A COMPARISON OF FRONT-END ANALYSES FOR THAI SPEECH RECOGNITION

Montri Karnjanadecha and Patimakorn Kimsawad

Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla, THAILAND 90110
montri@coe.psu.ac.th

ABSTRACT

The main purpose of this work was to find a suitable front-end analysis for Thai speech recognition by comparing the performance of the LPCC, MFCC and DCTC front-ends using several Thai continuous digit recognition tasks. HTK tools were used to build a word-based HMM recognizer which could handle several styles of digit string realization. Experimental results show that MFCC and DCTC perform equally well, and outperform LPCC. Dynamic parameters were also added to the feature sets to improve modeling accuracy. This work can be regarded as a baseline for further study of Thai speech recognition.

1. INTRODUCTION

Research work on Thai speech recognition started almost two decades ago, but little has appeared in the literature. Most work has focused on isolated word recognition and various classifiers. Recognition frameworks such as Dynamic Time Warping, Neural Networks, Hidden Markov Models (HMMs) and Fuzzy-Neural Networks have been studied [1][2]. Several front-ends have been tried ([3][4][5]), but there is no consensus on which feature extraction method is best suited to Thai speech recognition. Furthermore, a front-end that works well for isolated word recognition may not work as well for continuous speech.

Isolated word recognition has gained popularity in the Thai research community due to its implementation simplicity, and a lack of resources for developing a continuous speech recognition framework. The availability of the HTK Toolkit [6] has enabled Thai researchers to explore aspects of speech recognition and speech understanding without worrying about tool development. However, building a continuous Thai speech recognition system is not as difficult as it was a few years ago, as long as the required speech database for training and evaluation is available.

In this work, we investigate the performance of some well-know spectral features – Linear Predictive Cepstral Coefficient (LPCC), Mel-Frequency Cepstral Coefficient (MFCC) and Discrete Cosine Transform Coefficient (DCTC), and their derivatives on Thai speech recognition. The results of this work can serve as a baseline for further study in this area.

2. FRONT-END ANALYSIS

2.1. LPCC and MFCC Features

LPCC and MFCC are well-known, widely used speech features. Both front-end analyses have been successfully used in many speech recognition systems. The HTK Toolkit provides tools to extract these kinds of features from speech data, so the HTK implementations of these front-ends were used in our work [7]. The LPCC and MFCC parameters were computed using mostly HTK’s “default” configurations. For each front-end, 12 cepstral feature were extracted. These are denoted as LPCC12 and MFCC12 in Section 4.

2.2. DCTC Features

The DCTC front-end encodes or captures the global spectrum shape using a modified Discrete Cosine Transform (DCT). The DCT basis vectors were modified with a bilinear frequency warping function so that the spectrum at low frequencies was emphasized over that at high frequencies. The warping was applied to simulate human hearing. Figure 1 shows the modified DCT basis vectors, and how the amplitude of the basis vector is higher at low frequencies. A detailed explanation of the DCTC calculation can be found in [8] and [9].

Figure 1: The first three DCTC basis vectors, with a warping factor of 0.45

Twelve DCT coefficients were used to encode the spectrum of each speech frame. This feature set is denoted as DCTC12 in Section 4.
2.3. Dynamic Features

The most commonly used dynamic features are delta and acceleration terms derived from static features. For the HTK Toolkit, these terms are computed using the following regression formulae:

\[ d_t = \sum_{w} w(c_{t,w} - c_{t-1,w}) \]

\[ \Delta d_t = \sum_{w} w(d_{t,w} - d_{t-1,w}) \]

\[ \Delta^2 d_t = \sum_{w} w(\Delta d_{t,w} - \Delta d_{t-1,w}) \]

\[ (1) \]

where \( d_t \) is the delta term at time \( t \) computed in terms of the corresponding static coefficients \( c_{t,w} \) to \( c_{t-1,w} \). The value of \( W \) was set to 2. The same formula was applied to the delta coefficients to obtain acceleration coefficients.

Delta and acceleration terms were added to the corresponding static features to obtain a 36-element feature vector. This technique was applied to all the front-ends. The resulting feature sets are denoted as LPCC36, MFCC36 and DCTC36 in Section 4.

In [10], a Discrete Cosine Series (DCS) was successfully used to compute the dynamic features from DCT. In our work, each of the DCTC12 features was expanded over time by three DCS, resulting in 36 Discrete Cosine Series Coefficients (DCSCs) terms. Each DCSC was computed over 15 consecutive frames and a Kaiser window of beta=5 was multiplied to the DCT basis vector to emphasize the signal at the center of the block. This feature set is denoted as DCSC6 in Section 4. A detailed description of how to compute DCSC features can be found in [10].

3. TASK AND DATABASE

3.1. Task

Our task was to build a recognizer able to recognize continuous Thai digits in various pronunciation styles. A digit string can be pronounced in Thai in several ways. For example, the digit string “1989” can either be pronounced as “one nine eight nine” or “one thousand nine hundred eighty nine.” Note, however, that it is unlikely that Thais pronounce such a string as “nineteen eighty nine.”

We refer to the former pronunciation as PS1 and the latter as PS2. Digit strings with a decimal point must be pronounced only in the PS1 style. The digit strings before the decimal point can be pronounced using the PS1 style. The digit strings after the decimal point must be pronounced only in the PS2 style. All the digit strings were randomly generated, and all recorded utterances were manually time labeled.

Speech recording was conducted in an office environment. Speech was sampled at 16 KHz with 16 bit resolution through a medium quality close-talk microphone connected to a PC soundcard. The training data was taken from utterances spoken by 30 males and 30 females. The remaining utterances from 20 speakers were used as test data.

4. EXPERIMENTS, RESULTS, AND DISCUSSION

The HTK Toolkit version 3.0 [7] was used to build our word-based continuous digit recognizers. Each word was modeled with one continuous density HMM with 5 states, 3 Gaussian mixtures (diagonal covariance). Initially, HMM models were initialized and trained in isolated-word mode, using time labeling provided with the database. The models were then trained in embedded mode using word level transcription without the use of time labels. While the silence model was trained in the training phase and recognized in the recognition step, it was not used in the performance evaluation. In other words, all occurrences of silence in the reference transcriptions and recognized transcriptions were removed before the accuracy calculations were made. This resulted in a fairer performance evaluation, since silence does not convey any important information for our task, but its inclusion could significantly affect the performance figures.

4.1. Word Networks

A crucial step in building a continuous speech recognizer is to specify possible recognizable word sequences. For HTK, this can be achieved through the use of a word network. The word network forces the recognition tool to give only specified answers.

Several word networks were used to cover the desired pronunciation styles:

- Word network 1 (WN1) recognizes digits pronounced using the PS1 style. The digit strings could be of any length.

- Word Network 2 (WN2) is similar to WN1 except that the strings must be 9-digits long. The recognizer will always give 9-digit answers.

Footnote: For clarity, our examples use English, but pronunciation is in Thai in the actual task.
• Word Network 3 (WN3) recognizes digits pronounced using the PS21 style. The digit strings could be of any length.
• Word Network 4 (WN4) is similar to WN3 except that the digit strings must be in the NNNNNNN.MM format, i.e., 7 digits before the decimal point and 2 digits after.
• Word Network 5 (WN5) recognizes digit strings uttered using the PS1 and PS21 styles. Basically, this is equivalent to WN1 + WN3. The digit strings could be of any length.

4.2. Experiment I
We compared the performance of the LPCC, MFCC and DCTC parameters in their static form (denoted as LPCC12, MFCC12 and DCTC12, respectively). WN1-WN5 were used for each front-end. The experimental results are shown in Figures 2 and 3.

Figure 2: Sentence accuracy with static features using different word networks.

Figure 3: Word accuracy with static features using different word networks.

DCTC outperforms the other types of features in all cases. As can be expected, word networks that accept fixed-length digit strings yield higher accuracy than those that accept arbitrary-length strings: WN2 gives higher accuracy than WN1, and WN4 gives higher accuracy than WN3. Accuracy is much lower with WN5. Furthermore, word accuracy for each case is significantly higher than sentence accuracy. The highest word accuracies for WN1-WN5 are 92.4%, 94.6%, 85.0%, 94.8%, and 80.7%, respectively.

4.3. Experiment II
The front-end analyses with dynamic features were tested using various word networks. The results are shown in Figures 4 and 5.

Figure 4: Sentence accuracy with static features augmented with dynamic features.

Figure 5: Word accuracy with static features augmented with dynamic features.
Figures 4 and 5 display similar patterns to Figures 2 and 3. However, the performance figures are significantly higher due to the inclusion of the dynamic features. With the word networks WN2 and WN4, the MFCC front-end outperforms the DCTC front-end. The highest word accuracies for WN1-WN5 are 97.8%, 99.6%, 94.4%, 98.3%, and 93.3%, respectively.

4.4. Experiment III

We tested the DSC parameters described in Section 2.3 using the DSC36 feature set. In Figure 6 the results are compared with those obtained previously with the MFCC36 features.

Figure 6: Word accuracy with MFCC36 and DCS36 features.

Figure 6 indicates that the dynamic features based on DCS expansion can improve the performance of the DCTC front-end. In most cases, DCS36 parameters outperform MFCC36 parameters. With WN2 and WN4, the performances of the two front-ends are very close.

5. CONCLUSIONS

MFCC and DCTC are suitable front-ends for Thai speech recognition. DCTC outperforms MFCC in most cases, while LPCC performs less well. The inclusion of dynamic features greatly enhances recognition performance in all cases. Inclusion of temporal information using DCS expansion plays an important role in improving the DCTC front-end, and the same method could be applied to the MFCC and LPCC front-ends. Other types of features, e.g., PLP, need to be investigated in the future.

6. ACKNOWLEDGEMENTS

The author would like to thank Dr. Andrew Davison for his review of this manuscript, and Ms. Korada Khemapunmanus and Ms. Rungsiri Thiravithayanon for their hardwork collecting speech data.

7. REFERENCES


