Reducing the Corpus-based TTS Signal Degradation Due to Speaker's Word Pronunciations

Sérgio Paulo, Luís C. Oliveira

L^2F Spoken Language Systems Lab.
INESC-ID/IST
Rua Alves Redol 9, 1000-029 Lisbon, Portugal
{spaulo,lco}@l2f.inesc-id.pt

Abstract

The goal of producing a corpus-based synthesizer with the owner's voice can only be achieved if the system can handle recordings with less than ideal characteristics. One of the limitations is that a normal speaker does not always pronounce a word exactly as predicted by the language rules. In this work we compare two methods for handling variations on word pronunciation for corpus-based speech synthesizers. Both approaches rely on a speech corpus aligned with a phone-level segmentation tool that allows alternative word pronunciations. The first approach performs an alignment between the observed pronunciation and the canonical form used in the system's lexicon, allowing the mapping of the time labels from the observed phones into the canonical form. At synthesis time the unit selection is performed on the phone sequence predicted by the system. In the second approach, no modification is performed on the phone sequence generated by the segmentation tool. This way, at synthesis time, the words are converted into phones by using the speaker's word pronunciation, rather than the system's lexicon. Finally, both approaches are compared by evaluating the naturalness of the signals generated by each approach.

1. Introduction

The flourishing number of spoken language repositories has pushed speech research in multiple ways. Much of the best speech recognition systems rely on models created with very large speech databases. These repositories have allowed the development of high-quality concatenative speech synthesis systems ([1], [2], [3]) producing naturally sounding speech by using speaker's own voice with almost no modifications. However, some problems can arise from the speaker-specific acoustic realizations of the text prompts, because, unless the speaker performs exactly as the linguistic modules of the TTS predict, mismatches will always occur. Such mismatches can prevent the unit selection algorithm from retrieving the largest possible speech chunk from the corpus. The quality of the synthesized speech is highly dependent on the number of chunks required to synthesize a sentence: we would like to minimize the number of joins associated with audible discontinuities caused by the concatenation of segments coming from different contexts. Two approaches can be followed to reduce the concatenation artifacts of the synthesized speech. The first one consists of having a good concatenation cost estimate, including a good spectral mismatch metric, that can penalize the concatenation of segments producing audible discontinuities[4], [5], [6] or [7]. The second approach is to minimize the number of joins by allowing the selection of a small number of large chunks. This work is focused on the second approach, by trying to allow the use of longer speech segments even when there are mismatches between the realization of the texts predicted by the TTS and produced by the speaker. The mismatches can occur at the multiple linguist levels of the utterance. As the unit selection can eventually use information from every level, each level should be adapted to the speaker's way of speaking. This adaptation has been performed mainly on the phonetic level, by modelling the speaker-specific word pronunciation. Two different strategies to handle the mismatches arisen from badly predicted speaker's pronunciation will be presented and evaluated.

2. The speech synthesizer

Our synthesizer was developed under the scope of the project *Interactive Home of the Future*[8]. A spoken dialogue system was developed to allow the visitors of this museum installation to interact with some of the available devices and services through an audio-visual interface. This system represents the concept of a virtual butler, i.e. someone that is always available to execute the users requests. A visual interface based on a realistic animated face, the butler's face, is used to make the users communicate with the system in a more natural way. The butler's face is shown in Fig. 1. The speech synthesizer uses the unit selection technique in the Festival framework[9] and converts the butler's dialogue into an audio stream and a set of temporized phones. These phones are used to create the facial animation by concatenating the corresponding visemes through time. The audio stream is played back in synchronization with the facial animation.

2.1. Corpus creation

The corpus is composed mainly of butler's domain-specific recordings. The corpus also incorporates out-of-domain sentences so that a reasonable phonetic coverage was ensured.

Figure 1: *Butler's face when he is waiting for a request.*

3. Utterance representation

In the Festival system, the utterance's linguistic representation is stored in the corpus as *Heterogeneous Relation Graphs*[10]. Such graphs can be obtained either by using the synthesis system to perform the text analysis or using a set of speech signal annotations. Large chunks can be retrieved by the unit selection when many consecutive phones to be synthesized are held in the corpus. In order to produce a higher quality speech signal, we must reduce the number of chunks retrieved by the unit selection algorithm, unless we have a distance metric that can predict audible discontinuities accurately. The reduction of the number of chunks can be achieved by maximizing the matching between the phonetic sequences held in the corpus and the phone series to be synthesized by the TTS. When the speaker's word pronunciation is not in agreement with the TTS lexicon, several speech chunks must be concatenated to produce a word that actually is recorded. Moreover, the inaccurate prediction of the speaker's major phrase breaks within a text, can also cause the system to use several chunks for generating a recorded sentence. Now, we will present our proposals for dealing with the word pronunciation mismatches.

3.1. Lexicon pronunciation

The first approach that we tested in our system, consists in the projection of the phonetic sequences produced by the speaker onto the phonetic sequences generated by the lexicon of the synthesizer. This projection is performed by aligning the two phonetic sequences. Once the alignment is known, it is used for mapping the segment labels from the speaker's phonetic sequence to the lexicon-based phone series, as can be seen in Fig. 2. That figure shows the alignment between the canonical and speaker phonetic sequences in the case of the Portuguese word *"ministro"* ("minister"). There is a mismatch between the two phone series, a phone that was predicted by the lexicon was not actually uttered by the speaker, phone @[1]. In order not to break the word phonetic sequence, this phone is inserted within the graph with a zero duration. This way, when the system synthesize the word "ministro", its pronunciation lexicon can be used without generating a sequence break due to the phone "@", since despite its zero duration, the sequence is in the corpus. Thus, the phonetic sequences of the corpus utterances are always transformed into the canonical form, so that the words are marked in the corpus according to the lexicon. While this approach ensures, in some extent, that the domain specific sentences are produced by using large chunks, some care must be taken to prevent the system from placing a concatenation point at the boundaries of the phones that were not observed (e.g. "@", in Fig. 2). In our system, we deal with this problem by setting the huge concatenation cost at these phones.

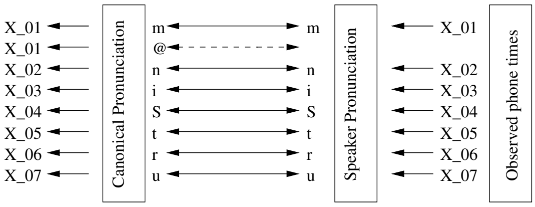

Figure 2: *Transformation of the observed phonetic sequence into the canonical form.*

3.2. Speaker-specific pronunciation

An alternative way to reduce the concatenation mismatches is by adapting the synthesizer's word pronunciation to the speaker-specific word pronunciation. Unfortunately, a word pronunciation can depend strongly on its neighboring words, due to cross-word co-articulations phenomena.

3.2.1. The building of the context-dependent lexicon

Since, in most cases the neighboring words interact at their boundaries only, we define the word context based one the neighboring phones. Fig. 3 shows the steps that must be taken to set each word context. We start by using the synthesizer's lexicon to get the canonical pronunciation for each word, so that a set of features can be assigned. The feature set includes:

- PreviousVowel, a binary feature to indicate whether the last phone of the previous word is a vowel ($value = 1$) or not ($value = 0$);;
- NextVowel, a binary feature to indicate whether the first phone of the next word is a vowel ($value = 1$) or not ($value = 0$);
- PrevCVoiced, a binary feature to indicate whether the last phone of the previous word is a voiced consonant ($value = 1$) or not ($value = 0$);
- NextCVoiced, a binary feature to indicate whether the first phone of the next word is a voiced consonant ($value = 1$) or not ($value = 0$);
- PrevEqual, a binary feature to indicate whether the last phone of the previous word is the same as the first phone of the current word ($value = 1$) or not ($value = 0$);

[1] The phonetic transcriptions that are shown throughout this text use the SAMPA phone set for European Portuguese, for further information refer to http://www.l2f.inesc-id.pt/˜lco/ptsam/ptsam.pdf

- NextEqual, a binary feature to indicate whether the last phone of the current word is the same as the first phone of the next word ($value = 1$) or not ($value = 0$);

During the development of the context-dependent lexicon, the TTS lexicon pronunciation is needed only for setting the word context. After knowing the way the speaker uttered each word of the corpus, and by knowing the word context, a Classification and Regression Tree[11] was used for clustering word instances with the same pronunciation. The clustering of word instances was performed by using the procedure used in [12], for clustering similar units for speech synthesis. Here, we use words instead of the speech units used in that work. Particular attention must be paid to the way the distance between the word instances are computed. Since we aim to group words with similar pronunciations in the same cluster, a distance between any two words must be based on their pronunciation. The selected distance measure between any two words is the cost of aligning their phone sequences. The phone distances were assigned by using a technique described in [13].

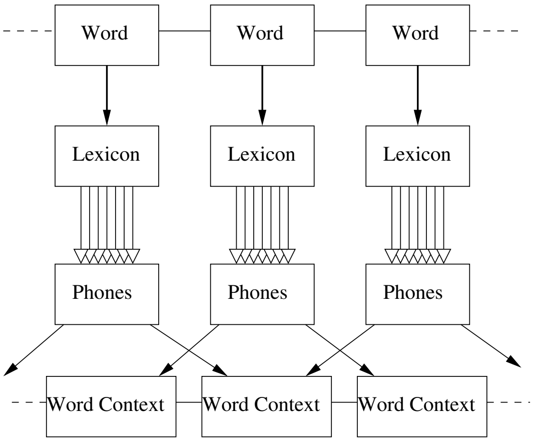

Figure 3: *Step sequence to generate the word context.*

3.2.2. Using the context-dependent lexicon

When Festival starts, the trees representing each word of the corpus are loaded into memory. For finding a word pronunciation, the system looks into the corresponding tree already in memory. If no such tree is found. Otherwise, the tree is used to provide speaker-specific pronunciation of the word in that context. Fig. 4 schematises all the steps the system has to perform. Firstly, the canonical pronunciation is produced using the system's lexicon. Then, word features are set according with that context. These features will be used later on to choose the most appropriate tree leaf for that word. Once the leaf is found, the pronunciation of the word instance that is closest to the cluster centroid, will be the elected one.

4. Evaluation

Both approaches were evaluated in their ability to reduce the number of concatenation points. Two scenarios were created. In the first, we evaluated only the improvements in the limited do-

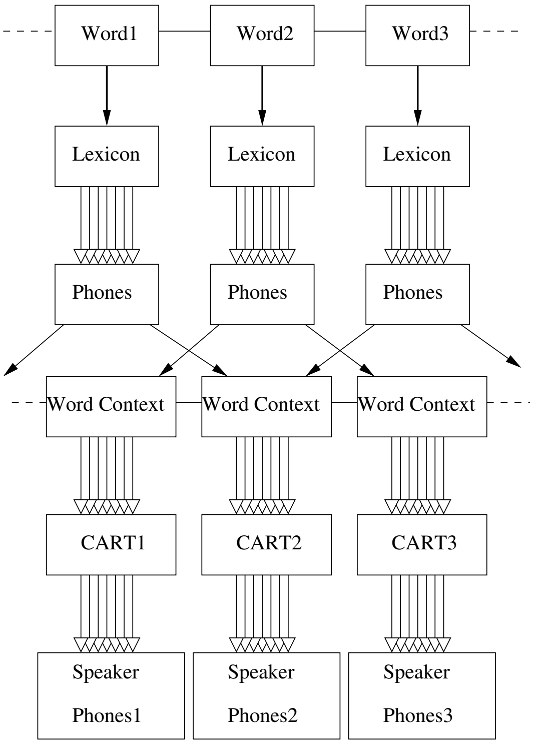

Figure 4: *Graphical representation of the generation of speaker-specific word pronunciations.*

main task of the butler's dialogue system and all the sentences selected for evaluation were generated by the system. In the second scenario, unrestricted sentences were used to compare the performance of the two approaches. A set of sentences was chosen from a newspaper text corpus. The number of concatenation junctures needed by each new approach was compared against the baseline system without adaptation to the speaker's pronunciation. Let A be baseline system, B the synthesizer using first approach described in this paper, and C the system the uses the speaker's own word pronunciations. Since the goal is to discover which system can produce fewer and larger speech chunks, we will use as evaluation measure the ratio between the number of phones and the number of non-adjacent chunks retrieved by the unit selection algorithm. This measure corresponds to the number of phones per chunk and can be expressed by:

$$f(X) = \frac{\#\ Phones}{\#\ Speech\_Chunks} \qquad (1)$$

Table 1: *Average number of units per speech chunk (units/chunk).*

	System		
Scenario	A	B	C
Domain-specific	2.9	4.1	5.0
Out-of-domain	1.9	1.8	2.1

Table 1 shows that in the limited domain scenario,

$system\ A$ performs poorly, but, in the other hand, both systems B and C can produce larger chunks, as expected. The approach that uses the speaker-specific word pronunciation outperform all others in both scenarios. The baseline system performs better than the approach using the canonical word pronunciations for out of domain utterances. A non-negligible factor that reduces the chunk length systematically, is the location of the phrase breaks in the sentence, since our synthesizer's prosodic modules still need some adaptation to the speaker's speaking-style.

In order to find out how the chunk's length relates with the resulting speech quality, we asked some colleagues to listen to a set of utterances for each different scenario. The evaluation consisted on listening to a pair of signals and to select the most appropriate answer out of four possibilities:

- Signal x sounds more natural than signal y;
- Signal y sounds more natural than signal x;
- Both signals sound very natural;
- None of the signals sounds good.

Table 2: *System A against system B*

Scenario	A Better	B Better	Both Good	Both Bad
Domain	22%	28%	39%	11%
OutofDom	0%	22%	33%	45%

Table 3: *System A against system C*

Scenario	A Better	C Better	Both Good	Both Bad
Domain	0%	50%	28%	22%
OutofDom	11%	33%	22%	34%

Table 4: *System B against system C*

Scenario	B Better	C Better	Both Good	Both Bad
Domain	6%	39%	55%	0%
OutfoDom	10%	67%	0%	23%

By observing the Tables 2, 3 and 4, the same conclusions can be drawn. $System\ C$ outperforms all others.

5. Conclusions

In our study, we evaluate three approaches in terms of both the speech chunk average lengths and perceived quality. The lexicon-based approach performs better than the baseline system that does not take into account the speaker's pronunciation variability. However, the best results were achieved by using an approach that uses a context-dependent lexicon adapted to the speaker-specific word pronunciation. This approach performed better than all others both in terms of average chunk length as well as in terms of speech naturalness.

6. Acknowledgements

This work is part of Sérgio Paulo's PhD Thesis sponsored by a Portuguese Foundation for Science and Technology (FCT) scholarship, and was partially funded by FCT project POSI/SRI/41071/2001.

7. References

[1] A. Hunt and A. Black, *Unit selection in a concatenative speech synthesis system using a large speech database.* In Proceedings of ICASSP 96, Atlanta, Georgia, USA, 1996.

[2] A. Black and P. Taylor, *CHATR: a generic speech synthesis system.* In Proceedings of COLING-94, Kyoto, Japan, 1994.

[3] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, *The AT&T Next-Gen TTS System.* 137^{th} Acoustical Society of America meeting, Berlin, Germany, 1999.

[4] M. Lee, D. Lopresti, and J. Olive, *A Text-to-Speech Platform for Variable Length Optimal Unit Searching using Perception Based Cost Functions.* International Journal of Speech Technologies, vol. 6, no. 3, pp. 347-356, 2003.

[5] R. Donovan, *A New Distance Measure for Costing Spectral Discontinuities in Concatenative Speech Synthesisers.* In Proceedings of 4th ISCA Speech Synthesis Workshop, Scotland, September 2001.

[6] N. Nukaga, R. Kamoshida and K. Nagamatsu, *Unit Selection Using Pitch Synchronous Cross Correlation for Japanese Concatenative Speech Synthesis.* In Proceedings of 5th ISCA Speech Synthesis Workshop, Pittsburgh, June 2004.

[7] A. Syrdal, A. Conkie *Data-driven Perceptually based Join Costs.* In Proceedings of 5th ISCA Speech Synthesis Workshop, Pittsburgh, June 2004.

[8] J. Neto and R. Cassaca, *A Robust Input Interface in the scope of the Project Interactive Home of the Future*, in Proceedings of ROBUST 04, Norwich, UK, 2004.

[9] A. Black, P. Taylor and R. Caley, *The Festival Speech Synthesis System.* System documentation Edition 1.4, for Festival Version 1.4.0, 17th June 1999.

[10] P. Taylor, A. Black, and R. Caley, *Heterogeneous relation graphs as a formalism for representing linguistic information.* in Speech Communication, 33, 2001.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth & Brooks, Pacific Grove, CA., 1984.

[12] A. Black and P. Taylor, *Automatically clustering similar units for unit selection in speech synthesis.* In Proceedings of Eurospeech 97, Rhodes, Greece, September 1997.

[13] S. Paulo and L. Oliveira, *Multilevel Annotation of Speech Signals Using Weighted Finite State Transducers*. In Proceedings of IEEE 2002 Workshop on Speech Synthesis, Santa Monica, California, 2002.