Phoneme-level Text to Audio Synchronization on Speech Signals with Background Music

Agnès Pedone, Juan José Burred, Simon Maller and Pierre Leveau

Audionamix
114, avenue de Flandre, 75019 Paris, France
{agnes.pedone,juan.jose.burred,simon.maller,pierre.leveau}@audionamix.com

Abstract

We address the task of synchronizing a given phoneme transcription with the corresponding speech signal, when the latter is linearly mixed with background music. To that end, we propose a new method based on Non-negative Matrix Factorization in the time-frequency domain, which models the speech as a source-filter factorization that includes a synchronization parameter matrix. Phoneme models, which consist of collections of basic spectral envelopes, are learned from a training set of isolated speech. The model is subjected to an iterative Maximum Likelihood optimization that concurrently estimates pitch, synchronization parameters and the contribution of the music part. Results show the feasibility of the system for application in text-informed audio processing and automatic subtitle synchronization.

Index Terms: voice synchronization, non-negative matrix factorization, source-filter model, information retrieval.

1. Introduction

Automatic text to audio synchronization can be used in several applications such as subtitling, karaoke and on-line retrieval of song lyrics. In addition, having the text aligned with the voice provides valuable information that can be exploited in text-informed audio processing for synthesis, separation or transformation applications. Given the spoken or sung text, synchronization consists in finding the time stamps aligning each synchronization unit (usually sentences, words or phonemes) with the audio signal. While high synchronization performances have been previously obtained for clean, isolated voice signals, we focus here on the more challenging task of dealing with voice signals mixed with a background signal. In particular, we address mixtures of speech with music or effects, such as film or TV soundtracks. Also, we aim at synchronizing at the phoneme-level, rather than at word or sentence level. Such a high precision is needed for the potential use of the system in phoneme-level informed tasks such as the ones mentioned.

Most previous work dealing with mixed signals concern singing voice recognition and synchronization in music. In [1, 2] synchronization is performed after a preprocessing step that segregates the voice signal. The aim is to reduce the accompaniment and to locate and extract the vocal part. After a re-synthesis of the vocal melody, the second step uses more traditional methods of voice processing (such as cepstral feature extraction, forced Viterbi alignment and Hidden Markov Models) to perform the actual alignment. An alternative approach is to perform synchronization directly on the mixture, without a previous voice separation stage. The more recent work of Fujihara *et al.* [3] goes in this direction: they identify the phonemes directly on the mixed signal by using specific voice and residual models and an iterative parameter estimation. They adopt a source-filter model for the voice, and use prior training to create a collection of spectral envelope templates.

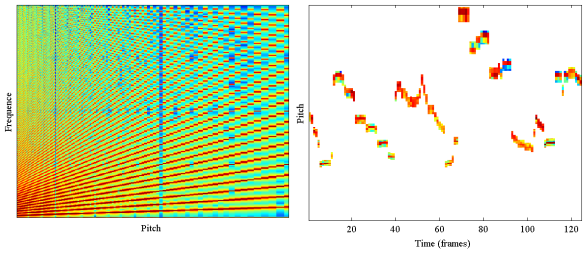
The approach we propose falls into the second category, namely the synchronization directly on the mixed signal. It is based on a matrix factorization model proposed by Durrieu *et al.* [4] for source separation purposes. In order to segregate the voice, they use a model of the mixture as the sum of two independent sources: the singing voice and the music. For the voice part, they formulate a source-filter model adapted to a matrix factorization framework, and for the music part a Non-negative Matrix Factorization (NMF) model is used. All matrices are then estimated with an iterative method based on gradient descent. This approach allows the separate treatment of the voice filter part, which corresponds to the spectral envelope characteristics of phonemes.

We extend that model by introducing a synchronization parameter matrix into the source-filter part of the model. As such, errors in synchronization contribute to the overal cost function to be minimized. Our approach is supervised: a set of phoneme models are learned from a database of isolated utterances. Apart from synchronization, we additionally evaluate the system in a blind phoneme recognition task, without any prior knowledge of the text. We anticipate that results on blind recognition are poor: the model is not accurate enough for that task. However, synchronization results with a given text are encouraging and are expected to help text-informed audio processing applications, as well as serving as a basis for word-level or sentence-level subtitle synchronization. In Sect. 2, we present the proposed synchronization model, and detail the methods used for learning (Sect. 2.1), recognition (Sect. 2.2) and synchronization (Sect. 2.3). Finally, Sect. 3 will present the evaluation method, databases and obtained results.

2. Proposed model

Our model is based on the one proposed by Durrieu *et al.* in [4], to which we add a synchronization constraint. The model is defined in the time-frequency domain in terms of matrices representing time-varying Power Spectral Densities (PSDs), defined as the squared magnitude of the Short Time Fourier Transform (STFT). The observed PSD corresponding to a mixture signal $x(t)$ is thus represented by matrix $\mathbf{X}(f,t) = |\mathrm{STFT}(f,t)\{x(t)\}|^{\cdot 2}$ of size $F \times T$, where F is the number of frequency bins and T is the number of time frames. The operator $\mathbf{A}^{\cdot b}$ denotes element-wise exponentiation.

We consider having two statistically independent sources: the speech and the music. Because of their assumed indepen-



(a) Glottal dictionary $\mathbf{W}_E$ (b) Pitch activations $\hat{\mathbf{H}}_E$

Figure 1: Voice source matrices.

dence (and therefore additivity of powers), the resulting modeled PSD ($\mathbf{D}$) is the addition of both voice PSD ($\mathbf{V}$) and music PSD ($\mathbf{M}$):

$$\mathbf{X}(f,t) \simeq \mathbf{D}(f,t) = \mathbf{M}(f,t) + \mathbf{V}(f,t). \quad (1)$$

The music part is modeled by a generic NMF (the element indices will be henceforth omitted):

$$\mathbf{M} = \mathbf{W}_M \mathbf{H}_M, \quad (2)$$

where $\mathbf{W}_M$ and $\mathbf{H}_M$ are respectively the basis matrix or *dictionary* of spectral bases (of size $F \times K_M$) and the temporal coefficients or *activation matrix* (of size $K_M \times T$), where K_M, the number of bases in the dictionary, is a parameter to be determined. Under such a general factorization model, the modeled signal is interpreted as a weighted sum of columns of the dictionary (spectral bases), where the weights are given by the activation matrix for each time frame. The rows of the activation matrix describe the temporal evolution of the contribution of each spectral basis to the observed PSD.

The voice part follows a source-filter model of the form

$$\mathbf{V} = \mathbf{E} \circ \mathbf{F} = [\mathbf{W}_E \mathbf{H}_E] \circ [\mathbf{W}_F \mathbf{H}_F \mathbf{S}], \quad (3)$$

where $\mathbf{E}$ (for *excitation*) is the PSD matrix contributed by the glottal source, $\mathbf{F}$ is the PSD matrix contributed by the spectral envelope filtering of the vocal tract, and the operator $\circ$ denotes the Hadamard (element-wise) product. The source and filter matrices are in turn subjected to a further factorization: $\mathbf{W}_E$ and $\mathbf{H}_E$ are respectively the dictionary and the activation matrix of the voice source part and $\mathbf{W}_F$ and $\mathbf{H}_F$ the dictionary and the activation of the voice filter part. In the filter part, we have introduced the square $T \times T$ *synchronization matrix* $\mathbf{S}$, whose effect will be detailed below.

The source dictionary matrix $\mathbf{W}_E$ is always known beforehand and fixed during model optimization. It contains a collection of glottal source harmonic combs for a range of fundamental frequencies, as generated by the KLGLOTT model [5] (see Fig. 1(a)). The source activation matrix $\mathbf{H}_E$ is initialized with a Gaussian distribution over the f_0s, with parameters set to reflect a typical range of the human voice, and updated during optimization. At the end of the optimization process, it should contain the degree of presence of each harmonic comb (and thus, of each f_0), as a function of time. In other words, it should visualize the melody or, in case of speech, the prosody (see Fig. 1(b)). The initialization and handling of the filter-part matrices depend on the desired task (recognition or synchronization), and will be detailed in the following subsections.

The number of components of the source-part factorization (K_E) is fixed and equal to the number of harmonic combs generated by the glottal model. The number of components of the filter-part factorization (K_F) is determined by preliminary experiments, as will be explained in Sect. 2.1.

The final model to be optimized is thus given by

$$\mathbf{D} = \mathbf{W}_M \mathbf{H}_M + [\mathbf{W}_E \mathbf{H}_E] \circ [\mathbf{W}_F \mathbf{H}_F \mathbf{S}]. \quad (4)$$

Parameter estimation is performed by an iterative gradient descent method comparing model $\mathbf{D}$ and observation $\mathbf{X}$ given a certain cost function

$$\mathcal{D}(\mathbf{X}|\mathbf{D}) = \sum_{f=1}^{F} \sum_{t=1}^{T} d([\mathbf{X}]_{ft} | [\mathbf{D}]_{ft}), \quad (5)$$

where $d(x|y)$ is a given element-wise distance measure. As shown in [6] for the case of NMF-based source separation, an appropriate choice for audio signal processing is the Itakura-Saito (IS) divergence:

$$d_{IS}(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1 \quad (6)$$

due to its property of scale invariance. Furthermore, gradient descent based on the IS divergence has been shown [6] to be equivalent to Maximum Likelihood (ML) optimization if the variables are supposed to be Gaussian and independent, and if the problem is defined for PSDs instead of STFT amplitudes (hence the power additivity assumption mentioned above).

2.1. Learning

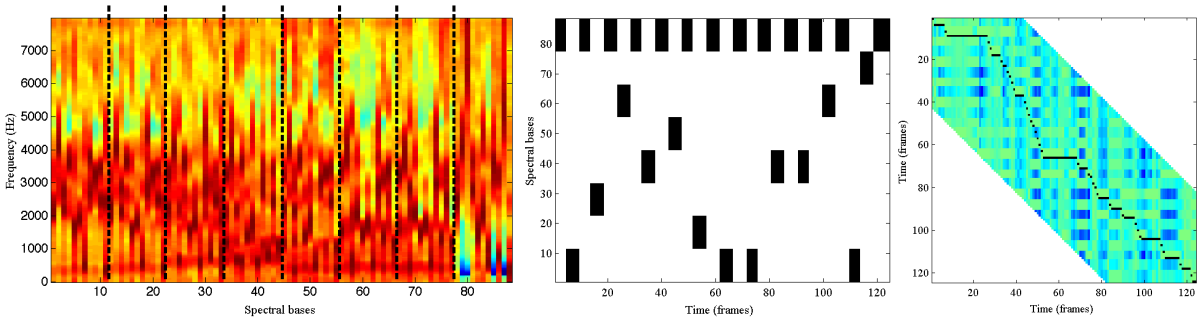
The phoneme models are learned as follows. First, we create sound files containing all concatenated occurrences of each phoneme in the training database. We subject each concatenated phoneme file to ML optimization according to the model of Eq. 4. To that end, source matrices $\mathbf{W}_E$ and $\mathbf{H}_E$ are initialized as explained before. The synchronization matrix is not used, so it is set to an identity matrix: $\mathbf{S} = \mathbf{I}$. All other matrices are updated after being randomly initialized. The parameter vector (i.e., the set of all matrices allowed to be updated between iterations) for the learning task is:

$$\theta_p^{learn} = \left\{ \mathbf{W}_{M_p}, \mathbf{H}_{M_p}, \mathbf{H}_{E_p}, \mathbf{W}_{F_p}, \mathbf{H}_{F_p} \right\}, \quad (7)$$

where p is the phoneme index. The remaining component number parameters K_{M_p} and K_{F_p} are determined by preliminary cross-validation tests. Even if the NMF part is supposed to model the musical accompaniment, which is absent from the training set, it was found useful to keep an NMF model with a single component ($K_{M_p} = 1$) in order to model potential generic noise present in the recordings.

The learned model for phoneme p is the estimated $\hat{\mathbf{W}}_{F_p}$ dictionary matrix, which contains spectral envelopes that, when combined, should correspond to typical spectral envelopes of that phoneme (the $\hat{}$ notation denotes estimation or learning). For vowels, the combination of the spectral basis should follow the characteristic formant structures. Finally, the final learned phoneme dictionary is constructed by concatenating all phoneme-wise dictionaries $\hat{\mathbf{W}}_{F_p}$ into the learned dictionary matrix $\hat{\mathbf{W}}_F$, with $K_F = P \cdot K_{F_p}$, where P is the total number of phonemes and K_{F_p} is supposed the same for all phonemes.

Learning was based on 13 vowel models and one additional *noise model* which corresponds to the aggregate effect of noises



(a) Phoneme envelope dictionary $\mathbf{W}_F$. Vertical dashed lines separate the individual phonemes.

(b) Phoneme activations $\mathbf{H}_F^B$.

(c) Synchronization matrix $\hat{\mathbf{S}}$ with DTW path.

Figure 2: Voice filter matrices.

and consonants, and thus presents a rather flat spectral envelope. Fig. 2(a) shows an example of learned spectral envelope matrix for $P = 8$ phonemes and $K_{F_p} = 11$ spectral bases per phoneme. The 11 rightmost columns of the matrix correspond to the consonant/noise model.

2.2. Recognition

For recognition, we suppose that the phonemes envelope dictionary is known and fixed, given by the learned matrix $\hat{\mathbf{W}}_F$. However, we do not know the phoneme sequence. Hence, the activation matrix of the filter part $\mathbf{H}_F$, as well as the fundamental frequency matrix $\mathbf{H}_E$, are randomly initialized and blindly estimated. The same applies to the NMF part. The synchronization matrix is again set to identity: we do not separate the activations and synchronization matrix, since the sequence of activations should be fully determined by the $\mathbf{H}_F$ matrix. The parameter vector for recognition is thus:

$$\theta^{recog.} = \{\mathbf{W}_M, \mathbf{H}_M, \mathbf{H}_E, \mathbf{H}_F\}\,. \tag{8}$$

Recognition results are obtained by observing the estimated matrix $\hat{\mathbf{H}}_F$. Adding the activations by phonemes and locating the maximum reveals the most likely phoneme at each frame.

2.3. Synchronization

Like for recognition, the phoneme envelope dictionary $\hat{\mathbf{W}}_F$ is known. But now the phoneme sequence is known and given by the text. This allows to initialize the matrix $\mathbf{H}_F$ following the expected phoneme sequence. This is done as follows: we suppose that each phoneme is equally distributed in time along the duration of each sentence, and we introduce blocks of ones in the concerned phoneme locations, the rest of the matrix consisting of zeros. We will denote a matrix initialized in this way by $\mathbf{H}_F^B$ (B stands for *binary*). An example of matrix $\mathbf{H}_F^B$ initialized from the text is shown on Fig. 2(b).

The synchronization matrix $\mathbf{S}$ allows to modulate the location and duration of the phoneme-wise blocks of the $\mathbf{H}_F$ matrix. Its effect is to temporally warp the matrices multiplying it from the left in order to find the best temporal match between model and observation. Matrix $\mathbf{S}$ is initialized as a band matrix $\mathbf{B}(W)$, i.e., a matrix with an "enlarged diagonal" filled with ones up to a distance of W elements from the diagonal, W being the maximum warping factor permitted. An example of optimized synchronization matrix is shown in Fig. 2(c).

Algorithm 1 ML update rules for synchronization

1: $\mathbf{H}_E \leftarrow \mathbf{H}_E \circ \frac{\mathbf{W}_E^T[(\mathbf{W}_F\mathbf{H}_F\mathbf{S})\circ\mathbf{D}^{\cdot-2}\circ\mathbf{X}]}{\mathbf{W}_E^T[(\mathbf{W}_F\mathbf{H}_F\mathbf{S})\circ\mathbf{D}^{\cdot-1}]}$

2: $\mathbf{H}_M \leftarrow \mathbf{H}_M \circ \frac{\mathbf{W}_M^T(\mathbf{D}^{\cdot-2}\circ\mathbf{X})}{\mathbf{W}_M^T\mathbf{D}^{\cdot-1}}$

3: $\mathbf{W}_M \leftarrow \mathbf{W}_M \circ \frac{(\mathbf{D}^{\cdot-2}\circ\mathbf{X})\mathbf{H}_M^T}{\mathbf{D}^{\cdot-1}\mathbf{H}_M^T}$

4: $\mathbf{S} \leftarrow \mathbf{S} \circ \frac{(\mathbf{W}_F\mathbf{H}_F)^T[\mathbf{D}^{\cdot-2}\circ\mathbf{X}\circ(\mathbf{W}_E\mathbf{H}_E)]}{(\mathbf{W}_F\mathbf{H}_F)^T(\mathbf{D}^{\cdot-1}\circ(\mathbf{W}_E\mathbf{H}_E))}$

The parameter vector for the synchronization task is

$$\theta^{synch.} = \{\mathbf{W}_M, \mathbf{H}_M, \mathbf{H}_E, \mathbf{S}\}\,. \tag{9}$$

The corresponding update rules are obtained from non-negative gradient descent (see [6]), and are given for this task in Algorithm 1. It should be noted that in this case we are concurrently estimating pitch (via $\mathbf{H}_E$) and performing synchronization (via $\mathbf{S}$) under the same optimization process. Table 1 summarizes all initialization and updating rules for this and the previous tasks.

After optimization, to obtain the synchronization results we proceed as follows:

1. **DTW of $\hat{\mathbf{S}}$.** Since only one column of $\mathbf{H}_F$ (one phoneme) is supposed to be observed at a time, we apply Dynamic Time Warping (DTW) to $\hat{\mathbf{S}}$ in order to find the highest similarity path. Fig. 2(c) shows (superimposed in black) an example of DTW path found from an estimated $\hat{\mathbf{S}}$ matrix. Note that horizontal segments in the path correspond to phoneme duration adjustments, and vertical segments are forbidden, again in order to avoid mixing several phonemes at a time.
2. **Warping of $\mathbf{H}_F^B$.** Once the DTW path has been found, the phoneme sequence activation matrix $\mathbf{H}_F^B$ is temporally warped by the product

$$\tilde{\mathbf{H}}_F^B = \mathbf{H}_F^B \cdot \mathrm{DTW}\{\hat{\mathbf{S}}\}. \tag{10}$$

Finally, the sequence of synchronized phonemes is given by the non-zero entries for each column of $\tilde{\mathbf{H}}_F^B$.

3. Evaluation and results

For learning, we use the concatenated occurrences of 13 vowels (aa, ae, ah, ax, axr, eh, er, ih, ix, iy, uh, uw, ux) from all speakers of the training subset of the TIMIT database [7] (which is

Task	Music		Source		Filter		
	$\mathbf{W}_M$	$\mathbf{H}_M$	$\mathbf{W}_E$	$\mathbf{H}_E$	$\mathbf{W}_F$	$\mathbf{H}_F$	$\mathbf{S}$
Learning	random	random	KLGLOTT	Gauss	random	random	$\mathbf{I}$
Recognition	random	random	KLGLOTT	Gauss	$\hat{\mathbf{W}}_F$	random	$\mathbf{I}$
Synchronization	random	random	KLGLOTT	Gauss	$\hat{\mathbf{W}}_F$	$\mathbf{H}_F^B$	$\mathbf{B}(W)$

Table 1: Summary of initialization rules. Shaded cells denote matrices being updated during optimization.

task	mix	ACC	PRC	RCL	F-score
Recognition	0 dB	**13** $\pm$ 5	**8** $\pm$ 5	**22** $\pm$ 14	**11** $\pm$ 7
	10 dB	**16** $\pm$ 5	**9** $\pm$ 5	**25** $\pm$ 14	**13** $\pm$ 7
Synchro.	0 dB	**62** $\pm$ 10	**35** $\pm$ 13	**51**$\pm$ 16	**41** $\pm$ 14
	10 dB	**66** $\pm$ 9	**39** $\pm$ 14	**50** $\pm$ 16	**43** $\pm$ 14

Table 2: Evaluation results (the values are in % $\pm$ standard deviation across mixes.)

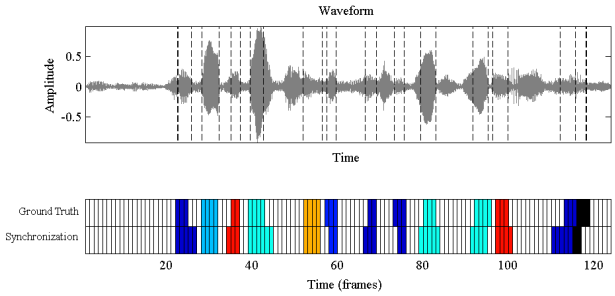


Figure 3: Example of phoneme synchronization for a 10 dB mixture. Different colors denote different phonemes.

annotated by phonemes), and the concatenation of non-vowel phonemes for the noise model. For the tests, we use linear mixtures of 8 sentences out of the TIMIT test subset (which does not overlap with the training set) with 9 randomly chosen music excerpts. We build two test groups: the first contains the voice mixed at the same average energy level as the music (mixtures at 0 dB) and the second contains the voice mixed 10 dB above the music (a more realistic case). This makes a total of 144 test mixtures containing one sentence each. All the files are monaural and sampled at 16 kHz. For PSD extraction, a Hamming window of 40 ms with a hop size of 20 ms is used.

For evaluation, we use the traditional criteria from Information Retrieval: accuracy, precision, recall and F-score, based on a frame-by-frame comparison of ground truth and estimated phonemes. In our case, the relevant outputs (positives) are the vowels, and the non-relevant outputs (negatives) are the consonant and music frames. Given the number of true positives (TP: correctly detected vowels), true negatives (TN: correctly detected consonant or music frames), false positives (FP: consonant and music frames incorrectly labeled as vowels) and false negatives (FN: vowel frames incorrectly labeled as consonants or music), accuracy is defined as $ACC = \frac{TN+TP}{TN+TP+FN+FP}$, recall is defined as $RCL = \frac{TP}{TP+FN}$, precision as $PRC = \frac{TP}{TP+FP}$, and F-score as $F = 2\frac{PRC \cdot RCL}{PRC+RCL}$. The final quality judgment should be mainly based on the F-score, since it gives the best compromise between PRC and RCL. Accuracy is often misleading: it can be high even if no vowels are detected, since usually most of the frames in the ground truth contain noise or non-relevant phonemes.

The results are presented on Table 2. As expected, the results for the 10 dB mixtures are better than for the the 0 dB mixtures. However, performance difference in terms of F-score is of only 2%, which indicates a good robustness against the background signal. It also can be seen that blind recognition is not feasible by this approach, resulting in low F-scores. However, knowing the text and performing synchronization improves the F-score by 30%, attaining a best average performance of 43% F-score and 66% accuracy. A visual example of a good synchronization result is shown in Fig. 3.

4. Conclusions

We have implemented and evaluated phoneme-level synchronization between text and speech mixed with music, under a matrix factorization ML framework. Synchronization at the phoneme level reaches an F-score of 43% and an accuracy of 66%. These results suggest that the resulting alignment of phonemes can help text-informed audio processing tasks. Also, we deem the performance adequate for the synchronization of subtitles, since aggregating the phoneme-level results to the word-level or sentence-level is expected to increase the F-score.

To further increase performances (and to make blind phoneme recognition feasible), better phoneme models will be investigated. A possibility is to add temporal models based on n-grams or Hidden Markov Models. Also, the currently used noise model is too general and will need to be refined. Finally, deviations from canonical pronunciations need to be addressed.

5. References

[1] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic Synchronization between Lyrics and Music CD Recordings Based on Viterbi Alignment of Segragated Vocal Signals," in *Proc. 8th IEEE Int. Symposium on Multimedia (ISM'06), San Diego, USA*, 2006.

[2] A. Mesaros and T. Virtanen, "Recognition of Phonemes and Words in Singing," in *Proc. 35th Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2010), Texas, USA*, 2010.

[3] H. Fujihara, M. Goto, and H. G. Okuno, "A Novel Framework for Recognizing Phonemes of Singing Voice in Polyphonic Music," in *Proc. IEEE Workshop on Application of Signal Processing to Audio and Acoustics, New Paltz, USA*, 2009.

[4] J.-L. Durrieu, G. Richard, and B. David, "An Iterative Approach to Monoral Musical Mixture De-Soloing," in *Proc. 34th Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2010), Taipei, Taiwan*, 2009.

[5] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *Journal of the Acoustical Society of America*, vol. 87, pp. 820–857, 1990.

[6] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative Matrix Factorization with the Itakura-Saito Divergence. With Application to Music Analysis," *Neural Computation*, vol. 21, pp. 793–830, 2009.

[7] J. Garofolo and coworkers, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *Linguistic Data Consortium, Philadelphia, USA*, 1993.