Linguistically-Motivated Features for Language Recognition

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

In this paper, we compare three systems using phonotactic features (two of them novel) against an equivalent i-vector system and an equivalent voice quality system for language identification. We demonstrate that systems built on phonotactic features exhibit good prediction performance while maintaining equivalent or better metrics with respect to resource consumption.

Index Terms: Language Identification, Phonotactic Modeling

1. Introduction

Automatic language identification of speech is the process by which the language or dialect of an utterance is automatically recognized by a computer. This task has broad applicability, including areas such as automatic speech recognition, multilingual translation systems or even emergency call routing, where the response time of a fluent native operator could be critical. Some of the most efficient approaches to language recognition have relied on language dependent phone models, which are largely based on the assumption that phonotactic constraints contain enough information to identify the language. These approaches have borrowed inspiration from linguistic theory. To linguists, differences in the number of sounds that exist in a language and how these sounds can be combined are extremely important clues to identifying a particular language or distinguishing between two similar languages. Even if we consider geographically close languages, language families, or dialects of the same language we find great variation across phonemic inventories. For example, Mandarin Chinese has a relatively average size consonant inventory (22 ± 3). However, the dialect *Wu*, spoken in southeast China, differs from Mandarin Chinese in preserving initial voiced stops (sounds formed with complete closure in the vocal tract), leading it to have a larger consonant inventory [1].

Accordingly, language identification systems have leveraged this linguistic knowledge and have built systems that rely heavily on phonemes. After many formal evaluations [2, 3, 4], research suggests that one successful approach to automatic language identification involves using phonotactic content of the speech signal to discriminate among a set of languages. In these approaches, phone recognizers are used to tokenize speech into phone sequences, which are then modeled using statistical language models, i.e. phone recognition followed by language modeling (PRLM) [5]. This work explores how to exploit and best leverage the output of phone recognizers. We do so by building multiple models, which each employ recognizer output in distinct ways. Our phone-based models include: a Phonemic Inventory model, a Parallel Phone Recognition to Language Model, and a Phone Variation model. We contrast our phone-based models with equivalent systems built on acoustic features on the basis of accuracy and speed.

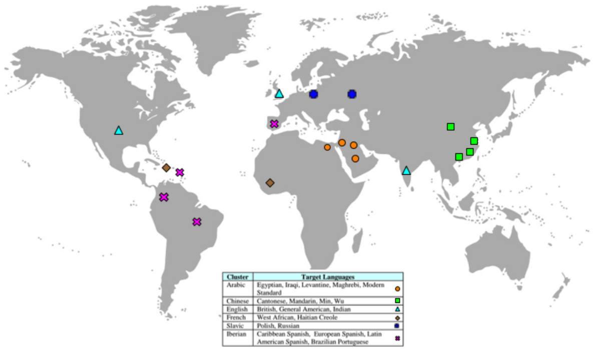
2. Related Work

Starting at least as far back as 1995, phonotactic approaches took advantage of distributional differences in phonology and became perhaps the most widespread state of the art technique for language (and dialect) detection [6]. This class of techniques has been used to distinguish between dialects [7] among several Arabic dialects [8], between two Spanish dialects [9] and among three Chinese dialects [10].

The PRLM approach [5, 8, 11] is a pipeline which starts with individual (vowel/consonant) phone recognition (PR) from an acoustic signal. During training, the output of this recognition is used to build an n-gram (language) model (LM) which captures the phonotactic probability distributions of the language/dialect. During testing, the output is used to compare against existing models, the most likely of which becomes the hypothesis for the label of the language or dialect. PRLM has an effective variant in which multiple parallel phone recognition systems, each trained on different languages or dialects, are used in parallel, with the output hypotheses of those systems combined with a classifier for final prediction [8].

Chen et al. [12] claim that phone recognizers fail to capture acoustic differences across dialects, such as the retroflex /d/ common in Indian dialects of English. To account for this, parallel PRLM, can be employed PRLM in multiple languages to tease out pronunciation differences among dialects. The "*marry-merry-Mary* merger" is an example: though most speakers of Standard American English hear these as homophones, some speakers in New England produce a distinct sound for each [13]. This distinction would be "inaudible" to a PRLM system built on Standard American English; however, other languages, and perhaps other dialects of English, may be sensitive to the distinction and therefore to the linguistic subculture from which the sounds are produced. Biadsy [14] proposed *discriminative phonotactics* as a method for handling the differences in phone realizations across languages. Under this technique, a system uses Gaussians' fit to the phone hypotheses from different languages to determine whether each phone is pronounced differently.

After more than a decade of the use of phonotactic approaches, the i-vector approach was proposed as a method for speaker diarization [15] as well as language identification [16]. The i-vector approach works by creating a "Universal Background Model" (UBM), *m*, representing the total variability of all speaker utterances in a supervector – a set of stacked mean vectors from a Gaussian Mixture Model (GMM) [17]. Each speaker utterance, *M*, can be described by the UBM, by a ma-



Cluster	Target Languages	
Arabic	Egyptian, Iraqi, Levantine, Maghrebi, Modern Standard	●
Chinese	Cantonese, Mandarin, Min, Wu	🟩
English	British, General American, Indian	▲
French	West African, Haitian Creole	◆
Slavic	Polish, Russian	●
Iberian	Caribbean Spanish, European Spanish, Latin American Spanish, Brazilian Portuguese	✖

Figure 1: *LRE15 Languages and Language Families.*

trix of features, T and a vector representing the identity of the speaker – the i-vector – w, all described in by (1). In language identification, it has been common to create a model for each language rather than each speaker, in [18] for example.

$$M = m + Tw \tag{1}$$

Several improvements have been suggested to the i-vector approach. For example, Shum et al. (2013) [17] describes a multi-step approach of post-processing i-vectors to derive the gender of the speakers. In this approach, Principal Components Analysis (PCA)-based projection is applied as a proportion of i-vector dimension. K-means clustering is applied to the output of this step based on cosine distance. (In this work, K was set only to 2 for the purpose of bifurcating gender. Additional refinements were applied to generate better segmentation for speaker diarization; these are omitted here.)

Our work borrows from and builds upon many of these existing approaches. Collectively, we implemented five different approaches, two of them novel: (1) a low-level acoustic-prosodic approach with a large set of acoustic features, (2) a novel phonemic inventory approach inspired by the PRLM approach [9], (3) a PPRLM approach [9], (4) a novel variation of the phonotactic approach described in [14], and (5) an i-vector approach [15, 16].

3. Data

For data, we used the training portion of the 2015 NIST Language Recognition Evaluation Plan (LRE15) corpus. This data was drawn from telephone and broadcast speech ("conversations") in 20 languages, each of which was assigned to one of 6 language families [4] due to the intra-family confusability of the languages. Languages and language families are shown in Figure 1 according to the stated or historical geographic origin of the language. All speech segments were provided in 16-bit, 8kHz linear PCM format in SPHERE file format.

We randomly held out 10% of the material for testing, with a minimum of one broadcast or telephone conversation per language. Hereinafter, we refer to this held out portion as the "development (dev) set." The remaining portion of the material was reserved for training our models as described in the following section.

Note that the LRE15 corpus also contains a separate test portion, sometimes recorded under different conditions in comparison to the training material. However, because the language labels for that portion of the corpus were not provided by the time of the writing of this work, we were unable to use that portion for all experiments. We do consistently report metrics on the development set, as described in Section 5.

4. Systems

We experimented with a total of five systems: one system built exclusively on comparatively low-level acoustic-prosodic features, three phonotactic systems of varying complexity and one system which used i-vectors. Each is described in this section. Across all systems, we used the entire contents of the conversation to produce a single prediction from one of our 20 languages. We did this to perform a fair comparison among these systems.

4.1. Low-Level acoustic-prosodic System

The Low-Level acoustic-prosodic System used approximately 6,373 "low-level" acoustic-prosodic features as described by the Interspeech 2013 COMPARE Challenge [19], extracted with OpenSMILE [20] using the baseline challenge configuration. Some of the low-level acoustic-prosodic system features include pitch (fundamental frequency), intensity (energy), duration, voice quality (jitter, shimmer, and harmonics-to-noise ratio). In two subsequent cases (the i-vector system and the Phone Variation system), we used MFCC features, which are a subset of the features in this system.

We trained models to hypothesize one of the 20 languages for which we had data and used the Weka [21] SMO classifier to generate language hypotheses from these features. All weka parameters were kept at their default values. We did not expect this system to outperform our phonotactic or i-vector systems, but we wanted to establish a baseline for good performance on this task on all our metrics (accuracy, confidence and resource consumption).

4.2. Phonotactic Systems

We created a total of three phonotactic systems of varying complexity. These systems targeted the different distributions and realizations of vowels and consonants for the different languages. To the best of our knowledge, our work is the first one that utilize phonemic inventory features, and our phone variation features are a unique variation of discriminative phonotactics. All of our phonotactic systems are built on phone hypotheses produced by the BUT phoneme recognition (PhnRec) tool [22], which supports four languages: Czech, English, Hungarian and Russian. We generate four language phone hypotheses using the trained model in PhnRec, and filter all the non-speech tokens('oth', 'pau', 'sil','spk','int') before feeding the transcripts to each system independently. The output of the PhnRec tool consists of phone hypotheses, durations and confidence scores; however, not all of these outputs were applied to each system.

4.2.1. Phonemic Inventory System

This system used 196 phoneme-based features. Using the PhnRec tool, we derived the features detailed in Table 1. For consonant types (C-Type) features, we tagged each consonant phone according to the following: affricates, fricatives, glottal stops, sonorants, and stops. These consonant types were determined using the Speech Assessment Methods Phonetic Alphabet (SAMPA) [23]. As with the above system, we used the Weka [21] SMO classifier (default parameters) to generate language hypotheses from these features.

Table 1: *Phonemic Inventory Features.*

Feature Class	Feature	Detail / Example
Inventory	Vowels	Unique number of vowels
	Consonants	Unique number of consonants
	C/V Ratio	Consonant to vowel ratio
	C-Type Ratio	Frequency by consonant type
Duration	Mean	Mean duration per phone
	Max	Maximum duration per phone
	Min	Minimum duration per phone
	Stdev	Standard deviation of duration per phone
	Variance	Duration variance per phone
	C-Type Dur.	Mean duration per consonant type
Confidence	Avg	Mean confidence score per phone
	C-Type Con.	Mean confidence score per consonant type

4.2.2. *Phone Recognition and Language Modeling (PRLM)*

First, we used the PhnRec tool to extract phone hypotheses as described earlier. For each of the 20 languages to be identified, we trained four 3-gram phoneme language models with Witten-Bell smoothing based on different phoneme hypotheses, using the SRILM toolkit [24].

This resulted in a total of 80 perplexity scores (2) as features.

$$10^{-\frac{logprob}{\#(words)}} \tag{2}$$

We then employed the Weka [21] SMO classifier to generate predictions 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.

4.2.3. *Phone Variation System*

This system used MFCC vectors (“raw”), their deltas (delta) and their double deltas (double-delta) as derived by OpenSMILE [20]. For each set of phoneme hypotheses as derived by the PhnRec tool as described above, the raw, delta and double-delta for MFCC vectors were determined and the following calculations were extracted as a total of 195 features:

- Min: Minimum for each MFCC[0..12] vectors
- Max: Maximum for each MFCC[0..12] vectors
- Mean: Mean for each MFCC[0..12] vectors
- Variance: Variance for each MFCC[0..12] vectors
- Stdev: Standard deviation for each MFCC[0..12] vectors

This system draws inspiration from discriminative phonotactics which combines features drawn from phone hypotheses with low-level MFCC features. However, our Phone Variation combines these features in different ways. Specifically, the output of this system was designed for faster calculation with a possible trade-off of being less accurate. We used the Weka SMO classifier to generate language hypotheses from these features. All parameters were kept at their default values.

4.3. i-vector System

Motivated by Joint Factor Analysis [25, 26], i-vector modeling was originally proposed in [16], showing the success of implementing an i-vector framework for language recognition. The i-vector exploits the concept of “total variability. It improves upon factor analysis by estimating a single low-dimensional subspace (i-vector space) where all variability is modeled, leading to improved accuracy and reduced computational complexity. Previous work exploited this framework to model speaker-specific variability; we develop a technique based on i-vectors to model the variability. This system used the Alize toolkit [27] to generate predictions for the held-out development set built from the training portion of the corpus. For each of the 20 target languages, a single speaker model was built from all conversations in training portion. All other parameters for Alize were kept at their default values.

5. Results

For each system, we report three metrics: the accuracy of language prediction on our held-out development set, the C_{llravg} of the language prediction on the held-out development set and the resource consumption (speed and memory usage) of the systems while building models and making predictions. We were also able to report the performance of some models on the LRE15 test set using the C_{llravg} metric. The C_{llravg} metric is the LRE15 shared-task metric [4]. As such, this is the only evaluation measure available for the test partition of the data. The majority of our systems output a probability distribution over the 20 language families. We then select the highest probability across the set as the language prediction. From these predictions, we report accuracy on a held-out development set. Accuracy represents our first metric. In addition to accuracy, we calculate the C_{llravg} on the development set for each system. This metric represents a mixture of the accuracy of the prediction within a language family as well as the confidence in that prediction. More accurately, it is defined in [28] and used in [29] as the expected cost of the prediction. The C_{llravg} metric is calculated as shown in (3), where N_L is the number of languages in the language family cluster.

$$\begin{aligned} C_{llravg} = \frac{1}{N_L} * \sum_{L_T} [P_{Target} * C_{llr}^{tar}(L_T) \\ + \sum_{L_N} P_{Non-Target} * C_{llr}^{non}(L_T, L_N)] \end{aligned} \tag{3}$$

Most of our systems use the Weka tool, which produces predictions for each language in forms of probabilities. For C_{llravg}, these probabilities needed to be converted to log likelihood ratios.

5.1. Re-scaling

In order to compare our systems' performance to other LRE15 participants [30], we chose to incorporate the C_{llravg} metric. Although, in some of our experiments we faced some issues regarding our predictions and the nature of the C_{llravg} metric. For example, if the system did not make any predictions above the 0.5 threshold, the metric treats the system as if no prediction is being made. Because the evaluation method only looks at pairs of languages within the same family, and because the evaluation uses a hard threshold on the likelihood ratios, we decided to re-scale the Weka predictions first. To re-scale, each

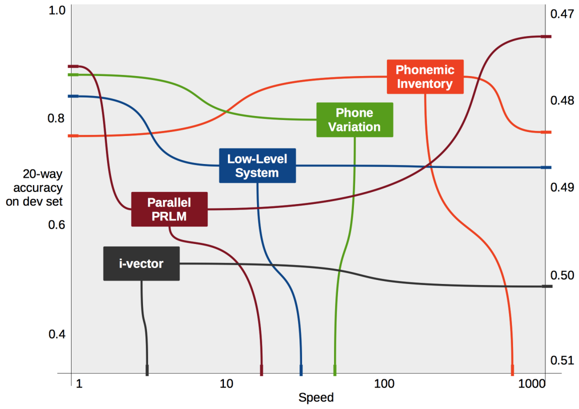


Figure 2: *Experimental Results.*

prediction probability was normalized such that within a language family, each language probability summed to one. This conditioned each language prediction probability to the cluster in which that language belongs. These probabilities were then converted to log likelihood ratios, with an equal prior for each language class. Re-scaling was performed for the Parallel PRLM and Phonemic Inventory systems prior to testing on the test set. As can be seen in Table 2, the Parallel PRLM and the Phonemic Inventory systems were performing at random on the development set. We hoped to mitigate this issue through re-scaling. We note that re-scaling seemed to mitigate the issue we faced with the C_{llravg} metric, improving performance for the Parallel PRLM and the Phonemic Inventory systems on the test set.

Table 2: *Performance of Language Prediction Systems. The Phone Variation and i-vector systems were not included in our submissions for the LRE15 shared-task, therefore the performance for those systems are listed as N/A.*

System	Accuracy (dev)	C_{llravg} (dev)	C_{llravg} (test)
Low-Level System	0.8312	0.1637	0.4863
Phonemic Inventory	0.7572	0.5000	0.4819
PRLM	0.7967	0.5000	0.4735
Phone Variation	0.8937	0.5000	N/A
i-vector	–	0.5109	N/A

5.2. Resource Consumption

The last metric considered is resource consumption. Each of our systems was executed on a different set of 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), each with 12GB RAM. All systems ran Ubuntu 15.04. In addition to speed, we report the peak memory usage of each system, all of which occurred during the model building phase. We report the User Time for each overall system as well as peak memory usage and for each system on Table 3.

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

We define the speed of language recognition as the sum of the duration of the speech for each conversation in the test portion of the LRE15 corpus divided by the total user CPU time. This is shown in Figure 2. We also provide the total (wall clock) time in Table 3, alongside peak memory usage, for reference.

Table 3: *Resource Consumption.*

System	User Time (seconds)	Peak Memory Usage (MB)
Low-Level System	127,438	1703.68
Phonemic Inventory	14,748	1213.30
PRLM	266,370	2226.92
Phone Variation	82,841	1495.53
i-vector	767,712	5986.39

5.3. Discussion

While all system performed well, as can be seen from the results in Figure 2, the overall trend for all systems appears to trade speed for accuracy in performing language identification. The fastest of the systems, Phonemic Inventory, is also the system with the lowest accuracy, albeit with surprisingly high C_{llravg} performance. Likewise, the slowest and most memory-intensive system, parallel PRLM, is also the one with the highest accuracy and lowest cost as measured by C_{llravg}.

We observe that our implementations of the phonotactic systems exhibit relatively high costs (all with C_{llravg} = 0.5000) but also with relatively high accuracy. However, the most interesting of the results could be the stellar performance of the low-level acoustic-prosodic system and the mediocre performance of the i-vector system. This is especially noteworthy given the popularity of systems based on i-vectors in the last few years.

6. Conclusion and Future Work

While the amount of memory required to perform language recognition is relatively consistent across the systems we implemented, one of the more striking observations from this work may be that the speed of language recognition varies widely enough to be noticed even with the caveats we mentioned. The Phonemic Inventory system is faster than all others by an order of magnitude, for which the trade-off is a small sacrifice in accuracy. Other phonotactic systems and our i-vector system all exhibit relatively similar performance with respect to speed and memory consumption. In short, where speed is a factor in language identification, one based on Phonemic Inventory features would be very useful. We also believe that phonotactic systems can be robust with respect to channel differences. Interestingly, we find that although i-vector approaches are currently included in many state of the art approaches to language identification, when used as a stand alone system do not outperform the phone based systems. In our future work, we aim to build systems which use phonotactic features as input and have architectures, perhaps inspired by the i-vector framework, which can exploit these inputs to the fullest. We also plan to evaluate these systems on dialect recognition tasks.

7. References

[1] M. S. Dryer and M. Haspelmath, Eds., *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. [Online]. Available: http://wals.info/

[2] A. F. Martin and C. S. Greenberg, "The 2009 nist language recognition evaluation." in *Odyssey*, 2010, p. 30.

[3] C. S. Greenberg, A. F. Martin, and M. A. Przybocki, "The 2011 nist language recognition evaluation," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[4] A. F. Martin, C. S. Greenberg, J. M. Howard, G. R. Doddington, and J. J. Godfrey, "Nist language recognition evaluation past and future," in *Proceedings of Odyssey: The speaker and language recognition workshop*, 2014, pp. 145–151.

[5] M. A. Zissman *et al.*, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[6] Y. Yan and E. Barnard, "An approach to automatic language identification based on language-dependent phone recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 5. IEEE, 1995, pp. 3511–3514.

[7] F. Biadsy, J. B. Hirschberg, and D. P. Ellis, "Dialect and accent recognition using phonetic-segmentation supervectors," 2011.

[8] F. Biadsy, J. Hirschberg, and N. Habash, "Spoken arabic dialect identification using phonotactic modeling," in *Proceedings of the eacl 2009 workshop on computational approaches to semitic languages*. Association for Computational Linguistics, 2009, pp. 53–61.

[9] M. A. Zissman, T. P. Gleason, D. M. Rekart, and B. L. Losiewicz, "Automatic dialect identification of extemporaneous conversational, latin american spanish speech," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 777–780.

[10] B. Ma, D. Zhu, and R. Tong, "Chinese dialect identification using tone features based on pitch flux," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[11] O. Koller, A. Abad, and I. Trancoso, "Exploiting variety-dependent phones in portuguese variety identification," 2010.

[12] N. F. Chen, W. Shen, and J. P. Campbell, "A linguistically-informative approach to dialect recognition using dialect-discriminating context-dependent phonetic models," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5014–5017.

[13] A. J. Dinkin, "Mary, darling, make me merry; say you'll marry me: tense-lax neutralization in the linguistic atlas of new england," *University of Pennsylvania Working Papers in Linguistics*, vol. 11, no. 2, p. 5, 2005.

[14] F. Biadsy, H. Soltau, L. Mangu, J. Navratil, and J. B. Hirschberg, "Discriminative phonotactics for dialect recognition using context-dependent phone classifiers," 2010.

[15] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[16] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction." in *INTERSPEECH*. Citeseer, 2011, pp. 857–860.

[17] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2015–2028, 2013.

[18] M. Najafian, A. DeMarco, S. J. Cox, and M. J. Russell, "Unsupervised model selection for recognition of regional accented speech." in *INTERSPEECH*, 2014, pp. 2967–2971.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *INTERSPEECH*, 2002.

[25] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[26] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, 2007.

[27] A. Larcher, J.-F. Bonastre, B. G. Fauve, K.-A. Lee, C. Lévy, H. Li, J. S. Mason, and J.-Y. Parfait, "Alize 3.0-open source toolkit for state-of-the-art speaker recognition." in *INTERSPEECH*, 2013, pp. 2768–2772.

[28] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2, pp. 230–275, 2006.

[29] N. Brummer and D. A. Van Leeuwen, "On calibration of language recognition scores," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–8.

[30] A. F. Martin, C. S. Greenberg, J. M. Howard, D. Bansé, G. R. Doddington, J. Hernández-Cordero, and L. P. Mason, "Nist language recognition evaluationplans for 2015," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.