INTEGRATED AUTOMATIC EXPRESSION PREDICTION AND SPEECH SYNTHESIS FROM TEXT

Langzhou Chen, Mark J.F. Gales, Norbert Braunschweiler, Masami Akamine, Kate Knill

Toshiba Research Europe Ltd., Cambridge research Lab, Cambridge, UK

langzhou.chen,mark.gales,norbert.braunschweiler@crl.toshiba.co.uk, masa.akamine@toshiba.co.jp

ABSTRACT

Getting a text to speech synthesis (TTS) system to speak lively animated stories like a human is very difficult. To generate expressive speech, the system can be divided into 2 parts: predicting expressive information from text; and synthesizing the speech with a particular expression. Traditionally these blocks have been studied separately. This paper proposes an integrated approach, sharing the expressive synthesis space and training data across the two expressive components. There are several advantages to this approach, including a simplified expression labelling process, support of a continuous expressive synthesis space, and joint training of the expression predictor and speech synthesiser to maximise the likelihood of the TTS system given the training data. Synthesis experiments indicated that the proposed approach generated far more expressive speech than both a neutral TTS and one where the expression was randomly selected. The experimental results also showed the advantage of a continuous expressive synthesis space over a discrete space.

Index Terms— expressive speech synthesis, hidden Markov model, cluster adaptive training, neural network, audiobook

1. INTRODUCTION

Synthesising expressive speech, e.g. to read novels expressively, is a very difficult task for TTS research. There are two key components: prediction of the expression from text; generation of speech with the selected expression. At synthesis, the predicted expression is used as a control input to the synthesiser to extract the correct expression.

The expression prediction task has always been treated as a computational linguistic problem in which the emotions of the text data are detected and classified e.g. happy, sad, angry [1]. Some systems allow the expressions to be manually specified. To handle more arbitrary and larger volume of texts, a number of automatic methods have been proposed including methods based on keywords which are related to the emotions [2], machine learning methods [3, 4, 5] and vector space methods [6].

The generation of expressive synthetic speech depends on the synthesiser. For this paper, only statistical parametric speech synthesis [7] will be considered. In this case the task becomes an acoustic modelling problem. Various approaches have been presented, e.g. model interpolation [8], multiple regression hidden semi-Markov model [9], decision tree based method [10], transform based method [11] and cluster adaptive training (CAT) [12]. Other multi-speaker methods - eigenvoice [13] and factor analysed voice models [14] - could also be adapted to model expressions.

The two components share a common set of pre-defined expressions/emotions. However, they have traditionally been trained independently. Using naturally expressive speech corpora for training the synthesiser, such as audiobooks [15, 16, 17], can yield a far greater range of expressions than training on an acted corpus with a fixed set of expressions such as happy, sad and angry. The classification of expressions in such 'found' speech corpora is non-trivial: a single utterance may contain multiple emotions [18]. This makes expression labelling time consuming and difficult to do consistently across labellers. Szekely et al [19] proposed an approach to automatically detect the presence of specific emotions in audiobooks. Emotion detection and acoustic modelling for a pre-defined set of emotions cannot cover the diversity of very expressive data such as audiobooks. To achieve diversity inexpensively automatic methods have been introduced to cluster the audiobook data into groups of similar expressions [20, 17]. However, since automatic data clustering makes no assumptions about the emotion labels of each expression state this makes the emotion prediction from text task even more difficult.

In this work, the expression predictor and an expressive statistical parametric speech synthesiser were investigated as a single unit at both the training and synthesis stages. An expressive synthesis space was constructed to provide expressive information to the synthesiser. The expression prediction from text was designed as a process to map linguistic features derived from the text to points in the same expressive synthesis space. A multi-layer perceptron (MLP) neural network (NN) was used to perform the mapping. Rich naturally expressive training data, taken from audiobooks, was shared across the two components.

The proposed method can alleviate the drawbacks of traditional methods. Firstly, it learns the mapping between linguistic space and synthesis space by NN, without caring how the expressions in synthesis space are labelled. This means that the expression states from automatic clustering can be used so the data preparation required can be significantly reduced. The proposed method also does not care what form the expressive information takes in synthesis space. It can be expression states from a discrete space, but also can be points in a continuous space. Therefore, the proposed method can potentially synthesise the speech with more detailed expressions using a continuous expressive synthesis space [12]. Finally, the expression predictor and speech synthesiser can be jointly optimised since they are linked together. Maximum likelihood optimisation of the whole process is possible.

2. INTEGRATING EXPRESSION PREDICTION AND SPEECH SYNTHESIS

This work considers an integrated expression prediction from text and speech synthesis process. In the proposed method, the two components are linked together by sharing an expressive synthesis space, as shown in figure 1. The expression synthesis space defines the set of expressions which can be processed by synthesiser. Meanwhile, it also defines the output of the expression predictor. In this work, the

expression prediction was designed as a MLP based mapping process. Given the text data, a linguistic feature vector was generated to represent the expressive information of this text. Then the MLP mapped this point in the linguistic feature space to a point in the expressive synthesis space. If the expressive synthesis space is discrete, as shown in figure 1, this mapping is equivalent to a classification process to determine which expression state should be assigned to the text data. Then, the selected expression state is used to synthesise the speech with the same expression. Furthermore, the expression predictor and speech synthesiser not only share the expressive synthesis space, but also share the training data. This is a big difference between traditional methods and the proposed method. In traditional methods, the emotion detector is trained purely on text data and the speech synthesiser is trained on speech data.

The sharing of the expressive synthesis space and training data allows the expressive synthesis space to be arbitrarily defined. In this case, the definition of expression states only influences the supervision of the training samples. It does not influence the process flow and the cost of system building. In traditional methods modifying the definition of expression states in one component will dramatically influence the other component.

In [12] it was shown that a TTS system based on a continuous expressive synthesis space can model much richer expressive information than discrete space. It is very hard to use continuous expressive synthesis spaces in traditional methods since an infinite number of different expressions need to be identified. Another advantage of the proposed method is that a continuous expressive synthesis space can be integrated.

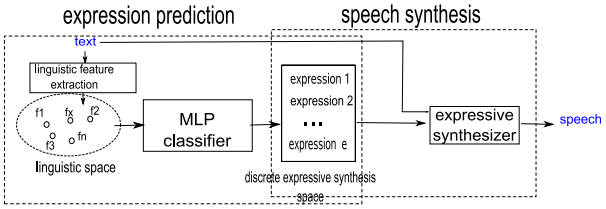

Fig. 1. *Expressive synthesis with discrete space*

Continuous expressive synthesis space assumes that each speech utterance contains unique expressive information. This unique expressive information can be heard from the speech data and can be read from the speech transcript, i.e. the text data as well. The expressive information in speech and in its transcripts is synchronised. Thus, for every point in expressive synthesis space, there is a point in expressive linguistic space which corresponds to it, and vice versa. Based on this fact, in the proposed method, the expression predictor based on a continuous space is equivalent to a non-linear transformation between the expressive linguistic space and the expressive synthesis space rather than an expression classification process. Again, a MLP was used to build this non-linear transformation, as shown in figure 2.

3. EXPRESSIVE SPEECH SYNTHESIS

An expressive synthesis space consists of all possible expressions which can be processed by the synthesiser. It can be discrete with a fixed number of expressions e.g. [17], or continuous which contains an infinite number of expressions, as proposed in [12].

To construct the discrete expressive synthesis space, the training speech utterances need to be grouped into a discrete set of expression

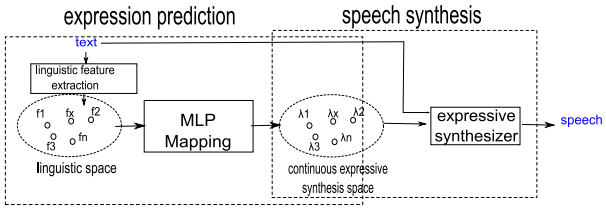

Fig. 2. *Expressive synthesis with continuous space*

states by manual labelling or automatic clustering. The expression state information can be modelled by various methods, e.g. decision tree based method [10], AESS method [17, 21] etc.

In the CAT method [22], the expression states can be modelled using CAT weight vectors. When a CAT model is used to calculate the likelihood of an observation vector, the mean vector to be used is a linear interpolation of all the cluster means, i.e.

$$p(\boldsymbol{o}_t|\boldsymbol{\lambda}^{(e)}, \mathbf{M}^{(m)}, \boldsymbol{\Sigma}^{(m)}) = \mathcal{N}(\boldsymbol{o}_t; \mathbf{M}^{(m)}\boldsymbol{\lambda}^{(e)}, \boldsymbol{\Sigma}^{(m)}) \quad (1)$$

where $\mathbf{M}^{(m)}$ is the matrix of P cluster mean vectors for component m,

$$\mathbf{M}^{(m)} = \left[\boldsymbol{\mu}^{(m,1)} \quad ... \quad \boldsymbol{\mu}^{(m,P)}\right] \quad (2)$$

$\boldsymbol{\lambda}^{(e)}$ is the CAT weight vector for expressive state e. It is simple to extend this form of representation to include multiple regression classes with each of the expressive states. In common with standard CAT approaches the first cluster is specified as a bias cluster, thus

$$\boldsymbol{\lambda}^{(e)} = \left[1 \quad \lambda_2^{(e)} \quad ... \quad \lambda_P^{(e)}\right]^{\mathsf{T}} \quad (3)$$

To construct an expressive space, CAT cluster models can be viewed as a basis of expressive synthesis parameters. The synthesis parameters with different expressions can be projected into this basis, while the CAT weights are the coordinates of this projection. Thus, based on the CAT method, the synthesis parameters for each expression is represented as a unique CAT weight vector.

In [12], experiments showed that CAT method generated significantly more expressive speech than the decision tree and AESS method. Therefore in this work the CAT method was used to construct the expressive synthesis space.

In a discrete expressive synthesis space, all the training data with the same expression state shares the CAT weights. By contrast, in a continuous synthesis space, each utterance is represented as an individual point. Therefore to construct the continuous expressive space, the sufficient statistics for CAT weight training should be calculated for each utterance individually and the prior information from discrete space can be used to smooth the CAT weights in continuous space [12].

4. SPEECH SYNTHESIS EXPRESSION PREDICTION

4.1. Expressive linguistic feature space

In this work, the bag-of-words (BoW) method was used to convert the text data into linguistic features. Latent semantic mapping (LSM) was used to compress the dimension of the feature vectors. Based on LSM technology, [6] presented a latent affective mapping method for emotion classification. In this work, the emotion classification was skipped. A linguistic feature vector which contains the

expression information of the text data is required, while mapping this vector to the expressive synthesis space is performed by a MLP.

Similar to [6], a LSM space which encapsulated the domain information was constructed using 50K paragraphs from the transcripts of 60 audiobooks. The size of vocabulary was 30K. Then each utterance which contained expressive information was projected into this domain LSM space as an individual document vector.

To introduce intra-utterance context information into the feature vectors, 3 types of frequency information were used, including word frequency $p(w)$, word pair frequency $p(w_1w_2)$ and word frequency with part-of-speech (POS) context $p(pos_1w_2pos_3)$. In addition, to introduce the inter-utterance context information, the vector of one utterance was glued with the vectors from its left and right neighbours to form the final expressive linguistic features.

Not only the word level knowledge, but e.g. the knowledge of different levels such as narration styles, full context phone sequences can be added into the linguistic feature as well.

4.2. Mapping from linguistic space to synthesis space

The expression prediction process in this work is mapping a linguistic feature vector to a point in expressive synthesis space. A MLP was used to do this mapping. For each expressive speech utterance in the training data, the transcripts were converted into a vector in expressive linguistic space. These linguistic vectors are used as input to the MLP. Based on the forms of the expressive synthesis space, i.e. discrete or continuous, the outputs of the MLP are different. For the discrete space, since the expressions are represented by a discrete set of expression states and each training utterance belonged to one of these states, the output of the MLP was designed as one/zero values corresponding to the expression state assignment to each utterance. The dimension of the MLP output layer is set equal to the number of expression states. With softmax as output layer activation function, the learning criterion for MLP with a discrete space is minimum cross-entropy, i.e.

$$e(\mathbf{W}) = -\sum_k \sum_j t^{(kj)} \log y^{(kj)}, \tag{4}$$

$$\hat{\mathbf{W}}^i = \mathbf{W}^i - \eta \frac{\partial e(\mathbf{W})}{\partial \mathbf{W}^i}, \quad i = 1...L \tag{5}$$

where $t^{(kj)}$ and $y^{(kj)}$ are the j^{th} value of target distribution and MLP output for training sample k respectively. $\mathbf{W}^i$ is the weight matrix of layer i and $\mathbf{W} = \{\mathbf{W}^1, \quad ... \quad, \mathbf{W}^L\}$ is the set of weight matrices.

For a continuous expressive synthesis space, the MLP directly outputs the CAT weight vector for a particular expression information. With maximum likelihood (ML) based supervised adaptation, each expressive speech utterance in the training data was projected into expressive synthesis space as a CAT weight vector which contains the expression of this utterance. The CAT weight vectors from training utterances were used as the target output to train the MLP.

Two types of training criteria were investigated in this work. The first one is the traditional least squared error (LSE) criterion, i.e.

$$e(\mathbf{W}) = \frac{1}{2} \sum_k \|\boldsymbol{\lambda}^{(k)} - \bar{\boldsymbol{\lambda}}^{(k)}\|^2 \tag{6}$$

where $\boldsymbol{\lambda}^{(k)}$ and $\bar{\boldsymbol{\lambda}}^{(k)}$ are the supervised trained CAT weight vector and MLP output CAT weight vector for training sample k respectively. The construction of the expressive synthesis space was based on the ML criterion instead. LSE training minimises the squared errors between target vectors and MLP outputs, but does not guarantee the output CAT weights from the MLP maximise the likelihood of training data. This work presented an alternative MLP training method based on ML criterion. In the new method, the cost function of MLP training is designed as the negative of the auxiliary function for ML based CAT weight training, i.e.

$$e(\mathbf{W}) = -\sum_k \frac{1}{|T_k|} (\bar{\boldsymbol{\lambda}}^{(k)\mathsf{T}} \mathbf{k}^{(k)} - \frac{1}{2} \bar{\boldsymbol{\lambda}}^{(k)\mathsf{T}} \mathbf{G}^{(k)} \bar{\boldsymbol{\lambda}}^{(k)}) \tag{7}$$

In equation 7, the cost from utterance k is normalised by the length of this utterance $|T_k|$, so that the contribution of each utterance is equal. $\mathbf{G}^{(k)}$ and $\mathbf{k}^{(k)}$ are the sufficient statistics for CAT weight training accumulated from utterance k which can be calculated as

$$\mathbf{G}^{(k)} = \sum_{m,t \in T_k} \gamma_t^{(m)} \mathbf{M}^{(m)\mathsf{T}} \boldsymbol{\Sigma}^{(m)\text{-}1} \mathbf{M}^{(m)} \tag{8}$$

$$\mathbf{k}^{(k)} = \sum_m \mathbf{M}^{(m)\mathsf{T}} \boldsymbol{\Sigma}^{(m)\text{-}1} \sum_{t \in T_k} \gamma_t^{(m)} (\boldsymbol{o}_t - \boldsymbol{\mu}^{(m,1)}) \tag{9}$$

The derivative of the cost function $e(\mathbf{W})$ w.r.t. the MLP output $\bar{\boldsymbol{\lambda}}^{(k)}$ can be calculated as:

$$\frac{\partial e(\mathbf{W})}{\partial \bar{\boldsymbol{\lambda}}^{(k)}} = -\frac{1}{|T_k|} (\mathbf{k}^{(k)} - \mathbf{G}^{(k)} \bar{\boldsymbol{\lambda}}^{(k)}) \tag{10}$$

Using equation 10 and a standard back-propagation algorithm, the MLP weight matrices $\mathbf{W}$ can be updated.

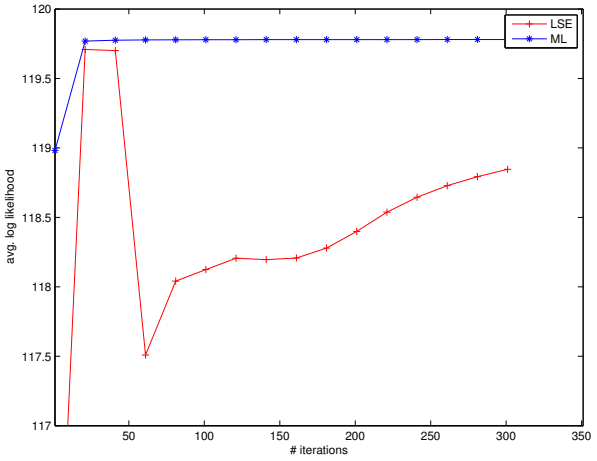

Fig. 3. *avg. log-likelihood, training data*

Figure 3 shows the change of average log-likelihood of the training data with respect to the iterations of NN training. The ML training used the LSE trained NN as its initial point. It can be seen that based on the LSE criterion, the likelihood of the training data did not increase monotonically with more training iterations. On the other hand, in ML based NN training the log-likelihood of the training data increased monotonically with more training iterations. This is due to the consistence of the optimisation between the MLP and the expressive synthesis space.

4.3. Joint optimisation of expression prediction and expressive synthesis space

The proposed method views the expression prediction process and the speech synthesis process as a single process. It makes the joint optimisation of the two components possible. In the proposed method, the MLP parameters $\mathbf{W}$ for expression prediction and the CAT cluster model $\mathcal{M}$ for expressive synthesis space building can be jointly trained with the ML criterion as follows.

1. Initial CAT model training with ML criterion, to generate $\mathcal{M}_0$ and $\mathbf{\Lambda_0}$, which represent cluster models and CAT weights respectively, set $i = 0$
2. accumulate statistics $\{\mathbf{G}_i^{(k)}, \mathbf{k}_i^{(k)}\}, \quad k = 1...K$ for each training utterance using equation 8 and 9.
3. Based on $\{\mathbf{G}_i^{(k)}, \mathbf{k}_i^{(k)}\}, \quad k = 1...K$, train the MLP $\mathbf{W}_i$ with equation 7 as cost function.
4. Generating CAT weights $\bar{\mathbf{\Lambda}}_i$ for training utterances from the output of MLP $\mathbf{W}_i$
5. $\mathbf{\Lambda_{i+1}} = \bar{\mathbf{\Lambda}}_i$, using $\mathbf{\Lambda_{i+1}}$ as input CAT weights, update the CAT cluster models to get $\mathcal{M}_{i+1}$
6. $i = i + 1$, goto 2 until convergence.

In this work, due to the limitation of calculation resources, only 1 iteration of joint training was performed. Based on the MLP output CAT weights, the joint training process significantly increased the likelihood of both training and test data.

5. EXPERIMENTAL RESULTS

The experiments in this work were based on the data of 4 audiobooks from Librivox.org read by John Greenman. The CAT model was trained on data from "A Tramp Abroad". This book contains 56 chapters, which were divided into 51 chapters for CAT model training and 5 chapters for evaluation. The CAT model training data includes 4.8k utterances. The average length of a training utterance is 6.8 seconds. The CAT model used in this work was the same as that used in [12]. It comprised five clusters, one bias cluster and four non-bias clusters. Given the CAT model, the speech utterances from 3 other audiobooks were projected into this "A Tramp Abroad" CAT weight space. Then, the utterances from all 4 audiobooks were used to train the MLP which builds the connection between the linguistic space and the expressive synthesis space. Since the calculation cost of CAT model training is expensive while projecting an utterance into an existing CAT weight space is cheap, this method provides a quick way to generate large amounts of MLP training samples. In this work, the MLP training data consisted of 10.3K utterances.

The synthesised speech data was evaluated by two types of listening test; utterance level ABX test and paragraph level preference test. The ABX test was based on 75 utterances from 5 test chapters of "A Tramp Abroad". The paragraph reading test was based on 15 test paragraphs. The average length of a paragraph was 3 utterances. In the preference test, the listeners were asked to indicate which of two English speech files expressed an appropriate emotion for the content of the paragraph.

The proposed method was investigated in both the continuous and the discrete expressive synthesis spaces. For the discrete space, the number of expression states was 20.

The first experiment was the ABX test. The proposed method was compared to a random CAT weights selection system. For the discrete expressive space, the random CAT weights selection system randomly selected the CAT weights from 20 expression states. For the continuous expressive space, the narration style information was used to guide the CAT weights selection, e.g. if the text to be synthesised was a direct speech, the system randomly selected a direct speech utterance from the training set, and used its CAT weights to do the synthesis. Table 1 gives the ABX test results. It indicates that the proposed approach achieved significantly better performance than the random selection method. In addition, table 1 showed that based on the proposed method, speech generated from continuous expressive space was more expressive than that from discrete space. The reason is that continuous expressive space can be used to model very detailed expressive information in speech, while in discrete space, the detailed information was smoothed. This is consistent with the supervised adaptation results presented in [12].

Table 1. ABX test, utterance reading

disc. rand.	disc. MLP	cont. rand.	cont. MLP	p
42.8%	57.2%			<0.001
		45.7%	54.3%	0.017
	46.8%		53.2%	0.057

In the second experiment, the proposed method was investigated on a paragraph reading task. It was compared to 2 baseline systems, randomly selected CAT weights and fixed CAT weights for neutral speech. A preference test was carried out for evaluation. The results are shown in table 2.

Table 2. Preference test, paragraph reading

neutral	disc. rand.	disc. MLP	cont. rand.	cont. MLP	no prefer	p
35.7%		50.2%			6.2%	<0.001
28.2%				69.1%	2.7%	<0.001
	41.5%	50.8%			7.7%	0.048
			41.6%	52.6%	5.8%	0.027
		42.3%		50.6%	7.1%	0.071

In the paragraph reading task, the proposed method again achieved the best performance in both continuous and discrete spaces. Consistent with the ABX test results, the proposed method based on a continuous expressive space achieved better performance than a discrete space.

6. CONCLUSIONS

This work presented a method to integrate the expression predictor and speech synthesiser to automatically generate expressive speech from arbitrary text. The proposed method alleviates the drawbacks of the traditional methods, so data preparation and labelling work can be significant reduced. In addition, the proposed method also supports continuous expressive synthesis spaces which can model the far richer expressive information found in human speech. Based on the proposed method, the joint optimisation of the expression extraction process and speech synthesis process can be performed. Experimental results showed that proposed method generated more expressive speech than a random CAT weight selection method and a neutral speech synthesiser. It also confirmed the advantage of continuous expressive synthesis space over a discrete space.

7. REFERENCES

[1] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: affective text," in *Proc. of 4th International Workshop on Semantic Evaluations*, 2007.

[2] C. Strapparava and A. Valitutti, "Wordnet-affect: an affective extension of wordnet," in *Proc. of LREC*, 2004.

[3] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text," in *Proc. of 2008 ACM Symposium on Applied Computing*, 2008.

[4] D. Das and S. Bandyopadhyay, "Sentence level emotion tagging," in *Proc. of Affective Computation and Intelligent Interaction and Workshops*, 2009.

[5] C. Ovesdotter Alm, D. Roth, and R. Sproat, "Emotion from text: machine learning for text-based emotion prediction," in *Proc. of Conf. HLT-EMNLP*, 2005.

[6] J.R. Bellegarda, "Further analysis of latent affective mapping for naturally expressive speech synthesis," in *Proc. of ICASSP*, 2011.

[7] H. Zen, K Tokuda, and A.W. Black, "Statistical parametric speech synthesis," *Speech communication*, vol. 51, pp. 1039–1154, 2009.

[8] M. Tachibana, J. Yamagishi, T. Masuko, and T.Kobayashi, "Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing," *IEICE Trans. on information and systems*, vol. 88, no. 11, pp. 2484–2491, 2005.

[9] T. Nose, J. Yamagishi, and T. Kobayashi, "A style control technique for speech synthesis using multiple regression hsmm," in *Proc. of Interspeech*, 2006.

[10] J. Yamagishi, K. Onishi, T. Masuko, and T.Kobayashi, "Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis," *IEICE Trans. on information and systems*, vol. E88-D, pp. 503–509, 2005.

[11] J. Yamagishi, T.Kobayashi, M.Tachibana, K. Ogata, and Y. Nakano, "Model adaptation approach to speech synthesis with diverse voices and styles," in *Proc. of ICASSP*, 2007.

[12] L. Chen, M.J.F. Gales, V. Wan, J. Latorre, and M. Akamine, "Exploring rich expressive information from audiobook data using cluster adaptive training," in *Proc. of INTERSPEECH*, 2012.

[13] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokura, T. Masuko, T.Kobayashi, , and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. of ICSLP*, 2002.

[14] K. Kazumi, Y. Nankaku, and K. Tokura, "Factor analyzed voice models for HMM-based speech synthesis," in *Proc. of ICASSP*, 2010.

[15] Y. Zhao, D. Peng, L. Wang, M. Chu, y. Chen, P. Yu, and J. Guo, "Constructing stylistic synthesis databases from audio books," in *Proc. of Interspeech*, 2006.

[16] K. Prahallad, A. Toth, and A. Black, "Automatic building of synthetic voices from large multi-paragraph speech databases," in *Proc. of Interspeech*, 2007.

[17] F. Eyben, S. Buchholz, N. Braunschweiler, J. Latorre, V. Wan, M.J.F. Gales, and K. Knill, "Unsupervised clustering of emotion and voice styles for expressive TTS," in *Proc. of ICASSP*, 2012.

[18] D. Das and S. Bandyopadhyay, "Labeling emotion in bengali blog corpus - a fine grained tagging at sentence level," in *Proc. of the 8th workshop on Asian language resources, COLING-2010*, 2010.

[19] E. Szekely, J. Kane, S. Scherer, C. Gobl, and J. Carson-Berndsen, "Detecting a targeted voice style in an audiobook using voice quality features," in *Proc. of ICASSP*, 2012.

[20] E. Szekely, J. Cabral, P. Cahill, and J. Carson-Berndsen, "Clustering expressive speech styles in audiobooks using glottal source parameters," in *Proc. of Interspeech*, 2011.

[21] J. Yamagishi, T.Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation method," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, pp. 66–83, 2009.

[22] H. Zen, N. Braunschweiler, S. Buchholz, M. J. F. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE Trans. on Audio Speech and Language Processing*, vol. 20, no. 5, 2012.