Speech Lab at Queens College: Language Recognition Evaluation 2015

Guozhen An[1], *David Guy Brizan*[1], *Felix Grezes*[1], *Min Ma*[1], *Michelle Morales*[2], *Andrew Rosenberg*[3]

[1]Department of Computer Science, CUNY Graduate Center, USA
[2]Department of Linguistics, CUNY Graduate Center, USA
[3]Department of Computer Science, Queens College (CUNY), USA
{gan, dbrizan, fgrezes, mma, mmorales}@gradcenter.cuny.edu, andrew@cs.qc.cuny.edu

Abstract

The Speech Lab at Queens College LRE 2015 submission contains one primary system and four contrastive systems submitted to the NIST Language Recognition Evaluation Plan (LRE15).

1. Low-level Descriptor Features System

1.1. Description

The primary system used approximately 6,373 low-level features as described by the Interspeech 2013 COMPARE Challenge [1], extracted with OpenSMILE [2] using the challenge configuration.

1.2. Training Data

This system was trained exclusively on the LRE15 training set.

1.3. Processing Speed

The system was executed on an 8-processor Intel Xeon system (each 3.0GHz) with 12GB RAM running Ubuntu 15.04. The speed of language recognition, defined as the total time duration of speech processed divided by the total (user) CPU time was 29.93. The maximum amount of memory used (during prediction) was 1,744,563.24 kbytes.

The system was executed in two distinct, consecutive phases: Feature Extraction and Prediction. Table 1 contains detailed timing for the Low-level Descriptor Features system.

Table 1: *Processing Time: Low-level Descriptor Features*

	User Time (sec)	System Time (sec)	Total Time (sec)
Feature Extraction	99470.50	21870.66	121341.16
Prediction	4717.60	1379.68	6097.28
Total Time	104188.10	23250.34	127438.44

We provide these metrics with the caveat that wall clock timing and memory usage are very unstable measures. They are extremely sensitive to even minor changes in architectures and load. Differences of less than an order of magnitude are likely insignificant. Comparisons between systems based on these numbers should be performed with this in mind.

2. Phonemic Inventory Features

2.1. Description

This system used 196 phoneme-based features. Using the PhnRec tool [3], phone hypotheses were extracted in 4 languages (Czech, English, Hungarian, and Russian). The output of the PhnRec tool consists of phone hypotheses, durations, and confidence scores. Using this output, we derived the following features: vowel and consonant inventory (unique number of vowels and consonants), consonant-vowel ratio, average confidence score, and statistical functionals applied to vowel and consonant durations, specifically: mean, maximum, minimum, standard deviation, and variance. Finally, we added frequency, confidence and duration functionals features for each consonant type, where consonant types include: affricates, fricatives, glottal stops, sonorants, and stops. Consonant types are determined using the Speech Assessment Methods Phonetic Alphabet, a machine-readable phonetic alphabet [4].

We used the Weka [5] SMO classifier to generate language hypotheses from these features. All parameters were kept at their default values.

2.2. Training Data

In addition to the LRE15 training set, this system used the phone hypotheses from PhnRec. These hypotheses are supplied by default in that system and were used without modification.

2.3. Processing Speed

The system was executed on different machines. The generation of phone hypotheses was performed on a 128-processor Intel Xeon system (each 2.8GHz) with 64GB RAM. All other portions of the system were executed on an 8-processor Intel Xeon system (each 3.0GHz) with 12GB RAM. Both systems ran Ubuntu 15.04. We report the User, System and Total Time for each because of the speed differences between the machines. The speed of language recognition, defined as the total time duration of speech processed divided by the total (user) CPU time was 673.80. The maximum amount of memory used (during prediction) was 1,242,416 kbytes.

Table 2 contains detailed timing for the Phonemic Inventory Features system in three distinct phases: Phone Hypotheses, Feature Extraction and Prediction.

We provide these metrics with the same caveat that wall clock timing and memory usage are very unstable measures.

Table 2: *Processing Time: Phonemic Inventory Features*

	User Time	System Time	Total Time
Phone Hypotheses	1853.88 s	10006.30 s	11860.18 s
Feature Extraction	2725.59 s	101.75 s	2827.34 s
Prediction	48.19 s	12.74 s	60.93 s
Total Time	4627.66 s	10120.79 s	14748.45 s

3. Parallel Phoneme Language Models

3.1. Description

This system used a two-tier approach to language detection: first, we used the PhnRec tool [3] to extract phone hypotheses in 4 languages (Czech, English, Hungarian and Russian). For each of the 20 language to identify, we trained four 3-gram phoneme language models with Witten-Bell smoothing based on different phoneme hypotheses, using the SRILM toolkit [6]. This resulted in a total of 80 perplexity scores ($10^{-\frac{logprob}{\#(word)}}$) as features.

We then employed the Weka [5] SMO classifier to make prediction from these features. Based on experiments performed on a held-out portion (13.5%) of the training material, we tuned the complexity parameter of SMO to 1000. All other parameters were kept at their default values. We re-scaled the Weka predictions so that within a cluster, each language probability summed to one. These probabilities were then converted to log-likelihoods, with an equal prior for each language class.

3.2. Training Data

In addition to the LRE15 training set, this system used the phone hypotheses from PhnRec. These hypotheses are supplied by default in that system and were used without modification.

3.3. Processing Speed

The system was executed on different machines. The generation of phone hypotheses was performed on a 128-processor Intel Xeon system (each 2.8GHz) with 64GB RAM. All other portions of the system were executed on three 8-processor Intel Xeon system (each 3.0GHz) with 12GB RAM. Both systems ran Ubuntu 15.04. We report the User, System and Total Time for each because of the speed differences between the machines. The speed of language recognition, defined as the total time duration of speech processed divided by the total (user) CPU time was 16.72. The maximum amount of memory used (during prediction) was 2,280,368 kbytes.

Table 3 contains detailed timing for the Parallel Phoneme Language Modeling system in three distinct phases: Phone Hypotheses, Perplexity Feature Extraction and Prediction.

Table 3: *Processing Time: PPRLM Features*

	User Time	System Time	Total Time
Phone Hypotheses	1853.88 s	10006.30 s	11860.18 s
Feature Extraction	184574.40 s	69874.40 s	254448.80 s
Prediction	48.26 s	12.93 s	61.19 s
Total Time	186476.54 s	79893.63 s	266370.17 s

4. Phone Variation

4.1. Description

This system used MFCC[0] vectors ("raw"), their deltas ("delta") and their double deltas ("double-delta") as derived by OpenSMILE [2]. For each set of phoneme hypotheses as derived by the PhnRec tool [3] across four languages (Czech, English, Hungarian and Russian), the raw, delta and double-delta for MFCC[0] vectors were determined and the following calculations were extracted as features: min, max, range, median, mean, variance, standard deviation.

We used the Weka [5] SMO classifier to generate language hypotheses from these features. All parameters were kept at their default values. We re-scaled the Weka predictions so that within a cluster, each language probability summed to one. These probabilities were then converted to log likelihood ratios, with an equal prior for each language class.

4.2. Training Data

In addition to the LRE15 training set, this system used the phone hypotheses from PhnRec. These hypotheses are supplied by default in that system and were used without modification.

4.3. Processing Speed

The system was executed on different machines. The generation of phone hypotheses was performed on a 128-processor Intel Xeon system (each 2.8GHz) with 64GB RAM. All other portions of the system were executed on an 8-processor Intel Xeon system (each 3.0GHz) with 12GB RAM. Both systems ran Ubuntu 15.04. We report the User, System and Total Time for each because of the speed differences between the machines. The speed of language recognition, defined as the total time duration of speech processed divided by the total (user) CPU time was 48.95. The maximum amount of memory used (during prediction) was 1,531,420 kbytes.

Table 4 contains detailed timing for the Phone Variation system in three distinct phases: Phone Hypotheses, Feature Extraction and Prediction.

Table 4: *Processing Time: Phone Variation Features*

	User Time	System Time	Total Time
Phone Hypotheses	1853.88 s	10006.30 s	11860.18 s
Feature Extraction	65020.62 s	1856.53 s	66877.15 s
Prediction	4076.52 s	27.29 s	4103.81 s
Total Time	70951.02 s	11890.12 s	82841.14 s

We provide these metrics with the same caveat that wall clock timing and memory usage are very unstable measures.

5. Ensemble

5.1. Description

This system combined the outputs of the four other systems described in Sections 1–4. Each trained model outputs a label prediction and a probability distribution for the 20 target languages. For each data instance, we concatenated the probability distributions for each of our systems and used this combined vector as our feature representation.

We used the Weka [5] SMO classifier to generate language hypotheses from these features. All parameters were kept at their default values. We re-scaled the Weka predictions so that within a cluster, each language probability summed to one. These probabilities were then converted to log likelihood ratios, with an equal prior for each language class.

5.2. Training Data

This system was trained on the probability distributions from each of the four systems above. Three of those systems used the phone hypotheses from PhnRec in addition to the LRE15 training set. The PhnRec tool hypotheses are supplied by default in that system and were used without modification.

5.3. Processing Speed

Each sub-system in the Ensemble was executed on different machines. The generation of phone hypotheses was performed on a 128-processor Intel Xeon system (each 2.8GHz) with 64GB RAM. All other portions of the system were executed on different 8-processor Intel Xeon system (each 3.0GHz) with 12GB RAM. All systems ran Ubuntu 15.04. We report the User, System and Total Time for each sub-system and for the Ensemble. The speed of language recognition, defined as the total time duration of speech processed divided by the total (user) CPU time was 8.51. The maximum amount of memory used (during prediction) was 2,769,148 kbytes.

Table 5 contains detailed timing for the Phone Variation system in three distinct phases: Sub-system Time (for the systems described in Sections 1-4), Feature Extraction and Prediction.

Table 5: *Processing Time: Ensemble*

	User Time	System Time	Total Time
Sub-system Time	366243.32 s	125154.88	491398.20
Feature Extraction	7.83 s	.86 s	8.69 s
Prediction	54.00 s	9.01 s	59.01 s
Total Time	366305.15 s	125164.75 s	491469.90 s

We provide these metrics with the same caveat that wall clock timing and memory usage are very unstable measures.

6. References

[1] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi *et al.*, “The interspeech 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism,” 2013.

[2] F. Eyben, F. Weninger, F. Groß, and B. Schuller, “Recent developments in opensmile, the munich open-source multimedia feature extractor,” in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 835–838.

[3] P. Schwarz, P. Matejka, L. Burget, and O. Glembek, “Phoneme recognizer based on long temporal context,” *Speech Processing Group, Faculty of Information Technology, Brno University of Technology.[Online]. Available: http://speech. fit. vutbr. cz/en/software*, 2006.

[4] J. C. Wells *et al.*, “Sampa computer readable phonetic alphabet,” *Handbook of standards and resources for spoken language systems*, vol. 4, 1997.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[6] A. Stolcke *et al.*, “Srilm-an extensible language modeling toolkit.” in *INTERSPEECH*, 2002.