Template-based ASR using posterior features and synthetic references: comparing different TTS systems

Serena Soldo [†,‡], Mathew Magimai.-Doss [†], Hervé Bourlard [†,‡]

[†] Idiap Research Institute, Martigny, Switzerland
[‡]École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
{serena.soldo, mathew, bourlard}@idiap.ch

Abstract

In recent works, the use of phone class-conditional posterior probabilities (posterior features) directly as features has provided successful results in template-based ASR systems. In this paper, motivated by the high quality of current text-to-speech systems and the robustness of posterior features toward undesired variability, we investigate the use of synthetic speech to generate reference templates. The use of synthetic speech in template-based ASR not only allows to address the issue of in-domain data collection but also the expansion of the vocabulary. On 75- and 600-word task-independent and speaker-independent setup of Phonebook corpus, we show the feasibility of this approach by investigating different synthetic voices produced by HTS-based synthesizer trained on two different databases. Our study shows that synthetic speech templates can yield performance comparable to the natural speech templates, especially with synthetic voices that have high intelligibility.

Index Terms: Speech recognition, template-based approach, posterior features, synthetic reference templates.

1. Introduction

In template-based Automatic Speech Recognition (ASR) systems [1], each speech unit (e.g., word) is represented by a set of reference templates. A template typically being a sequence of feature vectors for an utterance of the speech unit. Each test utterance is first transformed into a sequence of short-time spectral-based features and then compared against reference templates, using Dynamic Time Warping algorithm, to find the best match.

Recently, the use of phone class-conditional posterior probabilities estimated by an MultiLayer Perceptron (MLP) directly as speech features has been proposed [2, 3]. We refer to these features as *posterior features*. It was shown that, as a result of the training of the estimator, posterior features are robust to undesired variability and can generalize well, thus yielding significantly better performance than standard spectral-based features using a fewer number of templates.

One of the limitations of template-based ASR lies in the collection of in-domain data as templates. The high quality of the current Text-to-Speech (TTS) systems, together with the property of the MLP to generalize to unseen speech/condition [3], suggests that it could be possible to automatically produce reference templates, and thus build more flexible template-based ASR systems.

Usually, TTS systems are trained using databases recorded in controlled conditions (i.e. sound proof rooms/recording studios) by a small number of professional speakers, and including large amount of phonetically balanced speech data which is manually annotated. As a consequence, the cost to build such corpora is usually very high and only a limited number of voices is available. On the other hand, previous works have shown that speaker-adapted HMM-based speech synthesis systems (HTS) are robust to non-ideal conditions, such as various background noise, different microphones and lack of phonetic balance [4]. This allows to use ASR corpora to train TTS systems, thus providing the possibility of producing a large variety of good-quality synthetic voices [5].

In this paper, we aim at investigating the use of synthetic speech templates for template-based ASR, particularly focussing on the use of synthetic voices trained with different corpora. More precisely, we compare voices trained using TTS corpora versus voices trained with ASR corpora to investigate if and how much the training database influences the ASR system when no adaptation is performed.

Experiments on task-independent and speaker-independent isolated word recognition using Phonebook corpus with a small vocabulary show that synthetic voices can be successfully used as reference templates, provided that the voices have sufficiently good quality. The voices trained on a TTS corpus yield performance slightly higher than the performance obtained using voices trained on an ASR corpus. In addition, we found that average voices trained on the ASR corpus perform better than the voices adapted to a specific speaker.

The paper is organized as follows: we describe the ASR framework using posterior features in Section 2; we introduce and motivate the use of synthetic templates in

Section 3; we present the experimental setup and the results of the experiments in Sections 4 and 5, respectively; finally we conclude the paper with summary and future work in Section 6.

2. Posteriors template-based ASR

Formally, given a spectral-based feature vector, $\mathbf{x}$, and given a set of possible phoneme classes c_k with $k \in \{1, 2, ..., K\}$, the posterior features vector $\mathbf{y}$ is given by $\mathbf{y} = [P(c_1|\mathbf{x}), \ldots, P(c_K|\mathbf{x})]^{\mathrm{T}} = [y_1, \ldots, y_K]^{\mathrm{T}}$. As discrete distribution, the vector $\mathbf{y}$ has two properties: *a)* $y_k \in [0, 1], \forall k \in \{1, 2, ..., K\}$ and *b)* $\sum_{k=1}^{K} y_k = 1$.

In a previous work [3], these features have been investigated in the context of template-based ASR, showing that they generalize well to unseen data and yield better systems than standard spectral-based features.

Figure 1 shows the framework of a template-based ASR system using posterior features.

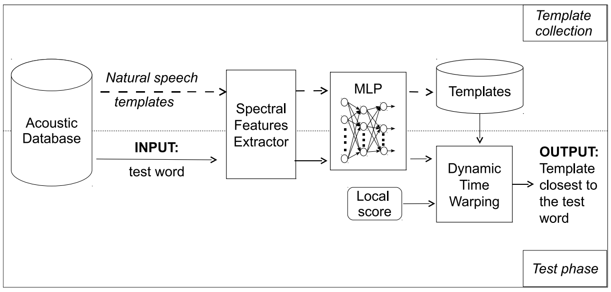


Figure 1: *Framework of a template-based ASR using posterior features*

Features extraction: In this framework, the features are extracted from the speech signal through two steps. First, the speech signal is transformed into a sequence of cepstral-based feature vectors. Then, each vector in the sequence (along with a temporal context) is provided as input to an estimator/classifier and transformed into a posterior features vector. A comparison of different posterior features estimators was performed in [3] and it was shown that, irrespective of the estimator, posterior features always yield better performance than spectral features. Specifically, MLP was found to yield consistently better systems.

Templates collection: In the training phase (dashed arrows in Figure 1), a number of reference templates are extracted from a database in the same domain as the test data. The templates are transformed into a sequence of posterior features and stored in memory.

Test phase: In the test phase (continuous arrows in Figure 1), a test word is first transformed into a sequence of posterior features and then compared to each template using the Dynamic Time Warping (DTW) algorithm. The best matching template is provided as output of the system.

DTW local scores: Taking into account the probabilistic nature of posterior features, DTW algorithm can be redefined using more suitable local distance measures (*local scores*). In previous works [2, 3], local scores such as Bhattacharyya distance, Kullback-Leibler divergence, dot product, cosine angle were found to yield significantly better performance when compared to Euclidean distance. In this paper, we focus in particular on a local score based on Kullback-Leibler divergence and on dot product which were found to be the best performing in terms of accuracy and the best performing in terms of computational effort, respectively. Formally, these local scores are defined as follow:

- *weighted symmetric Kullback-Leibler divergence ($wSKL$)*: In [2, 3], a local score based on Kullback-Leibler divergence, namely weighted symmetric Kullback-Leibler divergence, was found to yield the best performance. Briefly, if $\mathbf{y} = [y_1, \ldots, y_K]^{\mathrm{T}}$ denotes the posterior feature vector that belongs to the reference template and $\mathbf{z} = [z_1, \ldots, z_K]^{\mathrm{T}}$ denotes the posterior feature vector that belongs to the test template then $wSKL$ is computed as:

$$wSKL(\mathbf{y}, \mathbf{z}) = w_{\mathbf{y}} \cdot KL(\mathbf{y}, \mathbf{z}) + w_{\mathbf{z}} \cdot RKL(\mathbf{y}, \mathbf{z}) \tag{1}$$

where,

$$\begin{aligned} KL(\mathbf{y}, \mathbf{z}) &= \sum_{k=1}^{K} y_k \log \frac{y_k}{z_k}, \\ RKL(\mathbf{y}, \mathbf{z}) &= \sum_{k=1}^{K} z_k \log \frac{z_k}{y_k}, \end{aligned}$$

$$w_{\mathbf{y}} = \frac{\frac{1}{H(\mathbf{y})}}{(\frac{1}{H(\mathbf{y})} + \frac{1}{H(\mathbf{z})})}, \quad w_{\mathbf{z}} = \frac{\frac{1}{H(\mathbf{z})}}{(\frac{1}{H(\mathbf{y})} + \frac{1}{H(\mathbf{z})})}$$

$H(\mathbf{y})$ is the entropy of $\mathbf{y}$, and $H(\mathbf{z})$ is the entropy of $\mathbf{z}$.

- *Dot Product ($dotProd$)*: One way to compare two feature vectors is to ask if the two feature vectors belong to the same class. This can be possibly achieved by training an MLP with pair of feature vectors as input and output label as same class or not. In a recent work, it was shown that if such an MLP is trained with mean square error criteria then the optimal output is the dot product (also referred to as scalar product) of the associated pair of input vectors [6]. Another interesting aspect of this measure is that it requires a very low computational effort. Formally, given the posterior features vectors $\mathbf{y}$ and $\mathbf{z}$, then $dotProd$ is computed as:

$$dotProd(\mathbf{y}, \mathbf{z}) = \mathbf{y}^T \mathbf{z} = \sum_{k=1}^{K} y_k \cdot z_k$$

3. Synthetic References, Posterior Features and Template-based ASR

In the framework described in Figure 1, the templates were extracted from data of the same domain as the test data. In this section, we propose a framework that uses templates which are domain-independent and are generated using a TTS system. This would eliminate the issues related to in-domain data collection or vocabulary expansion. The new framework is illustrated in Figure 2.

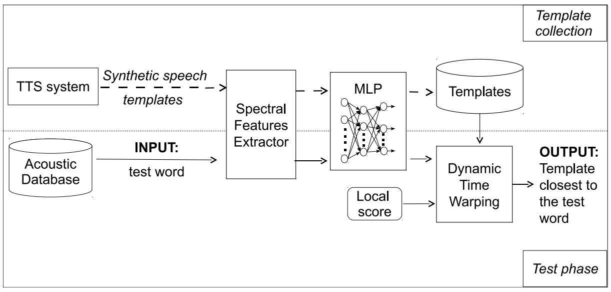


Figure 2: *Framework of the template-based ASR using posterior features and synthetic templates*

The use of synthetic templates has been already suggested in the past. In particular, in [7] the authors proposed a recognition system in which a speech production system (a rule-based TTS system) is used to generate a number of synthetic reference templates that are matched to the input test utterance at spectral level. The authors suggest several adaptation techniques to increase the match between natural and synthetic voices, such as speaker adaptation of the synthetic voice to the current test speaker or length adaptation of the synthetic utterance to the test utterance. However, the results of their experiment were far from competitive with systems based on natural speech templates. The failure of that work was mainly ascribed to the low quality of the voices produced by the rule-based TTS system and, thus, the lack of similarity between synthetic and natural speech.

Recently, new models for TTS has been proposed [8] and the quality of the synthetic voices has considerably increased, being now comparable with natural speech especially in terms of intelligibility [5]. Moreover, the representation of the speech signal in current TTS systems usually includes also information about the spectral envelope of the speech signal [9]. On the other hand, it has been shown that the MLP tends to learn information about the spectral envelope of the speech signal [10]. This suggests that MLPs could estimate reliable posterior features for synthetic speech as well. In addition, the posterior features have been found to be robust to speaker variability, thus dissimilarity between synthetic and natural speech could be effectively reduced when this features are used, possibly eliminating the need for speaker adaptation.

Besides overcoming the problem with data collection, one advantage in the use of synthetic speech templates lies in the possibility of automatically producing variation in the speech. In ASR the use of multi-condition data created, for example, by adding noise or collecting data from different domains has been well explored. The synthetic speech could be possibly used to systematically add variation such as, hyper/hypo articulation, age or accents in the training data. In this direction, the use of synthetic speech could also be exploited to explore the template space.

In the following section, we present the experimental setup and details about the different components of the system.

4. Experimental Setup

We perform speaker-independent task-independent isolated word recognition on Phonebook speech corpus. This corpus contains US English read telephone speech. The test set consists of 8 subsets of utterances, each containing 75 words uttered on average by 11 or 12 speakers once, for a total of 6598 word samples. For more details about the composition of this dataset, the reader may refer to [11].

We perform the experiments on two different tasks:

- 75-word task: the recognition is performed on each of the 8 subset (75-word lexicon each) separately and the average accuracy is presented as result.
- 600-word task: the 8 test subsets are merged to setup a task with 600 words lexicon.

In this work, we use exactly the same framework as in [2, 3], where one random utterance of each word was extracted from the test set and used as natural speech reference template [1]. There are two natural voices, namely, one female (denoted as *natVoice1*) and one male (denoted as *natVoice2*).

Text-To-Speech system

The synthetic reference templates were generated using Festival Speech Synthesis System [12]. We use *off-the-shelf* HMM-based Speech Synthesis System (HTS) voices, trained using the CMU ARCTIC databases [13] or Wall Street Journal (WSJ) database:

- CMU ARCTIC databases consist of phonetically balanced sentences selected from out-of-copyright texts recorded using a microphone in a sound proof room. This databases are specifically designed for speech synthesis. Among the different voices available, we use two US English male voices (*Bdl* and *Rms*) and two US English female voices (*Slt* and *Clb*).

[1] Our experiments show that there is no statistical significant difference in the performance when a different random selection of such utterances is made.

- WSJ database consists of read speech with texts drawn from a machine-readable corpus of Wall Street Journal news text. This database was originally built for large-vocabulary continuous speech recognition systems. In a recent work [5], it was shown that speaker-adaptive HMM-based speech synthesis is robust to non-ideal speech conditions typical of ASR corpora (such as, varying microphones, presence of background noise, lack of phonetic balance). This allows the use of ASR corpora to build good-quality synthetic voices. In the experiments we use four voices: two average voices (one male, named *59males*, and one female, named *60females*), and two synthetic voices corresponding to the male speaker *001* and the female speaker *002*.

For more details about the training system the reader may refer to [14, 15]. In these experiments, each of the synthetic voices has been used to produce one utterance of each word in the dictionary.

Posterior features estimation

We estimate posterior features using the MLP that yielded the best system in a previous work [3]. This MLP was trained with 232 hours of conversational telephone speech. The input to the MLP is a vector of 39-dimensional PLP features (c_0-$c_{12} + \Delta + \Delta\Delta$) along with a temporal context of 90ms. The MLP has 5000 hidden units and 45 output units, each corresponding to a context-independent phoneme.

In the case of synthetic speech, the speech was down sampled from 16 kHz to 8 kHz and posterior features were extracted without performing any kind of adaptation on the MLP.

Local Scores

In this work, we investigate the two local scores described earlier in Section 2. In previous studies [2, 3], $wSKL$, defined earlier in Equation (1), was found to yield the best system, whereas $dotProd$, defined earlier in Equation (2), was found to be faster than $wSKL$ but providing somewhat lower results. We investigate both local scores for natural and synthetic speech.

5. Results and Discussion

In Figures 3a to 3d, we show the results obtained with both natural and synthetic speech templates for 75- and 600-word tasks using $wSKL$ or $dotProd$ as local scores. The performances are expressed in terms of word accuracy. *Natural* denotes the system with natural speech templates, *Synthetic (ARCTIC)* denotes the system with synthetic voices trained on ARCTIC databases, and *Synthetic (WSJ)* denotes the system with synthetic voices trained on WSJ database.

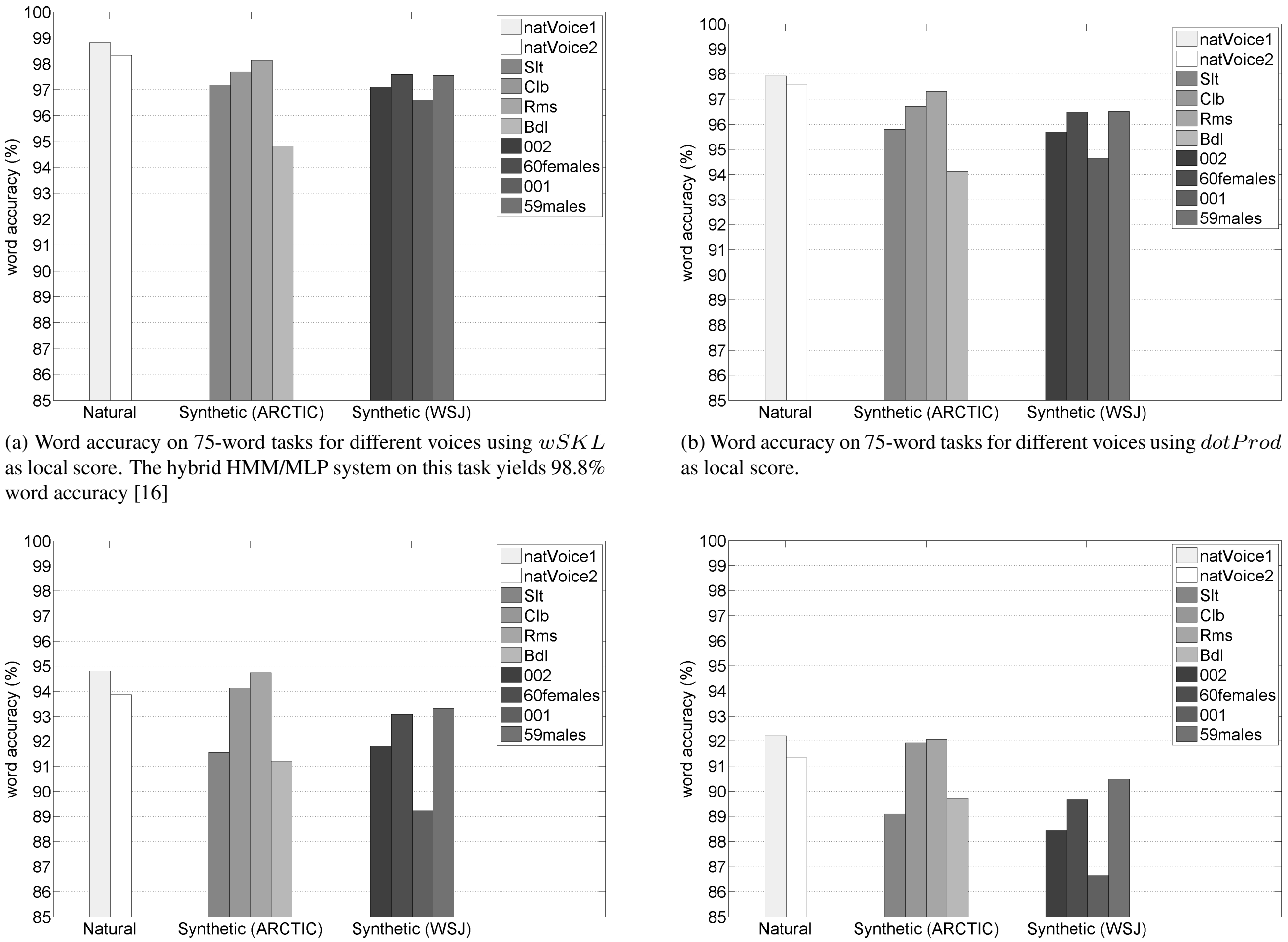
Using $wSKL$ as local score, the best performance obtained using natural templates are 98.8% and 94.8% on 75- and 600-word task respectively. Using ARCTIC synthetic voices, the best performance are 98.2% and 94.7% on 75- and 600-word tasks respectively. Using WSJ synthetic voices, the best performance are 97.6% and 93.3% on 75- and 600-word tasks respectively.

Using $dotProd$ as local score, the best performance obtained using natural templates are 97.9% and 92.2% on 75- and 600-word task respectively. Using ARCTIC synthetic voices, the best performance are 97.3% and 92.1% on 75- and 600-word task respectively. Using WSJ synthetic voices, the best performance are 96.5% and 90.5% on 75- and 600-word task respectively.

Similarly to what observed already on natural data, the results clearly confirm that $wSKL$ provides the best performance for all tasks. While on 75-word task the performance using $dotProd$ is comparable to the use of $wSKL$, on 600-word task the difference in the performance is more evident. However, for all the tasks the real-time factor using $dotProd$ as local score is less than half the real-time factor using $wSKL$.

Despite the differences between the database used in this ASR experiments and those used in the TTS training, it can be observed that the synthetic voices yield performance comparable to the use of natural in-domain voices. In our experiments we observed that synthetic voices exhibit more variabilities in terms of performance compared to the use of natural speech. This could be related to the different quality of the synthetic voices. For example, listening tests revealed that the synthetic voice *001* has a lower quality than the voice *59males* and this corresponds to lower performance in the experiments. Similarly, the results obtained using ARCTIC voices correlate well with the subjective quality evaluations of these voices reported in [17], especially in terms of intelligibility.

An error analysis of the ASR output indicated an inherent limitation that the TTS system could introduce in this ASR approach. Specifically, we found that the pronunciation of some words in the natural speech differ from the pronunciation in the synthesized speech. In particular, we observed two different cases. In one case, the phonetic transcription of the synthesizer does not match with the phonetic transcription of the natural speech. For example, using any of the synthetic voices, the word “gabardines” is transcribed by the TTS as /ˈgæbədaɪːnz/ whereas it is pronounced as /ˈgæbədɪːnz/ in the natural speech. In the other case, the phonetic transcription of the synthesizer corresponds to the phonetic transcription of the natural speech, but the synthesis process would eventually produce a speech sample with a slightly different pronunciation. For example, the word “externalize” is phonetically transcribed by the TTS as /ɪkˈstɜːnəlaɪz/ (corresponding to the pronunciation of the natural speech), but the synthesized speech sounds rather like /ɪkˈstɜːnəlɪz/ when the WSJ voice *001* is used (likely due to the low intelligibility of this voice).



(a) Word accuracy on 75-word tasks for different voices using $wSKL$ as local score. The hybrid HMM/MLP system on this task yields 98.8% word accuracy [16]

(b) Word accuracy on 75-word tasks for different voices using $dotProd$ as local score.

(c) Word accuracy on 600-word tasks for different voices using $wSKL$ as local score. The hybrid HMM/MLP system on this task yields 96.0% word accuracy [16]

(d) Word accuracy on 600-word tasks for different voices using $dotProd$ as local score.

Figure 3: Word accuracy on 75- and 600-word tasks using different voices and local scores.

From the results, it can be observed that ARCTIC voices provide better results than WSJ voices. This is probably related to the difference of the quality of the original training data for the TTS. As already mentioned in Section 4, the ARCTIC databases were recorded in a sound proof room and are phonetically well balanced, whereas the recording condition for WSJ was not consistent and presented a variety of environments and microphones. Concerning the voices trained on WSJ, it is interesting to notice that the average voices perform always better than the voices adapted to a specific speaker. This can be attributed to a quality reduction of synthetic speech that can be observed during the voice adaptation process [5]. This suggests that the use of an average voice might be sufficient in the present speech recognition approach, eliminating the need for speaker adaptation of the synthetic voices.

6. Summary

In this paper, we investigated the use of synthetic references for template-based ASR using posterior features. Our results show that the robustness of posterior features, together with the good quality of current TTS systems, allow to build flexible template-based ASR systems. Our studies show that using synthetic speech, without any adaptation of the posterior features estimator (trained on natural speech), we can achieve results comparable to the use of natural speech templates, both when the TTS system is trained on TTS corpora or ASR corpora. However, the quality of the synthetic speech influence the performance of the ASR system. Moreover, we found that average voices trained on the ASR corpus perform better than the speaker-adapted voices, suggesting that there may be no need to perform speaker adaptation for this task. We intend to further verify this on average voices trained on other databases.

In addition, in our future work we will investigate possible adaptation techniques for the MLP estimator in order to improve the quality of the posterior features also in case of low quality synthetic voices. In particular, we could use the hierarchical MLP-based task-adaptation approach proposed in [16], where a second MLP, trained on top of the first MLP, has the ability to learn confusions present at the output of the first MLP and also learn phonotactic constraints. The second MLP could be trained using only synthetic data or a mix of both natural and synthetic data.

7. Acknowledgements

This work was partly supported by the European Union under the Marie-Curie Training project SCALE, Speech Communication with Adaptive LEarning and the Swiss National Science Foundation under the National Centre of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). The authors would like to thank their colleagues for fruitful discussions, inputs, comments and suggestions. In particular, the authors would like to thank Hui Liang and Cassia Valentini-Botinhao for practical advices on the use of the TTS system.

8. References

[1] L. Rabiner and H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[2] G. Aradilla, H. Bourlard, and M. Magimai.-Doss, "Posterior Features Applied to Speech Recognition Tasks with User-Defined Vocabulary," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.

[3] S. Soldo, M. Magimai.-Doss, J. Pinto, and H. Bourlard, "Posterior Features for Template-based ASR," in *Proc. of ICASSP*, Prague, Czech Republic, 2011.

[4] J. Yamagishi, T. Nose, H. Zen, Z. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive HMM-based text-to-speech synthesis," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 6, pp. 1208–1230, 2009.

[5] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, Y. Guan, R. Hu, K. Oura, Y.-J. Wu, K. Tokuda, R. Karhila, and M. Kurimo, "Thousands of voices for hmm-based speech synthesis: analysis and application of tts systems built on various asr corpora," *Trans. Audio, Speech and Lang. Proc.*, vol. 18, no. 5, pp. 984–1004, Jul. 2010.

[6] B. Picart, "Improved Phone Posterior Estimation Through k-NN and MLP-Based Similarity," Idiap, Tech. Rep. Idiap-RR-18-2009, 2009.

[7] M. Blomberg, R. Carlson, K. O. E. Elenius, B. Granstrm, and S. Hunnicutt, "Word recognition using synthesized templates," Department of Speech Communication and Music Acoustics, KTH, Sweden, Tech. Rep. STL-QPSR 2-3/1988, 1988.

[8] S. King, "An introduction to statistical parametric speech synthesis," *Sādhanā*, vol. 36, no. 5, pp. 837–852, 2011.

[9] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[10] J. Pinto, G. S. V. S. Sivaram, H. Hermansky, and M. Magimai.-Doss, "Volterra series for analyzing mlp based phoneme posterior probability estimator," in *Proc. of ICASSP*, Taipei, Taiwan, 2009.

[11] S. Dupont, H. Bourlard, O. Deroo, V. Fontaine, and J.-M. Boite, "Hybrid HMM/ANN Systems for Training Independent Tasks: Experiments on 'PhoneBook' and Related Improvements," in *Proc. of ICASSP*, Munich, Germany, 1997.

[12] A. Black and K. Lenzo, "Building voices in the festival speech synthesis system," http://festvox.org/bsv, 2000.

[13] J. Kominek and A. W. Black, "CMU Arctic Databases for Seech Synthesis," Language Technologies Institute, Carnegie Mellon University, Tech. Rep. CMU-LTI-03-177, 2003.

[14] J. Yamagishi, H. Zen, Y.-J. Wu, T. Toda, and K. Tokuda, "The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge," in *Proc. of Blizzard Challenge 2008*, 2008.

[15] J. Dines, J. Yamagishi, and S. King, "Measuring the gap between hmm-based asr and tts," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1046–1058, 2010.

[16] J. Pinto, M. Magimai.-Doss, and H. Bourlard, "MLP Based Hierarchical System for Task Adaptation in ASR," in *Proc. of ASRU*, Merano, Italy, 2009.

[17] C. L. Bennett, "Large Scale Evaluation of Corpus-based Synthesizers: Results and Lessons from the Blizzard Challenge 2005," in *Proc. of Interspeech*, Lisbon, Portugal, 2005, pp. 105–108.