SPEAKER DEPENDENT EXPRESSION PREDICTOR FROM TEXT: EXPRESSIVENESS AND TRANSPLANTATION

Langzhou Chen, Norbert Braunschweiler, Mark J.F. Gales

Toshiba Research Europe Ltd., Speech Technology Group, Cambridge, UK
langzhou.chen,norbert.braunschweiler@crl.toshiba.co.uk, mjfg@eng.cam.ac.uk

ABSTRACT

Automatically generating expressive speech from plain text is an important research topic in speech synthesis. Given the same text, different speakers may interpret it and read it in very different ways. This implies that expression prediction from text is a speaker dependent task. Previous work presented an integrated method for expression prediction and speech synthesis which can be used to model the diverse expressions in human's speech and build speaker dependent expression predictors from text. This work extends the integrated method for expression prediction and speech synthesis into a framework for speaker and expression factorization. The expressions generated by the speaker dependent expression predictors can be represented in a shared expression space, and in this space the expressions can be transplanted between different speakers. The experimental results indicate that based on the proposed method, the expressiveness of the synthetic speech can be improved for different speakers. Furthermore this work also shows how important the speaker specific information is for the performance of the expression predictor from text.

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

1. INTRODUCTION

Synthesising expressive speech is a very important task in text-to-speech research. Usually, it can be divided into two components: expression prediction from text and expressive speech synthesis. In the traditional methods, the expression prediction from text is treated as a computational linguistic problem in which one of a predefined set of emotions associated with the text is selected [1]. This implies that for the same text data, the emotion prediction result is fixed without considering any intra-speaker and inter-speaker variabilities. Because all the intra-speaker and inter-speaker variabilities are ignored, usually, the traditional expression prediction from text can only deal with a small number of discrete expression states, i.e. 5 – 10 emotions, to reduce the ambiguity and the inter-annotator disagreement over different people. However, since the questions of how to interpret the text and how to convert it to expressions in speech are strongly dependent on the speaker's background, education, skill, etc., the inter-speaker variability is a factor that needs to be considered in expression prediction. Additionally, a speaker may repeat the same text in different ways. Thus the intra-speaker variabilities influence the expression prediction results as well.

In [2], a method of integrating the expression prediction from text and speech synthesis as a single system was presented. In this method, the two modules were integrated by sharing the expression space and the training data. This way, the continuous expression space which can represent infinite number of expressions is supported and more detailed expressive information can be modelled. A major difference between the integrated method and the traditional methods is that in the integrated method, both the expression predictor and speech synthesiser are trained using the speech data which can be shared by two modules. Since the training speech is speaker dependent, the speaker dependent expression predictor can be trained. Therefore, although the intra-speaker factor of expression prediction from text still cannot be investigated due to the plain text input, it is possible to investigate the inter-speaker factor of the expression predictor in the framework of the integrated method.

Although the integrated method presented in [2] can be used to train a speaker dependent expression predictor from text, the generated expressions can only be used to synthesise the speech of the training speaker. The generated expressions cannot be transplanted and it is also non-trivial to compare expressions from different speakers. The major reason is that the expression factors and the speaker factor are not distinguished from each other because a single transform contains both types of information. The methods of speaker and expression factorization (SEF) can be used to address this problem. Factorization techniques have been widely used in both TTS [3, 4, 5] and ASR [6, 7, 8]. The basic idea of factorization methods is training the orthogonal or independent transforms for different acoustic factors, e.g. speaker, expression, noise environment, etc. The orthogonality of the transforms from different factors allows the transforms from one factor to be transplanted to other factors directly. This way, the number of transforms that need to be trained to cover the various acoustic characteristics is much reduced and the transforms for new acoustic characteristics can be generated by simply composing the transforms from different factors. There are 3 types of methods to realise the orthogonality of the transforms from different factors, i.e. adding the implicit constraints either to the training data [4, 5] or to the type of transforms [6] and adding the explicit independence constraint to parameter optimization [7].

The current work combines the integrated method for expression prediction and synthesis presented in [2] with an SEF framework based on cluster adaptive training [4]. Based on this method, the expressions generated by the expression predictor of different speakers are represented in a common expression space. Thus the expressions generated by one expression predictor can be transplanted to different speakers. This dramatically widens the utilization of the generated expressions. The SEF method allows the speech data from different speakers to be projected into the common expression space, thus a speaker independent expression predictor can be trained using multi-speaker training data. Since in speaker independent expression predictors, the inter-speaker variabilities are assumed to be normalized, the impact of the speaker specific information on the expression prediction performance can be tested by comparing the speaker dependent and independent expression predictors.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2. INTEGRATED METHOD FOR EXPRESSIVE SPEECH SYNTHESIS FROM TEXT

Traditionally, expressive speech synthesis from text is usually seen as two distinct tasks: expression prediction from text and speech synthesis given an expression. Expression prediction from text was investigated as a computational linguistic problem while speech synthesis was considered as an acoustic modelling problem.

In [2], a joint method was presented which combined the expression prediction and speech synthesis into a single task, which is shown in figure 1.

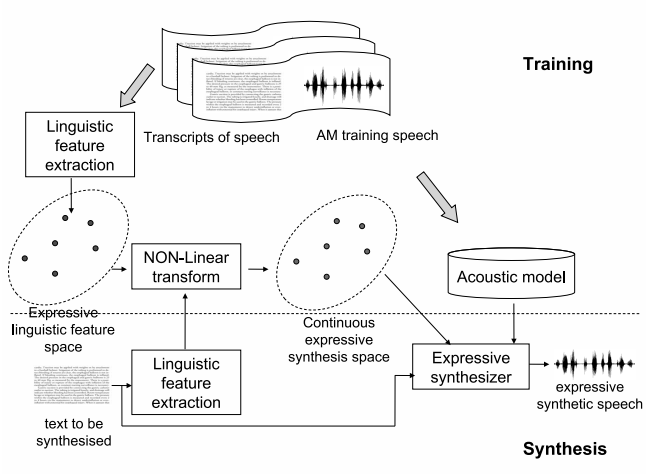

Fig. 1. *Integrated method for expressive TTS*

In this method, an expressive synthesis space is defined to contain all the expressions which can be generated by the synthesiser. This space can be discrete, i.e. containing only a limited number of expression states or it can be continuous, i.e. containing an infinite number of different expressions. While the expressive information in the text data forms a linguistic feature space in which every point represents the expressive information in a phrase. The task of expression prediction from text in [2] was designed as mapping the feature vectors in the linguistic feature space to the points in the expressive synthesis space through a non-linear transform. An MLP based neural network was used to build the non-linear transform. During the synthesis process, the text to be synthesised is converted as a point in the linguistic feature space, then the MLP maps it to a point in the expressive synthesis space. Since every point in the expressive synthesis space represents an expression that can be generated by the synthesiser, expressive synthetic speech can be generated.

2.1. CAT model and expressive synthesis space

Although in [2], the expressive synthesis space can be generated by various methods, in this work, the CAT model (Cluster Adaptive Training) was used to construct the expressive synthesis space. 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 synthesis 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 are represented as a unique CAT weight vector and it forms a point (or a state) in the expressive synthesis space.

2.2. Expression prediction from text

In this work, the linguistic feature vector which contains the expressive information in the text data was generated by the latent semantic mapping (LSM) method. The details can be found in [9]. Given the linguistic features, the task of expression prediction is building an MLP based non-linear transform f to map the linguistic feature vectors $\mathcal{L}$ to the expression vectors $\bar{\boldsymbol{\Lambda}}$ in the synthesis space, i.e.

$$\bar{\boldsymbol{\Lambda}} = f(\mathcal{L}, \mathbf{W}) \quad (4)$$

where $\mathbf{W}$ is the weight matrices of MLP.

To build the connection between the linguistic feature space and the expessive synthesis space, the input of the MLP was designed as the linguistic features extracted from the transcripts of the training utterance, while the output of the MLP was designed as the CAT weight vectors which contain the expression information of the speech utterances. The maximum likelihood (ML) criterion was used to train the MLP. Based on the standard EM algorithm, the cost function of MLP training was designed as the negative of the auxiliary function for CAT weight vector training, i.e.

$$e(\mathbf{W}) = -\sum_k \frac{1}{|T_k|}(\bar{\boldsymbol{\lambda}}^{(k)\mathsf{T}}\mathbf{k}^{(k)} - \frac{1}{2}\bar{\boldsymbol{\lambda}}^{(k)\mathsf{T}}\mathbf{G}^{(k)}\bar{\boldsymbol{\lambda}}^{(k)}) \quad (5)$$

$$\hat{\mathbf{W}}^i = \mathbf{W}^i - \eta\frac{\partial e(\mathbf{W})}{\partial \mathbf{W}^i}, \quad i = 1...L \quad (6)$$

where $\mathbf{W}^i$ is the weight matrix of layer i and $\mathbf{W} = \{\mathbf{W}^1, ..., \mathbf{W}^L\}$ is the set of weight matrices, $\bar{\boldsymbol{\lambda}}^{(k)}$ is the MLP output CAT weight vector for training sample k. $\mathbf{G}^{(k)}$ and $\mathbf{k}^{(k)}$ are the sufficient statistics for CAT weight training accumulated from utterance k which can be calculated as

$$\mathbf{G}^{(k)} = \sum_{m,t\in T_k} \gamma_t^{(m)}\mathbf{M}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\mathbf{M}^{(m)} \quad (7)$$

$$\mathbf{k}^{(k)} = \sum_m \mathbf{M}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\sum_{t\in T_k}\gamma_t^{(m)}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m,1)}) \quad (8)$$

In equation 6, the cost from utterance k is normalised by the length of this utterance $|T_k|$, so that the contribution of each utterance is equal.

From the framework mentioned above, the linguistic features $\mathcal{L}$ contain only the information from plain text, i.e. the input of the TTS system. Although the real speaker can repeat the same text in different speaking styles, for the TTS system from plain text, this part of information is lost. Thus the intra-speaker variability is impossible to be modelled in a TTS system with plain text input. For

the traditional methods which treat the expression prediction as a computational linguistic problem, the inter-speaker factors cannot be modelled either, since the text data is speaker independent. However, in the method mentioned above, the MLP based expression predictor is trained using the speech training data. Therefore if the speaker dependent training samples are used to train the MLP, what it learns is a speaker dependent way to convert the expressions encoded in the text data into speech. This makes the investigation of the inter-speaker factors of the expression prediction possible.

3. SPEAKER AND EXPRESSION FACTORIZATION

Factorization techniques have been widely used in both TTS and ASR. The basic idea is training the orthogonal or independent transforms for each individual acoustic factor. For speaker and expression factorization (SEF), the orthogonal speaker transforms and expression transforms are trained individually. The speaker tranforms can be arbitrarily combined to the expression transforms to fit the acoustic environment with paricular speaker and expression. The CAT model can be used to implement the SEF. CAT is a sub-space based method. Using CAT to model a single factor, e.g. expression, a subspace for expression is built and the very big dimensional expressive synthesis parameters are represented as a point in the low dimensional expression subspace. In the case of SEF, two subspaces have to be built, one for expression, the other for speaker [4]. Thus equation 1 can be re-written as

$$p(\boldsymbol{o}_t|\boldsymbol{\lambda}_{\mathrm{S,E}}, \mathbf{M}_{\mathrm{S,E}}^{(m)}, \boldsymbol{\Sigma}^{(m)}) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}^{(m,1)} + \mathbf{M}_{\mathrm{S}}^{(m)}\boldsymbol{\lambda}_{\mathrm{S}} + \mathbf{M}_{\mathrm{E}}^{(m)}\boldsymbol{\lambda}_{\mathrm{E}}, \boldsymbol{\Sigma}^{(m)}) \quad (9)$$

where $\boldsymbol{\lambda}_{\mathrm{E}}$ and $\boldsymbol{\lambda}_{\mathrm{E}}$ are the CAT weight vectors to model the expression and speaker respectively, and $\mathbf{M}_{\mathrm{E}}^{(m)}$ and $\mathbf{M}_{\mathrm{S}}^{(m)}$ are the cluster mean matrices for component m which are associated to the expression CAT weight vector and speaker CAT weight vector respectively.

The parameter estimation of SEF is based on ML criterion. The overlaps of the speakers and the expressions in the training data are required to ensure the orthogonality of the speaker and the expression transforms. That says, the training data for an expression should be from various speakers, so that the final trained expression transform is independent of any particular speaker. Similarly, the speaker transforms should be trained by the data with various expressions. The speaker parameters and the expression parameters are updated alternately. When the parameters of a factor are trained, the parameters of the other factor are assumed to be known and fixed, i.e.

$$\hat{\boldsymbol{\Lambda}}_{\mathrm{E}} = \arg\max_{\boldsymbol{\Lambda}_{\mathrm{E}}} p(\boldsymbol{O}|\mathcal{H}, \mathcal{E}; \mathcal{M}, \boldsymbol{\Lambda}_{\mathrm{S}}, \boldsymbol{\Lambda}_{\mathrm{E}})$$
$$\hat{\boldsymbol{\Lambda}}_{\mathrm{S}} = \arg\max_{\boldsymbol{\Lambda}_{\mathrm{S}}} p(\boldsymbol{O}|\mathcal{H}, \mathcal{E}; \mathcal{M}, \boldsymbol{\Lambda}_{\mathrm{S}}, \hat{\boldsymbol{\Lambda}}_{\mathrm{E}}) \quad (10)$$

4. TRANSPLANTATION OF PREDICTED EXPRESSION FROM TEXT

This work combines the integrated expression prediction and speech synthesis method with the SEF framework. That means, the SEF method is used to construct two sub-spaces, expression subspace and speaker subspace. Then, the expressions generated by different speaker dependent expression predictors are represented in the same expression subspace. This way, all the expressions in the expression subspace, can be arbitarily transplanted to any speakers. To achieve this, the sufficient statistics for MLP training need to be accumulated in the expression subspace only, i.e. replacing equation 7 and 8 with the following equations.

$$\mathbf{G}^{(k)} = \sum_{m,t\in T_k} \gamma_t^{(m)} \mathbf{M}_{\mathrm{E}}^{(m)\mathsf{T}} \boldsymbol{\Sigma}^{(m)\text{-}1} \mathbf{M}_{\mathrm{E}}^{(m)} \quad (11)$$

$$\mathbf{k}^{(k)} = \sum_{m} \mathbf{M}_{\mathrm{E}}^{(m)\mathsf{T}} \boldsymbol{\Sigma}^{(m)\text{-}1} \sum_{t\in T_k} \gamma_t^{(m)} (\boldsymbol{o}_t - \boldsymbol{\mu}^{(m,1)} - \mathbf{M}_{\mathrm{S}}^{(m)} \boldsymbol{\lambda}_{\mathrm{S}}^{(k)}) \quad (12)$$

Figure 2 shows the training process of the transplantable expression predictor from text. The expressions in training utterances for a particular speaker, e.g. speaker i, were extracted by projecting them into the expression subspace, given the speaker transform $\boldsymbol{\lambda}_{\mathrm{S}}^{(i)}$[1]. Meanwhile, the transcript of each utterance was converted into a linguistic feature vector in the linguistic space. Then, using the linguistic feature vectors as input, and the expressions in the expression space as target output, the MLP based expression predictor was trained using maximum likelihood criterion. Note, in figure 2, although the expression transforms in the expression subspace are speaker independent, the final trained expression predictor from text is speaker dependent, since different speaker dependent expression predictors may project the same linguistic feature into different points in the expression subspace.

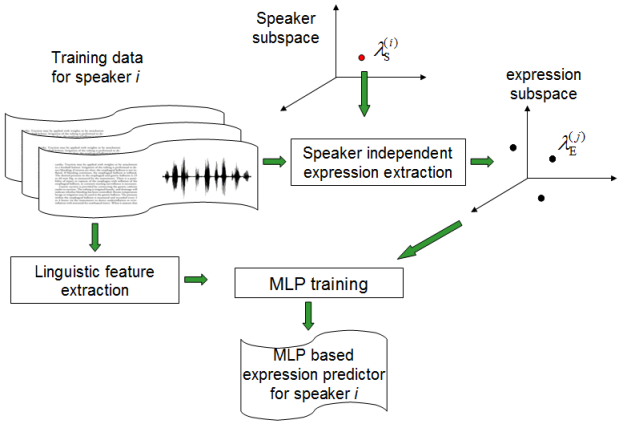

Fig. 2. *Training of transplantable expression predictor*

In the synthesis stage, the expression predictor maps the linguistic features from text data to the points in the expression subspace. Since the expressions in the expression subspace are speaker independent, they can be transplanted to different speakers in the framework of SEF. Thus the expressive synthetic speech from different speakers can be generated from plain text. This process is shown in figure 3.

5. EXPERIMENTAL RESULTS

In this work, the acoustic training data is from publicly available audiobooks from Librivox.org. It contains about 28 hours of speech from 4 audiobooks (2 male and 2 female speakers). The training data was selected by lightly supervised segmentation [10]. The sampling rate of the training speech was 16kHz and acoustic features consisted of 40 mel-cepstral coefficients, logF0, 21 (approximately bark scaled) BAP (band aperiodicity parameters) plus their delta and

[1] In this work, the neutral speech was used to estimate the speaker transforms based on equation 10.

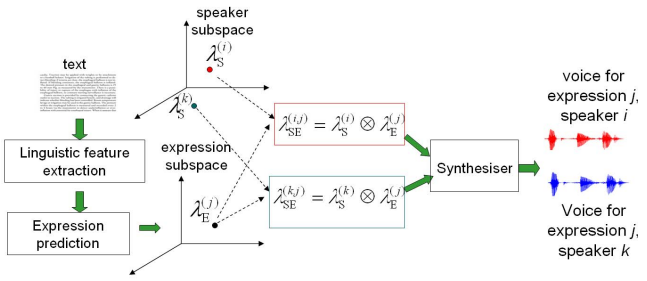

Fig. 3. *Synthesis with transplanted expressions*

delta-delta information. The models were 5 state left-to-right multi-space probability distribution hidden semi-Markov models. The unsupervised SEF method was used to train the CAT model which contains the independent expression space and speaker space. The CAT model used in this work contains 8 cluster models: 1 bias cluster model, 4 non-bias cluster models for speaker subspace construction and 3 non-bias cluster models for the expression subspace construction. The details of the model training can be found in [4].

Two speaker dependent expression predictors based on the integrated method were trained using the speaker dependent training speech. Expression predictor 1, which is named as "ep1" in this paper was trained by the training data from a male speaker, called, "speaker1", with 10.2k utterances. Expression predictor 2 (ep2) was trained by the data from a female speaker, called "speaker2", with about 6.8k utterances. Note, the expression prediction process in this work was combined with the SEF method. Thus the predicted expressions can be transplanted to different speakers.

The first experiment investigates the expressiveness of the synthetic speech based on the expression predictors. The expressions generated by the expression predictor from text were used to synthesise the expressive speech paragraphs and they were compared to the neutral TTS systems. This experiment only investigates the expressiveness. Thus the synthetic voice was generated for the training speaker, and the predicted expressions were not transplanted to the new speakers. A preference test using paragraphs was performed in this experiment. Subjects listened to the synthetic speech paragraphs from two system and were asked which one contains the appropriate expressions for the content of the paragraph. The test set includes 15 paragraphs with an average length of 3 utterances. The results were shown in table 1.

Table 1. Preference test for paragraph reading testing the expressiveness of expression predictors

spkr	ep1	neut	ep2	neut	nopref	p
1	53.3%	36.6%			10.1%	0.001
2			48.1%	36.6%	15.3%	0.017

From table 1, the synthetic speech from both of two expression predictors achieved significantly better scores than the neutral systems. It indicates that the integrated expression prediction method works well in the framework of SEF.

The second experiment investigates the portability of the expressions generated by the expression predictors. In the second experiments, the expressions generated from "ep1" and "ep2" were transplanted to a new speaker, called "speaker3", and the expressiveness of the synthetic speech was investigated based on speaker3's voice. Again, the preference test of paragraph reading was used with the neutral system as the contrast. The results are shown in table 2. Table 2 indicates that the expressions generated by both expression predictors can be transplanted to the new speaker and improve the expressiveness of the voice of the new speaker.

Table 2. Preference test for paragraph reading, testing expression transplantation

spkr	ep1	neut	ep2	neut	nopref	p
3	59.9%	33.0%			7.1%	<0.001
3			46.7%	38.8%	14.5%	0.088

Finally, the inter-speaker factors for the expression predictor from text were investigated. Through the SEF methods, the speaker independent expressions can be extracted from the speech data of various speakers. Thus the speech data from multiple speakers can be used to train the speaker independent expression predictor in which the inter-speaker variability is assumed to be normalized. The speaker independent expression predictor can only catch the common information over all the speakers. In this work, a speaker independent expression predictor was trained by 22.7k training utterances from 3 speakers and this speaker independent expression predictor was compared to the speaker dependent expression predictor in a paragraph reading preference test. The result is shown in table 3. The result in table 3 shows the impact of the inter-speaker factors to the performance of the expression predictor. In the expression predictor trained by multi-speaker data, the inter-speaker variability is normalized. As result, the expressiveness of the synthetic speech is significantly degraded.

Table 3. Preference test for paragraph reading, expression predictors with single speaker training data vs. multi-speaker training data

ep1	ep multi-spkr	No pref	p
54.8%	35.2%	10.0%	<0.001

6. CONCLUSIONS

Expression prediction from text is a speaker dependent task. Different speakers have different ways to interpret text and convert it into expressions in speech. In the work presented here, the integrated method for expression prediction from text and speech synthesis was performed in the framework of speaker and expression factorization (SEF), so that the expressions generated from different speaker dependent expression predictors can be represented in the same expression space. This allows the transplantation of the generated expression between different speakers. Experimental results showed that with the SEF framework, the expressions generated by the expression predictors can be used to improve the expressiveness of the synthetic speech not only for the training speaker, but also for other speakers by expression transplantation. Moreover, based on this framework, a speaker independent expression predictor was trained to normalize all the inter-speaker varibilities. The experimental result indicated that the speaker independent expression predictor generated less expressive speech than the speaker dependent expression predictor. This is evidence of the importance of speaker dependent information for expression prediction.

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] L. Chen, M. J. F. Gales, N. Braunschweiler, M. Akamine, and K. Knill, "Integrated automatic expression prediction and speech synthesis from text," in *Proc. of ICASSP*, 2013.

[3] 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.

[4] L. Chen and N. Braunschweiler, "Unsupervised speaker and expression factorization for multi-speaker expressive synthesis of ebooks," in *Proc. of INTERSPEECH*, 2013.

[5] J. Latorre, V. Wan, M. J. F. Gales, L. Chen, K. Chin, K. Knill, and M. Akamine, "Speech factorization for HMM-TTS based on cluster adaptive training," in *Proc. of Interspeech*, 2012.

[6] Y.Q. Wang and M. J. F. Gales, "Speaker and noise factorisation for robust speech recognition," *IEEE Trans. on Audio Speech and Language Processing*, vol. 20, no. 7, 2012.

[7] Y.Q. Wang and M. J. F. Gales, "An explicit independence constraint for factorised adaptation in speech recognition," in *Proc. of INTERSPEECH*, 2013.

[8] M. Seltzer and A. Acero, "Factored adaptation for separable compensation of speaker and environmental variability," in *Proc. of ASRU*, 2011.

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

[10] N. Braunschweiler, M. J. F. Gales, and S. Buchholz, "Lightly supervised recognition for automatic alignment of large coherent speech recordings," in *Proc. of Interspeech*, 2010, pp. 2222–2225.