A Template Based Voice Trigger System Using Bhattacharyya Edit Distance

Evelyn Kurniawati, Samsudin Ng, Karthik Muralidhar, Sapna George

STMicroelectronics Asia Pacific, Pte. Ltd. Singapore
evelyn.kurniawati@st.com, samsudin.ng@st.com, mr.karthik@st.com, sapna.george@st.com

Abstract

Dynamic Time Warping (DTW) is frequently used in isolated word recognition system due to their simplicity and robustness to noise. However, the computational effort required by DTW based solution is proportional to the number of words registered in the system. Vector Quantization (VQ) is employed to alleviate this by converting the spoken input to a sequence of discrete symbols to be matched with the stored word template. In this paper, we propose the use of Bhattacharyya distance as the cost function for this pattern matching problem. The template used is a string of discrete symbols, each modeled by Gaussian Mixture Model (GMM) representing context dependent sub-word unit. The system is tested on 100 template matching task from two registrations of 50 cable TV channel names to simulate voice-triggered remote control. An average of 92% accuracy is obtained. A scheme is also proposed to enable guest user without registration data to use the system efficiently.

Index Terms: dynamic time warping, template matching, edit distance, isolated word recognition.

1. Introduction

The two main approaches used in Isolated Word Recognition task are Hidden Markov Model (HMM) and Dynamic Time Warping. Although HMM in general has better performance due to its flexibility in modeling temporal structure and variability of speech, it requires significant computational resource incompatible with low power devices. Furthermore, the training process of HMM based speech recognizer needs a substantial amount of transcribed speech database to create a reliable acoustic model.

DTW on the other hand, stores multiple sequences of features corresponding to known words. During recognition process, the distance between the input to each of those templates are computed and the word with the least distance are chosen as output. Simplicity in implementation was one of the main reasons DTW based solution was chosen for our voice triggered remote control project. With the user having control of when to activate the microphone, the DTW no longer has the inherent problem of determining the start and end point of the input speech to be recognized [1]. The computational complexity however, increases in proportion to the number of template words used in the system. This is the main problem being addressed in this paper.

Our work builds on the solution presented in [2]. Instead of using maximum likelihood of the input resulting from each template as the distance measure, we propose to quantize the input signal such that the comparison between the input and the template is reduced to a simple string matching problem. A string edit distance [3] is then calculated for each template and we proposed the use of Bhattacharyya distance measure to quantify the mismatch between the input and the template.

This paper is organized as follows. Section 2 presents the description of the full system used in this experiment. The details of the proposed string edit distance with Bhattacharyya distance measure are explained in section 3. Section 4 discusses the results of the experiment, and section 5 presents the conclusions.

2. System Descriptions

This section elaborates on the full system description from the front end feature extraction, training process which consist of sub-word unit and word template generation, and the final recognition process. A section on how guest user could use the system is also presented.

2.1. Front End Feature Extraction

The feature vector used for this recognition task is 24 MFCC. The window size per frame is 20ms with 10ms overlap. All the speech data used in this experiment are 16 bit sampled at 16 kHz.

2.2. Sub-word Unit Generation

This is the first part of the training process where the user is required to record approximately two minutes of their speech. It is recommended to read phonetically rich sentences in order to obtain a more comprehensive sub-word unit. In this experiment, the user is asked to read a series of Harvard sentences [4].

After going through front end feature extraction described in section 2.1, the resulting MFCC are clustered (using k-means algorithm) into 64 distinct units, roughly corresponding to a collection of sub-word. Each of these clusters is then modeled using Gaussian Mixture Model of 4 mixtures. This is a similar modeling strategy as used in [5], but in this experiment, no re-clustering was done during word template generation.

For the proposed edit distance, further calculation is performed to generate the 64 by 64 Bhattacharyya distance matrix. Section 3 will present this scheme in more details. Figure 1 illustrates the sub-word unit generation process.

Figure 1: Sub-word Unit generation

2.3. Word Template Generation

In this step, the words that the user wants the system to recognize are registered. The user is asked to pronounce the words, and the template generation will convert those words into a sequence of sub-word unit index (obtained from the previous step) based on its maximum likelihood. To avoid over segmentation of the word, transitional heuristic [2] is employed by allowing the change of sub-word index only when there is a significant margin of likelihood difference.
with the neighboring state. The word template generation is illustrated in figure 2. This process has to be repeated for each words that the user wants to introduce to the system.

\[ m^* = \arg \max_m p_m(X_{input}) \] (1)

Note that the template can be viewed as a sequence of GMM, and this makes the \( p_m(X_{input}) \) calculation increasingly expensive with an increasing number of word template used. Figure 3 illustrates this recognition process.

2.5. Guest User scenario

The system descriptions above require a user to register both his voice and the words to be recognized. Each user has his or her own sub-word unit profile and word templates. Apart from that, the system also builds one guest profile and templates to be used by non-registered user. For example, in a voice trigger TV remote control use case, each member of a household can register their voice profile. However, if a guest arrived and wanted to use the system without registering their voice, this guest profile is the one that will be evoked.

Guest sub-word unit profile is built by combining all registration data from the entire user. The main issue now is how to build the word templates. Using the word template generation data from each user is not optimal for two reasons. Firstly, it is not generic enough as the template will be too closely tied with the way that particular user pronounce the word. Secondly, the size of the word templates will be huge because it needs to cover each word repeated by all registered user. This is where the solution from [6][7][8] to jointly decode multiple pattern can be adopted, but instead of multiple inputs, the scheme is used to aggregate the multiple pattern of each word template. The centroid of which is then used as the guest word template.

The speaker recognition task to select which user profile to use is addressed in separate publication.

3. Proposed Edit Distance

Instead of treating the input feature vector as possible observation generated by the template as described in section 2.4, we propose to convert the input feature to a sub-word unit index sequence, similar to the word registration process in section 2.3. Now what is needed is to calculate the distance between this string and those that are stored in the template. The chosen word is the one which has minimum distance. The process is illustrated in figure 4.

\[ Bhatt(p_1 \parallel p_2) = -\ln \left( \sqrt{\int p_1(x) \sqrt{\int p_2(x) dx}} \right) \] (2)

Each sub-word unit in our system is modeled using 4 mixtures GMM, so the distance between them is given by:

\[ Bhatt(p_1 \parallel p_2) = -\ln \left( \sqrt{\sum_{a=1}^{4} \int p_{1,a}(x) \sqrt{\int p_{2,a}(x) dx}} \right) \] (3)

This distance needs to be calculated for all 64 sub-word unit. However, this process can be done offline before the actual recognition process. The fact that this distance matrix is
pre-calculated also contributes to the speed-up factor obtained compared to the original matching process.

The edit distance is calculated following the Levenshtein distance pseudo-code below:

```
function edit_distance(inp[1..m],template[1..n])
for i = 1 to m
    distance[i,0] = del_penalty;
for j = 0 to n
    distance[0,j] = ins_penalty;
for j = 1 to n do
    for i=1 to m do
        distance[i,j] = minimum(
                        distance[i-1,j-1]+bhatt(inp[i-1],template[j-1],
                        distance[i-1,j]+del_penalty,
                        distance[i,j-1]+ins_penalty,
                        distance[i-1,j-1]+bhatt(inp[i-1],template[j-1] )
                    )
    return distance[m,n]
```

del_penalty, ins_penalty, and bhatt(x,y) correspond to deletion penalty, insertion penalty, and Bhattacharyya distance between sub-word x and y respectively. The above distance needs to be calculated for each word template in the system, and the minimum of which will be the chosen word output.

4. Experiment Results and Discussion

The proposed algorithm is tested on 50 vocabulary voice command system meant for changing channel in the remote control. Two registrations per word is used to increase the accuracy, totaling in 100 template. The lists of words used are listed in table 1.

<table>
<thead>
<tr>
<th>Table 1. Vocabulary used in the experiment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel5</td>
</tr>
<tr>
<td>Channel8</td>
</tr>
<tr>
<td>Suria</td>
</tr>
<tr>
<td>Vasantha</td>
</tr>
<tr>
<td>Channel9</td>
</tr>
<tr>
<td>Okto</td>
</tr>
<tr>
<td>StarPlus</td>
</tr>
<tr>
<td>StarGold</td>
</tr>
<tr>
<td>Asianet</td>
</tr>
<tr>
<td>Arirang</td>
</tr>
<tr>
<td>NHK</td>
</tr>
<tr>
<td>Eurosport</td>
</tr>
<tr>
<td>Racquet</td>
</tr>
</tbody>
</table>

The database was recorded using H2 Zoom microphone shown in figure 5 in a noiseless environment. 7 people were involved in this exercise (3 female 4 male), each recorded 4 set of the 50 words listed in table 1 (total 200 words per person). Two sets were used to create the word template and the other 2 sets were used to test the recognition process.

There was no significant difference in recognition rate between the original matching process and the use of edit distance. An average of 91.71% accuracy was obtained from the former, and 92% from the later. The main impact is in the running time as the edit distance was able to execute 3 times faster than the original method on this 100 template matching task.

Figure 6 shows the average run time of the recognition task, comparing the original pattern matching method with the proposed edit distance. The x axis shows the number of template being compared at each run. It is evident from the graph that the original pattern matching algorithm increases proportionally with the number of templates being compared. Its complexity however, is lesser than the proposed edit distance method when the number of template is relatively small (lower than 30). Above this number, the proposed edit distance measure has a speed advantage.

It was also observed that the running time of the proposed method is stable regardless of the number of template being matched. This can be explained by looking at the complexity breakdown of this method. Without taking into account the front end feature extraction which is a common process for all speech recognition, the majority (close to 98% in average) of the computational load in the new search method is actually spent in generating the sub-word unit index sequence of the input signal. This is the process described in section 2.3, involving multiple computation of probability of GMM distribution. For the same reason, the original pattern matching algorithm has a distinct increase in complexity as the number of templates increase, because what it does is also computing posterior probability given the sequence of GMM distribution template. The proposed edit distance manages to avoid this complex task by converting them into distinct symbol and pre-calculating the Bhattacharyya distance between each symbol so the only time posterior probability has to be computed from the GMM distribution is during the generation of the sub-word index sequence.

Table 2 summarizes the system accuracy for registered user test as well as the guest user test using the original and the proposed edit distance measure.
Table 2. Systems accuracy comparison.

<table>
<thead>
<tr>
<th>User sub-word</th>
<th>Original</th>
<th>Edit distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registered user</td>
<td>91.71%</td>
<td>92%</td>
</tr>
<tr>
<td>Guest sub-word</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guest user</td>
<td>76%</td>
<td>78%</td>
</tr>
<tr>
<td>Guest user with word registration</td>
<td>83%</td>
<td>86%</td>
</tr>
</tbody>
</table>

For the guest user scenario, only 78% accuracy is achieved with the edit distance method, and 76% on the original method. Note that this was tested on a user which was not involved in the sub-word unit generation process. Their registration speech was removed from the system, and their word registration was discarded in place of the centroid word template discussed earlier.

In the second part of the test, the guest word registration was introduced. However, the guest sub-word unit was not changed. It turns out significant improvement was achieved with 86% accuracy using the edit distance measure. This proves that the guest sub-word unit is representative enough to characterize the guest user pronunciation. It is possible that the low performance of the centroid word template for guest user was due to the high pronunciation variety of the registration word among the remaining users, given that all subjects involved in this exercise are not native English speakers.

Another interesting observation is the margin between the use of user’s own sub-word (92%) and guest sub-word (86%). In both cases, word registration is performed and the only factor causing this difference is the sub-word unit itself. These sub-word units can be seen as a collection of GMM that represents the speaker profile. It is possible to introduce an adaptation strategy to enhance the guest sub-word to better match his personal profile. Similar process exists in the domain of speaker recognition where MAP or ML method is employed to adapt from universal background model to a specific speaker profile. The only difference is that in our case, the adaptation data will be very limited, because it could only come from the test word when the guest user is using the system. This adaptation strategy and an improvement to the centroid word template to better accommodate variation in pronunciations will be part of our future work.

The system is also tested real time, using a front end built with portaudio library[15] and a simple energy based voice activity detector. A satisfactory performance similar to offline testing is achieved.

5. Conclusions

Although the use of dynamic time warping is fairly common for isolated word recognition, its complexity increases with the number of vocabulary introduced to the system, making it incompatible for low power devices. This paper proposed a new pattern matching algorithm using edit distance measure to find the most probable word template. Bhattacharyya distance matrix is computed prior to the recognition task to indicate the similarity between each sub-word which is modeled using Gaussian Mixture Model. The scheme was tested with 100 template matching task (2 registration of 50 words) to simulate the use of voice triggered remote control. Similar accuracy to the original pattern matching algorithm is obtained with substantial reduction in complexity in all the tests performed. On average, 92% accuracy is achieved when a registered user utilize the system. A scheme is developed for a guest user by building a guest sub-word unit from the available speech registered. The guest word template is built by combining all available word registration and finding the centroid. Only 78% accuracy is achieved for guest use case, but 86% can be obtained when guest word registration is employed. It is speculated that the poor performance of the guest word template is due to the variety of pronunciation in the available word registration. The development of a method to better accommodate this pronunciation variety will be part of our future work. The adaptation from guest sub-word unit to specific user sub-word unit will also be part of our future investigation.

6. References