Development of HMM-based Snoring Recognition System for Web Services

Youjung Ko\textsuperscript{1}, Insuk Hong\textsuperscript{1}, Hyunsoon Shin\textsuperscript{2,*}, Yoonjoong Kim\textsuperscript{1,*}

\textsuperscript{1}Dept. of Computer Engineering Hanbat National University, South Korea
\textsuperscript{2}Emotion Recognition IoT Research Section, ETRI, South Korea
\textsuperscript{1}{youlony, ishong, yjkim}@hanbat.ac.kr, \textsuperscript{2}hsshin@etri.re.kr
*Co-Corresponding Authors

Abstract—This study presents a Hidden Markov Models (HMM)-based snoring recognition system over the web services environment that consists of the snoring model generation, the recognition system, and the remote device. In design phase, a set of HMM model (snoring and non-snoring) is created from the MFCC feature vectors extracted from the sound corpus consisting of snoring sounds and non-snoring sounds. The recognition system is organized to provide the web services that can be called by the remote device in different platforms with any language. In the test, this system shows that snoring and non-snoring sounds were recognized as 93% and 95.2% for speaker-independent case and 98.3% and 99% for the speaker-dependent case, respectively.

Keywords—Snoring recognition, Snoring recognition web services, HMM

I. INTRODUCTION

Snoring is a breathing noise that is generated by the vibration of the uvula and surrounding structures by means of air suction passing through an airway that has become narrow due to many causes, including excessive muscle relaxation in the uvula and pharynx during sleeping [1]. Snoring, which is one of the typical symptoms of insomnia, can lead to obstructive sleep apnea caused by upper airway obstructions [2]. Persons with severe snoring can experience a headache on waking in the morning or sleepiness during the daytime as well as feel fatigued easily and disturb the sleep of others [3]. Thus, a number of studies have been conducted on how to verify snoring and prevent it continuously.

In a study on the Snoring detection using HMM and piezo snoring sensor, short-time Fourier transform and short-time energy were computed so they could be applied to HMMs. The data were classified as snoring, noise and silence according to their HMMs [4]. In another study of the snoring detection method suitable for a mobile environment to extract the frequency band of 300Hz or less, and the snoring is detected through comparison RMSE (Root Mean Square Error) [5].

In previous studies, equipment had to be attached or by using only a mobile environment in order to recognize the snore, it is difficult to apply the system to recognize a variety of platforms.

Thus, the present study aims to overcome the above problems by using micro-recorded voice file without a separate equipment recognizes snoring, and by providing a snoring recognition system with a web services, thereby being called from various devices to use the snoring recognition service.

The overall configuration of the present study consists of a design phase of snore/non-snore model generation, snoring recognition system with web services, and remote device as shown in Figure 1. In the design phase, the sound detection module accepts speech signals from the snoring sound corpus and detects speech region that is converted to MFCC feature vectors in the feature extraction module. The model estimation module creates snore/non-snore HMM (Hidden Markov Model) models using the extracted feature vectors. In the snoring recognition system, MFCC feature vectors are extracted from the signals that are transferred from the remote device. The model matching module matches the extracted feature vector to the snore/non-snore models based on the Viterbi algorithm. In the remote device, it records and detects sound region, call the web services of the snoring recognition offered by the recognition system in the web services environment, and accepts the results that is recognized by the model matching module. The web services is XML-based so that the snoring recognition system can be used regardless of platforms or implemented languages.

Figure 1. Configuration diagram of snoring recognition web services system
This paper is organized as follows. In Section 2, the creation of the snoring model is described, and in Section 3, the snoring recognition system and web services are presented. In Section 4, experimental results are presented to analyze the performance of the system, and in Section 5, conclusions are presented.

II. SNORING MODEL GENERATION

The voice model of snoring in the present study is developed by using a voice corpus consisting of snoring and non-snoring sounds, thereby extracting the snoring episode and MFCC-type voice features. Using these features, a snoring model, non-snoring model, and silence model are developed.

In the detection process of snoring episode, the energy is calculated via a moving average filter, and if the energy exceeds a limit value, it is considered the start of a sound, and if it becomes smaller than a limit value, it is considered the end of the sound for detecting a snoring episode. For the energy calculation, the length of the filter is set to the number of samples that are ranged from 100–800ms and a limit value is obtained through experiments.

In the snoring feature extraction process, a frame of 25ms is moved. The Mel Frequency Cepstral Coefficient (MFCC) feature vectors are calculated through the following processes: pre-emphasis to attenuate high-frequency distortion with regard to sample data s(n) that belongs to a single frame, a hamming window process to reduce noise occurring at the cut surface of both ends of the frame, and DFT, MEL scaling, the cepstral coefficient, and log processing for frequency characteristics considering perception characteristics. The snoring sound model is a process for training extracted sound feature vectors into a model. Forty snoring voice signals and 10 non-snoring voice signals are trained as follows, thereby generating snoring model T, non-snoring model F, and silence model sil. Table 1 shows a script for training the 40 snoring voice signals and 10 non-snoring voice signals.

<table>
<thead>
<tr>
<th>Sentence No.</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>sil T sil T sil T sil T sil T sil T sil T sil</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>T139</td>
<td>sil T sil T sil T sil T sil T sil T sil T sil</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>F100</td>
<td>sil F sil</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>F109</td>
<td>sil F sil</td>
</tr>
</tbody>
</table>

That is, voice signal file T100 consists of eight snoring sounds (T) and nine silent sound (sil) signals, and non-snoring voice file F100 consists of one non-snoring voice (F) and two silence signals (sil). Based on this script, a vocabulary list, master label, script for training, and vocabulary dictionary, which are required for recognition, are created. The MFCC file per voice file is created. The feature vectors used in the present study are 30-order vectors, including nine-order MFCC vectors, energy one-order, delta, and acceleration. To create three models (T, F, sil), the number of states per model is 15 and one Gaussian Mixture Model (GMM) is used per state. The initial HMM file, including three models, is created. Each model is trained iteratively using 50 data sets for training. Recognition experiments are conducted with regard to the improved HMM in every step. If a recognition rate is not improved, the iteration is terminated. The HTK tool is used for learning and recognition for feature extraction and model creation [6].

III. SNORING RECOGNITION SYSTEM

A. Snoring recognition system using HMM

An HMM model made by learning is used in the snoring recognition system. Voice features are obtained after detecting snoring episode using the method described in Section 2 with regard to inputted voice signals. Snoring recognition is conducted via the Viterbi search algorithm using extracted voice features and the given HMM model. A possible combination of models is defined as grammar, and a network is structured to find the optimum model combination among the possible models with regard to input voice feature vector O. Grammar and network are used to reduce the computation amount and limit the computation to only possible model combinations.

In the study, a grammar file is composed to recognize sound where silent sound comes first, followed by a repeated snoring silent sound or non-snoring silent sound. This grammar is converted into the network, and a model combination is searched by referring to the network.

B. Snoring recognition web services

The recognition system is configured as a web service to be employed by remote devices over the network on the internet [7]. The service is provided at “http://services.wins.or.kr/Snore/ WebService.asmx,” which consists of SnoreRecognizeAny audio and SnoreRecognize.

To use the remote application programming interface (API), a voice is recorded in the wave format and converted into a byte array. Once a parameter of SnoreRecognize is delivered, a recognition result is returned. The result will be T or F. If it is a snoring sound, it returns T otherwise, it returns F for sounds other than snoring. The second function, SnoreRecognizeAnyAudio, is a function for recognizing sound recorded via other formats than wave files.

IV. EXPERIMENT

To analyze the performance of the proposed system, we conducted three experiments such as test for training data, test for non-training data, and the remote device test. First, we suggest a way to recognize as follows. The recognition system as shown in the figure 1 accepts the sound, extracts MFCC feature vectors from the sound and matches it to the hmm models such as snoring(T) and non-snoring(F) and silence(sil), and produces an intermediate result that is a form of sequence composed of three models such as T, F, and sil. It counts three model numbers in the model sequence of the intermediate result and decides the majority model number as a final recognition result.

For the training data, recognition test was conducted as follows. The snoring sound source used in the data for training was collected with a 16kHz sampling frequency, mono, and wave format files. Table 2 shows the recognition results of 50 data sets for training. For 10 non-snoring sound sentences (F100-F109), each sentence was recognized as one non-snoring
model(F) and two silent sound models(sil). For 39 snoring sound sentences(T100-T126,T128-T139), each sentence was recognized as several snoring models(T) and several silent sound models(sil). One sentence(T127) was recognized as three snoring models, one non-snoring model, and five silent sound models, but the counts of snoring models was found to be more than that of non-snoring models so that this sentence is recognized as snoring model. Consequently this recognition system shows a 100% recognition rate in this experiment.

Table 2. Analysis of snoring and non-snoring models

<table>
<thead>
<tr>
<th>Sentence No.</th>
<th>Count of T models</th>
<th>Count of F models</th>
<th>Count of sil models</th>
</tr>
</thead>
<tbody>
<tr>
<td>F100–F109</td>
<td>0</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>T100-T126</td>
<td>228</td>
<td>0</td>
<td>267</td>
</tr>
<tr>
<td>T128-T139</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T127</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

For the non-training data, the second experiment was speaker-independent, in which 10 speakers who did not participate in training conducted 15 recognition tests for each model. The second experiment was speaker-dependent, in which eight speakers who participated in training conducted 10 recognition tests for each model. As shown in Table 3, snoring and non-snoring sounds were recognized as 93% and 95.2% for speaker-independent tests and 98.3% and 99% for the speaker-dependent case, respectively.

Table 3. Recognition results of snoring

<table>
<thead>
<tr>
<th></th>
<th>Speaker-independent</th>
<th>Speaker-dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snoring</td>
<td>93%</td>
<td>98.3%</td>
</tr>
<tr>
<td>Non-snoring</td>
<td>95.2%</td>
<td>99%</td>
</tr>
</tbody>
</table>

Finally, we conducted a recognition test of the remote device to analyze if the web services system works well. As shown in figure 1, the application SnoreRecSystem in the remote device records sound and calls a web services of the recognition system. As shown in figure 2, this application has a user interface that consists of 5 buttons and one image icon. The Record button is used to start recording sound and the Call button invokes a web services of the recognition system and the result is returned and displayed as one of icons based on the value of the result. Calling the web services can be invoked by the speech detection function and also by the manual click on the Call button. In the figure 2, (a) shows the recognition result of snoring sound and (b) shows the non-snoring sound result. It is confirmed that this system works correctly as the experiment above showed the application records and calls the web services correctly.

V. CONCLUSION

This study presents a HMM-based snoring recognition system over the web services environment that consists of the snoring model generation, the recognition system, and the remote device. In design phase, a set of HMM model (snoring and non-snoring) is created from the MFCC feature vectors extracted from the sound corpus consisting of snoring sounds and non-snoring sounds. The recognition system is organized to provide the web services that can be called by the remote device in different platforms with any language. In the test, this system shows that snoring and non-snoring sounds were recognized as 93% and 95.2% for speaker-independent case and 98.3% and 99% for the speaker-dependent case, respectively.

For future research, users will check the false recognition results by using a false recognition report function and sound sources, and results that are falsely recognized will be re-transmitted and stored in the server. An optimum snoring model will be created by constructing an environment where the server performs a verification process followed by training a model automatically to be used in improving the system.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of MSIP. (Development of Military Life Management System based on Emotion Recognition.)

REFERENCES