Articulatory Feature Extraction from Voice and Their Impact on Hybrid Acoustic Models

Jorge Llombart, Antonio Miguel, and Eduardo Lleida

ViVoLab,
Aragon Institute for Engineering Research (I3A),
University of Zaragoza, Spain
{jllombg,amiguel,lleida}@unizar.es
http://www.vivolab.es

Abstract. There is a great amount of information in the speech signal, although current speech recognizers do not exploit it completely. In this paper articulatory information is extracted from speech and fused to standard acoustic models to obtain a better hybrid acoustic model which provides improvements on speech recognition. The paper also studies the best input signal for the system in terms of type of speech features and time resolution to obtain a better articulatory information extractor. Then this information is fused to a standard acoustic model obtained with neural networks to perform the speech recognition achieving better results.

Keywords: Articulatory features, Neural network, Hybrid models.

1 Introduction

Speech production is a complex process which has attracted a wide research activity in the last decades to obtain articulatory information embedded in the speech signal. The motivation of this work is to study how articulatory information can improve phoneme classification accuracy. As shown in [1], [2] and [3], is possible to take advantage of phoneme similarities to build articulatory class specific classifiers, which we use to provide additional inputs for a phoneme classifier.

To deal with that amount of information and to obtain a better articulatory information representation, we propose to integrate neural networks in a hybrid recognizer. The performance of these models is very sensitive to the input features, therefore in this paper we study two different signal representations, Mel-Frequency Cepstrum Coefficients *MFCC* and a less processed representation, the Mel scaled Filter Bank. It is also important to study the impact of higher temporal resolution in the feature extraction process, that we think it may help convolutional networks to compensate time label misalignments, which usually degrade the performance of regular networks.

Once a good representation of the articulatory process is obtained as articulatory features, they may be included to get a hybrid model which takes advantage of them to reduce classification errors. We will show in the experimental

J.L. Navarro Mesa et al. (Eds.): IberSPEECH 2014, LNAI 8854, pp. 138–147, 2014.

section that the fusion of the articulatory information with more standard features reduces significantly recognition errors, thus validating the assumption of articulatory information being helpful to improve speech recognition.

This article is organized as follows. Section 2 describes articulatory features used in this work. Section 3 explains our representation of neural networks. Section 4 describes experimental procedure. In Section 5 the results are shown, and in Section 6 are exposed the conclusions extracted from this study.

2 Articulatory Features

Speech production involves three processes; initiation, phonation, and articulation. The first, initiation is the process in which air starts to flow through vocal tract. During the second process, phonation, vocal chords start to vibrate producing the sound. Finally, in the articulation process, some constrictions are made in the oral cavity to modify the produced sound.

During the articulation process, the constrictions made on the vocal cavity perturbate natural air flow. The location and type of constriction imprints specific information in the speech signal that we propose to extract and use to improve speech recognizers.

Constrictions could be done in different places, or by different manners. In order to study the speech production process, articulatory features have been described by phonology, which also studies their relation to human vocal tract. Those articulatory features explain all sounds related with speech and they are pictured on *International Phonetic alphabet (IPA)*. Regarding to the relation between articulatory features, for example the place where the main constriction is made or the shape of lips, they are clustered in independent groups, which offers an opportunity to classify sounds from different points of view and give us additional information to perform the phoneme classification. There are some previous works on speech recognition using only articulatory features to classify phonemes like [1] with good results, or combined with acoustic features [2]. Other works use them for speaker recognition [3]. Besides, there are works that use neural networks for speech recognition with articulatory features like [4].

We consider five different articulatory properties. The first property is **voicing**, which tells if a sound is *voiced* or *unvoiced*, and it is related to vibration of vocal cords. The second property used in this work is **place**, which indicates where is allocated the main constriction. Another property we deal with is the **manner**, which is referred to how the sound is generated, if it is a *nasal* sound or *fricative* and so on. More related to vowel sounds there are two properties, **rounding**, that describes if lips are rounded or not during pronunciation. And finally the **vowel location**, to express the position of the tongue, if it is on the *front* of vocal cavity or on the *back*. Table 1 shows all properties used in this work and the features which are classified in each property. It is important to point that *silence* is included in each property. The class *Silence* does not describe a phoneme but it is included to allow the classifier to deal with audio segments without speech. The same concept is applied to *not representative*, for instance if

the sound corresponds to a consonant, it does not make sense to speak in terms of *rounding*, which is a property that only take place on vowel sounds.

Table 1. Articulatory properties and theirs features

Property	Classes	N Classes
Voicing	Unvoiced, Voiced, Silence	3
Place	High, Medium, Low, Labial, Dental, Alveolar, Palatal, Velar, Glottal, Silence	10
Manner	Vowel, Nasal, Fricative, Aproximant-Lateral, Stop, Silence	6
Rounding	Rounded, Not Rounded, Not Representative, Silence	4
Vowel Location	Front, Middle, Back, Not Representative, Silence	5

With those features, now we have to label the database. This process should be done by manual labeling the database, but we propose a simpler labeling process based on the phonetic labels provided by a canonical pronunciation dictionary, since we have word level transcriptions. Then, using the phonetic description of those phonemes we set the attributes which correspond to the articulatory features described on Table 1. In this study it is used the *TIMIT Acoustic-Phonetic Continuous Speech Corpus* in which each phoneme was characterized and pictured in a table with the *Sampa* and *IPA* nomenclatures. The description of the phonemes is based on the work [5] and an extract from this table is shown on Table 2.

Table 2. Extract of *TIMIT* phonemes and their description on *Sampa*, *IPA* and articulatory features. (NR means *Not Representative*).

Mono-phonemes	Sampa	IPA	Voicing	Place	Manner	Rounding	Vowel Location
aa	A:	ɑ	Voiced	Low	Vowel	Not Rounded	Back
ae	{	æ	Voiced	Low	Vowel	Not Rounded	Front
ah	V	ʌ	Voiced	Low	Vowel	Not Rounded	Back
ay	aI	aɪ	Voiced	Low	Vowel	Not Rounded	Front
b	b	b	Voiced	Labial	Stop	NR	NR

3 Artificial Neural Networks

In recent years, there has been an increasing interest in neural networks. This work is based on multi-layer perceptron, a classical architecture of neural networks [6]. Equation (1) describes the mathematical model for an artificial neuron which is given a vector of M elements as input $X = [x_1, x_2, \ldots, x_M]$, where $W = [w_1, w_2, \ldots, w_M]$ are the weights for that input, b is a bias, and $\theta(\cdot)$ is the activation function that applies a non linearity to obtain the output. The training process is based on generalized gradient descent [6].

$$y = \theta\left(\sum_{m=1}^{M} (w_m \cdot x_m) + b\right) \tag{1}$$

The network is composed of tree layers. The input layer, one hidden layer with the internal parameters, and the output layer which transforms the internal parameters to human comprehensive information. Since in this work we use neural networks as classifiers, we choose the cross-entropy as cost function and *softmax*, shown in (2), as activation function for the output layer, where $X = [x_1, x_2, \ldots, x_i, \ldots, x_N]$ is a vector of N elements, where the output vector is normalized and it gives us the probability of being part of each class.

$$\theta(x_i) = \frac{\exp x_i}{\sum_{n=1}^{N} \exp x_n}, \tag{2}$$

Other activation functions used are the *Sigmoid* function for hidden layers, and the *Rectified Linear Unit (ReLU)* on input layer because this type of neurons can regularize the training process both in image [7] and on speech [8].

One of the problems which can be found while processing speech signal with neural networks is that neural networks are prepared for recognizing static patterns, but speech signal is a complex and non stationary signal. One phoneme has a temporal evolution, so in order to recognize better that phoneme it is useful to show to the network the temporal context of the speech signal during the phoneme. That means that it is important to give to the neural network the vector with the calculated features for this time, and a context which consists on the previous and posterior vectors. We suppose that the reference is on central vector, so the label of the overall network input is the label which is referred to the central vector. Using neural networks to learn spectro-temporal patterns makes them very sensitive to the exact label alignment, which can be inaccurate since it is obtained from a reference Hidden Markov Model *HMM*. This effect can be minimized with convolutional neural networks. This type of neural networks can be interpreted as if one neuron is a filter of a windowed input, then this filter is repeated for some displacements of that window. Finally, the maximum activation is selected as output of all those repeated filters. Mathematically, suppose that there is an input composed by feature vectors $X = [x_1, x_2, \ldots, x_M]$ that belongs to K temporal windows, so the input may be written as $X_k = [x_{1,k}, x_{2,k}, \ldots, x_{M,k}]$. There are J filters that we want to compute, therefore the filtering for each time index is $h_k = [h_{k,1}, h_{k,2}, \ldots, h_{k,J}]$ where each filter is represented as (3). To complete the convolutional layer, the output corresponding to the maximum temporal output per each filter as summarized in (4) is called *max pool* stage whose overall output is a vector $P = [p_1, p_2, \ldots, p_J]$ which is the input for the next stages on the neural network.

$$h_{k,j} = \theta\left(\sum_{m=1}^{M} (w_{m,j} \cdot x_{m,k}) + b_j\right) \tag{3}$$

$$p_j = \max_{k=1}^{K} (h_{k,j}) \tag{4}$$

This kind of layer has proved a good behavior on different type of inputs and applications where there is some spatial or temporal variability, and the

convolutional mechanism might regularize the input, like in image [9] or in speech in different ways [10] [11].

4 Experiment Description

This work uses the *TIMIT* Acoustic-Phonetic Continuous Speech Corpus, as mentioned on Section 2, and the phonetic classes are referred to that corpus. This corpus consists on 508 speakers from eight United States' regions, 462 on training set and 50 on testing set. As in previous works, we use the 3696 phrases marked as 'si' and 'sx', and in the testing set 192 *core test* phrases [12]. From the training set a 10% has been separated for development and validation set. For all experiments a bigram language model has been used, and the phone alignment has been obtained from reference *HMMs*. The database has 61 phoneme classes, which have been extended up to 183 classes by using the state index in the *HMM* as label, then each of those classes are composed by the phoneme class and the state, using three states per phoneme [11]. However, in the test process the labels are contracted to the 39 to calculate errors, as described on [13], [11] or [10]. One of the benefits of neural networks is that the forward evaluation at test time is inexpensive in computational terms, even though training in this experiments can take up to four days in a *graphic processing unit (GPU)* architecture.

As mentioned before, neural networks have to take care of the influence of temporal displacements. For the time-frequency analysis, a $25ms$ window is taken to get the frequency analysis, and then this window is displaced $10ms$ to calculate the next frame [14]. It has been suggested that there exists a context of $100ms$ with relevant information around each frame, as said in [15], who in [16] used mutual information to check that hypothesis, and [17] with the same method showed that information remains in cepstrum feature space. Therefore, all that information should be shown to the neural network by stacking a time-frequency matrix with 10 frames around the labeled frame, 5 each side to maintain symmetry, for a total of 11 frames, equivalent to $110ms$ of context. The temporal context used by the convolutional networks input is extended to 15 frames, $150ms$, to allow them to realign it while using a comparable effective context of 11 frames.

Another effect that has been taken into account is the time resolution. As mentioned before, the time-frequency analysis is made by transforming a window of the signal of $25ms$ and then displacing it $10ms$. In order to increase time domain resolution the displacement of analysis window is $5ms$, but to maintain the number of labels, and to allow a comparative between this two types of temporal resolution, the separation between two time-frequency matrices is $10ms$. In other words we can say that the time resolution of time-frequency input matrices has been increased but the number of matrices has been maintained as before. We show that effect in Figure 1, where it is pictured the same phoneme, labeled at the same time instant, but each one with different time resolution.

For the experiments, the *TIMIT* audios are processed in eight different ways taking into account the strategies mentioned before. We used the static and dynamic *MFCC*, and the Mel scaled filter bank with static and dynamic coefficients, which are resumed in Table 3

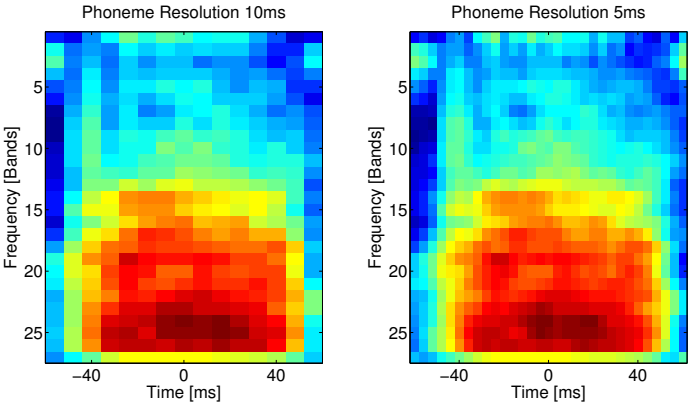

Fig. 1. Time-frequency analysis of a *TIMIT* phoneme. On the left a $10ms$ resolution matrix and on the right the $5ms$ resolution matrix.

Table 3. Input features description

Name	Coefficients description	Analysis displacement
Mfcc EZ 10	12 Mfcc coefficients, energy and cepstral mean subtraction	$10ms$
Mfcc EZ 05	12 Mfcc coefficients, energy and cepstral mean subtraction	$5ms$
Mfcc EACZ 10	12 Mfcc coef., energy, cepstral mean subtraction and dynamic coef.	$10ms$
Mfcc EACZ 05	12 Mfcc coef., energy, cepstral mean subtraction and dynamic coef.	$05ms$
Fb 26 E 10	26 Mel scaled filter bands with energy	$10ms$
Fb 26 E 05	26 Mel scaled filter bands with energy	$05ms$
Fb 26 EDA 10	26 Mel scaled filter bands, energy and dynamic coef.	$10ms$
Fb 26 EDA 05	26 Mel scaled filter bands, energy and dynamic coef.	$05ms$

The architecture of the system is composed of two different parts. The first part is a classificator, which is a neural network of three layers, an input layer, a hidden layer, and the output layer. For those classificators the first layer has 512 *ReLu* neurons, and it can be of two types: a regular mlp layer or a convolutional layer described in Section 3. The hidden layer is formed by 256 *logistic* neurons. And finally, the output layer has the same neurons than classes to classify, whose output is the probability of membership of the example in the input to the class which this output is referred to. The second part of this architecture is a fusion neural network. Once the classificators of phonemes and articulatory properties are obtained, the information that can be achieved from the articulatory properties is shown to the fusion network to improve the classification result. In this work three fusion philosophies are explored. The first one, *Output-Layer-Fusion*, consist on composing an input with the probabilities of the phoneme states and the articulatory features, then a neural network uses this information to perform a phoneme classification. The second strategy is to use the output of the hidden layer, instead of the output layer, to conform an input vector for the fusion network, the *Previous-Layer-Fusion*. The last strategy used is give to the neural network the context of the classified input by stacking the previous and

posterior classification outputs, the *Context-Fusion*. Moreover this fusion network can have two architectures, the fist one is only an input layer followed by an output layer, without hidden layer, and the second type has a hidden layer. These layers follow the philosophy of previous networks and use the same neuron types, with 1024 neurons on input and 512 in hidden layer.

5 Results

The first results obtained are the classification of phonemes as baseline, and the experiment with the first layer as convolutional one. In this experiment we analyzed the accuracy of the neural network output, and of the speech recognition system. The first is measured in terms of Frame by frame Error Rate, *FER*, which counts substitutions, while the second is measured in terms of Phoneme Error Rate, *PER*, which takes into account the substitutions, deletions and insertions. As it is shown on Table 4 a better performance is obtained when convolutional networks are used. Another important point is that when a better time resolution is used a better result is obtained in most of the cases, obtaining the best result, a *PER* of 30.52% on the Fb 26 EDA 05 case in the convolutional case.

Table 4. *FER* and *PER* in phoneme classification. *FER* is calculated for 183 classes and *PER* is calculated for 39 classes.

Features			Baseline		Convolutional	
			FER[%]	*PER*[%]	*FER*[%]	*PER*[%]
Mfcc	EZ	10	49.85	33.55	48.05	32.09
Mfcc	EZ	05	50.18	34.02	48.07	32.27
Mfcc	EDAZ	10	46.57	31.92	47.80	31.94
Mfcc	EDAZ	05	46.25	32.57	47.28	30.72
Fb 26	E	10	49.03	32.57	50.01	32.12
Fb 26	E	05	49.27	32.82	47.90	31.88
Fb 26	EDA	10	45.40	30.90	49.87	32.96
Fb 26	EDA	05	46.12	30.67	46.92	**30.52**

In Table 5 we show the classification *FER* for each articulatory property, for two configurations. In this case the better results are obtained on baseline architecture. This effect can be explained because in these features, the position of events may not be as important as in phoneme classification, since the articulatory classes are less specific. Nevertheless in this case it can be observed as in the phoneme case that using a higher temporal resolution may help classification. As we can see, the best performance has been obtained for higher temporal resolution in almost all cases.

For fusion experiments the Fb 26 EDA 05 convolutional network classificator has been selected to use it as baseline, which provides an accuracy of *PER* of 30.52%, and whose output is fused with the articulatory classificators. For this classificator we used the Fb 26 EDA 05 baseline classificator for each property in order to obtain comparable results and to know which property helps more to the

Table 5. *FER*[%] in articulatory classification. The properties are Voicing, Place, Manner, Rounding and Vowel Location. #Classes means the number of classes of these property.

Features		Baseline					Convolutional				
		Voice	Place	Manner	Roundig	Location	Voice	Place	Manner	Rounding	Location
#Calsses		3	10	6	4	5	3	10	6	4	5
Mfcc EZ	10	9.91	27.59	20.84	14.79	19.93	10.13	25.36	19.64	14.03	16.23
Mfcc EZ	05	9.21	27.23	20.26	14.82	16.60	9.40	25.92	19.32	16.36	16.36
Mfcc EDAZ	10	8.91	24.59	18.54	**13.37**	15.66	9.48	25.23	19.11	14.06	15.92
Mfcc EDAZ	05	8.73	24.63	18.67	13.59	15.24	8.85	25.20	18.20	14.23	15.32
Fb 26 E	10	9.51	26.95	20.05	14.90	17.40	9.66	26.12	17.90	15.14	17.39
Fb 26 E	05	9.67	26.32	20.29	15.13	16.94	9.67	24.90	18.42	13.92	15.82
Fb 26 EDA	10	8.54	**23.90**	18.07	13.51	15.34	9.56	26.09	19.33	14.47	16.65
Fb 26 EDA	05	**8.30**	24.61	**17.59**	13.51	**14.91**	8.96	24.91	18.12	13.62	15.60

Table 6. Fusion experiments. *FER* is calculated for 183 classes and *PER* is calculated for 39 classes.

Properties	Output-Layer-Fusion		Previous-Layer-Fusion		Context-Fusion	
2 Layers	*FER*[%]	*PER*[%]	*FER*[%]	*PER*[%]	*FER*[%]	*PER*[%]
Phoneme + Voicing	45.76	30.01	44.07	27.98	44.68	29.93
Phoneme + Position	45.35	29.92	43.49	27.26	44.39	29.63
Phoneme + Manner	45.31	29.93	43.57	27.79	44.27	29.48
Phoneme + Rounding	45.69	30.37	43.97	27.83	44.24	29.77
Phoneme + Location	45.57	30.12	44.02	28.11	44.10	29.74
Phoneme + All	44.61	29.67	43.21	27.03	43.82	29.66
3 Layers	*FER*[%]	*PER*[%]	*FER*[%]	*PER*[%]	*FER*[%]	*PER*[%]
Phoneme + Voicing	46.27	30.49	44.46	28.20	45.22	30.17
Phoneme + Position	45.67	29.85	43.71	27.64	44.84	29.92
Phoneme + Manner	45.57	29.67	43.61	27.52	44.53	29.77
Phoneme + Rounding	46.02	30.00	44.40	27.97	44.74	29.88
Phoneme + Location	45.92	29.93	44.23	28.09	44.58	29.89
Phoneme + All	45.20	30.06	43.04	**26.96**	44.05	29.86

classification for the same conditions. In Table 6 is shown that *Previous-Layer-Fusion* provides better accuracy than *Output-Layer-Fusion*. Using *Output-Layer-Fusion* might have as drawback that the classification has already been made, and errors can be propagated to the fusion stage. Other interesting effect shown in these results is that the addition of more context to the fusion network, like in *Context-Fusion*, improves the result, even though in these conditions it can not achieve as good results as *Previous-Layer-Fusion*. The motivation of the work is to study how articulatory information can improve phoneme classification accuracy. Since it is harder for a general phoneme classificator to take advantage of phoneme similarities. We propose different methods to train articulatory specific classifiers and to fuse their outputs to improve the accuracy of the system. One last impression of these results may be that *manner* or *position* provide more information for the classification than the other articulatory features. This may be because this two properties are present in all phonemes, so this property add extra information for all phonemes. When we fuse the phoneme classificator, all phonemes have extra information to improve classification. Although it can be seen that the fusion with *manner* or *position* attain the best improvement in accuracy individually. We show that fusion with all articulatory properties pro-

vides an 26.96%*PER* confirming our previous hypothesis. In Table 6 the relative improvement obtained using the different fusion methods ranges from 1% to 11% which is comparable to previous results in similar conditions in the state of the art [2].

6 Conclusions and Future Work

The main motivation of this work is study how articulatory information can improve phoneme classification accuracy, for that we propose to process articulatory information to build specific classifiers which can be used as additional information for the phoneme classifier. The first steps of the current study were to determine the manner in which articulatory features can be extracted from speech signal using neural networks as feature extractor, and use them to complement the acoustic model for using speech recognition. The results of this study indicate that using a less processed input in the frequency domain like Mel scaled filter bank, instead of cepstrum domain input, like *MFCC*, increases accuracy in acoustic neural network models. Moreover it seems that the higher time resolution, the better results, not only in convolutional neural networks which compensate misalignment, but also in simpler architectures, though this may be studied deeper in future works. The other aspect studied in this paper is how articulatory features may perform in an hybrid acoustic model, and the evidence from this study suggests that this kind of models provide a better representation which helps the speech recognition. It is shown that some properties like *position* or *manner* produce a mayor impact on hybrid models, but the relations among all of them in a unified hybrid model achieve the best results.

Further work needs to be done on both lines. It would be interesting to determinate how much time resolution is needed for each type of input to perform articulatory information extraction. The other aspect in which it is appropriated a further research is on fusing this articulatory information. The new techniques on deep neural networks may reach better representation of hybrid models by extracting higher levels of abstraction in the relations between articulatory features and phoneme acoustic models.

Acknowledgments. This work has been supported by the Spanish Government and the European Union (FEDER) through projects TIN2011-28169-C05-02 and INNPACTO IPT-2011-1696-390000.

References

1. Kirchhoff, K.: Robust Speech Recognition Using Articulatory Information. PhD thesis, University of Bielefeld (1999)
2. Kirchhoff, K., Fink, G.A., Sagerer, G.: Combining acoustic and articulatory feature information for robust speech recognition. Speech Communication 37, 303–319 (2002)

3. Leung, K.Y., Mak, M.W., Kung, S.-Y.: Applying articulatory features to telephone-based speaker verification. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, p. I858. IEEE, Montreal (2004)
4. Yu, D., Siniscalchi, S.M., Deng, L., Lee, C.-H.: Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, pp. 4169–4172 (2012)
5. Hieronymus, J.: ASCII phonetic symbols for the world's languages: Worldbet. Journal of the International Phonetic Association (1993)
6. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer
7. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, pp. 807–814 (2010)
8. Toth, L.: Phone recognition with deep sparse rectifier neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6985–6989. IEEE, Vancouver (2013)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
10. Toth, L.: Convolutional Deep Rectifier Neural Nets for Phone Recognition. In: INTERSPEECH, pp. 1722–1726. ISCA, Lyon (2013)
11. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.: Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, pp. 4277–4280 (2012)
12. Garofolo, J., et al.: TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Linguistic Data Consortium, Philadelphia (1993)
13. Lee, K.-F., Hon, H.-W.: Speaker-independent phone recognition using hidden Markov models. IEEE Transactions on Acoustics, Speech, and Signal Processing 37, 1641–1648 (1989)
14. Huang, X., Acero, A., Hon, H.-W.: Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall (2001)
15. Yang, H., van Vuuren, S., Hermansky, H.: Relevancy of time-frequency features for phonetic classification measured by mutual information. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1999, vol. 1, pp. 225–228. IEEE, Phoenix (1999)
16. Yang, H.H., Van Vuuren, S., Sharma, S., Hermansky, H.: Relevance of timefrequency features for phonetic and speaker-channel classification. Speech Communication 31, 35–50 (2000)
17. Segura, J., Benitez, M., Torre, A., de la Rubio, A.: Feature extraction from time-frequency matrices for robust speech recognition. In: INTERSPEECH, Aalborg, Denmark (2001)