Speaker and Expression Factorization for Audiobook Data: Expressiveness and Transplantation

Langzhou Chen, *Member, IEEE,* Norbert Braunschweiler, Mark J.F. Gales, *Fellow, IEEE*

Toshiba Research Europe Limited, Cambridge Research Lab, Cambridge, UK

langzhou.chen,norbert.braunschweiler@crl.toshiba.co.uk, mjfg@eng.cam.ac.uk

***Abstract*—Expressive synthesis from text is a challenging problem. There are two issues. First, read text is often highly expressive to convey the emotion and scenario in the text. Second, since the expressive training speech is not always available for different speakers, it is necessary to develop methods to share the expressive information over speakers. This paper investigates the approach of using very expressive, highly diverse audiobook data from multiple speakers to build an expressive speech synthesis system. Both of two problems are addressed by considering a factorized framework where speaker and emotion are modelled in separate sub-spaces of a cluster adaptive training (CAT) parametric speech synthesis system. The sub-spaces for the expressive state of a speaker and the characteristics of the speaker are jointly trained using a set of audiobooks. In this work, the expressive speech synthesis system works in two distinct modes. In the first mode, the expressive information is given by audio data and the adaptation method is used to extract the expressive information in the audio data. In the second mode, the input of the synthesis system is plain text and a full expressive synthesis system is examined where the expressive state is predicted from the text. In both modes, the expressive information is shared and transplanted over different speakers. Experimental results show that in both modes, the expressive speech synthesis method proposed in this work significantly improves the expressiveness of the synthetic speech for different speakers. Finally, this paper also examines whether it is possible to predict the expressive states from text for multiple speakers using a single model, or whether the prediction process needs to be speaker specific.**

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

I. Introduction

The expressive information in human speech is very rich and highly diverse. Previous work in expressive TTS usually focused on several predefined emotions, e.g. “happy”, “sad”, etc. [1], [2]. This allows users to generate synthetic speech with self-chosen but limited emotions. However, humans use a very rich space of expressiveness. In a complicated task like ebook reading, several pre-defined emotions can not cover the large range of human’s expressions and therefore a more complex expression space needs to be constructed from natural speech training data. This increases the challenge for the expressive TTS. For the expressive TTS system with multiple speakers, the challenge is even bigger. At first, the data sparseness problem is more serious in multiple speaker expressive TTS system training since it is impractical to collect the expressive speech with wide coverage in the expression space for every speaker. At synthesis time, the expressive information in the training data needs to be transplanted to the new speaker to generate the expressive synthetic speech. Finally, when the input of the TTS system is plain text, the complexity of a multiple speaker expressive TTS becomes even larger because the way speakers interpret emotion encapsulated in text and how they convert it into the expressions in speech depends on individual speaker’s background, education, skill, etc., and varies from speaker to speaker.

Research in statistical parametric speech synthesis widely uses adaptation methods for speaker and expression modelling, including model interpolation [3], [4], transform-based method [5], [6], CAT [7], [8], etc. All the methods mentioned above only deal with either speaker modelling or expression modelling. However, when both of the two factors have to be considered, directly modelling every combination of speaker and expression is often impractical since the expressive training data is not always available for every speaker.

A better solution for this problem is achieved by factorization techniques which model speaker and expression independently when using training data with multiple expressions and speakers. This way, different speakers voices can share the speaker independent expressions and produce expressive synthetic speech. Various forms of factorization can be used for speaker and expression factorization (SEF). For the methods based on linear transformation, a cascade of constrained maximum likelihood linear regression (CMLLR) transforms has been used in ASR to factorize the speaker and environment parameters in [9]. It appears interesting to investigate the similar method in TTS research, e.g. SEF. The subspace based methods such as eigenvoice method [10] and factor analyzed voice models [11] can be used for SEF as well. In this type of methods, the expressions and speakers are modelled in separate low dimensional subspaces. CAT is also a subspace based method and it has been used in supervised SEF in [12], based on an acted training corpus with

Manuscript received May 14, 2014; revised September 1, 2014 and December , 2014; accepted December 19, 2014.

L. Chen and N. Braunschweiler are with Toshiba Research Europe Limited, Cambridge, CB4 0GZ, U.K. (e-mail: langzhou. chen@crl.toshiba.co.uk, norbert.braunschweiler@crl. toshiba.co.uk). M. J. F. Gales is with Cambridge University, Engineering Department (e-mail: mjfg@eng.cam.ac.uk).

fixed expressions. Different to the eigenvoice method, the CAT method allows separate decision trees for different clusters. This yields a more complex expression and speaker space to be defined as any changes in the context-dependency of the speech with expressions and speakers can be modelled. Two types of transforms which are different in nature can be used to achieve the factorization of two different acoustic factors as well. [13] presented a speaker and noise factorization method for ASR. The speaker factor was modelled by maximum likelihood linear regression (MLLR) linear transforms while the noise was modelled by the non-linear vector Taylor series (VTS) method. For TTS, in [14], a speaker and language factorization (SLF) method was proposed which used CMLLR transforms to represent the speakers and the CAT weight vectors to represent languages. This can be extended to SEF by using CAT weight vectors to represent expressions rather than languages. The factorization methods mentioned above are based on labelled data, i.e. speaker and expression information in the training data is known.

Human speech contains a very large range of expressiveness. It is very hard to cover the very rich expressive information of human speech by a limited number of predefined emotions from an acted corpus. Nowadays, huge amounts of audiobook data is available and has been used for TTS system training [15], [16], [17]. This data source contains highly diverse speech which covers a wide range of speakers, expressions and character voices. This high diversity provides the opportunity to improve the performance of the TTS system in different aspects, e.g. the expressiveness of synthetic speech [7], [6], character voices [18], etc. Although the audiobook data contains very rich information to improve the performance of synthesis systems in different aspects, it is non-trivial to make use of it directly since different types of information are bonded together. That means, an utterance is typically associated with a particular expression and comes from a particular speaker. This makes the factorization techniques the key technology to explore different types of information from audiobook data. For audiobook data, it is a challenge to apply SEF techniques when multi-speaker training data is used. Manually adding expression labels to audiobook data is expensive and has typically poor inter-annotator agreement due to the high diversity of the data. This makes the standard SEF methods difficult to use for the audiobook data directly. To address this problem, two solutions are proposed in this work. The first one is a disjoint method, in which an independent expression clustering process is performed to automatically classify the audiobook data into different expressions; then expression clustering results are used as expression labels for the SEF process. The second method is a joint method in which the model parameter estimation and automatic expression clustering process are integrated into a single process based on the ML criterion. In this work, the advantages and the disadvantages of the two methods were analyzed and the performances of the two methods were compared as well.

In the SEF method, the expression subspace is shared by all the speakers. Thus every expression projected into this subspace can be transplanted to different speakers. This allows the same set of expressions to be used to generate the synthetic speech over different speakers. The expressions in human's language can usually be perceived in two ways: they can be heard in the speech data and they can be interpreted from the text data as well. Correspondingly, this work discusses the expression sharing and transplantation in a multiple speaker expressive TTS system in two distinct modes: a supervised adaptation mode in which the expression is extracted from adaptation speech and a full expressive synthesis mode in which the expression is predicted from text data.

In the first mode, the adaptation utterance from a speaker is given. Using the SEF framework proposed in this work, speaker information and expression information in the adaptation utterance can be projected to the points in the speaker subspace and the expression subspace separately. Thus, the expressive information in a particular speech utterance is represented as the projected point in the expression subspace. Using the expression transplantation method, synthetic speech for other speakers can be generated with the same expression as the adaptation data.

In the second mode, the adaptation speech is not provided and the input of the TTS system is plain text. Thus, it is a complete expressive text-to-speech synthesis system including expression prediction from text. Since the nature of how a reader interprets and reads the text varies from individual to individual, the expression prediction from text is actually a speaker dependent task. In this case, the speaker specific fashion to interpret and read the text was transplanted to other speakers. Traditional expression predictors based on computational linguistic methods [19], [2], [20], [21], [22] have not investigated the inter-speaker factors in the text to expression prediction. In this work, the integrated method for expression prediction and speech synthesis which was presented in [23] has been extended to the SEF framework. The expressive linguistic features extracted from the text data are mapped to the points in the expression subspace constructed by SEF using a non-linear transform based on an MLP neural network. Since the MLP based expression predictor is trained by speech data, the speaker dependent expression predictor can be trained by using the training data from a single speaker. Meanwhile, the predicted expressions are represented as the points in the expression subspace constructed by SEF. Thus they can be transplanted to other speakers. This work investigated if the fashion in which a particular speaker interprets and reads a text can be used to improve the expressiveness of the synthetic speech from other speakers.

Finally, the SEF method allows the speech data from different speakers to be projected into the common expression subspace, thus the speaker independent expression predictor can be trained using multi-speaker training data. Since in speaker independent expression predictors, the inter-speaker variability is assumed to be normalized, the impact of the speaker specific information on the expression prediction performance can be investigated by comparing the speaker dependent and independent expression predictors.

II. General idea of factorization

Adaptation technologies have been widely used in statistical parametric speech synthesis systems to adjust acoustic models (AM) to generate synthetic speech with some acoustic factors, e.g. speaker, expression, character etc. In order to adapt the AM to a particular acoustic factor, the adaptation data with the same acoustic factor is needed. The training data for a TTS system sometimes contains multiple acoustic factors simultaneously. This is especially true for highly diverse, complex speech data such as audiobook data. When using two acoustic factors, speaker and expression for example, and if the speech data with a particular speaker s and an expression e is used as adaptation data, the adapted AM will generate synthetic speech with two factors (s, e). Typically based on the ML criterion, the adaptation process can be represented as

$$\hat{\boldsymbol{\lambda}}^{(s,e)} = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{O}^{(s,e)}|\mathcal{H};\mathcal{M},\boldsymbol{\lambda}) \tag{1}$$

where $\boldsymbol{O}^{(s,e)}$ represents the adaptation data with acoustic condition s, e, and $\hat{\boldsymbol{\lambda}}^{(s,e)}$ represents the target transform for the same condition. $\mathcal{H}$ and $\mathcal{M}$ represent the transcripts of the adaptation data and the AM respectively. Based on the framework of equation 1, if the TTS system needs to generate the synthetic speech with m expressions from n speakers, the number of transforms need to be estimated is $m \times n$. This number can be very big when the values of m and n increase. Another problem is the availability of the adaptation data. When the adaptation data with a particular acoustic condition is not available, the TTS system is not able to generate the voice with the same acoustic factors.

To address the problem mentioned above, the factorization techniques were adopted to factorize a complex acoustic condition into several independent factors, i.e.

$$\boldsymbol{\lambda}^{(s,e)} = \boldsymbol{\lambda}_{\mathrm{S}}^{(s)} \otimes \boldsymbol{\lambda}_{\mathrm{E}}^{(e)} \tag{2}$$

where $\boldsymbol{\lambda}_{\mathrm{S}}^{(s)}$ and $\boldsymbol{\lambda}_{\mathrm{E}}^{(e)}$ are the independent transforms for speaker s and expression e respectively. Factorization techniques provide a better solution to deal with the complex acoustic conditions. Again, if the TTS system needs to generate synthetic speech with m expressions from n speakers, only $m + n$ transforms need to be trained with factorization techniques, i.e. m expression dependent transforms and n speaker dependent transforms. This number is much smaller than $m \times n$ when m and n increase. The factorization techniques assume that the transforms for different factors are "orthogonal", i.e. they should be independent to each other. Under this assumption, the transforms for different speakers and expressions can be arbitrarily composed, even when a combination never occurred in the training data. This means, to generate the synthetic speech with acoustic condition (s, e), only the speaker transform for s and the expression transform for e are needed and they are combined using equation 2. Furthermore, for the new speaker s', only the neutral data is needed to estimate the speaker transforms $\hat{\boldsymbol{\lambda}}_{\mathrm{S}}^{(s')}$. Then, the speaker transform can be combined with various of the speaker independent expression transforms to generate the synthetic voice from speaker s' with various forms of expressions. In this process, the adaptation data from speaker s' with different expressions is not necessarily required.

The ML based parameter estimation for factorization can be expressed as:

$$\hat{\boldsymbol{\Lambda}}_{\mathrm{S}}, \hat{\boldsymbol{\Lambda}}_{\mathrm{E}} = \arg\max_{\boldsymbol{\Lambda}_{\mathrm{E}},\boldsymbol{\Lambda}_{\mathrm{S}}} p(\boldsymbol{O}|\mathcal{H};\mathcal{M},\boldsymbol{\Lambda}_{\mathrm{S}},\boldsymbol{\Lambda}_{\mathrm{E}}) \tag{3}$$

where $\boldsymbol{O}$ are the observation vectors, $\hat{\boldsymbol{\Lambda}}_{\mathrm{S}}$ and $\hat{\boldsymbol{\Lambda}}_{\mathrm{E}}$ are the transforms for the speaker and the expression respectively.

The ML parameter estimation based on equation 3 can be solved in an iterative way. When the transforms of one factor are estimated, the transforms of the other factors are assumed to be known and fixed, and the transforms for different factors are updated alternately until the convergence. This process can be expressed as:

$$\begin{aligned} \hat{\boldsymbol{\Lambda}}_{\mathrm{E}} &= \arg\max_{\boldsymbol{\Lambda}_{\mathrm{E}}} p(\boldsymbol{O}|\mathcal{H};\mathcal{M},\boldsymbol{\Lambda}_{\mathrm{S}},\boldsymbol{\Lambda}_{\mathrm{E}}) \\ \hat{\boldsymbol{\Lambda}}_{\mathrm{S}} &= \arg\max_{\boldsymbol{\Lambda}_{\mathrm{S}}} p(\boldsymbol{O}|\mathcal{H};\mathcal{M},\boldsymbol{\Lambda}_{\mathrm{S}},\hat{\boldsymbol{\Lambda}}_{\mathrm{E}}) \end{aligned} \tag{4}$$

For the factorization techniques, the "orthogonality" between the transforms of different factors is the precondition. Thus how to keep this "orthogonality" is the question that every factorization technique needs to answer. In this work, the "orthogonality" was achieved by adding some implicit constraint to the training data, i.e. the speaker and expression overlaps in the training data, as shown in Fig. 1. The implicit constraint to the training data requires that for every training speaker, speech data with different expressions needs to be provided. While the training data for every expression should be from multiple speakers. Based on this constraint, the transform for an expression was trained by the speech data from multiple speakers, thus it can be guaranteed to be independent from a particular speaker. Similarly, the speaker transform was trained by the speech with multiple expressions. Thus it is independent of a particular expression.

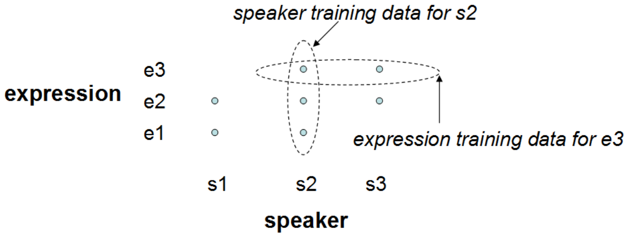


Fig. 1. *Speaker and expression overlaps in training data*

Although in this work, the implicit constraint for the training data was used to ensure the "orthogonality" of the speaker and expression transforms, some other methods can be used as well. For example, in [13], the transforms with different attributes were used to model the different factors. In [24] an explicit independence constraint method was proposed for factorized adaptation in speech recognition.

III. SEF based on CAT

The CAT model consists of a set of cluster models, each of which contain a set of Gaussian mean parameters while the

Gaussian variances are shared over all clusters. When this CAT model is used to calculate the likelihood of an observation vector $\boldsymbol{o}_t$, the mean vector to be used is a linear interpolation of all the cluster means, i.e.

$$p(\boldsymbol{o}_t|\boldsymbol{\lambda}, \mathbf{M}^{(m)}, \boldsymbol{\Sigma}^{(m)}) = \mathcal{N}(\boldsymbol{o}_t; \mathbf{M}^{(m)}\boldsymbol{\lambda}, \boldsymbol{\Sigma}^{(m)}) \tag{5}$$

where $\mathbf{M}^{(m)}$ is the matrix of P cluster mean vectors for component m, i.e. $\mathbf{M}^{(m)} = \begin{bmatrix}\boldsymbol{\mu}^{(m,1)} & \dots & \boldsymbol{\mu}^{(m,P)}\end{bmatrix}$ and $\boldsymbol{\lambda}$ is the CAT weight vector.

From equation 5, the CAT model is a subspace based method, which represents very high dimensional synthesis parameters (the concatenation of all Gaussian mean vectors) with a low dimensional subspace. When the CAT model is used for speaker modelling, a speaker subspace is constructed and each speaker dependent information is represented as a point in the speaker subspace which can be uniquely represented as a speaker CAT weight vector. Similarly, for expression modelling, each expression is associated with a point in an expression space which in turn is represented as an expression CAT weight vector. In the case of SEF, two subspaces were constructed separately for speakers and expressions. Thus the CAT weight vector contains both speaker and expression information. That means, some dimensions of the CAT weight vector represent the point in the speaker subspace while the others represent the point in the expression subspace. Based on CAT, the SEF in equation 2 is with the form of

$$\boldsymbol{\lambda}^{(s,e)} = \begin{bmatrix}\boldsymbol{\lambda}_{\mathtt{S}}^{(s)\mathsf{T}} & \boldsymbol{\lambda}_{\mathtt{E}}^{(e)\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \tag{6}$$

And, equation 5 can be re-written as

$$\begin{aligned} & p(\boldsymbol{o}_t|\boldsymbol{\lambda}_{\mathrm{S,E}}^{(s,e)}, \mathbf{M}_{\mathrm{S,E}}^{(m)}, \boldsymbol{\Sigma}^{(m)}) \\ & = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}^{(m,1)} + \mathbf{M}_{\mathrm{S}}^{(m)}\boldsymbol{\lambda}_{\mathrm{S}}^{(s)} + \mathbf{M}_{\mathrm{E}}^{(m)}\boldsymbol{\lambda}_{\mathrm{E}}^{(e)}, \boldsymbol{\Sigma}^{(m)}) \end{aligned} \tag{7}$$

where $\boldsymbol{\lambda}_{\mathrm{E}}^{(e)}$ and $\boldsymbol{\lambda}_{\mathrm{S}}^{(s)}$ are the CAT weight vectors to model the expression e and speaker s 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. With CAT based SEF, each speech utterance is projected into 2 subspaces separately and the CAT weights are the coordinates of these projections. That says, each speech utterance can be represented as two points in speaker subspace and expression subspace respectively. The cluster models only form the basis of the subspace, while they are not related to a particular speaker or expression. A particular speaker or expression is always related to a point in the subspace.

In the training process of SEF, the implicit data constraint described in section II is added to the training data to ensure the “orthogonality” of the expression transforms and speaker transforms. This means, there must be an overlap between speakers and expressions in the training data. Then, based on the ML criterion, the speaker transforms and the expression transforms are updated alternately using equation 4. Since the updating of speaker parameters and the updating of expression parameters work in a similar way, only the expression updating is discussed.

In the CAT framework, the auxiliary function for the CAT weight estimation can be expressed as

$$\mathcal{Q}(\hat{\boldsymbol{\Lambda}}; \boldsymbol{\Lambda}) = \sum_i \left(\hat{\boldsymbol{\lambda}}^{(i)\mathsf{T}}\mathbf{y}^{(i)} - \frac{1}{2}\hat{\boldsymbol{\lambda}}^{(i)\mathsf{T}}\mathbf{X}^{(i)}\hat{\boldsymbol{\lambda}}^{(i)}\right) + C \tag{8}$$

where i is the utterance index, C represents the terms independent to $\hat{\boldsymbol{\lambda}}$ and the sufficient statistics $\mathbf{X}^{(i)}$ and $\mathbf{y}^{(i)}$ are given by

$$\mathbf{X}^{(i)} = \sum_{m,t\in T_i} \gamma_t^{(m)}\mathbf{M}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\mathbf{M}^{(m)} \tag{9}$$

$$\mathbf{y}^{(i)} = \sum_m \mathbf{M}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\sum_{t\in T_i}\gamma_t^{(m)}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m,1)}) \tag{10}$$

where $\gamma_t^{(m)}$ is the occupancy probability of component m in time t, $\boldsymbol{\mu}^{(m,1)}$ is the mean vector of component m from the bias cluster.

For SEF, to calculate the new expression CAT weight vectors $\hat{\boldsymbol{\Lambda}}_{\mathrm{E}}$, given the old expression CAT weight vectors $\boldsymbol{\Lambda}_{\mathrm{E}}$ and the fixed speaker CAT weight vectors $\boldsymbol{\Lambda}_{\mathrm{S}}$, the equation 8 can be re-written as

$$\begin{aligned} \mathcal{Q}(\hat{\boldsymbol{\Lambda}}_{\mathtt{E}}; \boldsymbol{\Lambda}_{\mathtt{E}}, \boldsymbol{\Lambda}_{\mathtt{S}}) &= \sum_j\sum_{i\in e_j}\Bigg(\begin{bmatrix}\boldsymbol{\lambda}_{\mathtt{S}}^{(i)\mathsf{T}} & \hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)\mathsf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{y}_{\mathtt{S}}^{(i)} \\ \mathbf{y}_{\mathtt{E}}^{(i)}\end{bmatrix} \\ &\quad -\frac{1}{2}\begin{bmatrix}\boldsymbol{\lambda}_{\mathtt{S}}^{(i)\mathsf{T}} & \hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)\mathsf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{X}_{\mathtt{SS}}^{(i)} & \mathbf{X}_{\mathtt{SE}}^{(i)} \\ \mathbf{X}_{\mathtt{ES}}^{(i)} & \mathbf{X}_{\mathtt{EE}}^{(i)}\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda}_{\mathtt{S}}^{(i)} \\ \hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)}\end{bmatrix}\Bigg) + C \\ &= \sum_j\sum_{i\in e_j}\left(\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)\mathsf{T}}\mathbf{z}_{\mathtt{E}}^{(i)} - \frac{1}{2}\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)\mathsf{T}}\mathbf{X}_{\mathtt{EE}}^{(i)}\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)}\right) + D \end{aligned} \tag{11}$$

where $\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)}$ represents the expression CAT weight vector for expression j and $\boldsymbol{\lambda}_{\mathtt{S}}^{(i)}$ represents the speaker CAT weight vector of utterance i which is assumed to be known, D represents the terms independent to $\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(i)}$. The sufficient statistics are given by

$$\begin{aligned} \mathbf{X}_{\mathtt{EE}}^{(i)} &= \sum_{m,t\in T_i}\gamma_t^{(m)}\mathbf{M}_{\mathtt{E}}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\mathbf{M}_{\mathtt{E}}^{(m)} \\ \mathbf{X}_{\mathtt{ES}}^{(i)} &= \sum_{m,t\in T_i}\gamma_t^{(m)}\mathbf{M}_{\mathtt{E}}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\mathbf{M}_{\mathtt{S}}^{(m)} \\ \mathbf{y}_{\mathtt{E}}^{(i)} &= \sum_m \mathbf{M}_{\mathtt{E}}^{(m)\mathsf{T}}\boldsymbol{\Sigma}^{(m)\text{-}1}\sum_{t\in T_i}\gamma_t^{(m)}(\boldsymbol{o}_t - \boldsymbol{\mu}^{(m,1)}) \\ \mathbf{z}_{\mathtt{E}}^{(i)} &= \mathbf{y}_{\mathtt{E}}^{(i)} - \mathbf{X}_{\mathtt{ES}}^{(i)}\boldsymbol{\lambda}_{\mathtt{S}}^{(i)} \end{aligned} \tag{12}$$

Differentiating equation 11 with respect to $\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)}$ and equating to zero yields,

$$\hat{\boldsymbol{\lambda}}_{\mathtt{E}}^{(e_j)} = \Big(\sum_{i:i\in e_j}\mathbf{X}_{\mathtt{EE}}^{(i)}\Big)^{\text{-}1}\sum_{i:i\in e_j}\left(\mathbf{z}_{\mathtt{E}}^{(i)}\right) \tag{13}$$

IV. SEF FOR AUDIOBOOK DATA

This work investigates ways to apply SEF on audiobook data. Audiobook data is highly diverse data with very rich expressive information. Due to the high diversity, manually adding expression labels to the audiobook data is sometimes impractical. The parameter estimation algorithm of CAT based SEF mentioned above assumed that the speaker independent expression labels have been added to the training data for all speakers, and the data from one expression are distributed over different speakers, so that speaker independent expression parameters can be estimated. Thus for audiobook data, the standard SEF method described in the last section can not be used directly. Therefore, this article introduces methods of SEF for unlabelled audiobook data. Although the proposed method can be extended to the case that both speakers and expressions are unlabelled, in this work, only the case of audiobook data is discussed, i.e. the speaker is known, but the expression is unknown.

A. Disjoint approach

In order to perform an SEF process with unlabelled data, the straightforward approach is adding an expression clustering process before the SEF process as shown in Fig. 2.

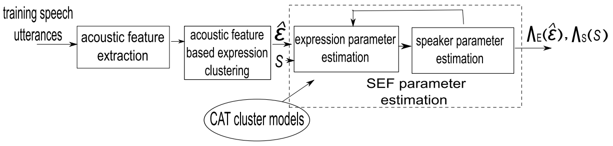

training speech
utterances
acoustic feature extraction
acoustic feature based expression clustering
$\hat{\mathcal{E}}$
S
expression parameter estimation
speaker parameter estimation
$\Lambda_E(\hat{\mathcal{E}}), \Lambda_S(S)$
SEF parameter estimation
CAT cluster models

Fig. 2. *Disjoint method for unsupervised SEF*

At first, for each utterance in the training data an acoustic feature vector was created representing the expressive information in an utterance. The feature vector includes various expression related features, e.g. mean of F0, voicing probability (p_v), local jitter and shimmer, logarithmic HNR, standard deviation of F0 and mean of absolute delta of F0 and p_v etc. Before clustering, all the feature vectors were standardised to zero mean and unit variance [6]. Then, a hierarchical k-means clustering was performed to classify the training utterance into different expression clusters based on the Euclidean distance metric between the feature vectors. The automatic expression clustering results were used as expression supervision information and the SEF process described in section III was performed. The speaker parameters and the expression parameters were updated alternately until the convergence. Note, Fig. 2 only includes the speaker and expression parameter updating parts in the training process. All the modules which are irrelevant to speaker and expression parameter updating, e.g. cluster model updating, decision tree construction etc., are not shown.

The expression clustering process in this work is the same as the one in [6]. The difference is that in [6], the expression clustering was performed on data from a single speaker, while in this work, data from multiple speakers was used. The expression clustering process groups the training speech utterances into a set of clusters $\hat{\mathcal{E}} = \{e_1, e_2, \cdots, e_k\}$, based on the distance measurement between acoustic feature vectors, e.g. minimize the within-class error. This process can be expressed as:

$$\hat{\mathcal{E}} = \arg\min_{\mathcal{E}} \sum_{j} \sum_{i \in e_j} \|\boldsymbol{v}_i - \boldsymbol{c}_j\|^2 \tag{14}$$

where $\boldsymbol{v}_i$ represents the acoustic feature vector of utterance i, $\boldsymbol{c}_j$ represents the centroid of cluster j.

Given the expression clustering results $\hat{\mathcal{E}}$ and the known speaker information, a standard SEF process was used to estimate the expression CAT weight vector for each expression cluster by the ML criterion, i.e.

$$\hat{\boldsymbol{\Lambda}}_{\text{E}}(\hat{\mathcal{E}}) = \arg\max_{\boldsymbol{\Lambda}_{\text{E}}(\hat{\mathcal{E}})} p(\boldsymbol{O}|\mathcal{H}, \hat{\mathcal{E}}; \mathcal{M}, \boldsymbol{\Lambda}_{\text{S}}, \boldsymbol{\Lambda}_{\text{E}}(\hat{\mathcal{E}})) \tag{15}$$

where $\boldsymbol{\Lambda}_{\text{S}}$ represents the speaker CAT weight vectors which are known and fixed, $\hat{\boldsymbol{\Lambda}}_{\text{E}}(\hat{\mathcal{E}}) = \{\hat{\boldsymbol{\lambda}}_{\text{E}}^{(e_1)}, \hat{\boldsymbol{\lambda}}_{\text{E}}^{(e_2)}, \cdots, \hat{\boldsymbol{\lambda}}_{\text{E}}^{(e_k)}\}$ represents the expression CAT weight vectors based on the expression clustering results $\hat{\mathcal{E}}$.

This method has two weak aspects. First, the acoustic features used for expression clustering, e.g. the mean of F_0 etc., are highly dependent on speakers, i.e. the expression clustering results may be influenced by the speaker factor. Second, the two processes are performed independently. The expression clustering is usually based on the distance measure in the acoustic feature space, e.g. the minimum within class error (MWCE), while the SEF is based on the ML criterion; the optimal expression clustering result in terms of equation 14 is not necessarily optimal for maximizing the likelihood of the training data as in equation 15. In other words, there is an inconsistency between the parameter estimation of the two processes. To address the first problem mentioned above, the speaker normalization approaches can be adopted, e.g. [25], [26] etc. These techniques can alleviate the influence of the speaker factors in expression clustering. However, they can not solve the second problem, i.e. the inconsistency of the training criteria.

B. Joint Approach

The weakness in the disjoint approach stems from the fact that the expression clustering process is independent of the SEF process. An alternative solution for SEF with unlabelled data is to integrate the expression clustering and parameter estimation into a single process. This means that the expression clustering and the expression dependent parameter estimation are strongly linked together, rather than 2 independent processes. It can be expressed as:

$$\hat{\mathcal{E}}, \hat{\boldsymbol{\Lambda}}_{\text{E}}(\hat{\mathcal{E}}) = \arg\max_{\mathcal{E}, \boldsymbol{\Lambda}_{\text{E}}(\mathcal{E})} p(\boldsymbol{O}|\mathcal{H}, \mathcal{E}; \mathcal{M}, \boldsymbol{\Lambda}_{\text{S}}, \boldsymbol{\Lambda}_{\text{E}}(\mathcal{E})) \tag{16}$$

To realize the joint SEF, the process in equation 16 was divided into two steps: given current expression parameters, clustering the training utterances into expressions and given current expression clustering estimating the expression parameters, i.e.

$$\hat{\mathcal{E}} = \arg\max_{\mathcal{E}} p(\boldsymbol{O}|\mathcal{H}, \mathcal{E}; \mathcal{M}, \boldsymbol{\Lambda}_{\text{S}}, \boldsymbol{\Lambda}_{\text{E}}) \tag{17}$$

$$\hat{\boldsymbol{\Lambda}}_{\text{E}}(\hat{\mathcal{E}}) = \arg\max_{\boldsymbol{\Lambda}_{\text{E}}(\hat{\mathcal{E}})} p(\boldsymbol{O}|\mathcal{H}, \hat{\mathcal{E}}; \mathcal{M}, \boldsymbol{\Lambda}_{\text{S}}, \boldsymbol{\Lambda}_{\text{E}}(\hat{\mathcal{E}})) \tag{18}$$

The training process of the joint method is shown in Fig. 3. Again, all the modules which are irrelevant to the expression and speaker parameter estimation are not shown.

There are 2 loops in the joint training process. The first loop generating the expression parameters, is a process of alternately clustering the expressions and estimating the expression parameters which correspond to equation 17 and 18.

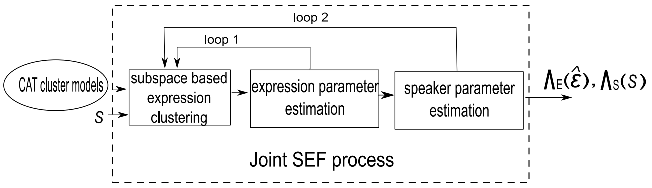

loop 2
loop 1
CAT cluster models
S
subspace based expression clustering
expression parameter estimation
speaker parameter estimation
$\Lambda_E(\hat{\mathcal{E}}), \Lambda_S(S)$
Joint SEF process

Fig. 3. *Joint method for unsupervised SEF*

Then the process of the first loop and the process of speaker parameter generation form the second loop, which optimize the speaker and expression parameters alternately as the standard SEF process. On the other hand, in Fig. 2, the disjoint approach only contains one loop, i.e. the speaker and expression parameters estimation which is a standard SEF process, while the expression clustering is an independent process of the SEF. That means, the expression clustering is performed independently before the SEF training and fixed during the whole training process. The other characteristics for the joint method is that the expression clustering is performed in the expression space only. Thus the speaker factors are isolated from this process and the results of expression clustering are speaker independent.

The utterance level auxiliary function for ML estimation can be defined as

$$\tilde{\mathcal{Q}}(\hat{\boldsymbol{\lambda}}_{\mathrm{E}}, \mathbf{X}_{\mathrm{EE}}^{(i)}, \mathbf{X}_{\mathrm{ES}}^{(i)}, \mathbf{z}_{\mathrm{E}}^{(i)}) = \hat{\boldsymbol{\lambda}}_{\mathrm{E}}^{\mathsf{T}} \mathbf{z}_{\mathrm{E}}^{(i)} - \frac{1}{2}\hat{\boldsymbol{\lambda}}_{\mathrm{E}}^{\mathsf{T}} \mathbf{X}_{\mathrm{EE}}^{(i)} \hat{\boldsymbol{\lambda}}_{\mathrm{E}} \tag{19}$$

Then, the equation 11 can be re-written as

$$\mathcal{Q}(\hat{\boldsymbol{\Lambda}}_{\mathrm{E}}; \boldsymbol{\Lambda}_{\mathrm{E}}, \boldsymbol{\Lambda}_{\mathrm{S}}) = \sum_{j} \sum_{i \in e_j} \tilde{\mathcal{Q}}(\hat{\boldsymbol{\lambda}}_{\mathrm{E}}^{(e_j)}, \mathbf{X}_{\mathrm{EE}}^{(i)}, \mathbf{X}_{\mathrm{ES}}^{(i)}, \mathbf{z}_{\mathrm{E}}^{(i)}) \tag{20}$$

The task of joint SEF is to find a partition of the training data $\hat{\mathcal{E}}$ and the expression specific CAT weight vectors associated to this partition $\hat{\boldsymbol{\Lambda}}_{\mathrm{E}}(\hat{\mathcal{E}})$ so that the value of equation 20 is maximized. This was realized by a k-means style algorithm. The k-means style algorithm can be divided in to two steps: the assignment step and the update step, which correspond to the optimization of equation 17 and 18 respectively.

In the assignment step, for each training utterance $\boldsymbol{O}^{(i)}$, an expression cluster $e(\boldsymbol{O}^{(i)})$ was assigned to it by

$$e(\boldsymbol{O}^{(i)}) = \arg \max_{e_j : j=1,\cdots,k} \tilde{\mathcal{Q}}(\hat{\boldsymbol{\lambda}}_{\mathrm{E}}^{(e_j)}, \mathbf{X}_{\mathrm{EE}}^{(i)}, \mathbf{X}_{\mathrm{ES}}^{(i)}, \mathbf{z}_{\mathrm{E}}^{(i)}) \tag{21}$$

In the update step, the expression CAT weight vector for each expression cluster was re-calculated, using equation 13. The assignment step and the update step were performed iteratively until convergence.

The joint method can alleviate the problems in the disjoint method. In the joint method, the expression clustering is performed in the expression subspace. That means, it optimizes the auxiliary function of SEF in which the speaker factor is explicitly removed; thus the speaker independent expression clustering can be achieved. At the same time, the expression clustering and CAT weight vector estimation are integrated into a single process based on the ML criterion and there is no inconsistency in the training process. Although the joint method can alleviate the weakness of the disjoint method, to implement joint SEF, an initial expression subspace needs to be constructed in which the utterance based statistics to be accumulated. The quality of this initial expression subspace may influence the performance of the joint SEF method. In this work, the expression space constructed by the disjoint methods was used to calculate the initial utterance based statistics for the joint method. Thus the weakness of the disjoint method may influence the performance of the joint method indirectly.

V. Expressive speech synthesis based on SEF

The expressions in human's language can be perceived in two ways: by listening to speech and by interpreting text. Correspondingly, the expressive speech synthesis system can work in two modes. One is extracting the expressive information from audio data. In this case, the adaptation technologies have been widely used to train the transforms for the expressions by maximizing the likelihood of the audio data which contains the expressions. The other is extracting the expressive information from plain text data. This case represents a complete expressive text to speech synthesis process, and the methods for expression prediction from text were developed to extract the expressions from the text data. In this work, the expressive speech synthesis system was investigated in both of the two modes. The expressive information is generated from either audio data or plain text. The generated expressions are represented as the points in the expression subspace under the framework of SEF. Thus the expressions are speaker independent and can be transplanted to different speakers.

A. Expression adaptation and transplantation with audio data

When the expression is obtained from audio data, the adaptation method was used to extract the expressions from the audio data. In the framework of SEF, the adaptation process can be expressed as equation 4, in which the expression CAT weight vectors and the speaker CAT weight vectors are updated alternately. When one factor is updated, the other is assumed to be known and fixed. After the expression CAT weight vector is trained, it can be composed with the speakers CAT weight vectors to generate the synthetic speech with the same expression as the adaptation data but with a new speaker's voice. This process can be illustrated as shown in Fig. 4.

Given the expressive adaptation speech with expression j, from speaker i, the adaptation process is performed to estimate a point $\boldsymbol{\lambda}_{\mathrm{E}}^{(j)}$ in the expression subspace to represent the expression j in the expressive adaptation speech. Meanwhile, the adaptation data from another speaker, i.e. speaker k, is fed to the system as well, and the speaker adaptation process is performed to estimate a point $\boldsymbol{\lambda}_{\mathrm{S}}^{(k)}$ in the speaker subspace to represent speaker k. Then the expression transform $\boldsymbol{\lambda}_{\mathrm{E}}^{(j)}$ and the speaker transform $\boldsymbol{\lambda}_{\mathrm{S}}^{(k)}$ can be composed to generate the synthetic speech for speaker k but with the same expression as the one in the expressive adaptation speech of speaker i. Note, in the process in Fig. 4, the process of speaker adaptation and the expression adaptation are exactly identical. The expression transform and the speaker transform are updated alternately using equation 4. The only difference is that after parameter

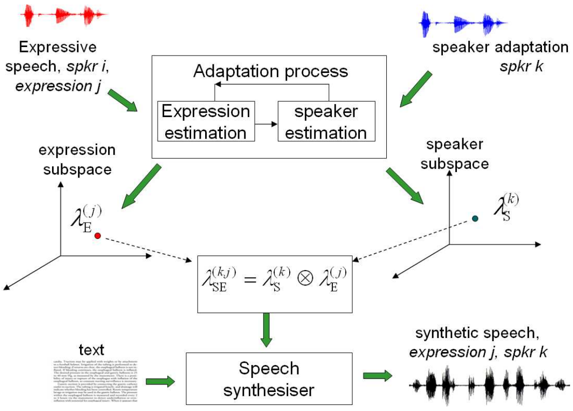

Expressive speech, *spkr i*, *expression j*
speaker adaptation data, *spkr k*
Adaptation process
Expression estimation
speaker estimation
expression subspace
speaker subspace
$\lambda_{\mathrm{E}}^{(j)}$
$\lambda_{\mathrm{S}}^{(k)}$
$\lambda_{\mathrm{SE}}^{(kj)} = \lambda_{\mathrm{S}}^{(k)} \otimes \lambda_{\mathrm{E}}^{(j)}$
text
Speech synthesiser
synthetic speech, *expression j, spkr k*

Fig. 4. *Expression adaptation and transplanting based on audio data*

updating, the speaker adaptation process keeps the speaker transform and discards the expression transform. On the other hand, the expression adaptation process keeps the expression transform and discards the speaker transform.

B. Expressive speech synthesis from plain text

Expressiveness of human language can not only be heard in speech data, it can be interpreted from text data as well. In a complete expressive speech synthesis process, the expressive speech synthesis system needs to generate the proper expressive speech from plain text and the methods for expression prediction from text are needed to generate the expressions for synthesis. How to interpret the emotion in text and how to convert it to the expressions in speech is strongly dependent on the speaker's background, education and skill etc. Therefore, expression prediction from text is a speaker dependent task. In this work, based on the framework of SEF, it was investigated how the speaker dependent expression predictor from text is constructed. At the same time, this work also investigates the transplantation of the way a speaker interprets the text and expresses the emotion in speech to other speakers, so that other speakers can read the text in the same fashion as the first speaker. Traditionally, the expression prediction is viewed as a computational linguistic task [19] and assumed to be speaker independent. That means, all the speaker specific factors are ignored in traditional expression prediction methods. In [23], a method of integrating the expression prediction from text and speech synthesis in a single system was presented. In this method, the task of the expression prediction was conducted as a mapping between the linguistic feature space which contains the expressive information from text data and the expressive synthesis space which contains the expressions extracted from audio data. Since in this method the expression predictor is trained by the speech data, the speaker dependent expression predictor can be trained by using speech from a single speaker. In [23], the linguistic feature vector which contains the expressive information in the text data was generated by the latent semantic mapping (LSM) method. In this work, a similar linguistic feature based on the bag-of-word model was used. 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 $P(w_1, w_2)$ and word frequency with part-of-speech (POS) context $P(pos_1, w_2, pos_3)$. The LSM was used to reduce the dimension of the feature vector. Finally, 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. The details can be found in [2]. Given the linguistic features, the task of the expression prediction is building an MLP based non-linear transform f to map the linguistic feature vectors $\mathcal{L}$ to the expression vectors $\bar{\mathbf{\Lambda}}$ in the synthesis space, i.e.

$$\bar{\mathbf{\Lambda}} = f(\mathcal{L}, \mathbf{W}) \tag{22}$$

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

In this work, the expression prediction method in [23] was combined with the framework of SEF by mapping the linguistic features to the points in the expression subspace in SEF. To build the connection between the linguistic feature space and the expression subspace for SEF, 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 obtained by projecting the training speech utterance into the expression subspace with the ML criterion. From the process mentioned above, the MLP based expression predictor is trained by the speech data and can be shared with the training of the speech synthesiser. Since the training speech can be from different speakers, a speaker dependent expression predictor can be trained.

The 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 the CAT weight vector training, i.e.

$$e(\mathbf{W}) = -\sum_i \frac{1}{|T_i|}(\bar{\boldsymbol{\lambda}}^{(i)\mathsf{T}}\mathbf{z}_{\mathrm{E}}^{(i)} - \frac{1}{2}\bar{\boldsymbol{\lambda}}^{(i)\mathsf{T}}\mathbf{X}_{\mathrm{EE}}^{(i)}\bar{\boldsymbol{\lambda}}^{(i)}) \tag{23}$$

$$\hat{\mathbf{W}}^k = \mathbf{W}^k - \eta\frac{\partial e(\mathbf{W})}{\partial \mathbf{W}^k}, \quad k = 1...L \tag{24}$$

where $\mathbf{W}^k$ is the weight matrix of layer k and $\mathbf{W} = \{\mathbf{W}^1, \ \dots \ , \mathbf{W}^L\}$ is the set of weight matrices, $\bar{\boldsymbol{\lambda}}^{(i)}$ is the MLP output CAT weight vector for training sample i. The normalization parameter $|T_i|$ represents the duration of utterance i, and it was used to ensure that the contributions of the different training utterances are equal. $\mathbf{X}_{\mathrm{EE}}^{(i)}$ and $\mathbf{z}_{\mathrm{E}}^{(i)}$ are the sufficient statistics for CAT weight training accumulated from utterance i. In this work, the expression prediction was performed in the framework of SEF. Thus the $\mathbf{X}_{\mathrm{EE}}^{(i)}$ and $\mathbf{z}_{\mathrm{E}}^{(i)}$ should be accumulated in the expression subspace only, using equation 12. The training process of the expression predictor in the framework of SEF is shown in Fig. 5.

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

[1] In this work, 'neutral' speech, i.e. speech which can be described as not expressing a particular emotion or speaking style, was used to estimate the speaker transforms based on equation 4

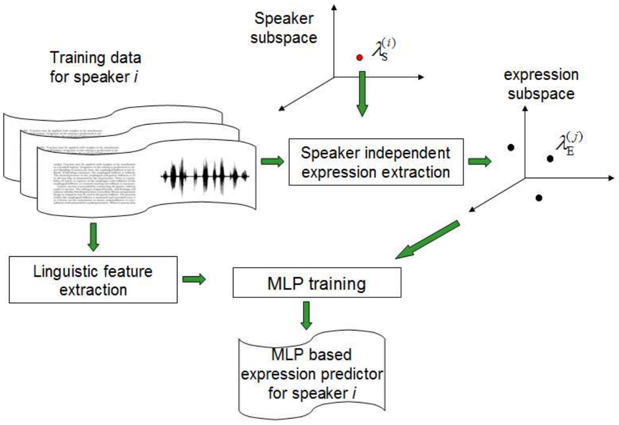

Speaker subspace
$\lambda_S^{(i)}$
Training data for speaker *i*
expression subspace
Speaker independent expression extraction
$\lambda_E^{(j)}$
Linguistic feature extraction
MLP training
MLP based expression predictor for speaker *i*

Fig. 5. *Training of transplantable expression predictor*

the linguistic feature vectors as input, and the expressions in the expression space as target output, the MLP based expression predictor was trained using the ML criterion. Note, in Fig. 5, although the expressions in the expression subspace are speaker independent, the finally trained expression predictor from text is speaker dependent. That means, different speaker dependent expression predictors may project the same linguistic feature into different points in the expression subspace. The expression predictor based on SEF projects the expressive linguistic features into the expression subspace of SEF which is shared by all the speakers. It means the predicted expressions can be transplanted between different speakers. Thus, the way in which a particular speaker interprets the text and expresses it as an emotion in speech can be transplanted to other speakers. This process is shown in Fig. 6. The expression predictor projects the linguistic feature from text to a CAT weight vector $\boldsymbol{\lambda}_E^{(j)}$ in the expression subspace, and $\boldsymbol{\lambda}_E^{(j)}$ can be composed with different speakers in the speaker subspace, e.g. $\boldsymbol{\lambda}_S^{(i)}$ and $\boldsymbol{\lambda}_S^{(k)}$, so that the expressive speech of speaker i and speaker k can be generated with similar expressions.

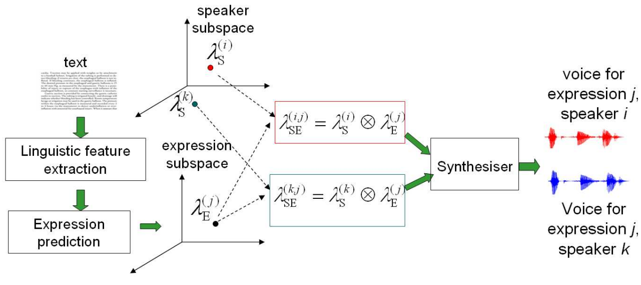

speaker subspace
$\lambda_S^{(i)}$
text
$\lambda_S^{(k)}$
Linguistic feature extraction
expression subspace
Expression prediction
$\lambda_E^{(j)}$
$\lambda_{SE}^{(i,j)} = \lambda_S^{(i)} \otimes \lambda_E^{(j)}$
$\lambda_{SE}^{(k,j)} = \lambda_S^{(k)} \otimes \lambda_E^{(j)}$
Synthesiser
voice for expression *j*, speaker *i*
Voice for expression *j*, speaker *k*

Fig. 6. *Synthesis with transplanted expressions*

VI. Experiments

A. Data preparation and model training

The experiments presented here are based on publicly available audiobooks from Librivox.org. The training data contained about 28 hours of recordings from 4 audiobooks (2 male and 2 female speakers). The lightly supervised sentence alignment and selection method [15] was used to transform the audiobooks into training data usable for speech synthesis purposes. The data was further segmented into 3 types of speech units or utterances: narration, carrier and direct speech [6]. A rule based neutral data selection was performed based on acoustic features such as f0-range, RMS-amplitude-range, etc [27]. This resulted in 5 hours of neutral training data which was used to initialize the speaker clusters and the speaker CAT weight vectors. The speech data from two extra audiobooks (1 male and 1 female speakers) was used as test data. Detailed information about the training and test data is given in table I.

TABLE I
Audiobooks used for training and testing

	spkr	length		audiobook	narrator
		full	neutral		
train	m1	8.65h	1.51h	“A Tramp Abroad” by Mark Twain	John Greenman
	f1	7.44h	1.53h	“The Beautiful and Damned” by F. Scott Fitzgerald	E. Tavano
	m2	8.40h	1.54h	“The Damnation of Theron Ware” by Harold Frederic	Greg W.
	f2	3.34h	0.49h	“What Katy Did” by Susan Coolidge	Karen Savage
test	m3	-	0.36h	“Bacon” by Richard W. Church	Bill Boerst
	f3	-	0.07h	“Olive” by Dinah Maria Craik	Arielle Lipshaw

Table II lists the speaking styles and recording conditions for each of the audiobooks used for training and testing. Recording quality of librivox audiobooks is often not at the same level as carefully conducted studio recordings. Typical problems are changes in recording level across sessions or changes in the distance to the microphone as well as noticeable background noises from page turns, mouse clicks, traffic noise, etc. However, recording quality is often sufficient for building synthetic voices e.g.the audiobook “A Tramp Abroad” read by John Greenman which was used in this article for speaker m1 has also been used successfully as training data in the Blizzard Challenge 2012 [16]. All the audiobooks selected for this article have reasonable recording quality and acceptable speaking styles as judged by the authors. All of the audiobooks are also solo recordings, i.e. a single speaker is narrating a whole book and for each of them there is usually more than 4 hours of speech data to choose from.

The sampling rate of the training speech was 16kHz and acoustic features consisted of 40 mel-cepstral coefficients, logF0, 21 (approximately bark scaled) BAP plus their delta and delta-delta information. The models were 5 state left-to-right multi-space probability distribution hidden semi-Markov models.

The CAT model used in this work consisted of 8 cluster

TABLE II
SPEAKING STYLES AND RECORDING CONDITIONS IN AUDIOBOOKS

spkr	speaking style	recording
m1	very expressive & character voices	small changes in rec-level across sessions, occasional background noise, changes in dist-to-mic
f1	some expressive speech & no distinct character voices	some variations in loudness & dist-to-mic, occasional background noise
m2	some expressive speech & no distinct character voices	very little rec-level variations across sessions
f2	some expressive speech & some character voices	very small rec-level variations across sessions
m3	not very expressive & no character voices	very little rec-level variations across sessions
f3	some expressive speech & no distinct character voices	changes in rec-level across sessions, some room reverberation & variations in dist-to-mic

models: 1 bias cluster model, 4 non-bias cluster models for speaker modelling and 3 non-bias cluster models for expression modelling. The CAT training process based on unsupervised SEF is shown in Fig. 7.

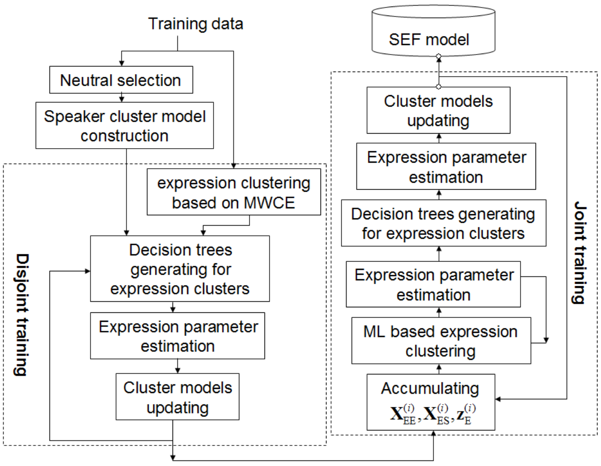

Training data
Neutral selection
Speaker cluster model construction
expression clustering based on MWCE
Decision trees generating for expression clusters
Expression parameter estimation
Cluster models updating
Disjoint training
SEF model
Cluster models updating
Expression parameter estimation
Decision trees generating for expression clusters
Expression parameter estimation
ML based expression clustering
Accumulating
$\mathbf{X}_{\text{EE}}^{(i)}, \mathbf{X}_{\text{ES}}^{(i)}, \mathbf{z}_{\text{E}}^{(i)}$
Joint training

Fig. 7. *Training process of SEF based on CAT*

At first, the automatically selected neutral training data was used to construct a speaker CAT model with a standard CAT training process, i.e. the speaker decision trees, speaker CAT weight vectors and speaker cluster models are iteratively updated until convergence. Then, a disjoint SEF training process was performed to build an initial expression space. In the disjoint training process, the minimum within class error (MWCE) based expression clustering was carried out to group the training speech into P_{E} clusters, where P_{E} is the dimension of expression CAT weight vectors. Based on the expression clustering results, the expression subspace of the disjoint method was constructed. Again, a standard CAT training process was used to update the decision trees, expression CAT weights and the cluster models alternately. After the disjoint training, the joint SEF training was performed. The disjoint SEF training constructs an initial expression subspace in which the statistics of joint SEF training, i.e. $\mathbf{X}_{\text{EE}}^{(i)}$, $\mathbf{X}_{\text{ES}}^{(i)}$ and $\mathbf{z}_{\text{E}}^{(i)}$ are accumulated. Then, the expression clustering and the expression CAT weight estimation were performed in a joint optimization process. It was followed by a standard CAT training process to construct the expression subspace with the joint method. Note, the disjoint SEF training only provides an initial expression space to accumulate the initial statistics for joint training. It does not define the initial expression states for joint SEF. After one iteration of joint training, a new expression subspace based on the joint method is constructed and the statistics for joint SEF can be re-calculated in the new expression space. The joint SEF training can be performed iteratively until convergence.

After the expression CAT weights and cluster models were trained, the speaker CAT weight vectors and cluster models can be re-estimated in a similar way. However, in this work, the re-estimation of the speaker part was skipped due to limited time for computing.

In the joint SEF training of this work, 242 expressions were generated from the training speech. A histogram of the expression distribution w.r.t. the number of speakers related to the expression is shown in Fig. 8. It shows that 46% of the expressions are related to all 4 speakers, 16% of the expressions are related to 3 speakers etc. It indicates the degree of overlap between speakers and expressions in the SEF training.

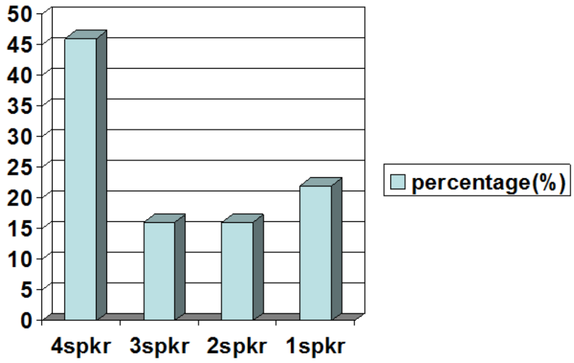


Fig. 8. *Expression distribution w.r.t. the number of speakers*

B. *Audio data based expression transplantation*

The first set of experiments investigated the performance of transplanting the expressions extracted from the audio data. Supervised adaptation was used to extract the expressions from audio data, as shown in Fig. 4. In this part of the experiments, the adaptation speech is from training speaker “m1”, which is a male speaker with very expressive speaking style. The expressions from the speech of ”m1” were transplanted to each training and test speaker. Both the within-gender transplantation (i.e. to ”m2”, “m3”) and cross-gender transplantation (i.e. to ”f1”, “f2”, “f3”) were investigated. In order to investigate the system performance more accurately in this part of the experiments, the results are shown for each training speaker and each test speaker separately.

In the first experiment, two SEF strategies for audiobook data were investigated. One is the disjoint method, the other is the joint method. Based on the two methods, the CAT models were trained separately. Then using the supervised adaptation process in Fig. 4, expressive speech with different speakers was generated using two CAT models separately. An ABX test was performed to evaluate the synthetic speech. The natural speech used for expression adaptation was used as reference in an ABX test. The subjects listened to the synthetic speech from 2 systems and were asked which one is expressively closer to the reference speech. The ABX test set contains 75 randomly selected evaluation utterances from “A Tramp Abroad” which were not used in model training, including 40 narrations, 10 carriers and 25 direct speech utterances. The results for the training speakers and the new speakers are shown in table III and table IV respectively.

TABLE III
ABX TEST FOR TWO SEF STRATEGIES, TRAINING SPEAKER

spkr	gender	joint	disjoint	p
m1	m	55.0%	45.0%	0.007
m2	m	56.7%	43.3%	<0.001
f1	f	51.6%	48.4%	0.241
f2	f	52.1%	47.9%	0.168
overall		54.3%	45.7%	<0.001

TABLE IV
ABX TEST FOR TWO SEF STRATEGIES, TEST SPEAKER

spkr	gender	joint	disjoint	p
m3	m	54.8%	45.2%	0.014
f3	f	54.1%	45.9%	0.029
overall		54.0%	46.0%	0.002

Table III and table IV indicate that the joint SEF method achieves significantly better results than the disjoint SEF method for the expressiveness of synthetic speech. According to these results, in all the remaining experiments, the joint SEF system was used.

Two aspects of the SEF are investigated here: (1) How close does the synthetic speech sound to the original natural speech when the expressions from the natural speech data are transplanted to different speakers, and (2) How close does the synthetic speech from different speakers sound when the same expression is transplanted to different speakers? When the same expressions are transplanted to different speakers, ideally, the synthetic speech from different speakers should sound expressively similar.

An ABX test was used to evaluate the expressiveness of the synthetic speech. The expressive speech based on SEF methods was compared to the neutral speech from the same speaker. To generate the neutral speech, a fixed, approximately expression free point in the expression subspace was defined by the automatically selected neutral training data. This expression free point was used to represent the expression parameters of neutral speech and it was interpolated to the speaker parameters to generate the synthetic neutral speech for different speakers. The reference speech of the ABX test is the natural speech from “m1”, i.e. the adaptation speaker. The results are shown in table V and table VI for training speakers and test speakers respectively. The p-value calculation can be found in [28].

TABLE V
ABX TEST: SEF VS. NEUTRAL, TRAINING SPEAKER

spkr	joint	neutral	p
m1	59.2%	40.0%	<0.001
m2	57.0%	43.0%	<0.001
f1	53.3%	46.7%	0.053
f2	49.1%	50.9%	0.34
overall	55.8%	44.2%	<0.001

TABLE VI
ABX TEST: SEF VS. NEUTRAL, TEST SPEAKER

spkr	joint	neutral	p
m3	59.3%	40.7%	<0.001
f3	58.3%	41.7%	<0.001
overall	59.2%	40.8%	<0.001

The results in table V and VI indicate that the synthetic speech generated by the SEF method is significantly closer to the original natural speech in expressions than neutral speech.

Although the results in table V and VI show that overall, the proposed method significantly improves the expressiveness of the synthetic speech, for different speakers, the improvement from the proposed method is inconsistent, e.g. the performance for speaker ”f3” is much better than ”f1” and ”f2”. This shows the difficulty of the cross-gender expression transplantation. The original expressions were extracted from “m1”, and it consistently improved the expressiveness of the synthetic speech of speakers with same gender, e.g. “m2” and “m3”. However, for the cross-gender transplantation, the performance is not as stable as the within-gender transplantation.

The ABX results in table V and VI are using the natural speech from “m1” as reference to evaluate the expressiveness of the synthetic speech from different speakers. The speaker factor may influence listeners’ judgement, e.g. listeners may prefer a voice with higher speaker similarity rather than the expressive similarity. To complement the ABX results, a preference test based on paragraph reading was performed to compare the voice before and after the expression transplantation. Since the reference speech is not needed in a preference test, the speaker factor can be removed from the listeners’ judgements. The preference test was based on 15 paragraphs with an average of 3 utterances per paragraph. The test paragraphs were randomly selected from the chapters of the book ”A Tramp Abroad”, which were not used in model training. The listeners were asked to choose the version which expressed an appropriate emotion for the content of the paragraph. The results are shown in table VII.

Table VII indicates that the transplanted expressions improved the expressiveness of the synthetic paragraphs. Similar

TABLE VII
PREFERENCE TEST FOR PARAGRAPH READING TESTING, SEF VS. NEUTRAL

spkr	predictor	neutral	nopref	p
m1	47.4%	28.8%	23.7%	0.007
m2	57.2%	41.6%	1.2%	0.02
m3	57.3%	36.7%	6.0%	0.02
f1	45.9%	42.9%	11.2%	0.351
f2	43.6%	42.9%	13.5%	0.469
f3	61.7%	35.3%	3.0%	<0.001

to the ABX results, the within-gender transplantation achieved more stable performance than cross-gender transplantation.

The next experiment was evaluating the consistency of the synthetic speech when an expression was transplanted to different speakers. If the expression extracted from the speech data of one speaker is transplanted to different speakers, the synthetic speech of different speakers should sound expressively close to the synthetic speech of the first speaker, meanwhile they should sound expressively close to each other as well. Thus the synthetic speech from a speaker, i.e. “m1” who provided the adaptation data was used as reference and the synthetic speech from other speakers, i.e. “m2”, “m3”, “f1”, “f2” and “f3”, was compared to it. A 5-point DMOS score was used, where 5 meant exactly like the reference and 1 meant completely different from the reference. The results are shown in Fig. 9.

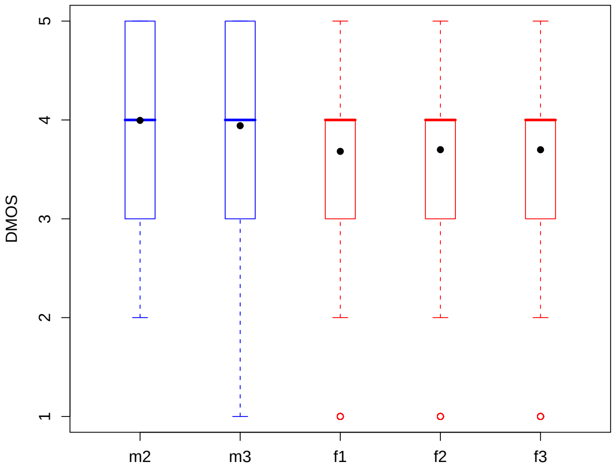


Fig. 9. *DMOS results for consistency*

Fig. 9 shows, that for all the speakers, the synthetic speech with transplanted expressions received quite good DMOS results when compared to the original speaker. An interesting phenomenon is that all the male speakers, i.e. “m2” and “m3” achieved almost the same scores, while, all the female speakers, i.e. “f1”, “f2” and “f3” achieved similar scores as well. This indicates that the transplanted expressions have very good portability over different speakers. Ideally, the speakers with different genders should also get similar scores. However, in this experiment, the male speakers achieved significantly better scores than the female speakers. One possible reason is that the reference data was from a male speaker “m1”. Thus, although the listeners were required to ignore the speaker difference, they still had the tendency to give higher scores to the speakers with the same gender. Another possible reason is again due to the difficulty of the cross-gender SEF. Transplanting the expressions from male voice to female voice does not achieve the same performance as the within-gender transplantation.

ABX and DMOS tests only check the similarity of synthetic speech to human speech. To evaluate the impact of the transplanted expressions on the intelligibility and naturalness of synthetic speech, a 5-point MOS test was performed. Listeners were required to score the synthetic speech in terms of voice quality, and the results are shown in Fig. 10. The results show that the transplanted expressions do not degrade the voice quality of the synthetic speech.

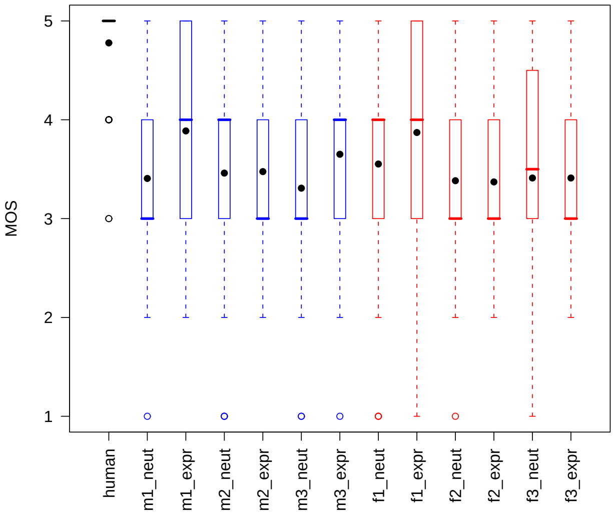


Fig. 10. *MOS results for voice quality*

C. Transplantation experiments based on expression prediction from text

In this set of experiments, the expressions generated from a speaker dependent text-to-expression predictor were transplanted. That means, the way in which a particular speaker interprets and reads a text was transplanted to other speakers. Listening tests were performed to evaluate how this speaker specific information can improve the expressiveness of the synthetic speech of other speakers. Since the SEF method allows the speech data from different speakers to be projected into the common expression space, the speaker independent expression predictor can be trained using multi-speaker training data. Because in speaker independent expression predictors, the inter-speaker variability is assumed to be normalized, the impact of the speaker specific information on the expression prediction performance can be investigated by comparing the speaker dependent and independent expression predictors.

Two speaker dependent expression predictors based on the integrated method were trained using the speaker dependent training speech. One predictor was trained by using training data from the male speaker “m1”, with 10.2k utterances. The other was trained using the data from female speaker “f1”, with 6.8k utterances. The MLP expression predictor includes 3 hidden layers, and 100 neurons for each hidden layer.

The output dimension is the same as the dimension of the expression subspace, i.e. 3.

The first experiment investigated the expressiveness of the synthetic speech based on the expression predictors. The expressions generated by the expression predictor from text were used to synthesise paragraphs and then they were compared to the neutral TTS systems. This experiment only investigated expressiveness. Thus the synthetic voice was generated for the training speaker, and the predicted expressions were not transplanted to the new speakers. The preference test was performed to evaluate the paragraph synthesis based on 15 paragraphs which are the same as the preference test in the audio based expression transplantation experiments. The results are shown in table VIII.

TABLE VIII
PREFERENCE TEST FOR PARAGRAPH READING TESTING THE EXPRESSIVENESS OF EXPRESSION PREDICTORS

spkr	predictor	neutral	nopref	p
m1	53.3%	36.6%	10.1%	0.001
f1	48.1%	36.6%	15.3%	0.017

Table VIII shows, that the synthetic speech from both expression predictors achieved significantly better scores than the neutral versions. This indicates that the integrated expression prediction method works well in the framework of SEF.

The second experiment investigates the transplantation of the expressions generated by the expression predictors. Here, the expressions generated from the speaker dependent expression predictors of “m1” and “f1” were transplanted to a new speaker, i.e. “m2”, and the expressiveness of the synthetic speech was investigated based on the voice of “m2”. Again, a paragraph based preference test was used with the neutral system as the contrast. The results are shown in table IX.

TABLE IX
PREFERENCE TEST FOR PARAGRAPH READING, EXPRESSION TRANSPLANTATION

spkr	predictor		neutral	nopref	p
	m1	f1			
m2	59.9%		33.0%	7.1%	<0.001
m2		46.7%	38.8%	14.5%	0.088

Table IX indicates that the expressions generated by both expression predictors can be transplanted to a new speaker and help to improve the expressiveness of the new speaker.

A speaker dependent expression predictor is trained by the natural speech of a particular speaker and models the fashion in which a particular speaker interprets and reads the text. Therefore the generated expressions are appropriate for the voice of this particular training speaker. One question that needs to be addressed is whether the expressions generated from the predictor of the same speaker are always the best choice or are there similar or even more appropriate expressions from the predictors of other speakers? To address this question, the expressions generated from the speaker dependent expression predictor of speaker “f1” were used to generate the expressive speech for speaker “f1”. Then, for contrast, the expressions from the speaker dependent expression predictor of “m1” were transplanted to the speaker “f1” and were used to generate expressive speech for “f1” as well. In this experiment, the cross-gender transplantation was performed. Ideally, even imperfect expression transplantation may lead to some loss in expressiveness, the very salient expressions from a speaker, e.g. “m1” can still add appropriate expressiveness to the voice of other speakers, e.g. “f1”. The paragraph based preference test result is given in table X.

TABLE X
PREFERENCE TEST FOR PARAGRAPH READING, EXPRESSION PREDICTOR FROM THE SAME SPEAKER VS. FROM ANOTHER SPEAKER

spkr	predictor		nopref	p
	m1	f1		
f1	38.9%	38.2%	22.9%	0.465

Table X shows that for speaker “f1”, the predicted expressions from another speaker, i.e.“m1” are equally appropriate compared to the predicted expressions from the same speaker, i.e. “f1”. This result indicates that for a particular speaker, the predicted expressions from the training speaker are not necessarily the best choice to synthesise the expressive speech. The transplanted expressions from other speakers may be appropriate as well.

Finally, the inter-speaker factors for the expression predictor from text are 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 this work, a multi-speaker expression predictor was trained by 22.7k training utterances from 3 speakers. Since multi-speaker training data was used, this expression predictor was assumed to be speaker independent. The multi-speaker expression predictor which was labelled as “MS” was compared to the speaker dependent expression predictor for “m1” and “f1” in a paragraph reading preference test. The result is shown in table XI.

TABLE XI
PREFERENCE TEST FOR PARAGRAPH READING, EXPRESSION PREDICTORS WITH SINGLE SPEAKER TRAINING DATA VS. MULTI-SPEAKER TRAINING DATA

spkr	SD		MS	nopref	p
	m1	f1			
m1	54.8%		35.2%	10.0%	<0.001
f1		38.4%	37.1%	24.6%	0.422

When the expression predictor is trained with multi-speaker training data there are two aspects to consider. On one hand, introducing multi-speaker training data increases the amount of training samples which leads to more reliable parameter estimation. On the other hand, the inter-speaker variability is normalized in the expression predictor trained by the multi-speaker data. Compared to the results in table XI, if the speech data from a speaker contains very salient speaker specific characteristics or styles, multi-speaker training data will normalize the speaker specific information which is very important to improve the performance of the expression predictor and degrade the expressiveness of the synthetic speech, as the result of speaker "m1" in the first row of table XI shows. When the speech data from a speaker does not contain salient style, multi-speaker training data will not degrade the expressiveness performance of the synthesis system, even if the speaker specific information is normalized, as the result of speaker "f1" in the second row of table XI shows. The results in table XI show how the speaker specific information influences the performance of text-to-expression prediction.

D. Speaker similarity experiments

In a good expression transplantation system, the transplanted expressions should improve the expressiveness of the synthetic speech of the target speakers. Meanwhile, the voice of the target speakers should be still identifiable after transplantation. Since multi-speaker speech data with the same content is not available from the audiobooks, this work generated the expressions from a text-to-expression predictor and transplanted the predicted expressions to other speakers. The speaker dependent expression predictor from "m1" was used to predict the expression for the utterances from different speakers using the transcripts of the speech utterances, and the predicted expressions were transplanted to the corresponding speaker to synthesise the expressive speech. An ABX test was performed to compare the speaker similarity between the synthetic speech with transplanted expressions and without transplanted expressions (i.e. neutral speech). The original natural speech utterances for each speaker were used as reference and the size of the test set is 40 utterances for each speaker. The listeners were asked to choose which speaker sounds more like the reference speaker. The results are shown in table XII and table XIII for the training and test speakers respectively.

TABLE XII
ABX FOR SPEAKER SIMILARITY: TRANSPLANTED VS. NEUTRAL, TRAINING SPEAKER

spkr	transplanted	neutral	p
m1	48.0%	52.0%	0.298
m2	50.0%	50.0%	0.500
f1	52.0%	48.0%	0.298
f2	66.7%	33.3%	<0.001
overall	53.3%	46.7%	0.025

TABLE XIII
ABX FOR SPEAKER SIMILARITY: TRANSPLANTED VS. NEUTRAL, TEST SPEAKER

spkr	transplanted	neutral	p
m3	60.3%	39.7%	0.005
f3	68.2%	31.8%	<0.001
overall	60.2%	39.8%	<0.001

In general, the transplanted expressions do not degrade the speaker similarity of the synthetic speech. It is surprising that for some speakers, the synthetic speech with transplanted expressions achieved a significantly better speaker similarity score than neutral synthetic speech. A possible explanation is that the test speech utterances were randomly selected. Thus, some expressive speech utterances may be included. When listeners are asked to judge the speaker similarity, they may prefer synthetic speech with transplanted expressions, if the reference speech is expressive. In order to investigate how big are the listeners choices influenced by expressiveness similarity in a speaker similarity test, following experiment was performed. The expressions from the natural speech of m1 was transplanted to the test speaker m3. The synthetic speech from m3 which was assumed to contain similar expressions as the natural speech of m1 was compared to the synthetic speech of m1, using the natural speech of m1 as reference. Again, the listeners were asked to indicate which speaker sounds like the reference speaker. The results are shown in table XIV. It indicates that although the expressions of the reference speech were transplanted to the new speaker m3, it does not influence the distinguishability of the synthetic speech from two speakers. This is consistent with the conclusions in table XII and table XIII, i.e. the transplanted expressions do not influence the speaker similarity significantly.

TABLE XIV
ABX FOR SPEAKER SIMILARITY: CROSS SPEAKER TEST

m1 neutral	m1 expressive	m3 expressive	p
80.1%		19.9%	<0.001
	79.3%	20.7%	<0.001

VII. CONCLUSIONS AND FUTURE WORK

This work investigates methods to build multi-speaker expressive speech synthesis systems. Instead of using acted speech corpora which contain a limited set of pre-defined emotions, this work chose highly diverse speech corpora with very rich expressive information, e.g. audiobook corpora as training data to model the very complex expressions in human speech. Factorization methods have been introduced into this work to factorize the complex acoustic characteristics into several independent components, e.g. speaker, expression etc. Within the CAT framework, a joint SEF method which integrates the expression clustering and parameter estimation into a single ML training process was proposed to build the orthogonal speaker and expression subspaces from the highly diverse unlabelled audiobook data. Based on the orthogonal speaker and expression subspaces, the expressions in the expression subspace are shared by different speakers and can be transplanted between speakers. Since the expressions in human's language can be perceived through two ways: from speech data and from text data, correspondingly, in this work, the multi-speaker expressive speech synthesis system has two work modes. In the first mode, the expressive speech utterances are given. The expressions from the expressive speech are extracted and transplanted to different speakers. In the second mode, the adaptation speech is not provided. Thus it is a full expressive synthesis system in which a speaker dependent text-to-expression predictor is used to extract the expressions from plain text and the predicted expressions are transplanted to different speakers. This is equivalent to transplanting the fashion in which a particular speaker interprets emotions in text and converts it into expressions in speech to other speakers. Experimental results showed that in both of the two modes, the generated expressions can be transplanted from one speaker to other speakers successfully and significantly improved the expressiveness of the synthetic speech for multiple speakers. Finally, the importance of the speaker specific information in the task of text-to-expression prediction has been investigated by comparing the performance between speaker dependent expression predictors and speaker independent expression predictors.

Although the joint method for the unsupervised SEF proposed in this work is based on the subspace based models, e.g. CAT, this framework can be generalized to other factorization schemes, e.g. methods based on linear transform or the combination of subspace-based methods and the method based on linear transform. In future work, the joint method to train the SEF system with unlabelled data will be applied to other factorization schemes. Meanwhile, based on the SEF framework, the speaker independent expression predictor can be trained. Although in this work, the experimental results showed that the speaker independent expression predictor did not achieve the performance of the speaker dependent expression predictor, it is assumed that the “canonical model” contains the general information to map the text to the expressions which is independent of speakers. In future work, the speaker dependent data can be used as “adaptation data” to adapt the “canonical model” to a “speaker dependent model” to predict the expression from text in the style of a particular speaker.

REFERENCES

[1] 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.

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

[3] 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.

[4] T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, “Speaker interpolation in HMM-based speech synthesis system,” *Journal of the Acoustical Society of Japan*, vol. 21, no. 4, pp. 199–206, 2000.

[5] 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.

[6] 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.

[7] 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.

[8] V. Wan, J. Latorre, K. Chin, L. Chen, M. J. F. Gales, H. Zen, K. Knill, and M. Akamine, “Combining multiple high quality corpora for improving HMM-TTS,” in *Proc. of Interspeech*, 2012.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] S. King and V. Karaiskos, “The Blizzard Challenge 2012,” in *Proc. of Blizzard Challenge Workshop in Portland, Oregon, USA*, 2012.

[17] S. King and V. Karaiskos, “The Blizzard Challenge 2013,” in *Proc. of Blizzard Challenge Workshop in Barcelona, Catalonia*, 2013.

[18] 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.

[19] C. Strapparava and R. Mihalcea, “Semeval-2007 task 14: Affective text,” in *Proc. of 4th International Workshop on Semantic Evaluations*, 2007.

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

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

[22] 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.

[23] 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.

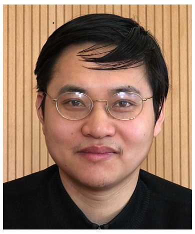
[24] Y.Q. Wang and M. J. F. Gales, “An explicit independence constraint for factorised adaptation in speech recognition,” in *Proc. of INTERSPEECH*, 2013.

[25] V. Sethu, E. Ambikairajah, and J. Epps, “Speaker normalization for speech based emotion detection,” in *Proc. of 15th international conference of digital signal processing*, 2007.

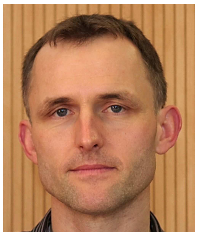
[26] C. Busso, A. Metallinou, and S. S. Narayanan, “Iterative feature normalization for emotional speech detection,” in *Proc. of ICASSP*, 2011.

[27] N. Braunschweiler and S. Buchholz, “Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality,” in *Proc. of Interspeech*, 2011.

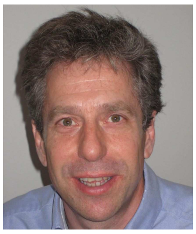
[28] S. Buchholz, J. Latorre, and K. Yanagisawa, “Crowd sourced assessment of speech synthesis,” in *Crowd Sourcing for Speech Processing: Applications to Data Collection, Transcription and Assessment*, M. Eskenazi, G.A. Levow, H. Meng, G. Parent, and D. Suendermann, Eds. Wiley & Sons, 2013.



Langzhou Chen received the B.E. degree from Nanjing University of Aeronautics and Astronautics, China in 1993, the M.A. from Huazhong University of Science and Technology, China in 1996, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China in 1999. From 1999 to 2005 he worked at LIMSI-CNRS, France. Since 2005 he has been a Research Engineer in the speech technology group at Toshiba Research Europe Ltd, Cambridge Research Laboratory, in Cambridge, UK. His research interests include statistical speech recognition, pattern recognition, language modeling, acoustic model training and adaptation, decoding, discriminative training, and HMM based speech synthesis.



Norbert Braunschweiler received the M.A. degree in physics and German language and literature from the University of Konstanz, Germany in 1994, and the Ph.D degree in applied linguistics from the University of Konstanz, Germany, in 2003. From 2000 to 2003, he has worked at the Institute of Natural Language Processing (IMS) at the University of Stuttgart, Germany as a Research Engineer in the SMARTKOM project. In 2004 he worked for Rhetorical Systems Ltd, in Edinburgh, Scotland as a Research Engineer. Since 2005 he has been a Senior Research Engineer in the speech technology group at Toshiba Research Europe Ltd, Cambridge Research Laboratory, in Cambridge, UK. His research interests include human-technology interaction, speech synthesis, expressive speech recognition and synthesis, automatic detection of prosodic cues and phonetics.



Mark J. F. Gales studied for the B.A. in Electrical and Information Sciences at the University of Cambridge from 1985-88. Following graduation he worked as a consultant at Roke Manor Research Ltd. In 1991 he took up a position as a Research Associate in the Speech Vision and Robotics group in the Engineering Department at Cambridge University. In 1995 he completed his doctoral thesis: Model-Based Techniques for Robust Speech Recognition supervised by Professor Steve Young. From 1995-1997 he was a Research Fellow at Emmanuel College Cambridge. He was then a Research Staff Member in the Speech group at the IBM T.J.Watson Research Center until 1999 when he returned to Cambridge University Engineering Department as a University Lecturer. He was appointed Reader in Information Engineering in 2004. He is currently a Professor of Information Engineering (appointed 2012) and a Professorial College Lecturer and Official Fellow of Emmanuel College. He is also technical advisor and consultant at Toshiba Research Europe in Cambridge, UK since 2009. Mark Gales is a Fellow of the IEEE and was a member of the Speech Technical Committee from 2001-2004. He was an associate editor for IEEE Signal Processing Letters from 2009-2011 and is currently an associate editor for IEEE Transactions on Audio Speech and Language Processing. He is also on the Editorial Board of Computer Speech and Language.