

Improving the accuracy of pronunciation prediction for unit selection TTS

Justin Fackrell, Wojciech Skut and Kathrine Hammervold

Rhetorical Systems Ltd
4 Crichton's Close, Canongate,
Edinburgh, UK
justin.fackrell@rhetorical.com

Abstract

This paper describes a technique which improves the accuracy of pronunciation prediction for unit selection TTS. It does this by performing an orthography-based context-dependent lookup on the unit database. During synthesis, the pronunciations of words which have matching contexts in the unit database are determined. Pronunciations not found using this method are determined using traditional lexicon lookup and/or letter-to-sound rules. In its simplest form, the model involves a lookup based on left and right word context. A modified form, which backs-off to a lookup based on right context, is shown to have a much higher firing rate, and to produce more pronunciation variation.

The technique is good at occasionally inhibiting vowel reduction; at choosing appropriate pronunciations in case of free variation; and at choosing the correct pronunciation for names. Its effectiveness is assessed by experiments on unseen data; by resynthesis; and by a listening test on sentences rich in reducible words.

1. Introduction

TTS systems based on unit selection have provided a major step forward in the quality of synthesized speech. Most such systems (e.g. [1, 2, 3]) have a broadly similar architecture in which the text to be synthesized is first converted into an intermediate phonological representation. The target sentence is thus defined in terms of phonemes, stress, phonotactic and prosodic features. A database of thousands of speech units is then searched to find the best sequence of units to represent the target. That sequence of speech units is then concatenated to produce output speech.

While the unit selection and waveform concatenation modules of such systems are speaker-specific, the preceding processing stage of letter-to-sound (LTS) conversion is, in most systems, speaker independent. It is understood, however, that in order to provide speaker-accurate pronunciations, the LTS process should be very much speaker-dependent.

This has rarely been a problem in the past, since most TTS systems have been developed for a single voice, and so the LTS module (lexicon, LTS rules, reduction rules, etc.) could be developed along with the rest of the system. However, with the advent of unit selection-based TTS systems which provide more than one speaker for an accent, lexical issues in general, and the issue of how to treat phenomena such as reduction and free variation in particular, have become more important.

The hand-crafting of reduction rules is in itself problematic, since it is easy to make rules which under-generate, or over-generate, but difficult to find the sweet spot between these extremes. In particular, it is extremely difficult to make good predictions of pronunciations for sequences of reducible words.

For example, the sentence "I am glad *to have been* in this country." contains a sequence of three reducible words (italized), yielding eight ($= 2^3$) possible phoneme sequences for those three words.

The need for speaker-dependent pronunciation prediction also manifests itself when the sentences which make up the unit database are themselves synthesized by the system ("resynthesis"). In an exploratory experiment on an RP English unit selection database, nearly 60% of sentences ended up with at least one difference in the symbolic representation. These differences can be assigned to the following causes: differences in vowel reduction; differences in the choice of pronunciation variant for words exhibiting free variation; and differences in the location and strength of phrase boundaries.

The quality of synthesized speech increases if the units are contiguous in the database (i.e. if the chunks selected are big). And clearly the likelihood of getting big chunks is will be improved if the transcription used in synthesis matches that in the database.

Previously, there has been some interest in the topic of matching synthesis to speaker characteristics. Jilka and Syrdal [4] describe how careful control of the recording process, and careful design of the labelling and synthesis lexicons, improves the accuracy of phoneme prediction, which in turn has a positive impact on speech quality for German.

Another way of improving the match between prediction and data is to bypass the LTS conversion process wherever possible. A tree-based approach proposed by Taylor and Black [5] offers a solution to the problem of integrating higher- and lower-level representations in unit selection, but there remain difficulties to do with the scoring of candidates on different temporal levels (e.g. scoring phonemes for comparison with whole words). In an approach using Finite State Transducers, Bulyko [6] entertains multiple pronunciations, resolving them during unit selection.

An alternative potential solution to these problems is to use data-driven techniques to derive the LTS system to ensure a high degree of match between speaker characteristics and synthesizer behaviour. This is attractive for two reasons: Firstly, it enables rapid tailoring of LTS systems for newly recorded speakers, encompassing phenomena such as reduction without great manual effort. And secondly, it improves the resynthesis behaviour of the system in retrieving large chunks of material *en bloc*.

However, to ensure sufficient coverage of the language, and to provide good generalization properties, the speaker-specific pronunciation database required to train such models would contain hundreds of thousands of sentences. But in reality, the only data usually available for a particular speaker is the unit

database for that speaker, in which the sentences number in the *hundreds*.

In this paper an alternative solution is sought which makes use of whatever data is available for a particular speaker, yet backs off where data is missing. In contrast to techniques based on higher-level representations, this technique retains a phoneme-based intermediate description, making it easy to integrate with the existing unit selection module.

The technique involves the addition of a context-dependent lexicon lookup stage before the existing lexicon lookup and LTS rules. The context-dependent lexicon is based on the recorded unit database.

In Section 2 the model is described in some more detail. In Section 3 experiments are described which assess how often the model fires (which measures its potential impact), how much it affects unit “retrieval” during resynthesis, and how much it improves TTS quality (judged by listening tests). The results are discussed and conclusions given in Section 4.

2. The model

The new model is inserted as a preprocessing stage to the existing LTS conversion module. It returns phonetic transcriptions of words by looking them up in the speaker’s recorded database. *Only* words with matching contexts have transcriptions assigned to them by this model. The remaining words have their pronunciations predicted by the usual context-free lexicon or, if the word is “out-of-vocabulary” (i.e. not in the lexicon), by speaker-specific LTS rules.

2.1. Training

The context-dependent lexicon L_c is indexed by tuples each consisting of an orthographic string w and its context c . For each (w, c) tuple, the lexicon contains the observed pronunciation p .

L_c is formed by simply moving a sliding context window over all word tokens in the recorded database, in order to make estimates of the probability $\hat{P}(w|c \Rightarrow p)$ that (w, c) is associated with pronunciation p . A pruning stage is then carried out to remove all but the most commonly occurring pronunciation for each (w, c) . Note that, depending on the definition of context used, this lexicon may have very limited coverage.

2.2. Prediction

Algorithm 1 describes the LTS procedure during synthesis: Given a word w , with context c and features (e.g. part-of-speech) f , the context-based lexicon L_c is consulted. If the word is found, the pronunciation is retrieved from L_c . If not, then the word, plus its features f are used to look up the word’s pronunciation in a word-based lexicon L_w . If it is not found there, then LTS rules R_{LTS} are used.

Algorithm 1 LTS conversion using context-based lexicon

```

if  $(w, c) \in L_c$  then
   $p = L_c(w, c)$ 
else if  $(w, f) \in L_w$  then
   $p = L_w(w, f)$ 
else
   $p = R_{LTS}(w, f)$ 
end if

```

Preliminary experiments have been carried out with the fol-

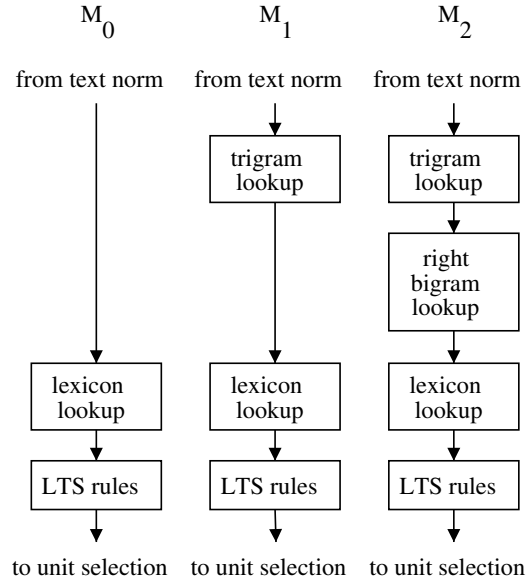


Figure 1: *Lexicon-based models of pronunciation*

lowing contexts: the left-word context only (“left bigram”), the right-word context only (“right bigram”), and left- and right-word context (“trigram”). Of these, the trigram is the most specialized and gives the best predictions in general, although it doesn’t fire very often. The firing rate can be greatly increased, and the overall performance improved, if the right bigram model is added as a back-off in case the trigram doesn’t fire. In this paper, results will be presented for three models, as shown in Figure 1: the default system (M_0); trigram (M_1); and trigram, with back-off to right bigram (M_2).

3. Results

To evaluate the technique, comparisons were made between the three models show in Figure 1.

Several data sets, each of approximately 10000 sentences, were used to compare the models. The first three of these sets were taken from web pages themed on *Finance*, *News*, and *Weather*. The fourth set, *Func*, contains sentences no longer than 15 words, which are rich in sequences of function words and reducible words which, as discussed in Section 1, are a particular problem for reduction prediction.

3.1. Firing rates

Figure 2 shows the percentages of word tokens in the various test sets for which the models fired (i.e. how often a word-in-context was found in the context-dependent lexicon L_c). Evidently, adding the bigram back-off condition to M_2 greatly increases the firing rate.

It is encouraging that the firing rate for the models is as high as it is, despite the small size of the database. The context-based lookup in M_2 finds an entry for approximately one-third of words. L_c seems to cover the *Weather* data set much better than it covers the other data sets. This turns out to be due to the limited variation in *Weather*, rather than to a preponderance of weather-related material in L_c .

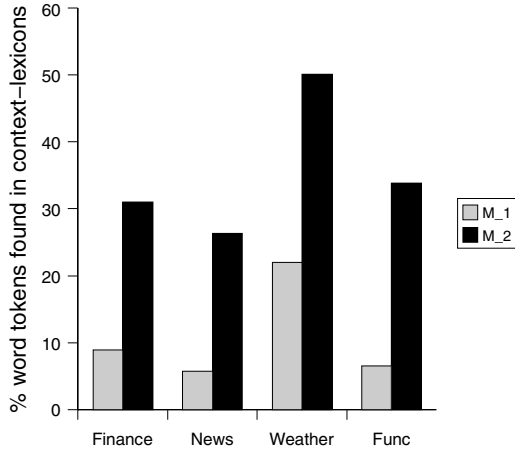


Figure 2: *Firing Rate*: the percentage of word tokens for which a context-dependent lexicon entry was found

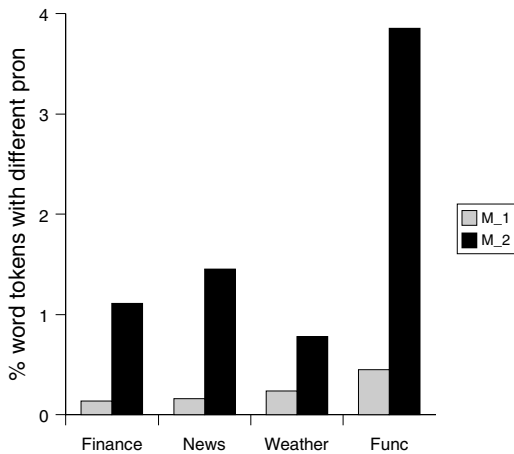


Figure 3: *Pronunciation differences* - the % of word tokens for which models M_1 and M_2 gave different pronunciation to the default model M_0

3.2. Pronunciation changes

Figure 3 shows the percentages of word tokens for the data sets in which the models M_1 and M_2 predict *different* pronunciations to the default system M_0 . The measures in both tables are calculated over *all* word tokens in the data sets, including words which have just a single pronunciation (such as most nouns). M_2 predicts a pronunciation different to M_0 for about 1 word in 100. It is encouraging to note that the pronunciations differ most for the *Func* data set, since reduction prediction is a primary aim of this work.

3.3. Reduction rules

Some examples of the behaviour of the new algorithm are shown in Table 1. This illustrates the overgeneration of the reduction in M_0 , and M_2 's behaviour in occasionally inhibiting reduction.

Although something of a side-effect, M_2 is also good at predicting pronunciations for first names, if they are in the unit selection database, without requiring specific homograph disambiguation rules to be written for them. For example

Table 1: *Some examples of differences between M_0 and M_2 . Reduced forms are shown in italics and full forms of reducible words are shown in bold*

M_0	"I don't think <i>that any of that</i> was directed to me."
M_2	"I don't think <i>that any of that</i> was directed to me."
M_0	Just as it <i>has been for the last 30 years.</i>
M_2	Just as it <i>has been for the last 30 years.</i>
M_0	Then we <i>have to</i> have a clear exit strategy."
M_2	Then we have to have a clear exit strategy."

"Goran" is correctly disambiguated to ¹ /j 3: r @ n/ for "Sven *Goran* Eriksson", and to /g o r @ n/ for "*Goran* Ivanisevic". This is by no means a replacement for a proper homograph disambiguation module, but it does give some positive effect even if no homograph disambiguation module exists.

3.4. Retrieval

The starting point for this work was the finding that the phoneme overlap during resynthesis of the material in the unit database was low. This problem has been investigated with the new pronunciation models.

The first data column of Table 2 shows the degree of symbolic overlap between the resynthesized sentences and the original database. The table shows that the percentage of sentences which are now rendered symbolically identical to the database sentence increases from 40.1% to 51.4% through the use of M_2 . This is an improvement, although it is still rather low, primarily because the pause prediction model currently predicts only about two-thirds of pauses correctly; This is a topic for future improvement. The word pronunciation rate (s_{word}), measuring how many word tokens in the database had their pronunciations correctly predicted improves from 94.8% for M_0 to 98.7% using M_2 .

The other data columns in Table 2 show the "retrieval" statistics for resynthesis of the database. They are calculated as follows; the database consists of sentences x_1, x_2, \dots, x_N . When the text of sentence x_1 is synthesized, a count is made of the number of selected speech units which came from the recording of x_1 . This count is expressed as a percentage relative to the number of units synthesized. This measure can be expressed in terms of speech units, words or sentences.

The symbolic match scores s_{sen} and s_{word} provide ceilings for the unit retrieval scores r_{sen} and r_{word} respectively. The improvements are not spectacular, but they do show that with the new approach the chance of retrieving recorded data *en bloc* is improved. The chunk size also increases with the new technique.

3.5. Listening Test

A listening test was conducted to determine if the changes to letter to sound prediction made by the best model M_2 do represent a real improvement over the default strategy M_0 . The listening test was carried out using a system based on a female RP English speaker, with 8 listener subjects, all experienced in the field of TTS.

Fifty sentences from the *Func* set, for which the two models produced different results, were used in the listening test. The listening test setup is as follows: For each sentence, the text is shown on the screen, with the words which have been assigned

¹in SAMPA notation.

Table 2: Measures of match between resynthesis of the database, and the database itself.

	symbolic match		unit retrieval			avg chunk size (units)
	sent	word	sent	word	unit	
	s_{sen} (%)	$s_{wrđ}$ (%)	r_{sen} (%)	$r_{wrđ}$ (%)	r_{unit} (%)	
M_0	40.1	94.8	6.8	45.2	55.9	3.80
M_1	48.9	97.6	8.0	48.6	58.4	4.07
M_2	51.4	98.7	8.1	49.1	58.7	4.11

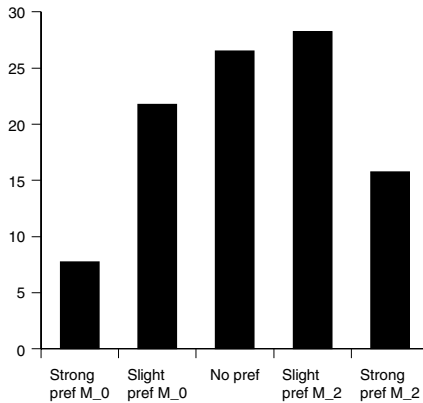


Figure 4: Listening test preference results

different transcriptions by M_0 and M_2 marked in square brackets, e.g. “If they say no I [would] like to know why.” The subject listens to the synthesized waveform for M_0 and M_2 , their order being randomized. If necessary the subject can repeat the playback of the waveforms any number of times. The user selects their preference on a 5-point scale (strongly prefer A, slightly prefer A, no preference, slightly prefer B, strongly prefer B). The order in which the sentences are presented to each subject is also randomized.

Figure 4 shows the average preference scores for the 50 sentences. By forming average scores for each listener, a Wilcoxon Signed Rank Test[7] was carried out: With a null hypothesis H_0 that the distributions of preference for M_0 and M_2 are the same, and an alternative hypothesis H_a that $M_2 > M_0$ (i.e. M_2 is preferred), the test yields $T_- = 5, T_+ = 31$ and $T_0(\alpha = 0.05) = 6$ for $n = 8$. The rejection region is $T_- \leq T_0$, so H_0 is rejected and H_a is accepted: $M_2 > M_0$. The subject agreement is not particularly high (average pairwise agreement 32.5%, Krippendorff’s $\alpha = 0.28$ [8]) reflecting that the subjects are rarely unanimous in preferring M_0 or M_2 . In fact, out of the 50 sentences, there were only four sentences for which all subjects preferred M_2 and just one for which all subjects preferred M_0 .

4. Discussion and Conclusion

This paper has presented a simple plug-in pronunciation prediction model, which overrides our default pronunciation model for roughly 1/3 of words, producing a pronunciation different to the default model for about 1% of word tokens. The model, which is automatically created from the unit selection database,

makes a positive impact on speech synthesis quality, predominantly by inhibiting vowel reduction. As a side-effect, the approach provides some limited disambiguation of name pronunciation.

The proposed model performs better than our current model insofar as unit retrieval is concerned: if a sentence or phrase to be synthesized has actually been recorded, then the likelihood of retrieving that material as a single chunk are increased. This is achieved without recourse to any modification to the unit search. Future work will investigate how the applicability of the model can be widened, perhaps by adding left-word context.

The main difference between the characteristics of our synthesizer output and the characteristics of the database on which it is based is to do with phrasing and pause prediction, and work on automatic training of phrasing modules is ongoing.

5. Acknowledgements

The authors would like to thank Maria Wolters and Matthew Aylett for their assistance with the interpretation of the listening test results.

6. References

- [1] Coorman, C., Fackrell, J., Rutten, P. and Van Coile, B. “Segment selection in the L&H Realspeak laboratory TTS system”, Proc ICSLP, Beijing, China, 2000.
- [2] Hunt, A. J. and Black, A. W., “Unit selection in a concatenative speech synthesis system using a large speech database”, Proc ICASSP, Atlanta, USA, 373-376, 1996.
- [3] Beutnagel, M., Conkie, A. and Syrdal, A. K., “Diphone synthesis using unit selection”, Proc 3rd ESCA/COCOSDA Workshop on Speech Synthesis, Jenolan Caves, Australia, 185-190, 1998.
- [4] Jilka, M. and Syrdal, A. K., “The AT&T German Text-to-Speech System: Realistic Linguistic Description”, Proc ICSLP, Denver, USA, Vol 1, 113-116, 2002.
- [5] Taylor, P. and Black, A. W., “Speech synthesis by phonological structure matching”, Proc. Eurospeech, Budapest, Hungary, 1999.
- [6] Bulyko, I. Ph.D. thesis “Flexible Speech Synthesis using Weighted Finite State Transducers”, University of Washington, USA, March 2002.
- [7] Scheaffer, R. L. and McClave, J. T., “Probability and Statistics for Engineers”, PWS-Kent, Boston, 1990.
- [8] Krippendorff, K. “Content Analysis”, Sage, London, 1980.