Approaches for adaptive database reduction for Text-To-Speech synthesis

Aleksandra Krul 1,2, Géraldine Damnati 1, François Yvon 2, Cédric Boidin 1, Thierry Moudenc 1

1 France Télécom R&D Division, TECH/SSTP
2 avenue Pierre Marzin, 22307 Lannion Cedex, France
{aleksandra.krul,geraldine.damnati,cedric.boidin,thierry.moudenc}@orange-ftgroup.com

2 GET/ENST and CNRS/LTCI
46, rue Barrault, 75624 Paris Cedex 13, France
yvon@enst.fr

Abstract
This paper raises the issue of speech database reduction adapted to a specific domain for Text-To-Speech (TTS) synthesis application. We evaluate several methods: a database pruning technique based on the statistical behaviour of the unit selection algorithm and a database adaptation method based on the Kullback-Leibler divergence. The aim of the former is to eliminate the least selected units during the synthesis of a domain specific training corpus. The aim of the later approach is to build a reduced database whose unit distribution approximates a given target distribution. We evaluate these methods on several objective measures.

Index Terms: text-to-speech synthesis, corpus reduction, database adaptation, KL divergence

1. Introduction
Current Text-To-Speech systems are based on concatenative methods [1]. Such systems use a large database of pre-recorded speech from which acoustic units are selected for concatenation. The scalability of the database is an important issue in unit selection based speech synthesis. Indeed, the use of the full database is not always suitable or even possible for some applications. The database has to be reduced so that the speech synthesis system can be integrated onto different devices.

Two approaches are commonly used for database reduction. In a “bottom-up” approach the database is examined in order to remove the redundant units. For instance, in [2] units are clustered according to some similarity measures concerning prosodic and phonetic contexts. Only units that are representative of each cluster are kept in the reduced database. The “top-down” approach is based on the investigation of the output of the synthesizer. One of the implementations consists in synthesizing a large amount of data and removing units which are not frequently used by the synthesizer. This approach is based on the statistical behaviour of unit selection algorithm and was originally proposed in [3]. The advantage of such a method is that no knowledge about speech units is needed. It is however closely dependent on the unit selection algorithm behavior.

However, the reduced synthesis systems are often used for specific applications such as in a mobile phone dedicated to partially-sighted or unsighted persons. The reduced database has to be adapted to the domain specific application.

In this paper we are interested in this particular paradigm. Our goal is to prune the generic database and to adapt it in order to synthesize a domain specific application corpus in different devices that do not support a large amount of data. We investigate two approaches: a variant of a “top-down” reduction method and a novel reduction method guided by the Kullback-Leibler measure.

The first reduction method that we use is a “top-down” approach. Instead of synthesizing a generic corpus, we propose to use a domain specific corpus that reflects the application for which the reduction has to be performed. We will show that even if the specific corpus is not very large we obtain better objective results than if we collect statistics by synthesizing a much larger generic corpus.

The second approach that we investigate is based on the Kullback-Leibler divergence and was introduced in [4]. This method was used for designing a textual corpus for speech synthesis applications. The main idea of this method is that the distribution of units in the constructed corpus should be close to an a priori defined distribution. In [4], the flexibility of this method is emphasized: the algorithm is able to accommodate different distributions which may prove more adequate for domain specific TTS synthesis applications.

In this paper, we use this method to construct a reduced database, whose unit distribution is close to the domain specific distribution: the distribution of words in the target domain differs from that in the original corpus and this has a bearing on the respective unit distributions. It may thus prove useful to preserve more representatives of those units that occur frequently in the target domain, and to reduce the number of representatives of those units that occur with a small frequency. This suggests to sample units in the reduced database in proportion to their frequency in the target corpus, to preserve the variability in the sampled sub-corpus. The distribution of units in the reduced database can be adapted to any domain.

This paper is organized as follows. In section 2, we present several approaches for adaptive database reduction. In section 3, we objectively evaluate all of the methods and present experimental results.

2. Overview of Reduction Methods
2.1. Database pruning based on the statistical behaviour of unit selection algorithm
The main idea of this pruning method is to keep the units that are the most often used to synthesize a representative corpus while the least selected units are pruned. Our system uses diphone as elementary unit. Each diphone (about 1200 in French) is present...
several times (from 1 to thousands) in the acoustic database: each acoustic realization is called a diphone variant or a unit. When synthesizing a message, each variant may or may not be selected. The number of times it is selected is called number of occurrences.

The first step consists in synthesizing a representative corpus and in counting the number of occurrences of each variant. Then the pruning step is performed independently for each diphone. All diphone variants are sorted according to their number of occurrences. The ones with the highest number of occurrences are kept while the ones with the lowest number of occurrences are pruned. The number of variants to be kept is calculated in order to reach a target coverage or a target reduction rate. This method is referred hereafter as Ps.

2.2. Method based on Kullback-Leibler divergence

The KL divergence [5] is a measure which assesses the similarity between two probability distributions. It is defined as:

$$D(P \parallel Q) = \sum_{i} p_i \log \frac{p_i}{q_i}$$

where $P$ and $Q$ are two discrete probability distributions.

The properties of this measure are the following. The divergence is positive or equal to zero. The two probability distributions are identical if and only if the KL divergence is null.

In the presented method all the speech database sentences are split into phrases also called breath groups. The KL based reduction method takes two steps. First we select phrases whose unit distribution approximates a target distribution from the corpus that was previously recorded for the database. Then we reduce the ordered phrases according to different reduction rates. The phrases are selected incrementally with a greedy algorithm. At a given iteration the unit distribution on the corpus that would be obtained by adding a candidate phrase is evaluated. The phrase for which this distribution results in the lowest KL divergence to the target is picked. The score of each candidate phrase is:

$$D(P \parallel Q) = \sum_{i, n_i \neq 0} n_i \left( \log \frac{n_i}{N} - \log q_i \right),$$

where $Q$ denotes the target distribution and $P$ is the constructed distribution. $n_i$ is the number of occurrences of a diphone $i$ in constructed corpus, and $N$ is the total number of units ($N = \sum n_i$).

In [4], we presented in details the behaviour of this algorithm. We also showed how to efficiently update, in an incremental manner, the Kullback Leibler divergence at each step of the algorithm.

The target distribution is estimated on a training corpus which is representative of a specific domain. The adaptation of the selected corpus to various distributions is easy to implement: what is only required is to obtain $Q$ from a given domain specific corpus and to set it as the target distribution in our algorithm.

We consider the diphone and triphone distributions. However, to ensure the full coverage of elementary units (diphones) we have to include the following constraint. Among the phrases that contain new diphones the algorithm selects the phrase that minimizes the KL divergence to the target diphone ($KLD_{dip}$ method) or triphone distribution ($KLD_{trp}$ method). This constraint insures that we will have at least one instance of each diphone in the reduced database. However, this method selects only units that occur in the target corpus. To tackle this problem, we use an $\epsilon$ smoothing unigram technique. A fixed value $\epsilon$ is allocated to units that are not present in the target corpus. The smoothing formulas are as follows:

$$q_i = \begin{cases} f_i \cdot [1 - \epsilon \cdot C_0] & \text{if } c(d_i) \neq 0 \\ \epsilon & \text{otherwise} \end{cases}$$

where $c(d_i)$ is the count of the diphone $i$, $f_i$ is the relative frequency of the unit $i$ and $C_0$ is the number of unseen units in the estimation corpus.

The selected phrases are iteratively included in the reduced database, until the desired reduction rate is attained.

2.3. Random method

This approach consists in randomly ordering phrases of the textual corpus. In order to ensure the diphone coverage in the reduced databases the same process was used as for the KL based method reduction. In the first step the random selection is made only among the phrases that contains new distinct diphones. When the full diphone coverage is achieved the selection process becomes completely random. We will refer to this method as "random."

3. Experimental results

3.1. Data

For our experiments we used a large database of a French speaker. The database contains about 7,000 sentences which correspond to 12,500 phrases and 252K units. In order to collect statistics on the use of diphone variants by the system and to estimate the distribution of diphones and triphones in the domain specific corpus we used a domain specific corpus $C_{ads}$. It contains 8,866 sentences. It is collected from the small ads from the real estate domain. $C_{ads}$ is split into a training corpus and a test corpus. The training corpus is used to perform the synthesis and to estimate the diphone and triphone distributions. Table 1 presents some corpora description.

<table>
<thead>
<tr>
<th></th>
<th>number of sentences</th>
<th>number of phrases</th>
<th>number of diphone types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADS TRAIN SET</td>
<td>6685</td>
<td>25039</td>
<td>1014</td>
</tr>
<tr>
<td>ADS TEST SET</td>
<td>1746</td>
<td>6067</td>
<td>955</td>
</tr>
</tbody>
</table>

3.2. Collecting statistics for the Ps database reduction

Two sets of statistics were collected for the Ps method.

Firstly, a corpus that contains about 359K newspaper text files was used. It corresponds to about 97M diphone occurrences. The selected variants represent 96% of the database. This method is referred hereafter as general Ps.

Secondly, we tried to run this reduction method on a smaller domain representative corpus. The $C_{ads}$ training corpus was synthesized. It corresponds to 403K diphone occurrences. Due to the limited size of the domain and the corpus, only 16% of the variants present in the generic database are selected at least once when synthesizing the corpus. This means that for reduction rates lower than 84% there is necessarily a random part in the algorithm in order to choose among the unused units.
Among these unused units it was therefore decided to keep the first variants of the database, i.e. in order they are stored in the database. It has to be noted that some unused units may be useful for the synthesis of the $C_{uds}$ test corpus. It is important to be able to target reduction rates smaller than 84%, even though there is a part of random in the reduction process. This method is hereafter referred as \textit{domain Ps}.

3.3. Objective evaluation

To evaluate the reduced databases we compare the aforementioned methods. The reduced databases are created by removing 10%, 20% .. 90% of the units.

We consider four objective measures which are given by the unit selection algorithm: the average length of selected segments, the average concatenation cost, the average target cost and the average cost. The target cost estimates how close a database unit is to the desired unit. The concatenation cost estimates how well two adjacent-ly selected units join together. The overall cost is a sum of the concatenation and the target costs. The selection algorithm minimizes the overall cost in order to find the optimal unit sequence. The average segment length measures the average number of units in the segment, i.e. a string of adjacentarily selected units. For instance, an average segment length equal to 1.0 means that none of the selected units are adjacent in the database. This measure is reverse proportional to the number of concatenations.

In [6], it has been shown that the average segment length and the average concatenation cost are highly correlated with MOS (Mean Opinion Score) tests. This measures are shown in the figure 1 and figure 2.

3.3.1. Average segment length

We investigate the average segment length in figure 1. The \textit{domain Ps}, $KL_{trip}$ and $KL_{dip}$ obtain significantly longer segments than the \textit{general Ps} and \textit{random} methods. At first sight, one may think that the fact the KL methods select segments is due to the fact that they keep entire phrases in the reduction process. However, the \textit{random} methods that also select entire phrases have poor average segment length, even worse than \textit{general Ps} method whose reduced databases are discontiguous. Therefore the adaptation to the specific context of the reduced database seems to be important to select adjacent units; the three adaptive reduction methods are equivalent for this criterion.

3.3.2. Average concatenation cost

We can look then to the average concatenation cost. For each synthesized sentence the average concatenation cost is the sum of all concatenation costs normalized by the total number of units in the sentence. The average concatenation cost that is shown in figure 2 is the average of the average concatenation costs of each sentence. In this figure as well as in the following ones a cost of 2 means that the cost is twice as high as the initial cost obtained on the whole database (0% reduction).

The lowest average concatenation cost, i.e. the best, is obtained with the \textit{domain Ps} method. Then $KL_{trip}$ is better than the $KL_{dip}$ method which is better than \textit{general Ps} method. The random methods are significantly worse than all other methods. The order is the same as for the average segment length but the three methods \textit{domain Ps}, $KL_{dip}$ and $KL_{trip}$ obtain distinct scores. The KL based methods obtain higher costs than the \textit{domain Ps} probably because they consider only basic units without taking into account concatenation cost criteria, i.e. acoustic features. We notice that $KL_{trip}$ have better concatenation cost than $KL_{dip}$, it seems to be correlated to the small difference between their average segment length.

3.3.3. Average target cost

The average target cost is computed in the same manner as the average concatenation cost. It is shown on figure 3. The best average target cost is obtained for \textit{domain Ps}. The second best average target cost is obtained for \textit{general Ps}, the KL based methods $KL_{dip}$ and $KL_{trip}$ seem to be equivalent with a higher cost than the two statistically based methods. As we consider only simple distribution of basic units the KL based methods do not use enough information about the units that are selected. To improve the KL based method, it may prove necessary to consider not only the phonetic nature of the units, but also features which characterize the units: length, stress, syntactic, lexical and phonetic context, etc. We are currently experimenting with such “contextualized” diphone sets.

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{figure1.png}
\caption{Average segment length.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{figure2.png}
\caption{Average concatenation cost, relative to the 0\% reduction concatenation cost.}
\end{figure}
than the average concatenation cost gain. We finally examine the overall cost in the figure 4 which is the sum of the target and concatenation costs. The best method is the KL based method. 

3.3.4. Average cost

We finally examine the overall cost in the figure 4 which is the sum of the target and concatenation costs. The best method is logically domain Ps, the second best is KLtrip, KL dip and general Ps are close to each other. This shows that the average target cost loss of KLtrip compared to general Ps is smaller than the average concatenation cost gain.

4. Discussion

In this study, we have investigated several reduction methods. The first observation is that the adaptive reduction methods overcome standard reduction general Ps and random reduction methods.

Even on the small domain specific corpus the domain Ps method seems to give the best results. This may be surprising as only 16% of the units have been used in the training corpus synthesis. Moreover, it has to be noted that some of the units that are used for test corpus synthesis were selected arbitrarily during the reduction process. That can be seen from figure 4 as the cost decreases between 80% and 0% reduction rate while only unused units are added arbitrarily to the reduced database.

The KL based method is almost equivalent to domain Ps for the average segment length. This result is promising as we have targeted distribution estimated on the basic types of units, i.e. diphones and triphones. No information about any other feature (acoustic, prosodic, linguistic features) was introduced whereas it is implicitly taken into account in the cost functions that guide the domain Ps. This raises the issue of which are the relevant features to describe the target distribution of a specific domain.

These methods might be combined in order to improve the reduction process, taking benefit from the close link of the domainPs method to the cost function and from the possibility to globally control the unit distribution on a variety of features by the KL based method.

5. Conclusions

In this study, we have presented approaches for the adaptive database reduction. We have adapted a classical reduction approach and we have proposed a method based on the Kullback-Leibler divergence. We have objectively evaluated the presented methods. Adaptive database pruning methods are promising reduction methods. It seems to be more suitable to use those techniques when the application for which the reduction has to be made is known.

The advantage of the presented reduction methods is that the reduced database can be adapted to any domain. For the statistically based approach it is simply a matter of collecting new statistics in the use of the database for a domain specific corpus. For the KL divergence based method, what is only required is to obtain Q from a given domain specific corpus and to set it as the target distribution in our algorithm.

Our plan include exploring the benefits this method on other domain specific corpora and examining other unit types, such as context dependent units, units enriched with prosodic features.

Complementary analyses also remain to be carried out in order to fully understand the pros and cons of these database reduction methodologies: for instance, we need to look closely at those sentences that are strongly affected by the database reduction, i.e. that yield high concatenation or target costs. Eventually, a complete subjective evaluation of the speech synthesis quality will have to be performed.

6. References