Phoneme recognition based on hybrid neural networks with inhibition/enhancement of distinctive phonetic feature (DPF) trajectories

Mohammad Nurul Huda, Kouichi Katsurada and Tsuneo Nitta
Graduate School of Engineering, Toyohashi University of Technology
1-1 Hibarigaoka, Tempaku cho, Toyohashi 441-8580, Japan
E-mail: huda@vox.tutkie.tut.ac.jp

Abstract
In this paper, we introduce a novel distinctive phonetic feature (DPF) extraction method that incorporates inhibition/enhancement functionalities by discriminating the DPF dynamic patterns of trajectories relevant or not. The trajectories of each DPF show a convex pattern when the DPF is relevant and a concave one when irrelevant. The proposed algorithm enhances convex type patterns and inhibits concave type patterns. We implement the algorithm into a phoneme recognizer and evaluate it. The recognizer consists of two stages. The first stage extracts 45 dimensional DPF vectors from local features (LFs) of input speech using a hybrid neural network and incorporates an inhibition/enhancement network to obtain modified DPF patterns, and the second stage orthogonalizes the DPF vectors and then feeds them to an HMM-based classifier. The proposed phoneme recognizer significantly improves the phoneme recognition accuracy with fewer mixture components by resolving coarticulation effects.

Index Terms: Distinctive Phonetic Feature, Recurrent Neural Network, Inhibition/Enhancement Network, Local Features, Multi-Layer Neural Network.

1. Introduction
A new vocabulary word or out-of-vocabulary (OOV) word often causes an “error” or a “rejection” in current hidden Markov model (HMM)-based automatic speech recognition (ASR) systems. To resolve this OOV-word problem, an accurate phonetic typewriter or phoneme recognizer functionality is expected [1][2][3].

Various methods had been proposed to accomplish this phoneme recognition [4][5] and some of them showed acceptable performance. However, most of them based on HMMs have several limitations. For example, a) they need a large number of speech parameters and a large scale speech corpus to negotiate coarticulation effects using context-sensitive triphone models, and b) they need higher computational cost to get acceptable performance.

To resolve the problems of current HMM-based phoneme recognizers, a lower computational cost algorithm with higher recognition accuracy is needed. An articulatory-based or a distinctive phonetic feature (DPF)-based system can model coarticulatory phenomena more easily [6]. In our previous work, a DPF-based feature extraction method was introduced [7], where a multi-layer neural network (MLN) was used to extract DPFs. The DPF-based system i) widens the margin of acoustic likelihood, ii) avoids the necessity of a large number of speech parameters, and iii) incorporates context-dependent acoustic vectors to negotiate dynamics. However, because a single MLN is unable to model longer context, it cannot resolve coarticulation effects precisely.

In this paper, we propose a DPF-based phoneme recognition method for an ASR system, which consists of two stages, to solve the problems of coarticulation. The first stage extracts 45 dimensional DPF vectors from local features (LFs) of input speech using a hybrid neural network (HNN) and incorporates an inhibition/enhancement (In/En) network by discriminating the trajectories of DPF pattern are convex or concave. Convex type patterns are enhanced and concave type patterns are inhibited. On the other hand, the second stage normalizes all the DPF vectors using the Gram-Schmidt orthogonalization procedure before connecting with an HMM-based classifier. Here, the HNN consists of a recurrent neural network (RNN) that represents dynamics in a sequence of acoustic features to resolve coarticulation effects [8][9] and an MLN that reduces fluctuation of DPF patterns. It is expected that the proposed system generates more precise phoneme strings at low computational cost and consequently gives a functionality of a high performance phonetic typewriter.

In this study, from the phoneme recognition performance point of view, we investigate and evaluate three types of DPF-based feature extraction methods together with the conventional method of MFCC. These methods are (i) DPF using MLN [7], (ii) DPF using RNN, (iii) DPF using HNN, and (iv) DPF using HNN with In/En.

The paper is organized as follows: Section 2 discusses the articulatory features. Section 3 explains the system configuration of the existing phoneme recognition methods with the proposed. Experimental database and setup are provided in Section 4, while experimental results are analyzed in Section 5. Finally, Section 6 draws some conclusion.

2. Articulatory features
A phone can easily be identified by using its unique articulatory features or distinctive phonetic feature (DPF)-set [10][11]. The Japanese balanced DPF set [7] for classifying Advanced Telecommunications Research Institute International (ATR) phonemes have 15 elements. These DPF

<table>
<thead>
<tr>
<th>DPF/Phone</th>
<th>a</th>
<th>e</th>
<th>f</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>mora</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>high</td>
<td></td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>low</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nil</td>
<td>-</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>anterior</td>
<td>-</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>back</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nil</td>
<td>-</td>
<td>+</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
values are mora, high, low, intermediate between high and low <nil>, anterior, back, intermediate between anterior and back <nil>, coronal, plosive, affricate, continuant, voiced, unvoiced, nasal and semi-vowel. Table 1 shows a part of this balanced DPF set. Here, present and absent elements of the DPFs are indicated by “+” and “-” signs, respectively.

3. Phoneme recognition systems

3.1 MFCC-based system

The conventional approach of ASR systems uses MFCC as feature vectors to be fed into an HMM-based classifier. This approach gives acceptable accuracy at the expense of higher number of mixture components. Moreover, phoneme recognition performance is below par in this approach.

3.2 DPF-based system using MLN

Figure 1 shows the DPF-based phoneme recognition method using MLN [7]. At the acoustic feature extraction stage, input speech is converted into LFs (LF/Δ, LF/ΔΔ), which represent a variance of spectrum along time and frequency axes by using three-point linear regression (LR) calculation. LFs are input to an MLN with three layers, including two hidden layers, after combining preceding (t-3)-th and succeeding (t+3)-th frames with the current t-th frame. Each frame of LF consists of 25 values (12Δ+12ΔΔ+ΔP). The MLN has 45 output units (15x3) corresponding to a context-dependent DPF vector that consists of three DPF vectors of a preceding context DPF, a current DPF, and a following context DPF with 15 dimensions each. The two hidden layers consist of 256 and 96 units, respectively. The MLN is trained by using the standard back-propagation algorithm. The DPF-based method yields comparable recognition performance to the MFCC-based, but requires fewer Gaussian mixture components in the HMM. However, the single MLN suffers from an inability to model dynamic information precisely.

3.3 Proposed system

Figure 2 shows a block diagram of the proposed HNN-based phoneme recognition system. The proposed phoneme recognizer consists of two stages. The first stage extracts 45 dimensional DPF vectors from LFs of input speech using an HNN and incorporates an In/En network to obtain modified DPF patterns, and the second stage normalizes all the DPF vectors using the Gram-Schmidt orthogonalization procedure before connecting with an HMM-based classifier.

3.3.1 Hybrid neural network

The HNN consists of a RNN that represents dynamics in a sequence of acoustic features to resolve coarticulation effects [8][9] and an MLN that reduces fluctuation of DPF patterns. The external input acoustic vector at time t, for the RNN, is formed by taking preceding (t-3)-th and succeeding (t+3)-th frames together with the current t-th frame. Each input frame is composed of 25 LF values that are same as the DPF-based phoneme recognition using MLN described in Section 3.2. The RNN outputs 45 DPF values of which 15 are for the preceding frame, 15 for the current frame and the rest for the succeeding frame. Next, the MLN outputs 45 DPF values for the current input frame by reducing DPF fluctuation.

3.3.2 Inhibition/Enhancement network

An In/En network is used to obtain modified patterns for the DPF patterns outputted by the HNN. An algorithm for this network is given below:

Step1: Apply a three-point median filter operation on each dimension of the DPF vectors to reduce abnormal fluctuation.

Step2: For each element of the DPF vectors, find the acceleration (∆∆) parameters by using three-point LR.

Step3: Check whether ∆∆ is positive (concave type pattern) or negative (convex type pattern) or zero (steady state).

Step4: The convex type pattern or the steady state pattern in-between convex type patterns is enhanced by multiplying it by c1. The concave type pattern or the steady state pattern in-between concave type patterns is inhibited by dividing it by c2.

3.3.3 Gram-Schmidt orthogonalization

The Gram-Schmidt orthogonalization procedure is used to
normalize preceding, current and following feature vectors with respect to current feature vector. The algorithm is given below:

Let \( \{ \mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2 \} \) be the context-dependent DPF vectors \{Preceding DPF, Current DPF, Following DPF\}.

(A) \( \mathbf{z}_1 \) is normalized by its norm.
\[
\mathbf{z}_1^N = \frac{\mathbf{z}_1}{||\mathbf{z}_1||}
\]

(B) \( \mathbf{z}_0 \) is decorrelated against \( \mathbf{z}_1^N \).
\[
\mathbf{z}_0^R = \mathbf{z}_0 - \left( \mathbf{z}_0 \cdot \mathbf{z}_1^N \right) \mathbf{z}_1^N
\]
\[
\mathbf{z}_0^N = \frac{\mathbf{z}_0^R}{||\mathbf{z}_0^R||}
\]

(C) \( \mathbf{z}_2 \) is decorrelated against \( \mathbf{z}_0^N \) and \( \mathbf{z}_1^N \).
\[
\mathbf{z}_2^R = \mathbf{z}_2 - \left( \mathbf{z}_2 \cdot \mathbf{z}_1^N \right) \mathbf{z}_1^N - \left( \mathbf{z}_2 \cdot \mathbf{z}_0^N \right) \mathbf{z}_0^N
\]
\[
\mathbf{z}_2^N = \frac{\mathbf{z}_2^R}{||\mathbf{z}_2^R||}
\]

4. Experiments

4.1 Speech Database

A subset of the Acoustic Society of Japan (ASJ) continuous speech Database consisting of 4513 sentences uttered by male and female speakers are used as training data. JNAS (Japanese Newspaper Article Sentences) [12] consisting of 284 sentences uttered by male and female speakers are used as test data. Sampling rate is 16 KHz.

4.2. Experimental setup

Frame length and frame rate are set to be 25 ms and 10 ms, respectively. MFCC consists of a vector of 38 dimensions (12 MFCC, 12\( \Delta \), 12\( \Delta \Delta \), AP and \( \Delta \Delta \Delta \), where P is log energy of raw signal). On the other hand, LF consists of a vector of 25 dimensions (12\( \Delta \), 12\( \Delta \Delta \), \( \Delta \)). For frame-by-frame phoneme evaluation, the Mahalanobis distance with a single state and a single mixture component, and the Euclidean distance metric are used for the MFCC-based and the DPF-based systems, respectively and the following methods are investigated.

i) MFCC.
ii) DPF using MLN [7].
iii) DPF using RNN.
iv) DPF using HNN.

For phoneme recognition, we use a standard monophone-based HMM classifier with 5-state 3-loop left-to-right models. The number of mixture components in the HMM is varied between mixture=1, 2, 4, 8, 16. The input features for the classifier are

i) MFCC.
ii) DPF using MLN [7].
iii) DPF using RNN.
iv) DPF using HNN.
v) DPF using HNN with In/En.

For the In/En network, c1 and c2 are set to 4.0 and 4.0, respectively observing the data patterns just before inserting into the HMM. The frame-level accuracy, comparing the evaluated phoneme sequence with the referenced phoneme sequence, is calculated by the following formula.

\[
\text{frame correct rate} = \left(1 - \frac{\text{No. of erroneous frames}}{\text{Total no. of frames}}\right) \times 100\%
\]

Figure 3: Phoneme distances for /ia/ utterance, using a) MFCC-based system b) DPF using MLN, and c) DPF using HNN.

Figure 4: The effect of second stage MLN.
Table 2: Frame-level accuracy for investigated methods.

<table>
<thead>
<tr>
<th>Investigated Methods</th>
<th>Frame Correct Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>27.93%</td>
</tr>
<tr>
<td>DPF using MLN</td>
<td>59.13%</td>
</tr>
<tr>
<td>DPF using RNN</td>
<td>69.37%</td>
</tr>
<tr>
<td>DPF using HNN</td>
<td>74.95%</td>
</tr>
</tbody>
</table>

Table 3: Phoneme recognition rate for investigated methods.

<table>
<thead>
<tr>
<th>Investigated Methods</th>
<th>Number of mixture components</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mix.1</td>
</tr>
<tr>
<td>MFCC</td>
<td>67.53%</td>
</tr>
<tr>
<td>DPF using MLN</td>
<td>70.68%</td>
</tr>
<tr>
<td>DPF using RNN</td>
<td>71.56%</td>
</tr>
<tr>
<td>DPF using HNN</td>
<td>74.84%</td>
</tr>
<tr>
<td>DPF using HNN+ In/En</td>
<td>87.60%</td>
</tr>
</tbody>
</table>

5. Experimental results and discussion

Table 2 shows frame correct rates (FCRs) of all the investigated methods. MFCC shows 27.93%, which is the worst over all the methods. The DPF using HNN provides the highest FCR 74.95%. The DPF extractor using MLN generates FCR 59.13%, which is 15.82% lower than the HNN based DPF. It may be mentioned here that RNN based DPF provides FCR 69.37%, but the additional MLN increases the rate by 5.58%.

Next, Table 3 shows phoneme recognition rates for all the investigated methods. The proposed method with In/En shows the highest accuracy at all the investigated mixture components and gives more than 90% accuracy at the mixture components of four, eight, and 16.

Figure 3 shows phoneme distances of five Japanese vowels for each input vector of utterance /aia/. Here, the Mahalanobis distance and the Euclidean distance metric are used for the MFCC-based and the DPF-based systems, respectively. The number of misclassified vowels for the MFCC-based system, the DPF using MLN and the DPF using HNN are 11, nine and four, respectively. Here, the HNN-based DPF extractor reduces vowel misclassifications at phoneme boundaries, while the MLN-based extractor shows more errors at boundaries. Therefore, the HNN-based extractor solves coarticulation effects more widely.

Lastly, Figure 4 describes the effect of the second stage MLN in the phoneme classification performance. The RNN shows greater deviation from ideal boundaries, while the HNN minimizes the deviation and hence, reduces vowel misclassifications.

Moreover, it is claimed that the proposed method requires less computation than the other existing methods because higher recognition accuracy is achieved with fewer mixture components. On the other hand, the MLN-based and the MFCC-based methods require an additional number of mixture components to get higher accuracy. It may be mentioned that each HMM-based classifier needs \(O(mST)\) computation time to calculate output probability at Viterbi search, where \(m\), \(S\) and \(T\) represent the number of mixture components, the number of states and the number of observation sequences, respectively.

Because the proposed method uses an HNN and monophone-based HMMs instead of triphone models to resolve coarticulation effects, it does not need a large number of speech parameters. On the other hand, a triphone model needs at least \((2F+1)mS\) speech parameters, where \(F\), \(m\) and \(S\) represent the feature vector dimensions, the number of mixture components, and the number of states, respectively. Moreover, a large number of triphone models need a large scale speech corpus.

6. Conclusion

This paper has presented a phoneme recognition system that provides high recognition accuracy with fewer mixture components by resolving coarticulation effects using an HNN and an In/En network. Moreover, it is expected that the system will avoid the necessity of a large scale speech corpus and a large number of speech parameters. Future works are further improvement of the algorithm by incorporating Japanese syllable-based sub-words as language models.

Acknowledgement

This work was supported in part by Global COE program “Frontiers of Intelligent Sensing”, MEXT, Japan.

REFERENCES