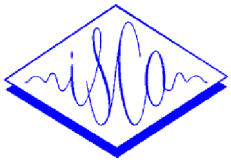
ISCA Archive
http://www.isca-speech.org/archive



Odyssey 2008: The Speaker and
Language Recognition Workshop
Stellenbosch, South Africa
January 21-24, 2008

Building language detectors using small amounts of training data

David A. van Leeuwen

TNO Human Factors
Soesterberg, the Netherlands
david.vanleeuwen@tno.nl

Niko Brümmer

Spescom Datavoice
Stellenbosch, South Africa.
nbrummer@za.spescom.com

Abstract

In this paper we present language detectors built using relatively small amounts of training data. This is carried out using the modelling power of a Linear Discriminant Analysis back-end for the languages which have a small amount of training data. We present experiments on NIST 2005 Language Recognition Evaluation data, where we use a jackknifing technique to remove well-trained language knowledge from the LDA back-end, using only sparse trials for training the LDA. We investigate three systems, which show different levels of loss of language detection capability. We validate the technique on an independent collection of 21 languages, where we show that with less than one hour training we obtain an error rate for 'new' languages that is only slightly over twice the error rate for languages for which the full 60 hours of CallFriend data is available.

1. Introduction

Spoken language recognition is an area of research that has attracted increasing interest in recent years. Not only does the series of 'Odyssey' workshops carry language recognition in the title since 2004, but also the National Institute of Standards and Technology (NIST) language recognition evaluations (LREs) have increased in frequency and numbers of participants. As in other areas of speech research, the NIST series of evaluations play an important role in the direction of research in stimulating researchers to develop better algorithms that deal with the challenges specific to language recognition.

The initial LRE-1996 still has an important impact on the research direction since it was accompanied by a major data collection effort that resulted in the CallFriend speech database [8], which is distributed by the Linguistic Data Consortium (LDC). This database consists of telephone recordings of conversations in 12 languages (and three dialects within those).[1] For each of these languages/dialects there is about 60 hours of training data[2] of about 100 speakers. Since current spoken language detectors benefit from large amounts of speech, CallFriend still acts as a basis for most systems. General language recognition algorithms are not hand-tailored to any particular language, instead they are data-driven and should work with any collection of speech that is comparable in size to CallFriend.

The big advantage of language recognition data collections is that they do not require costly phonetic or orthographic transcriptions—accurate language labels are sufficient. Adding another language to the CallFriend training collection may at first sight seem easy: just record a lot of speech from many people speaking the language. However, if the data collection characteristics of this new language are different from CallFriend, a system may learn unwanted differences in these collection characteristics (channel or session effects), and not the language characteristics. This is yet another reason why LDC's CallFriend is such an important collection.

In the series of NIST LREs, therefore, the collection of languages under evaluation have always been centered around the 12 CallFriend languages. It has been difficult to deal with new languages for which there is little training data available. In 2003 Russian was introduced as a 'surprise' non-target language in the open set condition—hence no specific training for this language was expected. In 2005 a different situation arose, where it was announced that as part of the English trials, Indian accented English would be included. There was a limited set of 40 speech segments (just 20 minutes total duration) available for training. As turned out in the evaluation, these Indian accented English trials were responsible for most of the errors in well performing systems, even though the American English models for such systems were very good due to the vast amounts of telephone speech data available.

In this paper we try to address the problem of building language detectors for 'new' languages, for which there is only a small amount of training data available. We will do this in the NIST LRE framework, typically

[1] The so-called 'CallFriend languages' comprise the 10 languages of the earlier recorded Oregon Graduate Institute (OGI) corpus [9].

[2] The actual amount of speech is about half of this, because these are recordings of two speakers in a conversation.

using CallFriend languages and NIST LRE-2005 as evaluation database. Of the many approaches of language recognition, such as phonotactics of phone lattices [6], or binary decision trees [10] we use the approach of direct modelling of the acoustic space using Gaussian Mixture Models [14], concentrating on small amounts of training data. The paper is organized as follows. In Section 2 our baseline system is described and characterized. Then, in Section 3 general approach is explained, followed by experimental results which are discussed in the final section.

2. Baseline system description

Our basic system design consists of a set of Single Language Detectors (SLDs) trained for the twelve CallFriend languages, followed by a Gaussian back-end trained on the languages under evaluation [13]. This is more-or-less the design of our system submission to NIST LRE-2005 [15], where we fused 4 subsystems in a Linear Discriminant Analysis (LDA) back-end. For this research, we concentrate on the best performing of these subsystems, as improved with recent insights in modelling learnt from speaker recognition technology.

2.1. Data processing and feature extraction

We use the same feature processing as described in [15]. Five PLP coefficients plus log-energy are derived every 16 ms using 32 ms analysis windows. From these, shifted-delta-cepstra (SDC) feature vectors are constructed by stacking $d = 1$ frame span derivative vectors over $p = 2$ frames, $k = 4$ times. Frames were selected based on energy, where for at least one of the original frames involved in constructing the SDC the energy should exceed a level of 30 dB under the maximum frame energy measured in the speech segment under analysis. Note that our $(6, 1, 2, 4)$ SDC features span about the same duration as the celebrated MIT SDC settings $(7, 1, 3, 7)$ [13], where a frame shift of 10 ms is used, but have considerably fewer parameters (24 versus 49). In earlier experiments we did not observe a degradation in performance with this reduced parameterization [15].

2.2. Data resources

We discriminate three types of speech data: (i) data used for training SLDs, (ii) data used for training the LDA back-end, and (iii) evaluation data to determine the performance. For training the basic generative and discriminative SLDs we exclusively use all available data from the CallFriend database. For training the back-end, we use the 30 s duration trials from the NIST databases lid96d1, lid96e1, lid03e1 and lid05d1[3]. Note that lid96d1 and lid96e1 have some overlap with the SLD data, but we have found that including these trials in the LDA training helps performance on independent evaluation data. For evaluation, we use lid05e1, which we also refer to as the LRE-2005 data.

2.3. Gaussian Mixture Model Single Language Detector

The basis of every SLD in this paper is a language-independent Gaussian mixture model (LI-GMM), trained on all 12 CallFriend languages. We can train one LI-GMM on all data available for each language, or alternatively we can train several different LI-GMMs, each on a subset of the data, conditioned on automatically determined sex and/or channel labels c. A given LI-GMM $\mathcal{M}^c$ (similar to the Universal Background Model [12] in speaker recognition) can then be adapted to a specific training language using a Maximum A-Posteriori (MAP) criterion [5, 12] to obtain a language-dependent GMM. Each MAP-adapted GMM $\mathcal{M}^c_L$, for each of the training languages L, then acts as a 'Single Language Detector,' where the log-likelihood-ratio of the language-dependent against the language-independent GMM acts as the *score* for a given test speech segment.

2.4. GMS Single Language Detector

In 2006, an important leap in performance in speaker recognition performance was reported by MIT [2], which was based on using Support Vector Machines (SVMs) to discriminate between speakers by using the *means* of segment-dependent GMMs as feature vectors. For speaker recognition, this performance boost turned out quite important, not only giving lower detection errors for the '1-side' train/test conditions, but also allowing for Nuisance Attribute Projection [3], a technique that very effectively [1] can compensate for channel and session variability.

It is almost obvious[4] to build a language detector in the same way. Rather than doing a MAP adaptation of language-independent model $\mathcal{M}^c$ using all available data for language L, we MAP-adapt from the LI-GMM to obtain a segment-dependent GMM for every test and for every training segment T_L separately. The means of the GMM $\mathcal{M}^c_{T_L}$ obtained this way now represents the test or training utterance. Using these mean-supervectors of all training segments, we train a language-dependent (linear-kernel) SVM for every language. Each SVM discriminates one language from the other 11 languages. As in [2] we normalize each mean vector component μ^j_i by $\sqrt{w_i}/\sigma^j_i$, where w_i and $(\sigma^j_i)^2$ are the weight and feature component j of the (diagonal) covariance of Gaussian component i of the LI-GMM. Since we use a linear-kernel SVM, the model can be represented a single supervector $\mathbf{S}_L$ and a scalar constant b_L.

[3] We use the NIST nomenclature lid$yys1$, where yy is the LRE year and $s \in \{\mathrm{d}, \mathrm{e}\}$ denotes development and evaluation data.

[4] as we learned, after submission of this manuscript, was presented in Ref. [4]

In scoring a test segment t, we first form the mean-supervector as described above and then score it against each SVM by taking the inner product of this test-segment supervector and the model supervector: $s_L(t) = \mathbf{t} \cdot \mathbf{S}_L + b_L$. We use the abbreviation GMS (meaning GMM Means SVM) for this approach.

Because we use 'fast MAP adaptation' [1], only scoring the top-5 mixtures for every frame in the 'expectation' step of the MAP adaptation, the scoring of a test segment is very fast. By proper alignment of language model supervectors and offsets b, a full set of model scores can be obtained by a single matrix-vector multiplication. Obtaining scores for all trials in an LRE then becomes a single matrix-matrix multiplication.

2.5. LDA Back-end

In order to obtain a language decision of a test segment t for a target language L, we utilize the modelling power of an LDA (a.k.a. Gaussian) back-end [13]. A Linear Discriminant Analysis (LDA) classifier is trained using a set of supervised training trials $\{B_i\}$ which consist of speech segments x_i labelled with a language L_i. A vector of scores $\mathbf{s}$ is formed by stacking the outputs of a set of SLDs for the speech segment x_i. The classifier finds the projection of this input space that maximizes the ratio of between-class variance to the within-class variance. The output coordinates (representing the language classes) are transformed such that they can be interpreted as 'posterior' probabilities (i.e., summing to unity) taking into account a language prior.

For the LRE-2005 data we work with several SLDs for each of the twelve CallFriend languages, resulting in anything from 24 up to 71 scores per LDA trial B_i, and reduce the results to the seven LRE-2005 languages by using a prior $p_L = 1/N_L$ for each of these languages and 0 for other languages, where $N_L = 7$ is the number of languages in the test. Then, in order to make a decision of target language L, we set the threshold θ for its posterior probability $p(L|x_t)$ to $\theta = 1/N_L$.

2.6. Performance measure and baseline performances

We use the NIST language detection cost C_{DET} [11] as our evaluation measure. This measure is, in its current definition, insensitive to the relative proportions of trials for the languages in the evaluations database. We prefer this measure over commonly reported Equal Error Rates (EER), because we believe the pooling of correlated score distributions, that is necessary for determining the EER, is wrong [16].

In this paper we consider only the 30 s test-segment subset of the LRE-2005 evaluation, and we concentrate on the closed-set detection task, using all trials. Thus, we know each trial is in one of seven languages, with a prior probability of $\frac{1}{2}$ for the target language. The interpretation of the LDA back-end posteriors and threshold setting described above should be optimal for this task.

Table 1: Baseline performance statistics for the three systems described here.

Target L system	C_{DET}(%)		
	Chan-GMM	GMS	Chan-GMS
English	10.0	11.0	9.57
Hindi	13.0	6.61	11.49
Japanese	7.22	4.51	4.34
Korean	9.61	5.85	5.90
Mandarin	4.93	5.67	5.20
Spanish	7.77	6.67	7.25
Tamil	10.6	7.77	7.92
Mean	9.01	6.88	7.38

We use three systems:

Chan-GMM A 'channel-conditioned' GMM system. This system was reported on in [15], and functioned as our best performing individual subsystem at LRE-2005. It was also used in a study of open-set language recognition experiments [16]. It consists of 6 'gender-channel' conditioned language-independent GMMs resulting in 71 generative SLDs.[5]

GMS A basic GMM means supervector SVM system. Here we condition the language independent GMM only on speaker sex[6], resulting in 24 discriminative SLDs.

Chan-GMS A combination of the above. Here, we use the same sex-dependent language independent GMMs. In MAP adapting the means we condition the available CallFriend training data on the same gender-channel conditions as for the Chan-GMM system. These means are used to train the language-detecting SVMs. The result is 71 SLDs, which are discriminative within same-channel groups of 12 (or 11) languages.

The performance figures for our three language recognition systems are given in table 1.

3. Experiments

For each of the languages in LRE-2005 there was an abundance of training data available, except as mentioned above that the Indian English accented trials were not well-represented in the training data. In this section we will present results on attempts to build detectors for a language with limited training data.

[5] One language-channel combination, female Japanese cellular phone, did not get populated by the CallFriend data using our channel-classifier [15], and hence we have only 71 SLDs.

[6] Where the sex of speakers of the CallFriend training data was not known, we used a gender discriminator to determine this automatically.

3.1. Experimental approach

The goal of the experiment is to investigate how well a language can be detected when no SLD has been trained for it, relying only on (i) the language information carried by SLDs trained on *other* languages and (ii) on the modelling power of the LDA back-end.

Recall that our SLD's are trained on the many and long speech segments in CallFriend, whereas the LDA back-end is trained on the fewer and shorter (30s) test segments of pre-2005 LREs. Languages which are recognized just in the LDA back-end are therefore effectively trained with sparse training data.

In order to test the effect of LDA modelling of a new language, we use a left-out jackknifing rotation over the LRE-2005 evaluation [16]. The procedure may be best explained in the following pseudo-code:

- For each target language L_i
  - select the subset of segments from the evaluation with language L_i,
  - Remove SLD scores (columns in the LDA matrix) corresponding to L_i from all LDA training data and the L_i-subset of evaluation data,
  - Build the LDA, compute target and non-target scores for this L_i-subset of evaluation data, and make decisions for these language-segment trials.
- Pool decisions over all languages and all jackknives and then calculate C_{DET} with the appropriate language-conditional weighting as specified in the LRE-plan.

The decisions made in this process are made such, that there is no SLD conditioned to the language used in the test trial, whether it be a target or a non-target trial. All knowledge to recognize the language of the test trial is encoded in the back-end. Therefore, this cross validation simulates the situation where the recognition system is tested with a new language, for which only limited amount of training is available. We call this type of modelling of the new language Gaussian Score-Space Modelling (GSSM).

The *amount* of training data for each test language is a lot less than what we are used to with CallFriend, which is of the order of 30–60 hours of speech per language. The total duration of training trials for the back-end varies from 225 for Tamil to 917 for English, which corresponds to about 1.9–7.6 hours. For the Indian English accent only 20 minutes worth of trials is available.

Table 2: Results of the GSSM experiment on LRE-2005, leaving the test segment language out of the SLD's scores.

Target L system	C_{DET}(%)		
	Chan-GMM	GMS	Chan-GMS
English	9.63	11.3	9.56
Hindi	12.7	11.0	15.0
Japanese	9.88	9.34	5.96
Korean	10.3	12.3	8.52
Mandarin	5.80	9.49	5.53
Spanish	9.19	11.6	7.84
Tamil	9.36	10.9	7.29
Mean	9.54	10.9	8.53

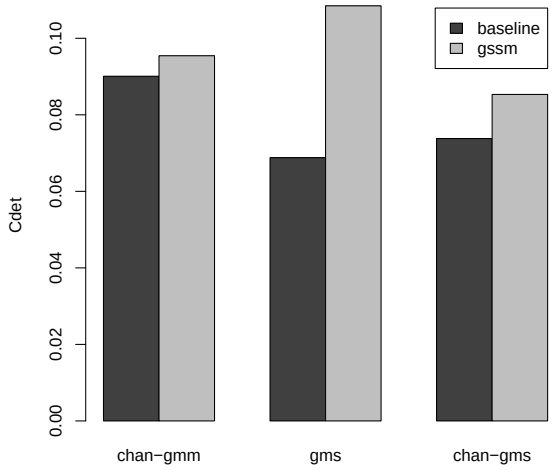


Figure 1: Comparison of full training (baseline) and GSSM training (LDA back-end only), for the three systems. Shown is the average C_{DET}.

3.2. Results

In Table 2 the results for the GSSM experiment are given, for the three language recognition systems described in Section 2.6. A comparison between the baseline systems and the short language training of the average C_{DET} is shown in a bar-graph in Figure 1.

Our generative channel-GMM system with 71 SLDs appears most robust to not having explicitly modelled a test language in the set of SLDs. The value of C_{DET} increases only 2.4 % by missing this information. On the other hand, it has the worst baseline performance. The more recently developed GMS system has a much better baseline performance, but takes a large hit from missing the test language information in the SLDs. The average C_{DET} increases by over 50 %, to become even higher than the GMM system. A potential reason for this could

be the fact that there are much less SLDs for the LDA back-end, which has to do all the modelling of the test language, for the GMS system (24) than for the channel-GMM system (71).

When we combine both technologies of channel conditioning and GMM means in SVM approach (Chan-GMS), we can observe that the loss of information by not explicitly modelling the test language in the SLD has a smaller effect, increasing C_{DET} by only 10 %. Despite the fact that the baseline Chan-GMS system performs worse than the GMS system, its performance for the sparse training condition is the best of the three systems studied here. This suggests that some of the potential of the back-end to train efficiently with small amounts of data lies in the high dimensionality of the score vectors.

In order to investigate the influence of the number of SLD on the ability of the LDA to model a language on its own, we reduced the number of SLDs in one sample system, Chan-GMS. We varied the number of SLDs per language, r, from 1–6. We did this by randomly selecting r SLDs for each language from the available 6 SLDs for that language. Because there may be many ways to choose r SLDs from the available 5 or 6 SLDs, we averaged over 10 selections for each r. The results are shown in Figure 2 for both the baseline and sparse training condition. For this system, the increase in C_{DET} due to no direct modelling of the test language appears constant w.r.t. the baseline condition. Note that the drop in C_{DET} is not only attributable to the mere increase of SLDs, since each SLD is conditioned on a different part of the Call-Friend database. Hence the total available training time increases with more SLDs.

Another variation we can introduce in the available training time for the LDA is the number of trials. In the current set-up the number of trials is actually quite different for every language, due to the different focuses of the NIST evaluations in the past, and the availability of speakers for the data collection. We will not go into the detail of the individual language detection performance here, but just look at the overall effect if we reduce the number of trials for the LDA. We selected a random sample of fraction f from the available trials for training the LDA back-end, and determined C_{DET}. We chose f ranging from 2^{-5} to 1, doubling at each step. In Figure 3 the effect of the LDA training size on C_{DET} for the GMS and Chan-GMS systems is shown. We took the mean C_{DET} over 10 runs, to average out the effect of sampling. An interaction can be observed between the sensitivity to missing SLDs of the test language (sparse training) and the system type. The many SLDs make the LDA back-end more robust against missing the test segment's SLD, but these need more training trials.

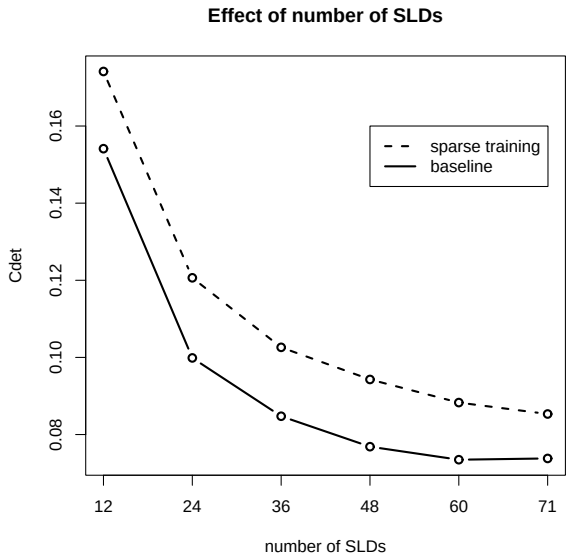


Figure 2: Effect of the number of SLDs available to the LDA back-end, for the Chan-GMS system, for the baseline and GSSM condition.

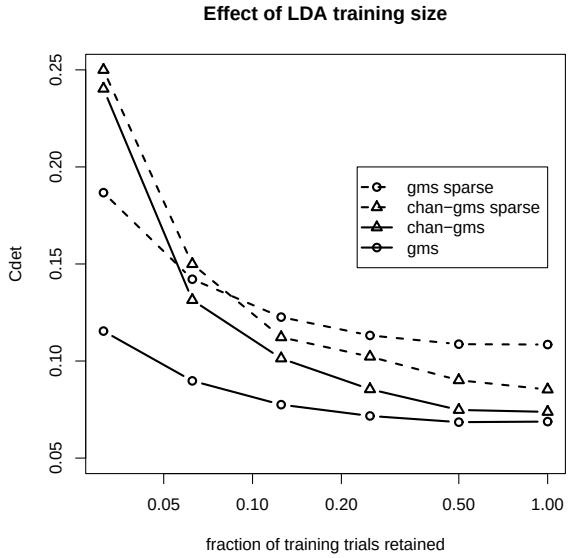


Figure 3: Effect of LDA training size on C_{DET} for GMS and Chan-GMS system, in baseline and sparse training condition. Note that the x-axis is logarithmic in scale.

3.3. Increasing the number of languages

So far we have investigated the modelling power of the LDA back-end by looking at a subset of the CallFriend languages. In this section we will investigate the performance on more languages. To this end, we use the CSLU22 speech database [7], which contains telephone recordings of over 2000 speakers in 21 languages, which forms almost a superset of the CallFriend languages, only (Canadian) French is missing. For this experiment we used the 'story' sentences of the database.

We employed the same SLDs trained on full CallFriend, and produce scores for all 2037 'story' segments in CSLU22. The segments are used both for training the LDA back-end and in testing, in a 10-fold cross-validation scheme. For decisions we used priors and decision thresholds based on $N_L = 21$.

In Table 3 the results of this test are shown, for the two GMS systems. The LDA training duration for this GSSM is an average 37.13 s nominal speech per segment, which amounts to approximately 54 min of speech per language. The average C_{DET} performance over the languages evaluated in LRE-2005 is lower than what is observed with LRE-2005 data (cf. Table 1), which suggest that this test data is slightly 'easier.' Also notice how the Chan-GMS system outperformes the GMS system for the CSLU22 data, even when conditioned on LRE-2005 languages alone, which is contrary to the results shown in Table 1.

Table 3: Performance of GMS and Chan-GMS systems on the CSLU22 database, for the 21 languages. Separate average C_{DET} values are show for the 7 'LRE-2005' languages, 11 'CallFriend' languages, and 10 'new' languages.

Language	C_{DET}	
System	GMS	Chan-GMS
Arabic	10.2	7.42
Bportuguese	11.3	8.64
Cantonese	6.54	5.93
Czech	19.1	12.8
English	0.56	0.30
Farsi	3.76	2.64
German	7.06	4.18
Hindi	9.90	7.22
Hungarian	17.0	14.1
Indonesian	14.2	9.80
Italian	15.7	10.3
Japanese	5.94	5.75
Korean	4.08	4.78
Mandarin	3.60	4.70
Polish	17.5	11.4
Russian	17.4	11.4
Spanish	7.80	4.87
Swahili	21.6	13.8
Swedish	15.5	11.3
Tamil	4.51	3.95
Vietnamese	10.9	5.00
Mean LRE-05	5.20	4.51
Mean CallFriend	6.21	4.62
Mean new	15.1	10.6
Overall mean	10.7	7.63

4. Discussion

We have shown that an LDA back-end can be quite effective for modeling a new language. This new language does not have to be represented in the Single Language Detector scores that feed the LDA back-end. With as little as 2 hours of training data for some languages, we have shown, in a jackknifing scheme, that the performance of a system with only LDA back-ends training is not very much worse than that of a fully trained system with 60 or more hours per language. We have observed that the weaker, generative, Chan-GMM system is less susceptible to missing language training data in the SLDs than the more discriminatively operating GMS systems. A hybrid system, carrying both the SLD diversity of the Chan-GMM system and the discriminability of the GMS system gains robustness of modeling a new language with little data, while only moderately losing performance in the full training condition.

In scanning the number of SLDs necessary for good performance of the Chan-GMS system (cf. Figure 2), there appears to be little difference between the baseline and GSSM condition. The latter system appears to gain a constant penalty in C_{DET} from missing the language information from the SLDs. Comparing the low score dimension GMS system with the higher score dimension Chan-GMS (cf. Figure 3), we see that it is more robust to small amounts of LDA training trials for the baseline condition. This is probably because the LDA has less dimensions and the covariance matrix is more stable. Also, as the GMS system is more discriminative the basic SLD will carry more of the modeling power and the LDA has to 'correct' fewer errors.

We have also shown that outside the jackknifing paradigm the LDA back-end can model new languages, as for the data from the CSLU22 database. Here, the penalty of not having discriminatively trained SLDs for the new languages is bigger, about a factor two for the Chan-GMS system. A reason for this might be that there is too little data even for the LDA to train the new language properly. In Figure 3 we can see there still is a benefit in the last doubling of amount of LDA training data for the sparse training condition—the data at fraction 0.50 is representative for the approximate one hour speech available for the new CSLU languages. However, the gap to be bridged is fairly large, and one might

question whether this can be reached using LDA training alone. If more training data becomes available for the LDA, one might consider using this for training an SLD instead.

Our experimental protocol, as described in Section 3.1, is somewhat artificial in the sense that *all* test data is presented as a sparsely modeled language. This implies the LDA is not tested with 'normal' languages for which SLDs are trained. If the LDA would somehow have a tendency to produce high scores for languages without SLD scores, this protocol would show artificially low C_{DET} values. Our system, however, operates with fixed priors and threshold values for the LDA posterior (each $1/N_L$), and we typically observe a miscalibration towards higher miss rate, which is the opposite direction. We have repeated the experiments described in this paper, adding all remaining 'normal' segments from the evaluation for each jackknife slice. The result is that the effects of sparse modeling of languages are the same as described above in Figures 1–3, but less pronounced.

We have not looked into other language modelling techniques that may be well suited to deal with sparse training. A good candidate might be the binary decision tree phonotactic modeling [10], as binary decision trees can be efficient in modeling sparse and inhomogeneous data. The jackknifing approach used here can serve as a methodology for testing other sparse training approaches. We aim at utilizing some of these techniques for the upcoming NIST LRE 2007, where seven new languages are introduced with moderate amounts of training data, typically 6.5 hours per language.

5. Acknowledgements

The authors wish to thank Doug Reynolds from MIT Lincoln Labs for his critical reading of the manuscript. This work was supported in part by the European Union 6th FWP project AMIDA, 033812.

6. References

[1] Niko Brümmer, Lukáš Burget, Jan Černocký, Ondřey Glembek, František Grezl, Martin Karafiát, Pavel Matějka, David A. van Leeuwen, Petr Schwarz, and Albert Strassheim. Fusion of heterogeneous speaker recognition systems in the STBU submission for the nist speaker recognition evaluation 2006. *IEEE Transactions on Speech, Audio and Language Processing*, 15(7):2072–2084, 2007.

[2] William Campbell, Douglas Sturim, and Douglas Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, 2006.

[3] William Campbell, Douglas Sturim, Douglas Reynolds, and Alex Solomonoff. SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In *Proc. ICASSP*, pages 97–100, Toulouse, 2006. IEEE.

[4] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair. Acoustic language identification using fast discriminative training. In *Proc. Interspeech*, pages 346–349, Antwerp, 2007. ISCA.

[5] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori esitimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. Speech Audio Processing*, 2:291–298, 1994.

[6] J. L. Gauvain, A. Messaoudi, and H. Schwenk. Language recognition using phone lattices. In *ICSLP*, 2004.

[7] T. Lander. CSLU: 22 languages corpus. Linguistic Data Consortium, 2005.

[8] Linguistic Data Consortium. Callfriend corpus. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC96S46`, 1996.

[9] Y. K. Muthusamy, R. A. Cole, and B. T. Oshika. The OGI multi-language telephone speech corpus. In *Proc. International Conference on Spoken Language Processing*, Banff, Canada, 1992.

[10] Jiří Navrátil. Recent advances in phonotactic language recognition using binary-decision trees. In *Internat. Conference on Spoken Language Processing*, Pittsburgh, October 2006.

[11] The 2007 NIST language recognition evaluation plan. `http://www.nist.gov/speech/tests/lang/2007/`, 2007.

[12] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[13] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and R. A. Reynolds. Acoustic, phonetic, and discriminative approaches to automatic language identification. In *Proc. Eurospeech*, pages 1345–1349, 2003.

[14] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, Richard J. Greene, Douglas A. Reynolds, and J. R. Deller Jr. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *ICSLP*, 2002.

[15] David A. van Leeuwen and Niko Brümmer. Channel-dependent GMM and multi-class logistic regression models for language recognition. In

Proc. Odyssey 2006 Speaker and Language recognition workshop, 2006.

[16] David A. van Leeuwen and Khiet P. Truong. An open-set detection evaluation methodology applied to language and emotion recognition. In *Proc. Interspeech*, pages 338–341, Antwerp, August 2007. ISCA.