Variable-length acoustic units inference for text-to-speech synthesis

Olivier Boëffard

IRISA – Institut de Recherche en Informatique et Systèmes Aléatoires, Université de Rennes 1, ENSSAT Lannion, France
boeffard@enssat.fr

Abstract

The best voices in text-to-speech synthesis are currently obtained via acoustic units concatenation-based systems. In such systems, the choice of units whose concatenations will produce an acoustic message is a crucial stage. Moreover, it can be observed that current TTS systems use acoustic units which most often correspond to variable-length phonetic descriptions. In this article, an original framework is proposed which allows the automatic determination of an optimum set of variable-length acoustic units.

1. Introduction

The most effective systems of text-to-speech synthesis, with regard to both voice quality and algorithmic complexity, are based on a technique of acoustic units concatenation. It is indeed possible to define for any given language a minimum set of acoustic units which permit the reproduction of any sound message in that language. These units, as well as the contexts which are most likely to allow their recording, are defined during a stage of phonetics analysis. Historically, the first systems of text-to-speech synthesis proceeded from the concatenation of diphones (sequences of two half-phones); these were followed by systems integrating longer units: triphones, polyphones or longer units. A new generation of synthesis systems goes even further in accepting several acoustic or prosodic variants for a given synthesis unit [1] [2].

This leads to the following quandary: although long units are very interesting if they exhibit an acoustic continuum, their number increases exponentially with their length. In other words, given a finite storage capacity and a limited access time, which acoustic units are to be kept as the most interesting? The usual way of solving that problem is to keep only the most frequent units. The major drawback of this approach is that it does not relate explicitly to the unit length and its a priori probability. A long unit can prove more interesting than a shorter and more frequent one; indeed, it makes possible to avoid some concatenations. We offer a solution which is both theoretical and experimental to the question of search for variable-length acoustic units, which we formulate as a problem of statistical inference. The formal framework is that of multigrams already proposed in the context of language models, phonetic transcription and speech recognition [3] [4].

This work is part of a more general framework where the search for acoustic units for speech synthesis is considered as a minimal set covering problem. The main objective consists in taking into account the likelihood of a sentence given a set of phonetic units during the estimation stage of this covering.

2. Hypotheses

Our objective is to automatically determine a list of acoustic units from a corpus of phonetized sentences. An acoustic unit is here characterized by a sequence of phonemes which will be referred to as a phonetic unit.

The search for a set of variable-length acoustic units therefore consists in selecting a subset of relevant sequences from a set of possible combinations. The universe of observed phonemes combinations corresponds to a corpus of 311,572 textual sentences that were subjected to automatic phonetization.

Our methodological framework is based on the following hypotheses:

- We consider an observed phonetic sequence as being made of the juxtaposition of phonetic units which must be discovered. These phonetic units are the linguistic labels of the acoustic units which must be recorded.
  - This set of phonetic units is not a priori known. It will be determined through the analysis of observable sequences forming a learning database.
  - Each phonetized sentence corresponds to a random sequence of symbols. These symbols are the language phonemes.
  - A phonetic unit is a sequence of n phonemes. We consider the hypothesis that a n-length phonetic unit captures a statistical correlation on an horizon of n consecutive symbols. This hypothesis replaced in a probability framework corresponds to the definition of an n-gram.
  - The set of phonetic units to be kept is made of variable-length units. However the definition of an n-gram model makes the assumption of a constant n. It is nevertheless possible to define a set of polygrams or multigrams as a heterogeneous n-gram set by making n vary.

3. Multigram theoretical framework

The theoretical framework used here has been already proposed by [4]. It is applied here to a context of speech synthesis. We postulate that:
- A phonetic sequence is made of a concatenation of phonetic units, which can be variable-length phonetic units.
- The process describing the concatenation of phonetic units is a hidden random process.
- The random variables constituting that random process are independent and identically distributed.

3.1. Notations

Let \( \mathcal{O} \) be the set of all the observed sequences belonging to a learning database. A particular observation \( O^k \in \mathcal{O} \) is considered as a discrete time random process.
3.2. Model

Let \( \mathcal{Z} \) be any set, the joint probability of the observed and hidden processes is as follows:

\[
P(O^i, Z^i | \mathcal{Z}) = \prod_{i=1}^{[\mathcal{Z}]} P(O_i | Z_i) P(O^i_{\mathcal{Z}_i} | Z_i) \tag{1}
\]

with \( i(Z_i) \) are non-overlapping intervals linking an element \( Z_i = z_i \) with a sub-sequence of the observation \( O_i \).

We consider that \( P(O^i_{\mathcal{Z}_i} | Z_i) \) follows a Dirac's distribution which amounts to 1 if the sequences of symbols taken from \( \mathcal{A} \) are identical for \( O^i_{\mathcal{Z}_i} \) and \( z_i \), and which is worth 0 otherwise.

The parameters of this model correspond to the probabilities of the a priori observation of the elements \( z_i \) of \( Z' \). Let \( \Theta \) be the set of possible parameters for this model, \( \Theta \subset \mathbb{R}^{Z'} \), \( \theta \in \Theta \) corresponds to a particular distribution of the elements of \( Z' \).

The point is then to find an optimal distribution \( \theta^* \), characterizing the set \( Z^* \), which maximizes the following marginal probability:

\[
\theta^* = \arg \max_{\theta} P(O | \theta) \tag{2}
\]

We suppose that all the observations are statistically independent:

\[
\theta^* = \arg \max_{\theta} \prod_{i=1}^{[\mathcal{Z}]} P(O_i | \theta) = \arg \max_{\theta} \prod_{i=1}^{[\mathcal{Z}]} \sum_{m} P(O^i_{\mathcal{Z}_i}, Z^i | \theta) \tag{3}
\]

Where \( Z_{O_i} \) is the set of all the possible segmentations of the sequence \( O^i \).

The optimization is carried out via an iterative Expectation-Maximization, EM, technique. An iteration is composed of two stages: in the first one, the inference of a distribution on the model parameters is calculated; the second stage determines a set of parameters which maximizes the likelihood of the observations.

### 3.3. Estimation of parameters

The algorithmic solution presented is based on the following heuristic: from a first set \( Z^0 \), we form iteratively a sequence of sets \( Z^p \). The set \( Z^{p+1} \) is to be initialized by the set \( Z^p \). This strategy guarantees a decrease of the entropy for each \( Z^p \) until the sought-for optimal set \( Z^p \) is reached. For each set \( Z^p \), the distribution \( P_{Z^p}(z_i) \) which maximizes the likelihood of the observations is inferred.

#### 3.3.1. Algorithm

Based on the previous remarks, here are the steps of the algorithm:

- \( Z^0 = n \)-multigram initial set with \( 1 \leq n \leq N \)
- \( \text{repeat} \)
  - \( \text{repeat} \)
    - inference step for \( P_{Z^p}(z_i | O, \theta^p) \)
    - maximization step for \( \theta^p \)
  - until EM has converged
- \( Z^{p+1} = \text{reduction}(Z^p) \)
- until \( Z^p \) is reached

#### 3.3.2. Initialization of parameters

The initial \( Z^0 \) set is formed by keeping all the \( n \) phonemes sequences present in the learning database. \( n \) goes from 1 to \( N \). In order to limit combinatorial problems – space complexity in proportional to \( |\mathcal{A}|^n \) – only phonetic units present in the learning database are kept.

#### 3.3.3. EM optimization

The main problem with the theoretical framework adopted in equation 1 lies in the asynchronism of states sequences between the observed and the hidden processes. A marginalization such as that described by equation 3 would imply the calculation of all the sequences \( Z^p \) representing an observation \( O^i \). This calculation has an exponential complexity depending on the length \( T |k| \) of the observed sequence.

The only algorithmic solution is to express that inference problem through a forward-backward-type form. This technique is widely used to infer hidden Markov random processes. However, standard recurrent formulas found in literature are expressed for a hidden process which is synchronous with the observed process.

Let \( t \) be the temporal indice of variable \( O_t \) taken from the observed process \( O^i \). We define \( \alpha(t) \) as the probability to observe the sequence \( (O_1, \ldots, O_t) \). There is therefore a segmentation of \( O^i \) at the reference indice \( t \) whatever the \( z_i \) units are.

Let \( \beta(t) \) be the probability to observe the \((O_{t+1}, \ldots, O_{T})\) sequence and a segmentation at time \( t \). Applying a Lagrangian relaxation technique leads to the following estimation of \( \theta^p \), a component of vector \( \theta \):

\[
\theta_{t+1}^p = \frac{1}{|O| \times T(k)} \sum_{k} \sum_{t} \alpha(t - |z_i|) \times \theta_t^p \times \beta(t) \tag{4}
\]

We point out that \( \theta_{t}^p = P(z_i) \) at iteration \( p \).

#### 3.3.4. \( Z^p \) reduction

The point is now to define the set \( Z^{p+1} \) from a set \( Z^p \). The distribution of elements of set \( Z^p \) has just been optimized during an EM stage.
\(Z^{p+1}\) is defined by taking from \(Z^p\) all the units \(z_i\) whose probability \(P(z_i)\) is greater than a given threshold. This threshold is determined by the minimal probability on the initial set \(Z^p\); in other words, it is determined from the distribution preceding the EM stage. In that way, units which have been excluded from the composition of observation sequences are eliminated.

The convergence towards the optimal set \(Z^p\) is not easy to control. Indeed, on the learning database, the entropy of sets \(Z^p\) decreases until no units remain to be eliminated between two iterations. This monotone decrease is no longer valid for the whole test. Therefore we are left two possibilities:

- Either set a heuristic that will make it possible to stop before reaching an empty set of units to eliminate - for instance stop on a ratio between the number of units to eliminate and the number of sets units \(Z^p\).
- Either use a validation corpus on top of the learning and test corpora. This validation corpus makes it possible to determine an experimental inflexion point for the entropy of sets \(Z^p\).

4. Experimentation methodology

The experimentation we have carried out was designed to characterize the sets of \(n\)-multigrams obtained as well as to observe a convergence of the learning mechanism presented in paragraph 3.3.

The concept of perplexity allowed us to assess the performance of a set of \(n\)-multigram. This concept is linked to entropy. Indeed, let us postulate a large enough number of observed sequences and an ergodic source process. The entropy calculated through the model is denoted \(H = \frac{1}{|Q|} \sum_k P(Q) \log_2 P(Q)\). This entropy is always greater or equal to the actual entropy — assuming we know the actual set \(Z^p\). Therefore, the lower the entropy, the better the model. The perplexity is expressed through \(PP = 2^H\) [5]. As for the entropy, the lower the perplexity, the better the language modelling.

4.1. Learning and test databases

The corpus of sentences we have selected includes varied texts extracted mainly from dialogues. We have a total of 311,572 sentences phonetized via an automatic system. Sentences are divided into breath groups defined on major syntactic boundaries.

The learning database is a subset of sentences taken randomly within the complete corpus. These sentences are used for the learning of \(n\)-multigram models. The test database is complementary to the learning database in that it makes it possible to check the model behaviour. We have set a ratio of 80\% between the size of the learning database and the complete corpus size.

4.2. Initialization of sets

The initialization of a \(n\)-multigram set is made by keeping from the complete corpus all the phonetic units whose length vary from 1 to \(n\). The more the length of the phonetics units increases, the more the frequency of the unit decreases (exponential-type decreasing shape).

In order to avoid too much memory storage during the optimization process, we have set a minimum threshold of occurrences for a phonetic unit to be kept. This threshold is set at 5 occurrences regardless of unit length. It corresponds to relative frequencies of about \(10^{-5}\).

5. Results

5.1. Convergence

The convergence of the loglikelihood is relatively quick. The convergence threshold was fixed at \(10^{-4}\). We observed that for \(p > 0\) the EM algorithm records very few iterations (less than 3).

A complementary experiment on a limited number of units, 83, and a limited number of learning sentences, 5, allowed us to observe that EM does not reach a global, but a local minimum. The protocol of this experiment consists in determining a sampling of parameters space in the neighbourhood of optimal point \(\theta^*\), here with the constraint \(\sum \theta_k = 1\). It is a random sampling which is made through a Gibbs sampler on a Dirichlet distribution. The surface of the loglikelihood in the neighbourhood of the optimal point is very chaotic; nevertheless the optimal point is situated under that surface. We can expect a better convergence results if we use for instance stochastic variants of the EM algorithm [6].

5.2. Perplexity

Diagram 1 is a synthetic presentation of perplexity measures on learning and test databases for different sets of \(n\)-multigrams with \(n\) varying from 1 to 10. Let us remind that a set of 10-multigrams contains a mix of \(p\)-grams with \(p\) varying from 1 up to 10.

![Diagram 1](image)

**FIG. 1 –** Perplexity on the learning and test databases for different \(n\)-multigram sets, \(n\) varying from 1 up to and including 10.

We can observe that perplexity value decreases according to \(n\). The perplexity value calculated on the test database are only slightly superior to those calculated on the learning database.

5.3. Cardinality of \(n\)-multigram sets

Diagram 2 shows the evolution of the cardinality of the different \(n\)-multigram sets studied. On the x-axis we find the index \(n\) which characterizes the set, and on the y-axis the number of distinct \(p\)-grams. We can observe an asymptotic behaviour which differs between the initial \(n\)-multigram set and the optimized set. The number of elements is thus reduced by a factor two for the 10-multigram set.
5.4. Analysis of a set

Diagram 3 show the p-gram composition of the 6-multigram set after the optimization process. The x-axis corresponds to the fraction of the set on which p-grams have been calculated. The set was arranged according to $P(z_i)$: the unit is the percentage. The y-axis, on a log scale, represents the number of p-grams parameterized by 6 curves. The curves respectively represent from the origin point the number of 1/2/3/4/5/6-gram for different proportions of the complete set. We can observe that the optimization helps to keep more balanced proportions than those of the initial set.

Diagram 4 shows the perplexity value for the 6-multigram set. The x-axis enumerates the different proportions on the set used to calculate the perplexity in two configurations: the initial and the optimized set. The perplexity measured with the overall initial set, 9.95, is the same as the one measured on the optimized set with a 10% ratio. The amount of units is composed of 322,219 for the former and 19,145 for the latter.

6. Conclusion

In this article we have shown that the multigram framework can be applied successfully to find linguistic units in the text-to-speech synthesis field. From a huge amount of phonetized sentences, an optimal set of variable-length units can be inferred. These units embed the main statistical regularities observed in the learning database. Some additional informal experiments allow to conjecture that the linguistic structure of the discovered units is closely related to the syllabic structure of the language. Indeed, applying a Viterbi algorithm to decode the best segmentation of sentences into units, we have found that the segmentation is mostly syllabic. A next step on this research will be to connect this work with those concerning the design of reference databases allowing the best covering of acoustic units [8].

7. References


