COMPOSITE BACKGROUND MODELS AND SCORE STANDARDIZATION FOR LANGUAGE IDENTIFICATION SYSTEMS

Terry P. Gleason, Marc A. Zissman

Lincoln Laboratory
Massachusetts Institute of Technology
244 Wood Street
Lexington, MA 02420–9108 USA
Voice: +1 781 981-3915
Fax: +1 781 981-0186
E-mail: TPG@SST.LL.MIT.EDU

ABSTRACT

This paper describes two enhancements to our language identification system. Composite background (CBG) modeling allows us to identify target language speech in an environment where *labeled* background training data is unavailable or limited. Instead of separate models for each of the background languages, a single composite background model is created from all the non-target training speech. Generally, the CBG system performed about as well as a baseline system containing a separate model per background language. The average equal error rate for 12 CBG tests was 13.6% versus 13.4% for the baseline. We have also developed and tested a standardized confidence scoring function based on a single-layer perceptron which has proven to be capable of robust modeling of score distributions.

1. INTRODUCTION

As speech recognition systems proliferate at locations frequented by speakers of many languages (e.g. hotel lobbies, international airports), language identification (LID) systems will be used as preprocessors to determine how to route utterances for speech recognition. We have reported previously that our **P**honeme **R**ecognition followed by **L**anguage **M**odeling performed in **P**arallel (PRLM-P) system provides state-of-the-art language identification performance on conversational, telephone speech [1]. In this paper, we describe a composite background (CBG) model that can be used effectively in place of multiple single-language background language models. We also discuss ways of producing a standardized confidence score.

The rest of the paper is organized as follows: Sections 2 and 3 review the PRLM-P algorithm and CALLFRIEND speech corpus, respectively. Section 4 introduces composite background models and reports on their performance compared to conventional single-language models. The standardized confidence score function is described in Section 5. Finally, we draw some conclusions in Section 6.

2. LANGUAGE ID ALGORITHM

Because the basic PRLM-P LID algorithm has been described in detail elsewhere ([1],[2]), we only provide a quick summary here.

2.1. Basic Algorithm

Figure 1 shows a block diagram of the PRLM-P system. HMM-based phone recognizers are trained using a phonetically labeled subset of the OGI training speech [3] in each of six languages: English, German, Hindi, Japanese, Mandarin, and Spanish. Each phone recognizer takes as input a stream of mel-weighted cepstra and delta cepstra computed from the incoming digitized speech and produces a stream of phone symbols as output. Interpolated n-gram language models (LMs) designed to capture the phonotactic statistics of each language are created by passing the training speech for each of the languages to be recognized through each of the six front-end (FE) phone recognizers and recording the unigram and bigram counts. Though we can only build FE phone recognizers in languages for which we have orthographically or phonetically transcribed speech, we can use the PRLM-P system to perform LID even on languages for which no orthographically or phonetically transcribed speech is available.

2.2. Combining Scores

The final likelihood scores for each language for each utterance can be calculated any number of ways. The simplest approach is to set the log likelihood that the utterance is spoken in language L equal to the arithmetic sum of the log likelihoods emanating from each of the six n-gram models for language L. The underlying assumption of this simple technique for combining the scores is that the various phone recognizers and corresponding language models operate independently from one another. Summing the log likelihoods is equivalent to multiplying linear likelihoods, and multiplying the linear likelihoods is appropriate if events are independent. Although this was our approach through 1995, we have since adopted the strategy used by Yan [4], whereby we consider the linear likelihoods output by the various language models as elements of a feature vector. If there are N_F FE phone recognizers and N_L languages to recognize, then there are $N_F \times N_L$ elements in the feature vector. During development training and testing, we train N_L Gaussian models with the multi-dimensional mean and variance of these likelihood feature vectors. During final (evaluation) testing, we compute the likelihood given each language and select as our language hypothesis that language whose Gaussian model yields the highest likelihood. We usually use a diagonal, grand covariance matrix, meaning that all models share the same covariance matrix that has non-zero elements only along its diagonal. Presumably we would obtain better performance with language-dependent, full-covariance matrices, but we rarely have enough development data to estimate so many parameters accurately.

3. CALLFRIEND SPEECH CORPUS

The Linguistic Data Consortium (LDC) CALLFRIEND corpus contains speech for 12 languages collected from telephone conversations between friends. We used the training partition to build the LMs, the development partition to create Gaussian backend and score standardization models, and the 30-sec segments from the evaluation partition for testing. English, Mandarin, and Spanish had approximately 160 test segments each. The other 9 languages had approximately 80 test segments each.

THIS WORK WAS SPONSORED BY THE DEPARTMENT OF DEFENSE UNDER AIR FORCE CONTRACT F19628-00-C-0002. OPINIONS, INTERPRETATIONS, CONCLUSIONS, AND RECOMMENDATIONS ARE THOSE OF THE AUTHORS AND NOT NECESSARILY ENDORSED BY THE UNITED STATES AIR FORCE.

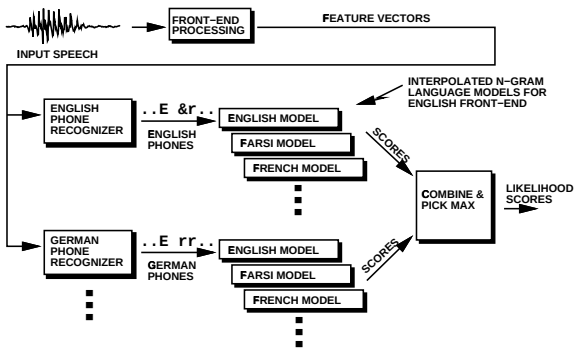

Fig. 1. The PRLM-P algorithm.

4. COMPOSITE BACKGROUND LANGUAGE MODELS

In some LID applications, test utterances may be spoken in languages for which adequate training data is not available. This leads us from conventional closed-set forced-choice language identification towards language verification. For example, if a LID system is used as a front-end to a bank of speech recognizers, the LID output should indicate whether the speech utterance is spoken in one of the "target" languages (i.e., one of the languages for which we have a speech recognizer available) or is spoken in a "background" language (i.e., a language which is not one of the target languages).

In language verification applications, it is clear that we need a model for each target language. However, we have at least two options for modeling the background. In the conventional approach, we build a separate LM for each background language. This is especially appropriate when we have enough data and knowledge to train a separate LM for each possible background language that could occur during testing. However, in cases when we either do not have enough training data for individual background LMs, or when we do not have enough knowledge to identify all possible background languages, we propose a second approach that uses a composite background (CBG) model, i.e., a single model that is trained from all the (unlabeled) background data.

4.1. CBG Versus Conventional Background Models

In our baseline system (Figure 1), training data is used to build LMs for the individual target and background languages expected in the test data. Our goal is to determine if one CBG model, built from unlabeled background training data, can be as effective as a baseline system that uses individual background LMs, one per background language. A CBG system that performed as well as the baseline system would offer several advantages. First, less effort and expertise would be required to label the training data because only the target speech would need to be identified. Any speech that is not target is labeled as background. Secondly, in cases where background training speech is limited, additional partitioning for training separate background LMs might result in undertrained models, whereas one CBG model would use all the background training speech. Finally, the single CBG model would save a modest amount of CPU run time over using multiple background models. Though we are interested in the general case of multiple targets and multiple background languages, the experiments run so far have used only a single target language with multiple background languages.

4.2. Experiments and Results

In order to determine how effective and how robust CBG LMs are compared to separate background LMs, a set of experiments was run where we varied the languages, the number of languages, and the amount of training for the background languages. In all our CBG experiments we used four target languages (Arabic, English, Farsi, and Hindi), testing each target language separately against a variety of randomly selected background language sets. Table 1 lists the 20 test suites used in our initial CBG experiment. Each target language was combined five times with two background languages to make up the 20 test suites. This allowed us to determine if performance was dependent on the specific target or background languages.

i	Target-BG1-BG2	i	Target-BG1-BG2
1	Ara-Fre-Jap	11	Far-Ara-Tam
2	Ara-Fre-Tam	12	Far-Eng-Spa
3	Ara-Ger-Man	13	Far-Eng-Vie
4	Ara-Kor-Spa	14	Far-Fre-Ger
5	Ara-Man-Vie	15	Far-Ger-Jap
6	Eng-Ara-Far	16	Hin-Ara-Jap
7	Eng-Ara-Spa	17	Hin-Far-Fre
8	Eng-Hin-Jap	18	Hin-Fre-Vie
9	Eng-Hin-Vie	19	Hin-Ger-Man
10	Eng-Tam-Vie	20	Hin-Jap-Tam

Table 1. Initial 20 "Target-BG1-BG2" test suite indices. Target languages: Arabic, English, Farsi, Hindi. BG languages: all 12 CALLFRIEND languages.

To evaluate CBG LMs, we compared the performance of the CBG system with its corresponding "baseline" system in which each background language was modeled explicitly. Performance was measured by computing the equal error rate (EER), i.e., the point in the receiver operating curve where the fraction of misses is equal to the fraction of false alarms. When we check the performance of the CBG or baseline system we only measure its ability to distinguish target from background speech, i.e., we are not interested in discriminating among the various background languages.

4.2.1. Initial 20 Test Suites

Figure 2 compares baseline and CBG EERs for the 20 test suites. Generally, the CBG system performed about as well as the baseline. The average EER for the 12 CBG tests was 13.6% versus 13.4% for the baseline. As the chart shows, performance is highly dependent on the target language. For example, the five results with English as a target language (indices 6-10) were among the best scores, whereas the five Hindi target results (indices 16-20) were all among the worst scores. This target-dependent performance was observed for both the baseline and CBG results.

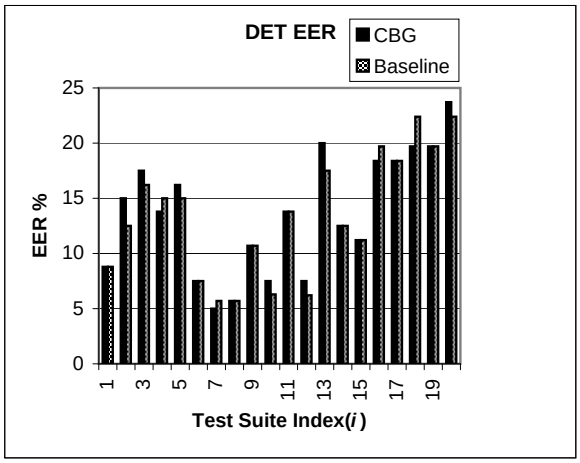

Fig. 2. Twenty Test Suite Results: Baseline Vs. CBG

4.2.2. *Increasing the Number of Background Languages*

Using the same four target languages, we ran five more experiments per target language where we increased the number of background languages from 3 to 7. Again, we found results to be comparable between the baseline and CBG systems and mostly dependent on the target language. Also there was a slight degradation in performance as more BG languages were added. Overall, the average EER was 15.8% for the CBG versus 15.3% for the baseline.

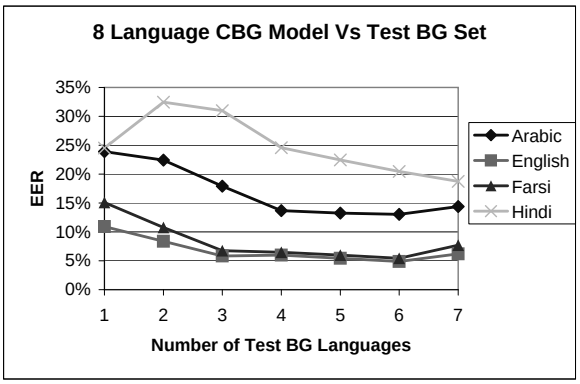

Fig. 3. CBG models with extraneous training.

4.2.3. *CBG LMs With Extraneous Training*

We also trained one CBG with all eight background languages and then varied the background test set to include 1 to 7 of the background languages in the CBG model. We wanted to see if performance was affected when BG training data included extraneous languages not encountered in testing. Figure 3 shows that performance for three of the four target languages is unaffected as long as the test language set comprises at least 4 (50%) of the languages used to train the CBG.

4.2.4. *CBG LMs With Disproportionate Training*

We measured how critical the proportions of training speech are for two background languages. Using four test suites, each consisting of one target and two background languages, we trained using all of the first background ($BG1$) training data while varying the proportion of the second background ($BG2$) training data for the CBG model. Figure 4 shows that performance degraded for all four target test suites when the $BG2$ proportion of training was less than about 30% of the total training. Note that as we decreased the amount of $BG2$ training, we also decreased the amount of overall training for the CBG LM, and it is not clear what effect that alone would have had.

4.2.5. *CBG LM With Target Speech Training*

Given that performance was fairly robust even when the CBG training mix did not match the background test mix, we tried one more mismatch experiment where we included target training speech in the CBG LM. We used our original 20 test suites (Table 1) and compared the EER for each test suite with and without target training. Even though the target training made up 33% of the total CBG training, it did not degrade performance: 13.7% EER for the CBG with target training vs. 13.6% for the CBG without target training.

4.2.6. *CBG Plus Gaussian Backend*

All our experiments to this point compared baseline and CBG systems without a Gaussian Backend (GBE). Because the PRLM-P

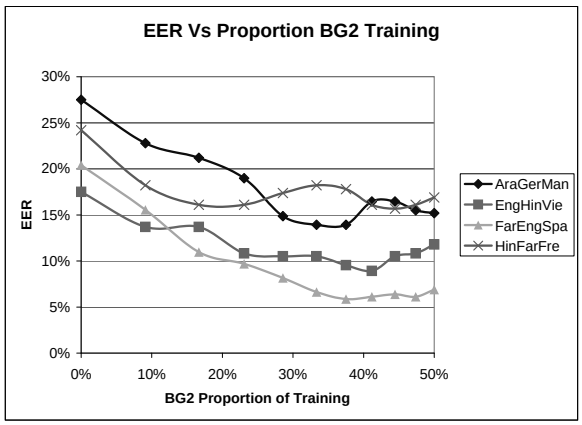

Fig. 4. CBG model training mix.

	Baseline	CBG
Without GBE	15.3%	15.8%
With GBE	13.2%	15.3%

Table 2. GBE Performance Improvement: Baseline versus CBG. Average of 20 tests with 3 to 7 background languages.

GBE uses all the FE phone recognizer LM scores as inputs[1], we were curious to see if the reduction of inputs to the GBE in the CBG system would affect its performance. The GBE in our CBG test systems would always only have 12 inputs: two LM scores (target and CBG) from each of the six FEs. However, with six FE phone recognizers and N_L LMs (target + background) the GBE in the baseline would have $6 \times N_L$ inputs. Using the same 20 test suites where we increased the background languages from 3 to 7 for our four target languages (Section 4.2.2), we rescored the baseline and CBG systems with our GBE. The GBEs in the CBG systems had 12 inputs, whereas the baseline GBEs had 24 to 48 inputs.

As Table 2 shows, with the GBE, the CBG system only improved its overall EER to 15.3% from 15.8% without a GBE. The baseline system, saw a bigger improvement: 13.2% with a GBE compared to 15.3% without a GBE. The result is consistent with our experience that additional LM scores usually help the GBE.

5. STANDARDIZED SCORES: GAUSSIAN VERSUS A SINGLE-LAYER PERCEPTRON MODEL

In many language verification problems, we would like the LID system to output a confidence score associated with a particular hypothesis. The range of this score should be between 0 and 1, and it should reflect the system's estimate of the likelihood that its hypothesis is correct. Experience has shown that LM likelihoods, LM likelihood ratios, GBE likelihoods and GBE likelihood ratios cannot provide, in their raw form, this confidence information. We describe below two approaches to producing confidence scores: Gaussian modeling and single-layer perceptron modeling. Both of these approaches require training.

5.1. Gaussian Models

Standardized scores can be obtained by modeling the log likelihood ratios (LLRs) using Gaussian probability densities $N(\mu, \sigma)$ for target (t) and background (BG) scores:

$$P(LLR_t) = N(\mu_t, \sigma_t) \quad (1)$$

$$P(LLR_{BG}) = N(\mu_{BG}, \sigma_{BG}) \quad (2)$$

Estimates of the density parameters are obtained by using a set of development target and background messages.

Given an unknown message and its target LM LLR, the standardized confidence score is derived by the posterior probability of the target language:

$$Score = \frac{P_t(x)Pr(t)}{P_t(x)Pr(t) + P_{BG}(x)Pr(BG)} \quad (3)$$

where $Pr(t)$ and $Pr(BG)$ are the a priori probabilities for the target and BG messages.

The Gaussian model works well when the background and target score distributions are Gaussian and have comparable variances. If, however, a distribution is skewed or the variances are not comparable, the Gaussian model can do a poor job mapping the scores, especially near the tails. In some cases the Gaussian model creates a non-monotonic mapping with the undesirable effect that high LLR scores map to low standard scores.

5.2. Single-Layer Perceptron Model

We have developed a new model for computing posterior scores that uses a single-layer perceptron (SLP). Compared to a Gaussian classifier, an SLP classifier [5] has two main advantages. First, the SLP is modeling the observed posterior probability directly whereas the Gaussian method uses two models to derive the posterior probability. Secondly, SLPs (no hidden layers) guarantee a monotonic relationship between input (in our case, the likelihood ratio scores) and output (the standard scores).

5.3. CALLFRIEND Hindi Example

To illustrate the advantages of the new SLP model over the Gaussian, consider the case of trying to detect CALLFRIEND Hindi from a set of 11 background languages. Figure 5 shows how LLR scores from development data are used to create a Gaussian mapping to the standard confidence scores. The two jagged curves are the actual histograms (percent) of Hindi LLRs showing the target (diamonds) and background (triangles) score distributions. The two smooth curves are the computed Gaussian models for the data. In this example, prior probability of the target was approximately 8% (1 of 12 languages). As can be seen, the background distribution is non-Gaussian, resulting in a background Gaussian model tail that exceeds the data curve it attempts to model.

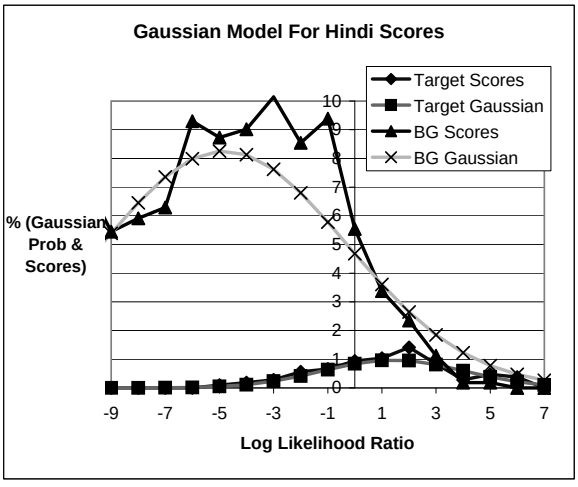

Fig. 5. Gaussian Modeling of Hindi PRLM-P LLR Scores.

The measured target posterior score, computed from the target and background histograms, is shown as the curve marked with diamonds in Figure 6 . The Gaussian posterior score, computed using equation 3 and the Gaussians of Figure 5 is shown in Figure 6 by the curve marked with squares. We can see that the Gaussian posterior model does a poor job of modeling the actual posterior data due to the ill-effects of the background Gaussian model.

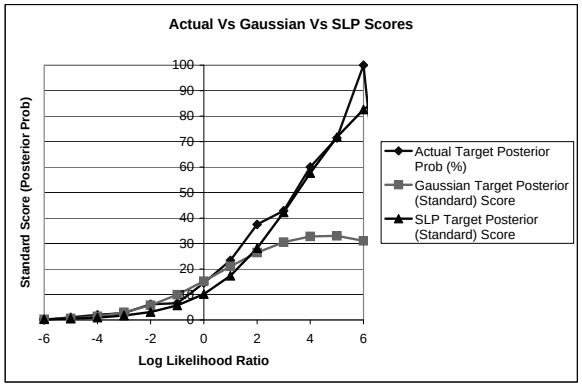

Fig. 6. Posterior Scores For Hindi.

Using the same input scores, an SLP model was also trained for Hindi. As seen in Figure 6 the SLP model (shown as the curve with triangles) is both monotonic and maps the tail LLR scores well. Although not shown here, the SLP model worked well across all the CALLFRIEND languages.

6. CONCLUSIONS

CBG LMs are effective and robust. Without Gaussian BEs, they performed comparably to a baseline system with individual background models over a variety of languages. Performance was not affected by small mismatches in training versus test language mixes. Even including target speech in the CBG training did not degrade performance. Adding Gaussian BEs improved performance for both systems, but the baseline system saw a bigger improvement, likely due to its richer set of inputs. CBG LID systems can be useful when background training data is unlabeled or when there is insufficient data to build separate background LMs.

Our experiments showed that the SLP standardization model is more robust than Gaussian standardization. In addition, it guarantees the desired monotonic mapping of LLR scores to the standardized confidence scores.

7. REFERENCES

[1] M. A. Zissman, "Predicting, diagnosing, and improving automatic language identification performance," in *Proceedings of Eurospeech 97*, September 1997, vol. 1, pp. 51–54.

[2] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech and Audio Proc.*, vol. 4, no. 1, pp. 31–44, January 1996.

[3] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika, "The OGI multi-language telephone speech corpus," in *ICSLP '92 Proceedings*, October 1992, vol. 2, pp. 895–898.

[4] Y. Yan and E. Barnard, "Experiments for an approach to language identification with conversational telephone speech," in *ICASSP '96 Proceedings*, May 1996, vol. 2, pp. 789–792.

[5] M. D. Richard and R. P. Lippmann, "Neural network classifiiers estimate bayesian a posterior probabilities," *Neural Computations*, , no. 3, pp. 461–483, 1992.