VOCABULARY-INDEPENDENT WORD CONFIDENCE MEASURE USING SUBWORD FEATURES

Li Jiang and Xuedong Huang

Microsoft Research
One Microsoft Way
Redmond, Washington 98052, USA

ABSTRACT

This paper discusses how to compute word-level confidence measures based on sub-word features for large-vocabulary speaker-independent speech recognition. The performance of confidence measure using features at word, phone and senone level is experimentally studied. A framework of transformation function based system using sub-word features is proposed for high performance confidence estimation. In this system, discriminative training is used to optimize the parameters of the transformation function. In comparison to the baseline, experiments show that the proposed system reduces the equal error rate by 15%, with up to 40% false acceptance error reduction at various fixed false rejection rate. The combination of multiple features under the proposed framework is also discussed.

1. INTRODUCTION

An accurate confidence measure is critical for practical spoken language conversational systems. With an accurate confidence measure for each recognized word, the conversational back-end can repair potential speech recognition errors to improve user's conversational experience. In a speaker-dependent or speaker-adaptive system, the confidence measure can be used to help user enrollment (eliminate misspoken training words) as well as unsupervised speaker adaptation.

The problem of computing the confidence measure can be regarded as a statistical hypothesis testing problem or two-class pattern classification problem. Traditionally, there are two kinds of likelihood used for alternative hypothesis in hypothesis testing. One is the likelihood derived from filler models [1,2], which have been proven to be simple and effective for many spoken language applications. Another is the likelihood derived from so-called anti-keyword models [3,4], which can be built with anti-class samples. Recently some researchers reported that the information of the decoding process is very useful for the confidence measure [5,6,7]. In these systems, features such as N-best hypotheses, the number of active hypotheses during decoding, language model scores, acoustic scores are combined to produce the most reliable confidence measure. These features are typically fed to a sophisticated classifier [6,7,8,9] to obtain the classification result.

Discriminative training has been explicitly used in a number of systems to compute confidence measures [3,4]. They are particularly effective in small-vocabulary applications for tasks such as keyword spotting or utterance verification. Most of them have been applied to word-level features.

It has been reported that phone-level features can improve confidence estimation performance [6,10]. In [10], the phone level score is performing significantly better than the corresponding word-level score. In [6], the use of phone-level predictor variables achieves very good cross-entropy reduction. These experiments indicate that finer granularity in feature might be able to help us distinguish the correct and incorrect hypothesis, especially for confusable words.

In this paper, we focus on the features at sub-word level and how to effectively use them for general-purpose large vocabulary speech recognition. Our study is focused on the application of confidence measures for user enrollment of speech recognition systems. The performance of confidence measure using features at word, phone and senone level is experimentally studied. Our experimental results indicate that that the finer granularity is generally helpful.

In most conventional systems, when a decision is made at the word level with sub-word level features, each sub-word unit is typically weighted equally. In fact, different phones clearly have different impacts on our perception of words. In this paper, A framework of transformation function based system using sub-word features is proposed. The transformation function can be optimized by certain objective functions. For example, discriminative training can be used to optimize the parameters of the linear transformation functions. Experiments show that performance is significantly improved with linear transformation of features.

This paper is organized as follows. In Section 2, we explain our system setup and the data we used in our experiments. In section 3, we compare the performance of using features at different levels, namely, word, phone and senone level. In Section 4 we discuss the framework of transformation of sub-word feature to achieve better word-level performance. In linear transformation case, discriminative training is used to optimize the parameters. In section 5, we combine more than one feature under the same framework. Finally we summarize our major findings and outline our future work.

2. EXPERIMENTAL SETUP

Microsoft's WHISPER speech recognition system [2,12] is used in our experiments. It processes 16kHz PCM data using a MEL-scale cepstrum along with its dynamics into a multi-dimensional feature vector. The acoustic model we used here is a simplified version – a set of HMMs with continuous-density output probabilities consisting of 3000 senones [11]. A mixture of 4 Gaussian densities with diagonal covariances is used for each

senone. The phonetic modeling in the system consists of position and context dependent within-word and crossword triphones. A more complete description of the Whisper speech recognition system can be found in [2,12].

In this experiment, confidence is derived for each word given transcription. The speaker-independent portion of the North America Business News (NAB) is used. About half the data are used to train the acoustic model. The rest are reserved as development and test sets (no speaker overlap). The development and test set contains 4251 and 1082 sentences respectively. Two test cases are formed:

- *Random Test:* Every sentence is used twice in the test set; one with correct transcription and another with totally random transcription.
- *Confusable Test:* Recognition is performed for every sentence. The decoding result is used as the transcription.

The *Confusable Test* is used for most of the experiments, which is far more difficult than the *Random Test*. Unless specified explicitly, *Confusable Test* is used.

3. BASELINE SYSTEMS

3.1 Word and Phone Based Filler-Model

The filler-model is a fully connected all-phone network [1,2]. It can be evaluated using a Viterbi beam search with its own beam width. Transitions between phones can be weighted by phonetic bigram probabilities that could be trained using a typical lexicon and language model. The best path determined by the evaluation of the all-phone network can be considered as alternative hypothesis. The ratio between the length-normalized score obtained from Viterbi alignment based on given transcription and the length-normalized score obtained from all-phone network is the feature used for confidence measure.

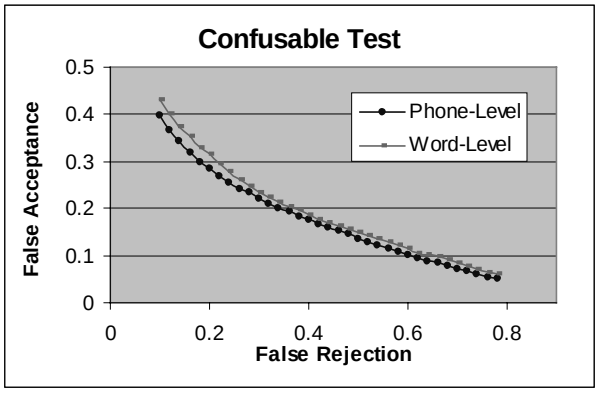

Figure 1: Comparison of filler-model performance at word and phone level

Both context-independent (CI) and context-dependent all-phone (CD) network have been considered in our experiments. Surprisingly, the CD all-phone network does not provide any extra help except the fact that the interpolation of CI and CD all-phone network scores demonstrates modest improvement.

The filler-model can be applied in the word-level or phone-level. In the word-level, the feature is directly compared to a threshold and the decision can be made as CORRECT/ INCORRECT. In the phone-level experiment, the features are accumulated for all the phones in the word and then normalized over the number of the phones in the word. Figure 1 illustrates the performance curve of both phone-level and word-level results.

3.2 Phone and Senone Based Anti-Model

The anti-model [3,4] for a class is typically trained with all the data that are not associated with the class. It is often used with a set of specific model trained for confidence measure. In this experiment the confidence model is trained on segmental features for every sub-word unit, which consists of cepstrum feature, its dynamics and segment duration. A Gaussian is built for each sub-word unit. For anti-model, instead building the model with all the data of anti-classes, we used the best Gaussian selected from the context-independent sub-word units which do not belong to the class. The ratio of length normalized likelihood between sub-word unit model and anti-model is used as the feature to compute the confidence score.

The anti-model can be applied either at the triphone level or at the senone level. Since the number of triphones is generally large, we group the triphones to clustered-triphones based on senones – each clustered triphone has a unique HMM in terms of senones. For 3000 senones, we have about 15000 clustered triphones. Some of the triphone models are untrained, but they are not frequently used and can be backed off to context-independent models. Figure 2 shows that finer granularity feature offers slightly better performance. In these experiments, it is clear that anti-model performs worse than the baseline filler-model.

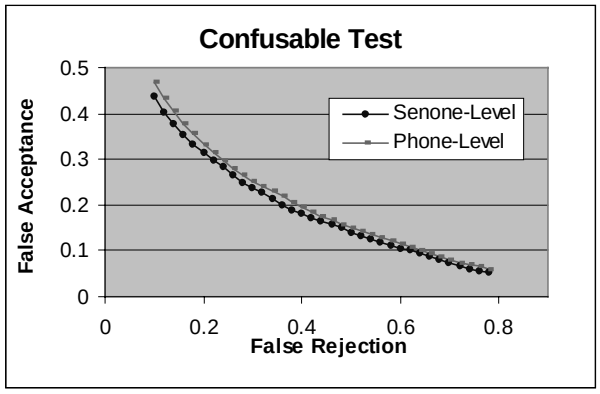

Figure 2: Comparison of anti-model performance at phone and senone level.

4. FEATURE TRANSFORMATION

It is clear that sub-word level features are better than word-level features in above experiments. In these experiments, when a decision is made at the word level with sub-word level features, each sub-word unit is weighted equally. In fact, different phones clearly have different impacts on our

perception of words. The simplest modification would be assigning each sub-word unit a different weight. The weights can be optimized from the training and development set. In general, we can use a transformation function *f(x)* to map features at sub-word level.

$$CS(w) = \frac{1}{N}\sum_{i=1}^{N} f_{class(U_i)}(x_i)$$

Where *CS(w)* is the confidence score for word *w*, *xi* is the feature (e.g., log likelihood) for *ith* sub-word unit in word *w* and *N* is the number of phones in word *w*. *Ui* is the id of the *ith* sub-word unit and the mapping function *class(Ui)* will allow us to determine how many transformation functions we want to use. As in the baseline, typically function *f(x)* is defined as:

$$f(x) = x$$

Here we propose to have a linear transformation function:

$$f(x) = ax + b$$

In this case, we can use discriminative training to optimize the parameters *a* and *b* [3,4]. A cost function can be defined as a sigmoid function on top of *CS(w)*. Then gradient descent algorithm can be used to optimize parameters in function *f(x)* to minimize the expectation of cost function over all the samples used in optimization. In our experiments development set data are used to train the transformation parameters.

4.1 Context-Independent Transformation

Starting with the *Random Test* case, we have a linear transformation function for each context-independent phone class. Filler-model is used to generate feature in this experiment as it performs better than anti-model. From Figure 3 we can clear see that the linear transformation of the feature helps greatly.

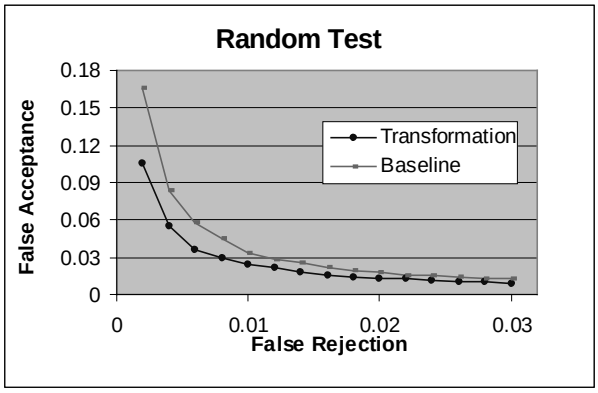

Figure 3: Comparison of performance on *Random Test* case with and without linear transformation (CI class).

To illustrate the effects of transformation, the transformation parameters of *a* is illustrated in Figure 4 (only some of the phones are labeled due to resolution). We notice that on average the weight for consonants is greater than the weight for vowels.

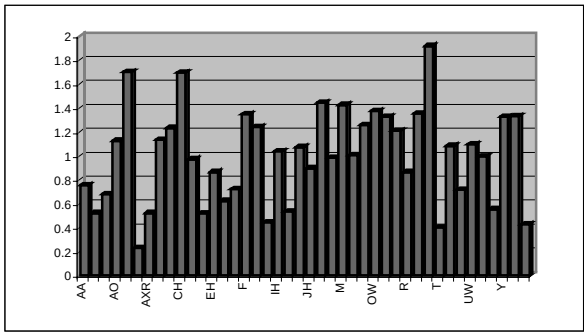

Figure 4: Transformation parameter *a* for each phone class

4.2 Context-Dependent Transformation

As the *Confusable Test* set is much more difficult than the *Random Test* set, we suspect that we might need more parameters in transformation functions. Experiment shows that using context-independent transformation functions on the *Confusable Test* set only yields some marginal improvement. That seems to validate our hypothesis. Therefore we extend the transformation function from context-independent class to context-dependent class. In our experiments we have one function for each clustered triphone. The added parameters work extremely well to model the context-dependencies. In Fig 5 we can clearly see the significant performance gain by using context-dependent transformation functions. In comparison to the baseline, it shows that the feature transformation based system reduces the equal error rate by 15% with up to 40% false acceptance error reduction at various fixed false rejection rate.

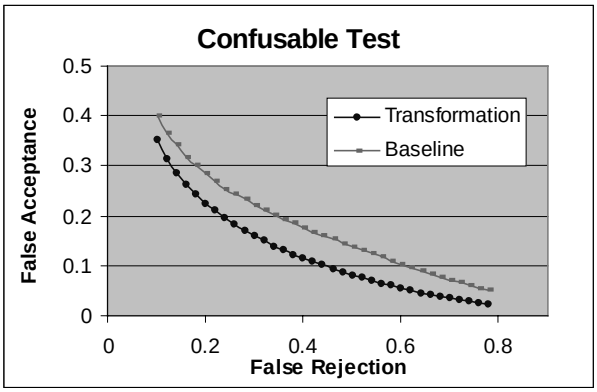

Figure 5: Comparison of performance on *Confusable Test* case with and without linear transformation (CD class)

5. FEATURE COMBINATION

In many cases, we need to use more than one feature in the confidence measure. Combination of the features typically requires a careful weighting scheme. The most frequently used approach includes: ad-hoc tuning, linear classifier [7], generalized linear model [6] and neural networks [6, 7] etc. With feature transformation function, it is very straightforward to incorporate multiple features. All we need to do is to take all

the features as the parameters of the transformation function. The advantage is, the parameters are class-dependent so they could be more flexible compared to the approaches mentioned above. For example, if we want to integrate the feature generated by the filler-model and feature generated by the anti-model, we can have function:

$$f(x, y) = ax + by + c$$

Where x is the feature generated by filler-model and y is the feature generated by anti-model.

In our experiment, we use the features generated by phone-level filler-model and senone-level anti-model. Note that features are also generated at clustered-triphone level with the senone-level anti-model so transformation can be applied. The result with feature combination is shown in Figure 6. Although the anti-model is performing considerably worse than the filler model, the experiment demonstrates that combining them together can further improve the performance.

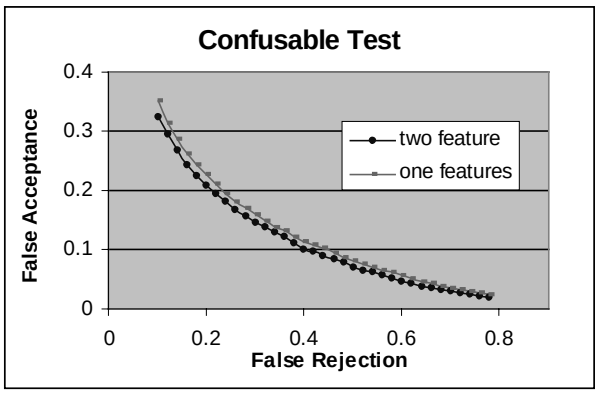

Figure 6: Comparison of transformation of one feature vs. two features

6. SUMMARY

In this paper we have demonstrated the effectiveness of using sub-word features in word confidence measure. In particular, we have shown that the linear transformation of features works significantly better than the widely used all-phone baseline system. By using linear feature transformation, we have achieved 15% equal-error rate reduction, with up to 40% false acceptance error reduction at various fixed false rejection rate. Also we have demonstrated that multiple features can be easily incorporated with linear feature transformation functions.

It is clear that the sub-word features are important and modeling of context-dependency also plays an important role. We will continue to focus on this direction. In the future, it will be interesting to integrate more sophisticated features to the proposed confidence measure framework, especially the information carried or produced by decoding process.

7. ACKNOWLEDGEMENTS

The authors would like to thank Milind Mahajan and Hsiao-Wuen Hon of our group for the useful discussions.

8. REFERENCES

[1] Asadi A., Schwartz R. and Makhoul J., "Automatic Modeling of Adding New Words to a Large-Vocabulary Continuous Speech Recognition System", *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1991

[2] Huang X., Acero A., Alleva F., Hwang M.Y., Jiang L. and Mahajan M. "Microsoft Windows Highly Intelligent Speech Recognizer: Whisper". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, May 1995.

[3] Sukkar R., Setlur A.R., Rahim M.G. and Lee C.H., "Utterance Verification of Keyword Strings using Word-Based Minimum Verification Error (WB-MVE) Training", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May, 1996

[4] Rahim M.G., Lee C.H., Juang B.H. and Chou W., "Discriminative Utterance Verification using Minimum String Verification Error (MSVE) Training", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May, 1996

[5] Eide E., Gish H., Jeanrenaud P. and Mielke A., "Understanding and Improving Speech Recognition Performance through the Use of Diagnostic Tools", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, May 1995.

[6] Chase L., "Word and Acoustic Confidence Annotation for Large Vocabulary Speech Recognition". *European Conference on Speech Communication and Technology*, Rhodes, Greece, September, 1997

[7] Schaaf T. and Kemp T., "Confidence Measures for Spontaneous Speech Recognition", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, May, 1997

[8] Weintraub M, Beaufays F., Rivlin Z., Konig Y. and Stolcke A., "Neural Network Based Measures of Confidence for Word Recognition", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, May, 1997

[9] Siu M., Gish H. and Richardson F., "Improved Estimation, Evaluation and Applications of Confidence Measures for Speech Recognition", *European Conference on Speech Communication and Technology*, Rhodes, Greece, 1997

[10] Rivlin Z., Cohen M., Abrash V. and Chung T., "A Phone-Dependent Confidence Measure for Utterance Rejection", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May, 1996

[11] Hwang, M.Y. and Huang, X. and Alleva, F. "Predicting Unseen Triphone with Senones". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, pages 311-314. April, 1993.

[12] Alleva F., Huang X. and Hwang M., "Improvements on the Pronunciation Prefix Tree Search Organization", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May, 1996