estimated using the $S$ incremental strategy and the baseline non-incremental ($NI$) approach and $F$ the number of frames in the utterance. These coefficients were derived from the HNM-model of the spectrum (section 4.1) using the SPTK.
For each metric (approach, the ‘Default’ approach, and the proposed ‘Joker’ sentence synthesized respectively with the non-incremental "naturalness”. These sound samples correspond to the same was asked to sort 3 sound samples, according to its listening test, similar to [25] and [26]. For each trial, the subject was presented in Table 1.

Table 1: Objective differences between ‘Default’ and ‘Joker’ for spectrum, f0 and phone duration, averaged across the test set (± standard deviation).

<table>
<thead>
<tr>
<th>Strategy</th>
<th>MCD (dB)</th>
<th>Eₙ (cents)</th>
<th>Eₐₙ (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.78 ± 0.26</td>
<td>197.4 ± 88.7</td>
<td>0.20 ± 0.06</td>
</tr>
<tr>
<td>Joker</td>
<td>0.94 ± 0.15</td>
<td>178.2 ± 78.4</td>
<td>0.17 ± 0.04</td>
</tr>
<tr>
<td>Joker vs. Default</td>
<td>***</td>
<td>NS</td>
<td>***</td>
</tr>
</tbody>
</table>

As expected, the best-ranked samples were those obtained with the non-incremental approach (i.e. with a complete set of contextual features). The proposed ‘Joker’ strategy outperforms significantly the baseline (‘default’) strategy (2.8 vs. 1.5, p<0.005). This supports the benefit of considering explicitly the uncertainties about right context when building the synthetic voice. Interestingly, no statistically significant difference was observed between the non-incremental and the ‘Joker’ strategy. Amongst other possible causes, this could be explained by a ‘ceiling effect’, due to the intrinsic quality of the baseline HMM-based synthesis (with a mean score of 3).

5. Conclusion and perspectives

This study describes a strategy for dealing with missing contextual features, for incremental HMM-based speech synthesis. This strategy consists in integrating a potential uncertainty on some contextual features when training the HMM set. This approach was compared to the baseline technique proposed in [7]. Both strategies were implemented in an HMM-based TTS system for French. A perceptual test shows that the proposed strategy outperforms the baseline technique for that language. Future work will focus on the evaluation of the proposed strategy for other languages, such as English and German (which were considered in [7]). In order to build a complete incremental TTS system, we will also combine the proposed technique with and ‘incremental text parsing’ front-end. Such module could be inspired by some approaches developed for incremental text processing and syntactic parsing [27].
6. References


