A Resource-Dependent Approach to Word Modeling for Keyword Spotting

I-Fan Chen and Chin-Hui Lee

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
ichen8@gatech.edu, chl@ece.gatech.edu

Abstract

A hierarchical framework is proposed to address the issues of modeling different type of words in keyword spotting (KWS). Keyword models are built at various levels according to the availability of training set resources for each individual word. The proposed approach improves the performance of KWS even when no training speech is available for the keywords. It also suggests an easier way to collect training data for these resource-limited words. Experimental results show that the proposed framework improves performance in KWS in a figure-of-merit (FOM) metric regardless of the number of training instances for each keyword. For words with abundant speech data, the proposed method exploits the training data better than the conventional modeling technique and boosts the system FOM from 9.79% to 42.78%. For words with a small amount of training data, the new method increases the system FOM from 29.05% to 49.06%. Even for keywords without any training examples, the new modeling scheme improves the system FOM from 60.96% to 66.51%.

Index Terms: keyword spotting, hidden Markov model, hierarchical modeling, resource-limited acoustic modeling

1. Introduction

Keyword spotting (KWS) [1][2] is a task of detecting a set of preselected keywords in continuous speech. The technology has been used in various applications, such as spoken term detection [3][4][5], spoken document indexing and retrieval [6], speech surveillance [7], spoken message understanding [8], etc. In general, they can be categorized into two groups, namely: (i) large vocabulary continuous speech recognition (LVCSR) based KWS [3][4][5][9], which uses either vocabulary-dependent word lattices [3][4][5] or vocabulary-independent phone lattices [5][9] for keyword search, and (ii) classical keyword-filler based KWS [1][2] for keyword detection.

Although the LVCSR-based KWS systems often deliver a good performance, they require a large amount of training data, usually only available for resource-rich applications [3][4][5]. On the other hand, the keyword-filler based KWS systems are flexible on the selection of the vocabulary words, and can be built with little or no training data. Furthermore it requires only simple or no language models. These features make keyword-filler based KWS an attractive choice for resource-limited languages and applications [10][11] that run faster and with lower computational cost than the LVCSR-based systems for mobile devices. Recently Intelligence Advanced Research Projects Activity (IARPA) in the United States announced their goals to focus on rapid implementation of effective speech search systems for resource-limited languages in the BABEL program [10]. This shows that searching ways to build good KWS systems under a resource-limited circumstance is becoming a research trend as well. It thus motivates us to find ways to improve the classical keyword-filler based KWS techniques. The term KWS in following paragraphs indicates the classical keyword-filler based KWS if no other particular description is made.

In the conventional KWS framework, keyword modeling generally can be categorized into two methods. First we train whole-word hidden Markov models (HMM) directly from scratch using the available tokens for the particular keywords of interest [1][12]. Since the models are trained purely on the speech data of the keywords, the models capture the variability of the words well and often deliver a good system performance. On the other hand, for words that speech data cannot be easily collected the keyword models are usually constructed from phone models for an existing LVCSR system [2]. The models built in this way often provide an adequate accuracy. However, because of parameter sharing in phone models such keyword models are usually not optimized for these particular words, i.e., they do not fully capture the word pronunciation variation as well as the keyword models trained with the whole-word approach.

It has been shown that an LVCSR system can be improved if we modify its phone models by constructing whole-word model for each word using the word's own speech data via model adaptation [13]. This technique can be adopted for keyword modeling as well to improve the detection accuracy. However, keyword adaptation cannot be used for words with no training data. Such a situation is quite common in most KWS tasks. Finding ways to better model unseen keywords is still a research issue for KWS system design. Meanwhile, for words with plenty of training data, exploiting all the data to achieve better KWS performance is also of great interest.

In this paper, a hierarchical framework is proposed to address the issues of modeling different type of words in keyword spotting. Keyword models are built at various levels according to the availability of training set resources for each individual word. The proposed approach improves the KWS performance even when no training speech is available for the keywords. For words with plenty of training data, the new method exploits the data better than the conventional modeling method and achieves a better system performance.

2. Keyword Categorization and Modeling

In this study, the set of keywords are divided into three groups: (i) Group 1: words without any training data, (ii) Group 2: words with some training data (but not enough for training a whole-word model from scratch), and (iii) Group 3: words with plenty of training data. Two modeling methods, one for all words and the other for Group 3 words only, are proposed.

2.1. Hierarchical keyword modeling

In acoustic modeling for LVCSR system design, it is common that different word/sub-word models share parameters with each other, e.g., at the HMM state level. With more available training data, parameter estimation is often more reliable. However, since the KWS systems only care about spotting a few preselected keywords, parameter sharing in the conventional LVCSR systems may often contaminate the
models for the specific keywords with the acoustic distributions from other words. The resulting keyword models are thus unable to precisely capture their own pronunciation detail. Previous research shows that if we do not utilize this parameter sharing structure, at the word level, in the LVCSR phone models, the word error rate of the LVCSR system can be improved about 4~13% absolute when no language models were used [13].

For a keyword $W$, assume that we have some keyword speech samples $x_w$ and prior knowledge about how the keyword model should be. To estimate $\lambda_w$, a vector of the HMM parameters for the keyword, the problem can be solved by maximum a posteriori (MAP) adaptation proposed in [14] in which a conjugate prior density of an HMM is a combination of Dirichlet and normal-Wishart probability density functions (p.d.f). Parameters used in this prior density are often called hyperparameters which can be estimated from training data.

Following the approach in [14], assume $g$ is the prior p.d.f of $\lambda_w$, $f(\lambda_w)$ is the p.d.f of $x_w$, and $g(x_w)$ is the posterior p.d.f of $\lambda_w$, keyword adaptation in [13] can be represented as:

$$
\lambda_w^* = \arg \max_{\lambda_w} g(\lambda_w | x_w, \lambda, \alpha) = \arg \max_{\lambda_w} f(x_w | \lambda_w) g(\lambda_w | \lambda, \alpha),
$$

where $\lambda_w$ and $\alpha$ are the vectors of hyperparameters of the prior density. More specifically, $\lambda_w$ denotes the hyperparameter set which refers to the LVCSR phone models, while $\alpha$ represents the additional hyperparameters needed to be further estimated from the training data. Since $\alpha$ does not play any role in the formula derived in this paper, it will be ignored in later equations for simplicity. Eq. (1) shows that the new keyword model $\lambda_w^*$ is optimized on the keyword training data $x_w$ with the hyperparameters $\lambda_w$ (and $\alpha$) in the prior density. Since the optimization in Eq. (1) is performed over the keyword speech data $x_w$, the model $\lambda_w^*$ is likely to capture the speech characteristics of the keyword $W$ better than $\lambda_w$. Eq. (1) also shows the advantage of using MAP adaptation for keyword modeling. If the keyword $W$ has no available training data, then Eq. (1) can be reduced to

$$
\lambda_w^* = \arg \max_{\lambda_w} g(\lambda_w | \lambda_w). \tag{2}
$$

The solution to Eq. (2) is the mode of the prior density. In other words, $\lambda_w^*$ would be equal to $\lambda_w$, the original model parameter setting in the LVCSR phone model set.

### 2.1.1. Hierarchical model optimization

Eq. (1) optimizes the keyword model at the word level to address the parameter sharing issue in LVCSR. However, we find that parameter sharing occurs not only at the word level but also at the phone level. Phones with different contexts may share the same model distribution in the LVCSR phone model. Therefore, besides word-level, a phone-level optimization of the keywords is proposed in this study.

For a phone $P$, assume that the vector of the phone model parameters provided by the LVCSR phone model is $\Theta_P$, which is shared by the other phones, and the speech data for phone $P$ is $x_p$, phone-level optimization over the speech data is then:

$$
\theta_P^* = \arg \max_{\theta_P} g(\theta_P | x_p, \Theta_P) = \arg \max_{\theta_P} f(x_p | \theta_P) g(\theta_P | \Theta_P). \tag{3}
$$
perception (MLP) [17] with deep structure, is proposed for rescoring the putative hits of the HMM-based keyword detectors created in the previous section.

Usually, the input features for keyword MLP models are confidence score outputs of the HMM-based KWS systems [18] or the acoustic features of the putative segments [19]. Here the input feature vector to the MLP model is the state-alignment-warped MFCC (mel-frequency cepstral coefficient) [20] features of the putative segment. The output of the MLP model is the posterior probability of the given putative hit being a true hit or a false alarm. For each keyword, the MLP model will be trained on the putative hits generated by performing keyword detection on the training data, instead of utilizing the whole training set.

Using the keyword MLP models to rescoring the putative hits of the HMM-based KWS systems is not a new technique [18][19]. However, different from the previous methods, the number of hidden layers of our MLP models is allowed to be more than one in this study and is determined by the empirical performance on the development data. Here, the average depth of the MLP models is 3 hidden layers. Each hidden layer contains 512 hidden nodes. An unsupervised pre-training [16] method is adopted before regular backpropagation training to make sure the MLP models are initialized at a good starting point. Since the MLPs are trained on putative hits instead of the whole speech corpus, the average training time for the proposed keyword MLP models with 3 hidden layers takes less than 6 minutes on a single i5-2400 CPU@3.10GHz personal computer. No GPU acceleration is used.

3. Keyword Spotting Experiments

All the experiments were conducted on the TIMIT corpus [21] divided into three parts: a training set (3296 utterances, 2.79 hours), a development set (400 utterances, 0.34 hours), and a test set (1344 utterances, 1.14 hours). The training and development sets are subsets of the standard TIMIT training set, while the test set is the standard TIMIT test set. Dialect utterances (SA1 and SA2) were not used in the experiment. The acoustic features used in the experiments are 12 MFCCs plus energy and their first and second time derivatives. The CMU/MIT phone set, which contains 39 phonemes, was used. Triphone HMMs were used to construct the initial phone models of the baseline LVCSR system.

3.1. Keyword selection

Thirty words in the TIMIT vocabulary were selected as the keywords in the KWS experiments. They are divided into three groups: (i) words having no speech data in the TIMIT training corpus (14 words), (ii) words having some training tokens in the training set (7 words), and (iii) words having plenty of training samples in the training data (9 words). We used a threshold of 50 samples to determine whether a word belong to Group 2 or 3. The threshold is empirically set. Table 1 lists the chosen keywords in the three groups. For each keyword, a KWS system is built for detection.

3.1.1. Performance measures

A keyword occurrence is considered correctly detected, or hit, if the mid-point of the detected reference falls within the time boundaries of the hypothesis. To evaluate the performance of the overall system, the figure-of-merit (FOM) [22] metric, which is an upper-bound estimate of word spotting accuracy averaged over 1 to 10 false alarms per hour, was used. An average FOM over all keywords was used as the overall performance measure.

Table 1. Keywords used in this study. They are divided into three groups according to the count of their training instances.

| Group 1 | overalls, potatoes, greg, tooth, shore, products, silly, prestige, avoid, popular, pretty, expense |
| Group 2 | before, after, people, these, always, without, money |
| Group 3 | was, with, his, this, from, not, but, every, often |

3.1.2. Baseline systems

Both of the two baseline KWS systems adopted the monoword modeling method proposed in [13] for the keyword HMMs. The difference between these two systems is that the keyword model in the second baseline system is word-level optimized. The filler models used in the two baseline systems had 9 states and 78 Gaussian mixture components per state. The whole TIMIT training set was used for filler model training. The confidence scores for the detected keywords were evaluated as the log likelihood ratio (LLR) [23] between the keyword and the filler models on the detected keyword segments.

3.2. Parameter sharing analysis

To illustrate the situation of parameter sharing in the LVCSR phone models, an analysis was conducted on the TIMIT set. We define a parameter sharing factor (PSF) at the word and phone levels as follows: (1) Word-level PSF – the average number of words sharing the same triphone. For example, the triphone /ay-n+d/ is shared by the words "blind" and "finds". This factor can be computed using the TIMIT dictionary; and (2) Phone-level PSF – the average number of triphones sharing the same HMM state. This number can be calculated by examining the baseline LVCSR phone models.

Table 2 shows that, for the 6237 words in the TIMIT vocabulary, a triphone is shared by 5.59 words on the average. And in the LVCSR phone models, which contain 35468 triphones and 397 Gaussian mixture states, each state is expected to be shared by 268 triphones1. Ideally, these 268 triphones model different pronunciations. Yet due to parameter sharing, the model is losing the ability to differentiate these sounds. Even expand the LVCSR model to 3000 states, which is the number used by most of the LVCSR systems, each state is shared by about 35 triphones. This PSF is still considered relatively high.

3.3. Experimental results

Table 3 lists the average FOM performance of the baseline and the proposed KWS systems. The first to third columns show the average system FOM of the keywords in Groups 1, 2, and 3.

1 This number is calculated by 35468/(397/3) since each phone has three states.
Phone-level parameter sharing plays a key role in parameter optimization, the overall FOM can be improved. For the Group 1 words, significant FOM improvement was 45.33% after word-level optimization. If we further built keyword-specific MLP models for the Group 3 keywords, the FOM performance can be improved to 42.78%. Finally, the overall FOM was 55.32% averaged over all 30 keywords. Using phone-level optimization is beneficial to all the KWS systems. For the keywords in Group 2, doing word-level optimization after phone-level optimization is significantly better than doing word-level optimization directly. This shows that phone-level optimization provides a better prior for modeling the Group 2 keywords. On the other hand, for the keywords in Group 3, additional phone-level optimization did not help word-level keyword modeling much. This might be because the Group 3 keywords have so much training data that the model prior had no significant effect on the optimized parameter values. On the contrary, using these data to build discriminative keyword MLP models significantly improved the KWS system performance of the Group 3 keywords.

Table 3. Performance comparison of the baseline and the proposed KWS systems at different optimization levels: numbers in gray are the performance using keyword models at the lower optimization level because the systems are lacking training data and cannot be optimized on the current level.

<table>
<thead>
<tr>
<th>KWS Systems</th>
<th>FOM (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group 1</td>
</tr>
<tr>
<td>Baseline 1</td>
<td>60.96</td>
</tr>
<tr>
<td>Baseline 2 – Word-level optimization</td>
<td>(60.96)</td>
</tr>
<tr>
<td>Phone-level optimization</td>
<td>66.51</td>
</tr>
<tr>
<td>Phone- + Word-level optimization</td>
<td>(66.51)</td>
</tr>
<tr>
<td>MLP rescoring</td>
<td>(66.51)</td>
</tr>
</tbody>
</table>

From the results in the two baseline systems shown in the first and second rows, we can find that word-level modeling is indeed helpful for improving KWS performance. The overall FOM was increased from 38.16% to 45.33%. However, despite the significant FOM improvement for the keywords in Groups 2 and 3, the word-level optimization is not helpful for the keywords in Group 1 due to the lack of training data.

The third to fifth rows show the performance of the KWS systems using the proposed hierarchical modeling method on the different optimization levels. The keyword modeling started at the phone-level optimization (shown in the third row); the word-level optimization was then applied to the phone-level optimized models (the fourth row); if the keywords have sufficient training data, the keyword MLP models were built and used for rescoring (the fifth row). If the keywords did not have enough training data for the optimization on the current level, the models constructed on the previous level were kept; the performance is then shown in gray texts in Table 3. The third row shows that after using the phone-level optimization, the overall FOM can be improved from 38.16% to 46.19%, which is slightly better than the FOM 45.33% of the second baseline system. This is because, in addition to the improvements of the KWS systems of the keywords in Group 2 and 3, the FOM performance of the KWS systems of the Group 1 keywords is also improved by the phone-level optimization.

Notice that, for the keywords in Groups 2 and 3, phone-level optimization provided a similar FOM improvement to the second baseline KWS system, which performed word-level optimization directly. This result agrees with the finding that phone-level parameter sharing plays a key role in parameter sharing in a word model discussed in Section 3.2 because it improves the performance of the KWS systems significantly. This also provides us a direction for collecting training data for keywords whose speech data not easy to get, i.e., we may collect data of other words whose triphones are also used in modeling a particular keyword in addition to using the keyword's own speech data since phone-level optimization exploits the speech data of those words.

When it comes to word-level optimization, KWS for the Group 1 words could not be improved due to the lack of training data, while the KWS performance for the keywords in Groups 2 and 3 can be further improved. The overall FOM was 49.05% after word-level optimization. If we further built keyword-specific MLP models for the Group 3 keywords, their FOM performance can be further improved to 42.78%. Finally the overall FOM was 55.32% averaged over all 30 keywords.

Using phone-level optimization is beneficial to all the KWS systems. For the keywords in Group 2, doing word-level optimization after phone-level optimization is significantly better than doing word-level optimization directly. This shows that phone-level optimization provides a better prior for modeling the Group 2 keywords. On the other hand, for the keywords in Group 3, additional phone-level optimization did not help word-level keyword modeling much. This might be because the Group 3 keywords have so much training data that the model prior had no significant effect on the optimized parameter values. On the contrary, using these data to build discriminative keyword MLP models significantly improved the KWS system performance of the Group 3 keywords.

In this paper, for each KWS system, about 83 seconds were used to process the whole TIMIT test set (1.14 hours). The real time factor, defined as a ratio of the runtime for processing the test data to the length of the test set, of the systems was 0.02. For the Group 3 keywords, since the MLP models only rescore the putative hits, the additional rescoring only took less than 0.01 seconds.

4. Discussion and Future Work

We propose a resource-dependent hierarchical approach to keyword modeling which includes phone- and word-level model optimization and keyword MLP model construction. The proposed framework allows keywords with different amounts of training resources to be optimized at different levels. Experimental results show that the proposed approach significantly improves the performance over conventional KWS systems, even for keywords having no training data.

The experiments in this study are conducted on TIMIT corpus because TIMIT data is clean-read, phone-balanced, which allows us to test keywords with different pronunciations as many as possible, and a controlled test environment. Although the contexts are manually designed, since our KWS needs no language models, the experimental results on the TIMIT data are still convincing. For future work, we plan to test our approach on more natural speech, e.g., the Switchboard data, to study the performance of the approach under circumstances of spontaneous speech and noises. Also, we believe our proposed framework can be further improved by expanding the hierarchy to the speech attribute level. These attributes, such as place and manner of articulation [24], characterize acoustic properties at a lower level than phones. They have been shown to improve LVCSR system performance in lattice rescoring [25], bottom-up speech recognition [26], and multi-lingua speech recognition [27]. For resource-limited keywords we may explore attribute-level optimization, which provides the possibility of keyword modeling using resources from other languages. Furthermore, higher acoustic levels such as phrases might also be used to help resource-abundant keywords.
5. References


