Exploiting Phoneme Similarities in Hybrid HMM-ANN Keyword Spotting

Joel Pinto [12], *Andrew Lovitt* [13], *Hynek Hermansky* [12]

[1] IDIAP Research Institute, Martigny, Switzerland
[2] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[3] University of Illinois at Urbana-Champaign, IL., USA
joel.pinto@idiap.ch, bcseiny@gmail.com, hynek@idiap.ch

Abstract

We propose a technique for generating alternative models for keywords in a hybrid hidden Markov model - artificial neural network (HMM-ANN) keyword spotting paradigm. Given a base pronunciation for a keyword from the lookup dictionary, our algorithm generates a new model for a keyword which takes into account the systematic errors made by the neural network and avoiding those models that can be confused with other words in the language. The new keyword model improves the keyword detection rate while minimally increasing the number of false alarms.

Index Terms: keyword spotting, confusion matrix, HMM-ANN hybrid decoding

1. Introduction

Keyword spotting refers to identifying a word (or phrase) of interest in unconstrained speech recording. Keyword spotting approaches are broadly classified into two categories. One based on large vocabulary continuous speech recognition (LVCSR) and the other based on acoustic match between the keyword (modeled by its phonetic string) and the data. In LVCSR based word spotting, the keywords are spotted from the word lattices generated by the ASR. While this approach is more accurate for words in the ASR dictionary, it is not suitable for out-of-vocabulary (OOV) words and is computationally expensive. On the other hand, acoustic word spotting can be used for any keyword but this approach generates a large number false positives, especially for shorter words as each word is processed independently and language constraints are not exploited.

In this paper, we will discuss the acoustic keyword spotter (described in section 2) based on hybrid hidden Markov model - artificial neural network (HMM-ANN) paradigm. Here, a multi-layered perceptron (MLP) neural network is discriminatively trained to estimate the posterior probability of phonemes with acoustic evidence as its input. The keyword is represented by its phonetic string and each phoneme in the keyword is modeled by an HMM. The phoneme posterior probabilities are used as the emission probabilities for the HMM state and Viterbi algorithm is applied to spot the keyword.

An advantage of this approach is that the MLP is able to estimate the phoneme identity with sufficient accuracy because it is trained using sufficiently long temporal context and has learned to discriminate between phonemes. Moreover, the errors by the MLP are systematic and can be captured in the form of a confusion matrix [3]. However, a disadvantage of the above approach is that if there is a mismatch between the phonetic string of the keyword (obtained from the lookup dictionary) and the phoneme posteriors from the MLP, the keyword detection rate falls. This mismatch is due to the following:

- Speaker Error: Speaker did not pronounce the word according to the dictionary
- Machine Error: MLP classification error due to acoustic confusability

Dictionaries with multiple pronunciations capture the speaker error to some extent but the dictionary entries are based on the expected way to pronounce the word. In this study, we also incorporate the knowledge about the systematic errors made by the MLP. This knowledge is captured in the form of acoustic confusion matrix. Given a base pronunciation for a keyword, the acoustic confusion matrix hypothesizes different alternatives for phoneme in the keyword based on the confusability among phonemes. The language confusion matrix will prune out those candidates that are likely to be confused with other words and hence minimize the increase in false alarms.

Phoneme confusion matrix has been used in phoneme recognition based spoken document retrieval. In [1], phoneme confusion matrix has been used in query expansion. In [2], it has been used for document expansion. This work is similar to the query expansion in the sense that we expand the phonetic string of the keyword to increase the keyword detection. We also use a language confusion matrix to minimize false alarms.

2. Acoustic Keyword Spotting

In acoustic keyword spotting, the keyword is modeled by its phonetic string and all non-keyword speech is modeled by a garbage model connected in parallel to the keyword model as shown in Fig. 1. Additionally, there is a transition from the end of this parallel model to its beginning to enable spotting more than one keyword in the utterance.

A garbage model is a generic model to absorb all speech and satisfying the following inequalities:

$$p(X_{W_i} \mid M_{W_i}) > p(X_{W_i} \mid M_G) \quad (1)$$

$$p(X_G \mid M_G) > p(X_G \mid M_{W_i}) \quad (2)$$

where X_{W_i} and M_{W_i} is the acoustic evidence and the model respectively for the keyword W_i. Similarly, X_G is the speech corresponding to non-keywords and M_G is the garbage model. Eqn (1) controls the keyword detection rate and Eqn (2) controls the number of false alarms.

The keyword model is the concatenation of the hidden Markov models (HMMs) corresponding to the constituent phonemes in the keyword. However, garbage models are obtained in different ways. One way of obtaining smoothed garbage model is to train a GMM or HMM explicitly on non-keyword speech. Multiple garbage models could also be trained

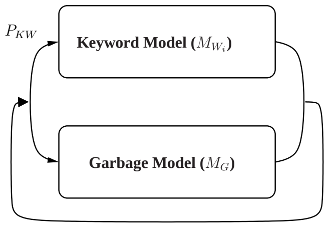

Figure 1: Acoustic keyword spotting architecture. P_{KW} is the keyword entrance probability.

for different classes of sounds (vowels, plosives, nasals etc). A garbage model could also be a generic word model modeled as an ergodic network of context dependent or independent phonemes. Smoothing can also be done at the score level rather than model level as done in the online garbage model [6]. Here, the likelihood of a garbage HMM state is the average of the top likelihoods at that time frame. The observation likelihood for an HMM state could be obtained by a Gaussian mixture model (GMM) or from an MLP. In our experiments, we have used hybrid HMM-ANN decoding with an online garbage model.

3. Baseline System

The baseline system consists of a keyword model connected in parallel with the garbage model as shown in Fig. 1. The HMM for the keyword is the concatenation of the HMMs of the constituent phonemes. The phonetic string for the keyword is obtained from a lookup dictionary with multiple pronunciations. Each phoneme is modeled as an HMM with 3 emitting states (minimum duration 30 ms). The self and next state transition probability is fixed at 0.5 each. The emission probability in each state is obtained from an MLP. The MLP estimates the posterior probability of the 46 output classes with multi RASTA [5] features as input. The garbage model has 5 states and the output likelihood in the garbage state for a given frame is the average of top $N = 3$ scaled likelihoods for that frame. Scaled likelihoods are obtained by normalizing the posterior probability vector from the MLP by the phoneme prior probabilities. The Viterbi algorithm is used to find the best path through the trellis and to spot the keywords.

4. Our Approach

We propose a systematic approach to derive a pronunciation model for the keyword that takes into account the errors by the MLP in phoneme posterior estimation. An LVCSR language model and dictionary are also analyzed to avoid pronunciation models that could be confused with other valid phoneme sequences in the language in order to minimize the increase in false alarms.

The new pronunciation model is obtained by adding acoustically confusing phonemes in parallel to the phonemes in the base pronunciation (see Fig. 2). The new pronunciation model for keyword $M_{W_i}^{'}$ will also include the base pronunciation M_{W_i} and in a Viterbi decoding framework, $p(X_{W_i}|M_{W_i}^{'}) \geq p(X_{W_i}|M_{W_i})$. Hence from Eqn (1), we get:

$$p(X_{W_i} \mid M_{W_i}^{'}) \geq p(X_{W_i} \mid M_{W_i}) > p(X_{W_i} \mid M_G) \qquad (3)$$

It is evident from Eqn (3) that the keyword detection rate for the new model increases or remains the same. However, Eqn (4) must also be satisfied to ensure the minimal or no increase in false alarms

$$p(X_G \mid M_G) > p(X_G \mid M_{W_i}^{'}) \geq p(X_G \mid M_{W_i}) \qquad (4)$$

Depending on the words spoken, there is a risk that the false alarm rates could increase as $p(X_G \mid M_{W_i}^{'})$ approaches $p(X_G \mid M_G)$ in Eqn (4). We try to minimize this risk by avoiding those models that are likely to be confused with other words. This information is obtained by analyzing the similarity of the keyword to other words in the language. The language confusion matrix captures this information and is derived from LVCSR dictionary and the language model. This is explained in section 4.2

4.1. Acoustic Confusion Matrix (ACM)

An acoustic confusion matrix (ACM) captures the systematic errors made by the MLP. Given P phonemes, the ACM is a P by P matrix and each element is the probability $P(p_j|p_i)$ that phoneme p_j was reported by the recognizer when phoneme p_i was said. To obtain the acoustic confusion matrix, phoneme recognition (described in 5.4) is performed on development set and the recognized phoneme sequence is time aligned to the true phoneme sequence. The substitution counts from the alignment are then normalized to obtain the acoustic confusion matrix.

Dynamic time alignment is done in two stages [8]. In the first stage, Levenstein's algorithm with equal substitution costs for all phoneme pairs is used to obtain a preliminary confusion matrix. This matrix will not capture the true confusions because of the equal substitution cost assumption. To improve the alignment, the preliminary confusion matrix is used to update the substitution costs for the Levenstein's algorithm and new confusion matrix is obtained. Table 1 shows the top confusions and the probability for the phonemes /ah/, /ax/, /s/ and /m/.

Table 1: *Table illustrating the top confusions for the phonemes /ah/, /ax/, /m/ and /s/*

Phoneme Said	Phoneme Recognized	Probability
/ah/	/ah/	0.367
	/ax/	0.324
	/ae/	0.036
/ax/	/ax/	0.698
	/ih/	0.093
/m/	/m/	0.766
	/n/	0.053
/s/	/s/	0.878
	/t/	0.017

The acoustic confusion matrix is not a symmetric matrix. For example, $P(/ax/|/ah/) = 0.32$ but $P(/ah/|/ax/) = 0.02$. Also, some phonemes e.g. /ah/ are easily confused $P(/ah/|/ah/) = 0.36$ but some other phonemes e.g /s/ are not easily confused $P(/s/|/s/) = 0.87$.

4.2. Language Confusion Matrix (LCM)

While we have a single acoustic confusion matrix for a task, we derive a different language confusion matrix for each keyword. Suppose that the keyword has the base pronunciation $\vec{p}^0 = p_1^0, ...p_k^0, ...p_K^0$. The language confusion matrix is obtained in the following steps.

[1] Compute the joint probabilities of word triplets in the tri-gram language model. Typically language models give conditional probabilities.

[2] Estimate the joint probability $P(\bar{p})$ of all phoneme K-tuples $\bar{p}$ using the joint probability of the word triplets and the lookup dictionary. Denote this set as P_K. K is the number of phonemes in the keyword.

[3] By dynamic time alignment (dta), compute the edit distance $d(\bar{p}^0, \bar{p})$ and alignment between the base pronunciation $\bar{p}^0$ and every phoneme K-tuple $\bar{p}$ in the set P_K.

[4] Extract a subset $P_K^E \subset P_K$ such that $0 < d(\bar{p}^0, \bar{p}) \leq E$, where E is the threshold on the edit distance. The set P_K^E will contain all valid phoneme sequences with edit distances $1, 2 \ldots, E$ from the keyword phoneme sequence. The threshold E can be set higher for keywords with more number of phonemes.

[5] Align every phoneme sequence $\bar{p}$ in P_K^E to $\bar{p}^0$ and if $dta(p_k^0)$ is the phoneme aligned to p_k^0 in the base pronunciation, update the language confusion matrix as follows:

$$LCM(p_k^0, dta(p_k^0)) = LCM(p_k^0, dta(p_k^0)) + P(\bar{p})$$

The entries in the LCM are row normalized so that they sum upto one. In step-4, alternative pronunciations for the keyword should not be included in set P_K^E. The top language confusions for the word 'something' is listed in the third column in Table 2

4.3. Pronunciation Model

For every phoneme p_k^0 in the keyword with base pronunciation $\bar{p}^0 = p_1^0, ...p_k^0, ...p_K^0$, the acoustic confusion matrix gives a list of phonemes $A(p_k^0)$ that the MLP is likely to have misclassified. Similarly, the language confusion matrix gives a list of phonemes $L(p_k^0)$ that should not be associated with a phoneme in the keyword.

$$A(p_k^0) = \{1 \leq i \leq P, p_i \neq p_k^0 \mid ACM(p_k^0, p_i) > A_{tr}\} \quad (5)$$

$$L(p_k^0) = \{1 \leq i \leq P, p_i \neq p_k^0 \mid LCM(p_k^0, p_i) > L_{tr}\} \quad (6)$$

A_{tr} is the threshold on the acoustic confusion matrix, L_{tr} is the threshold on the language confusion matrix and P is the number of phonemes. A phoneme is added in parallel to p_k^0 only if it is present in the list $A(p_k^0)$ and not present in $L(p_k^0)$. The depth of the pronunciation model is controlled by the parameters A_{tr} and L_{tr}. Table 2 illustrates how pronunciation is obtained for the word 'something' and Fig. 2 shows the final pronunciation model for different values of the thresholds A_{tr} and L_{tr}.

Table 2: *Table illustrating the construction of the pronunciation model for the word 'something'. The phoneme /n/ is not present in the final phonetic string because LCM hypothesizes it as a conflicting phoneme*

keyword 'something'	top acoustic confusions	top language confusions	final pronunciation
/s/	-	/r/	/s/
/ah/	/ax/	/ey/	/ah/,/ax/
/m/	**/n/**	**/n/**,/ch/, /ax/	/m/
/th/	/t/	/b/	/th/, /t/
/ax/	/ih/	/iy/	/ax/,/ih/
/ng/	/n/	/dx/, /t/, /v/	/ng/,/n/

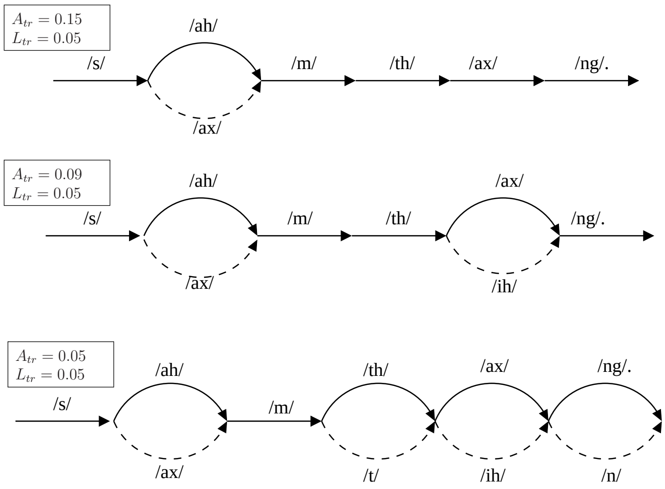

Figure 2: *Pronunciation models for the word 'something' for different values of A_{tr} and L_{tr}. The bold line is the base pronunciation from the dictionary. Dotted lines are the phoneme links added by the algorithm.*

5. Experiments

5.1. Database

Experiments were conducted on the conversational telephone speech (CTS) development data distributed by NIST for the 2006 Spoken term detection task. Of the six hours of two channel speech, 4 hours was used for development and the rest was used for evaluation. A set of 25 single-word keywords were selected from the search list distributed by NIST. The keywords include words like *different, getting, everything, something and affected*. Each of the keyword had at least 4 phonemes in it.

5.2. Acoustic Features

Speech was first segmented to speech/silence classes using a neural network based phoneme recognizer [9] where all the phonemes were linked as speech class. Multi-resolution RASTA [5] features were used to obtain the phoneme posteriors. Critical band spectral analysis (Auditory Spectral Analysis step in the PLP technique) is first performed on the speech signal with a window length of 25 msec and step size of 10 msec. The resulting critical band spectrogram is then filtered using a bank of 2-D filters with varying temporal resolution to obtain a 448 dimensional feature vector every frame.

5.3. Phoneme Posteriors

The phoneme posteriors were estimated every 10 ms by a discriminatively trained MLP trained on 30 hours of Fishers conversational telephone speech (CTS) [12] data using multi-RASTA features of 448 dimension. The 46 output classes included 41 phonemes, a silence class and 4 classes for speech artifacts. The hard target labels for training were obtained by forced alignment. There were 2000 hidden layers and a softmax non-linearity function was used. 10% of the training data was used for cross validation. The MLP training was done using the Quicknet software [11].

5.4. Phoneme Recognition

To estimate the acoustic confusion matrix, phoneme recognition was performed on development set. A hybrid HMM-ANN [4] phoneme recognizer was used where each phoneme was modeled by a standard 3 state left-right HMM. The transition probabilities are fixed a priori. The emission probability for the HMM state was estimated from an MLP trained on MRASTA features. We assume equal prior distribution of the phonemes to obtain the scaled likelihoods. On 41 phoneme set, using a uniform language model, the phoneme error rate (PER) was 47%.

5.5. Dictionary and Language model

To obtain the language confusion matrix, the AMI dictionary with 50000 entries and the AMI language model with approximately 40 million trigram entries were used [7]. From this LM, 1 million trigrams with the highest word triplet joint probability were used for the language confusion analysis.

6. Results

The performance of the keyword spotter is evaluated using the figure-of-merit (FOM) [10] measure which is the average of the keyword detection rates at false alarm rates of $1, 2, ...10$ false alarms per keyword per hour of speech. FOM approximates the keyword detection rate of the system at 5 false alarms per keyword per hour of speech. ROC is computed by varying the word entrance probability (P_{KW} in Fig. 1). The baseline system gives a FOM of 64.30%.

The performance of the proposed method is compared against the baseline system for different values of A_{tr} and L_{tr}. In deriving the language confusion matrix, the threshold $E = 2$ was used for all keywords.

Table 3: *The FOM of the baseline system compared to the proposed method for different values of A_{tr} and L_{tr}*

experiment number	threshold A_{tr}	threshold L_{tr}	FOM
1 (baseline)	1.00	1.00	64.3
2	0.09	1.00	61.9
3	0.15	0.05	65.5
4	0.09	0.05	66.3
5	0.05	0.05	66.5

Experiment 1 is the baseline system as a high threshold of $A_{tr} = 1.0$ does not hypothesize any alternative phonemes and the base pronunciation is used. In experiment 2, only the acoustic matrix is used to generate the new pronunciation as a high L_{tr} will not prune out any pronunciation that could lead to false alarms. The poor FOM compared to the other experiments confirms the importance of the language confusion matrix. Experiment 3, 4 and 5 show that using both acoustic and language confusion matrices for generating the keyword pronunciation gives better performance than the baseline system.

7. Conclusions

In this paper, a systematic approach to building a pronunciation model in a hybrid HMM-ANN keyword spotter is presented. The systematic errors by the neural network in estimating the posterior probability of phonemes is captured in the form of a confusion matrix. This information is used to expand the pronunciation to improve the keyword detection rate. The risk of increase in false alarms is minimized by analyzing the confusion of the keyword to other words in the language captured in the form of language confusion matrix.

8. Acknowledgements

This work is supported in parts by IM-HOST project from Hassler foundation, Switzerland, KEYSPOT project from Deutsche Telekom, Germany and DIRAC integrated project from the European Community. The authors would like to thank Speech@FIT group, Brno for the speech silence segmentation.

9. References

[1] Yuk-Chi, L. et al, “Query Expansion using Phonetic Confusions for Chinese Spoken Document Retrieval”, Proc. of the 5^{th} International Workshop on Information Retrieval with Asian Languages, Hong Kong, 2000, 89–93.

[2] Srinivasan, S. and Petkovic, D., “Phonetic Confusion Matrix based Spoken Document Retrieval”, Proc. of the 23^{rd} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, 2000, 81–87.

[3] Lovitt, A., Pinto, J., and Hermansky, H., “On Confusions in a Phoneme Recognizer”, IDIAP Research Report, IDIAP-RR-07-10, 2007.

[4] Bourlard, H. and Morgan, N., “Connectionist Speech Recognition - A Hybrid Approach”, Kluwer Academic Publishers, Boston, 1994.

[5] Hermansky, H. and Fousek, P., “Multi-Resolution Rasta Filtering for Tandem based ASR”, Proc. of Interspeech, September, 2005, 361–364.

[6] Bourlard, H., D’hoore, B., and Boite, J.-M, “Optimizing Recognition and Rejection Performance in Wordspotting Systems”, Proc. of International Conference on Acoustics and Signal Processing, 1994, vol-1, 373–376.

[7] Hain, T. et al, “The Development of the AMI System for the Transcription of Speech in Meetings”, Proc. of Machine Learning for Multimodal Interaction (MLMI), 2005.

[8] Lovitt, A., “Correcting Confusion Matrices for Phone Recognizers”, IDIAP Communication, No.03, 2007.

[9] Schwarz, P., Matejka, P., Cernocky, J., “Towards Lower Error Rates in Phoneme Recognition”, Proc. of th 7^{th} International Conference on Text, Speech and Dialoque, Brno, 2004.

[10] Rohlicek, J.R., Russell, W., Roukos, S., and Gish, H., ”Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting”, Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol 1 1989.

[11] “The ICSI Quicknet Software Package”, http://www.icsi.berkeley.edu/Speech/qn.html

[12] Zhu, Q., Chen, B., Morgan, N. and Stolcke, A., “On using MLP Features in LVCSR”, Proc. of International Conference on Spoken Language Processing, Korea, October 2004.