interspeech 2008 incorporating SST 08

Unsupervised adaptation for HMM-based speech synthesis

Simon King[1], *Keiichi Tokuda*[2], *Heiga Zen*[2], *Junichi Yamagishi*[1]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]Nagoya Institute of Technology, Japan

Simon.King@ed.ac.uk

Abstract

It is now possible to synthesise speech using HMMs with a comparable quality to unit-selection techniques. Generating speech from a model has many potential advantages over concatenating waveforms. The most exciting is model adaptation. It has been shown that supervised speaker adaptation can yield high-quality synthetic voices with an order of magnitude less data than required to train a speaker-dependent model or to build a basic unit-selection system. Such supervised methods require labelled adaptation data for the target speaker. In this paper, we introduce a method capable of unsupervised adaptation, using only speech from the target speaker without any labelling.
Index Terms: speech synthesis, HMM-based speech synthesis, HTS, trajectory HMMs, speaker adaptation, MLLR

1. Introduction

1.1. Speech synthesis using HMMs

In recent Blizzard Challenge speech synthesis evaluations [1, 2], HMM-based systems have been found to have comparable quality to state-of the art concatenative systems. These HMM-based system use the so-called 'Trajectory HMM' [3] which is an algorithm for generating an observation sequence from an HMM that uses delta and delta-delta coefficients in the observation vector. By generating output that has the correct statistics with respect not only to the static coefficients, but also with respect to the delta and delta-delta coefficients, a smooth trajectory in observation space is produced. The observation vector used must contain sufficient information to generate a speech waveform. Typically, this might be spectral envelope information (represented as a cepstrum), a small number of coefficients of a multi-band noise model, and F0.

1.2. Comparison with speech recognition using HMMs

The models used for speech synthesis are essentially conventional HMMs, as used for automatic speech recognition (ASR). Therefore, it is relatively straightforward to apply techniques developed for ASR to models used for synthesis. Perhaps the most exciting of these is speaker adaptation, which is able to produce speech synthesis voices using as few as 100 sentences; compare this to a minimum of 1000 sentences to build a simple unit-selection system, and 10 000 to build a good system.

There are of course a number of significant differences in the way models are *configured* for speech synthesis, compared to ASR, although the underlying statistical model is the same is most respects. The observation vector for speech synthesis contains a more detailed spectral envelope (we use 40th-order Mel-cepstral features derived from STRAIGHT spectral envelope estimation), plus coefficients not required for ASR (we use the log energy of the aperiodic component of the signal in 5 frequency bands, plus log F0). Delta and delta-delta coefficients are appended: total observation vector dimension is 138.

Other differences include partitioning the observation vector into streams, explicit duration models, multi-space probability distributions to handle voiced/unvoiced regions and 5-state (rather than 3-state) phone models. These are not central to this work, so the reader is referred to [3] and references therein. The models we used have the same configuration as the HTS entry to the 2005 Blizzard Challenge [4], but built using HTS 2.1beta.

1.3. Speech synthesis uses highly context-dependent models

The subword units for synthesis are context-dependent phones, as in ASR, but the context is much richer. We use quinphones plus supra-segmental features: position of segment in syllable, position of syllable in word/phrase, position of word in phrase, stress/accent/length features of current/preceding/following syllables, distance from stressed/accented syllable, POS of current/preceding/following word, length of current/preceding/following phrase, end tone of phrase, length of utterance measured in syllables/words/phrases. These 'full context' models make unsupervised adaptation harder for synthesis than for ASR.

1.4. Supervised adaptation

The use of adaptation to create new voices for speech synthesis makes HMM-based speech synthesis very attractive. To date, supervised adaptation has been used: full context labels are required for the adaptation data (i.e., all supra-segmental feature values must be known). These labels are produced in the same way as those for the training data: they are predicted from the text using a TTS front-end. Note that no attempt is made to ensure these detailed labels accurately match what the speaker actually said (apart from checking that the words match), although that would be expected to improve results.

The work reported here uses Constrained Maximum Likelihood Linear Regression (CMLLR), because it performed well in initial experiments. More sophisticated schemes have recently become available [5]. CMLLR adapts HMMs by applying a linear transform to each Gaussian; the same transform is applied to the mean and variance of any particular Gaussian. This transform is learned using labelled adaptation data. Because the adaptation data are generally limited in quantity, and context-dependent models are used, there will not generally be an example of every model in the adaptation data. So, it is necessary to share transforms between groups of states, known as 'regression classes'. The grouping of states into regression classes is part of the model training process; the classes are arranged into a tree, so that the number of classes can be varied to suit the amount of adaptation data available. In this work, we

Accepted after peer review of full paper

use only regression trees for adaptation, since they are computationally cheaper than decision trees. In reality, the regression classes do not contain entire states, but just the parameters for an individual stream in a state (spectral envelope, noise bands or F0). For clarity, we will omit this detail in the remainder of this paper and describe the method only in terms of states.

For speech synthesis, a model trained on multiple speakers' data is called an 'Average Voice model' [6]. It can be adapted into a speaker-specific model using CMLLR (or some other method), using a small amount of adaptation data from the target speaker to estimate the adaptation transforms.

2. Unsupervised adaptation for synthesis

In ASR, unsupervised adaptation can be as simple as running a recogniser using unadapted models to obtain an initial phonetic transcription of the adaptation data. This transcription is then used as the labelling for supervised adaptation.

2.1. Obtaining full context labels for the adaptation data

The first approach that we considered was analogous to that for ASR: obtain full context labels for the adaptation data. ASR using full context models would produce the required full context labels for the adaptation data but is unfortunately not feasible. It would be computationally very expensive (even lattice rescoring would produce vast lattices when expanding from triphones to full context); it is also likely to produce very inaccurate labels because recognition of some features will be hard or impossible. Instead, we could perform ASR to obtain words, then predict full context labels from this (as for the training data, or adaptation data with known word labels). However, word errors in ASR output may cause significant errors in the full context labelling, since effects will spread beyond the word boundary.

2.2. Using only phonetic labels for the adaptation data

Obtaining full context labels for the adaptation appeared to be difficult. Therefore, we considered an alternative, in which only phonetic labels would be obtained for the adaptation data (using standard ASR techniques). The problem then changes to one of performing adaptation using labels that do not match the model set being adapted: we have phonetic labels, but full context models. Our proposed method offers a way to achieve this.

2.3. The proposed method

2.3.1. Learning adaptation transforms using a reduced-context labelling of the adaptation data

We need to estimate an adaptation transform for every state in the full context model set. However, we only have adaptation data labelled with triphones. Rather than attempt to produce full context labels for the adaptation data, we decided to learn adaptation transforms for a triphone model set, and then apply these transforms to the full context models. The transform for a full context model state is taken from the corresponding triphone model state (the mapping is trivial: just drop most of the context). The outline of the method is presented in Figure 1.

A set of Average Voice triphone models is required; they must have to have the same number of emitting states as the full context models (5) and the same observation vectors. These models could be trained from scratch, or they could be derived from the trained full context models. We chose the latter: it is quicker because it requires fewer training steps; more importantly, the parameters of the triphone models are more likely

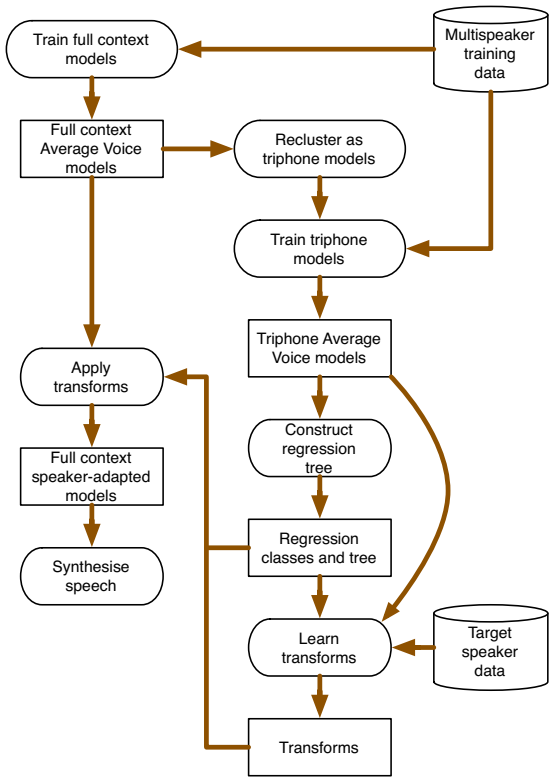

Figure 1: Adapting full context models using transforms learned by triphone models

to be close to those of the corresponding full context models, which should lead to more transferable adaptation transforms.

2.3.2. Creating triphone models

The method used to convert full context models (one Gaussian per state) to triphone models (also one Gaussian per state) is to untie the full context model states and recluster them using only a triphone question set. Since the triphone models we require must be HMMs rather than HSMMs,[1] we converted the full context HSMMs to HMMs by reinstating the transition matrix and training it using EM. The model states are untied and trained for one iteration, obtaining the required state occupancy statistics. The clustering is performed on these untied states of the full context model, but using only questions about their triphone context. The result is a tree from which a complete list of triphone models can be synthesised. Finally, these triphone models were trained for a few more iterations of EM and a triphone model adaptation regression tree is learned.

2.3.3. Estimating adaptation transforms for triphone models

A conventional phone recogniser was built using these single-Gaussian triphone acoustic models and a simple bigram phone language model (estimated from the phonetic transcription of the acoustic training data). This recogniser is very simple: we did not attempt to build the most accurate recogniser possible. This is because we were interested in the performance of

[1] This is only for technical reasons: decoding using HSMMs was not supported by the code at the time this work was done. As a consequence, we are not adapting the duration model.

speaker adaptation when there is a significant error rate in the labelling of the adaptation data, as would be expected in real applications. The recogniser had a phone accuracy of around 50-60%. The output of the recogniser is a sequence of triphone labels for the adaptation data.

Given these triphone labels, adaptation transforms were estimated for the triphone models, using the triphone regression tree and corresponding regression classes. The result is that every triphone model state (which belongs to a single regression class) has a transform associated with it. Note that any triphone model states that have their parameters tied will necessarily belong to the same regression class.

2.3.4. Transferring adaptation transforms from the triphone models to the full context models

The mapping from a triphone model to the corresponding group of full context models is trivial, but in order to actually apply the transforms to the full context model parameters, we must first deal with differences in the parameter tying structure between the triphone and full context model sets.

Recall that the triphone states are parameter-tied by decision tree clustering, using conventional triphone questions. The full context model states are also parameter-tied, but using a tree containing quinphone and prosodic questions. As a consequence, there is no guarantee of any simple correspondence between the clusters of triphone model states and clusters of full context model states. Because the same adaptation transform will be applied to all members of any given cluster of tied states, it is essential that each cluster is associated with exactly one transform.

The method we devised does not compromise the quality of the models. We manipulate the state tying scheme of the full context model so that it is compatible with the triphone model *regression classes*. We take each cluster of tied states in the full context model and partition it into a number of smaller clusters such that, within each new cluster of full context model states, all the corresponding triphone model states are in the same regression class. This is performed on fully trained models, so the only effect is to increase the storage space required by the model by a small factor. The parameter values of each state are unchanged. Of course, instead of partitioning the state clusters of the full context model, we could simply untie all full context model states, but the resulting model would be very large.

3. Experimental conditions

The experimental test protocol was the same as that of the Blizzard Challenge 2007 and used the same web-browser interface. Each of the different configurations of our system (Figure 2) was assigned an identifying letter and was entered as a 'participant' in this mini Blizzard Challenge. We omitted the section on pairwise "same or different" judgements (intended for Multidimensional Scaling analysis). The test sentences used were taken from a previous Blizzard Challenge. 40 listeners (native speakers of English, paid subjects, no self-reported hearing problems) were divided into 10 blocks of 4 listeners. Using a balanced 10-by-10 (systems -by- test sentences) Latin square design within each section, each listener block was assigned a row of the Latin Square, and was thus presented with 10 different test sentences, each produced using a different system.

Different sentences were used in each of the three sections of the test, making a total of 30 different test sentences. The audio files for the test sentences are included with this paper.

The first section of the listening test provides listeners' judgements on the similarity of the synthetic speech to the original speaker. Three reference natural utterances (different sentences to the synthetic speech) were provided. No natural examples were used in the remainder of the experiment because we were primarily interested in comparisons between speaker-dependent, supervised speaker-adapted and unsupervised speaker-adapted systems. The second and third sections of the test provide listeners' Mean Opinion Scores (MOS) for sentences from 'conversational' and 'news' domains respectively. The final section provides Word Error Rate (WER) using Semantically Unpredictable Sentences (SUS).

In general, the speaker-adapted systems performed slightly worse than the speaker-dependent one. This was because we used a relatively simple version of the Average Voice scheme, and did not utilise all the recent improvements to this technique [5].

We used the ARCTIC corpus [7]. Speakers awb, clb, jmk, rms and slt were used as training data and speaker bdl was used as the target speaker. The amount of data used is summarised in Table 1. There were two reference systems: System A was a speaker-dependent system trained on all the available bdl data, and system B was a speaker-adapated system trained on all the available data from the training speakers and adapted using the correct full-context labels on all the available bdl data.

Data set	Speakers	Sentences	Minutes
all except bdl	awb, clb, jmk, rms, slt	5648	310
all bdl	bdl	1131	50
10% of bdl	bdl	114	5
1% of bdl	bdl	12	$\sim$1

Table 1: The data

4. Results

We present results in Figure 2, which summarises the results graphically using the presentation method described in [2] with the same scales. MOS values are not directly comparable, since listeners' judgements are relative. The somewhat small listening test (40 listeners) was not able to find statistically significant differences between the different systems, so we can only interpret differences as trends.

Adapting using phonetic labels vs. full context labels: In order to separate the effects of errors in the labels for the adaptation data, from the effects of learning the adaptation transforms using triphone models, we compared full context models in a supervised setting with triphone models in a supervised setting (B vs. C). The comparison can also be made using less adaptation data (E vs. F, or H vs. I). There is generally a small reduction in similarity, naturalness and intelligibility with supervised triphone adaptation vs. full context adaptation.

Supervised vs. unsupervised adaptation: The primary goal of our experiments was to compare unsupervised adaptation against supervised adaptation. It is already known that supervised adaptation can result in equal or better performance than speaker-dependent HMM-based synthesis. We wished to find out how much degradation would result from unsupervised adaptation. The comparison of configurations C vs. D tells us about the effect of a 40-50% phone error rate in the adaptation labels. The comparison can also be made using less adaptation data (F vs. G, or I vs. J).

Amount of adaptation data: We compared use of all data with 10% of the data and 1% of the data for full context mod-

	Training data	Adaptation data	Adaptation labels	Supervised?	MOS sim	MOS news	MOS conv	WER (%)
A	all bdl	none			2.9	2.8	2.8	10.9
B	all except bdl	all bdl	full context	Y	2.4	2.7	2.5	9.3
C	all except bdl	all bdl	triphone	Y	2.2	2.4	2.7	11.5
D	all except bdl	all bdl	triphone	N	2.1	2.5	2.3	14.6
E	all except bdl	10% of bdl	full context	Y	2.5	2.7	2.7	10.6
F	all except bdl	10% of bdl	triphone	Y	2.3	2.2	2.7	13.7
G	all except bdl	10% of bdl	triphone	N	2.2	2.3	2.5	18.3
H	all except bdl	1% of bdl	full context	Y	2.0	2.4	2.5	15.8
I	all except bdl	1% of bdl	triphone	Y	2.0	2.3	2.5	15.5
J	all except bdl	1% of bdl	triphone	N	1.9	2.1	2.5	14.6

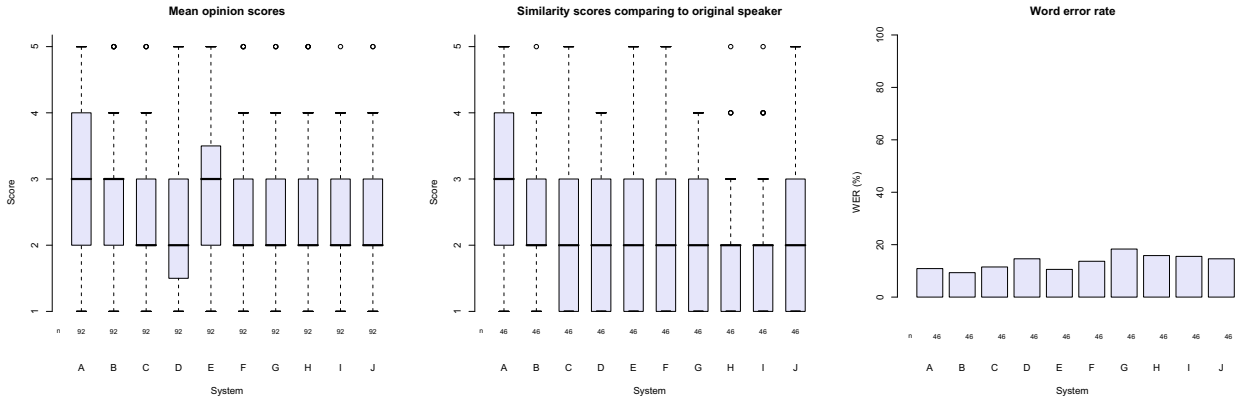

Figure 2: The table shows the various configurations of the system that were compared in the listening test. System A is speaker-dependent; all other systems are speaker-adapted. The right hand part of the table summarises MOS and WER results. (sim: similarity to original speaker; news: naturalness on news domain sentences; conv: naturalness on conversational domain sentences; WER: word error rate on semantically unpredictable sentences). The results are displayed graphically in the three plots

els with supervised adaptation (B vs. E vs. H), triphone models with supervised adaptation (C vs. F vs. I) and for triphone models with unsupervised adaptation (D vs. G vs. J). As expected, quality decreases as data is reduced, but the decline is graceful and, even with only around 100 sentences of adaptation data, the system quality is still at a usable level.

5. Conclusions

We presented a simple method for adapting full context HMMs when only phonetic labels are available for the adaptation data, which enables unsupervised adaptation. The method still adapts to the general supra-segmental characteristics of the target speaker, although adaptation to specific characteristics is limited by the use of triphone models to learn the transforms. There is a small degradation in quality from correct full context labels to correct triphone labels (e.g., B vs. C). There is a further degradation when adaptation is unsupervised, because of errors in the triphone labels (e.g. C vs. D). WER increases particularly when using unsupervised adaptation, although similarity and naturalness are less severely impacted. With more accurate triphone labels, the performance of the unsupervised system would approach that of the supervised triphone system, which is not far below that of the supervised full context system. The gap in performance between the adapted and speaker-dependent systems could be removed by use of recent improvements to the Average Voice method [5].

Acknowledgements: This work was carried out when SK was an invited researcher at NIT. SK holds an EPSRC Advanced Research Fellowship. Vasilis Karaiskos ran the listening test. This work used the Edinburgh Compute and Data Facility (`www.ecdf.ed.ac.uk`) which is partially supported by eDIKT (`www.edikt.org`).

6. References

[1] M. Fraser and S. King, "The Blizzard Challenge 2007," in *Proc. Blizzard 2007 (in Proc. Sixth ISCA Workshop on Speech Synthesis)*, Bonn, Germany, August 2007.

[2] R. A. J. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the Blizzard Challenge 2007 listening test results," in *Proc. Blizzard 2007 (in Proc. Sixth ISCA Workshop on Speech Synthesis)*, Bonn, Germany, August 2007.

[3] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech and Language*, vol. 21, no. 1, pp. 153–173, January 2007.

[4] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Information and Systems*, vol. E90-D, no. 1, pp. 325–333, January 2007.

[5] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorihms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech and Language Processing*, 2008 (Accepted for publication).

[6] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Trans. Information and Systems*, vol. E90-D, no. 2, pp. 533–543, February 2007.

[7] J. Kominek and A. Black, "The CMU Arctic speech databases," in *Proc. 5th ISCA speech synthesis workshop*, Pittsburgh, USA, 2004, pp. 223–224.