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

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

The ICSI 2007 Language Recognition System

Christian Müller, Joan-Isaac Biel

International Computer Science Institute,
Berkeley, CA
{cmueller, biel}@icsi.berkeley.edu

Abstract

In this paper, we describe the ICSI 2007 language recognition system. The system constitutes a variant of the classic PPRLM (parallel phone recognizer followed by language modeling) approach. We used a combination of frame-by-frame multilayer perceptron (MLP) phone classifiers for English, Arabic, and Mandarin and one open loop hidden Markov Model (HMM) phone recognizer (trained on English data). The maximum likelihood language modeling is substituted by support-vector-machines (SVMs) as a more powerful, discriminative classification method. Rank normalization is used as a normalization method superior to mean-variance normalization. Results are presented on the NIST 2005 language recognition evaluation (LRE05) set and a test set taken from the LRE07 training corpus. The average NIST cost of the system on the LRE05 set is 0.0886.

1. Introduction

The ICSI 2007 language recognition system constitutes a variant of the classic PPRLM (parallel phone recognition followed by language modeling) approach which is commonly used for this task. The basic idea of PPRLM is to model the phonotactic characteristics of the languages l_1,...,l_n in the test by means of a statistical language model. As frontends, either a single or multiple phone recognizers are used. In the former case the approach is called PRLM (without "parallel"). It is beneficial to use an *open loop* phone recognition, i.e. to not apply (language-specific) phonotactic constraints during the decoding. Using parallel phone recognition proved to be beneficial over using a single one [1].

The novel aspect of our approach is that we used a combination of multiple frame-by-frame multilayer perceptron (MLP) phone classifiers trained on English, Arabic, and Mandarin data and one hidden Markov model (HMM) open loop phone recognizer (trained on English data). Taking into account a recent enhancement of the PPRLM approach on the backend (see for example [2, 3]), we used the n-gram counts of phones as features to train support vector machines (SVM) instead of building an actual maximum-likehood language model. Besides a more sophisticated decision function, this variant is characterized by supporting a combination of different n-grams, say bigrams and trigrams. It also supports an immediate combination of multiple frontends on the feature level.

Discriminative training as a modification of the PPRLM approach is also described by [4]. The authors used three streams of phones produced by recognizers trained on Arabic, English, and Spanish data, respectively. The performance of SVMs (discriminative training) is compared with maximum likelihood language modelling. A 0.7 % absolute improvement (5.2 % EER with LM and 4.5 % EER with SMVs) on the LRE 03 evaluation set in the 30 seconds condition is reported. [5] postulate a short-term cepstral system using shifted delta cepstral (SDC) coefficients in conjunction with an SVM backend. Although the SVM system alone was inferior to the baseline Gaussian Mixture Model (GMM) on the 30 seconds NIST LRE 03 test (6.1 % versus 4.8 % EER), the two system could be effectively combined obtaining an EER of 3.2 %.

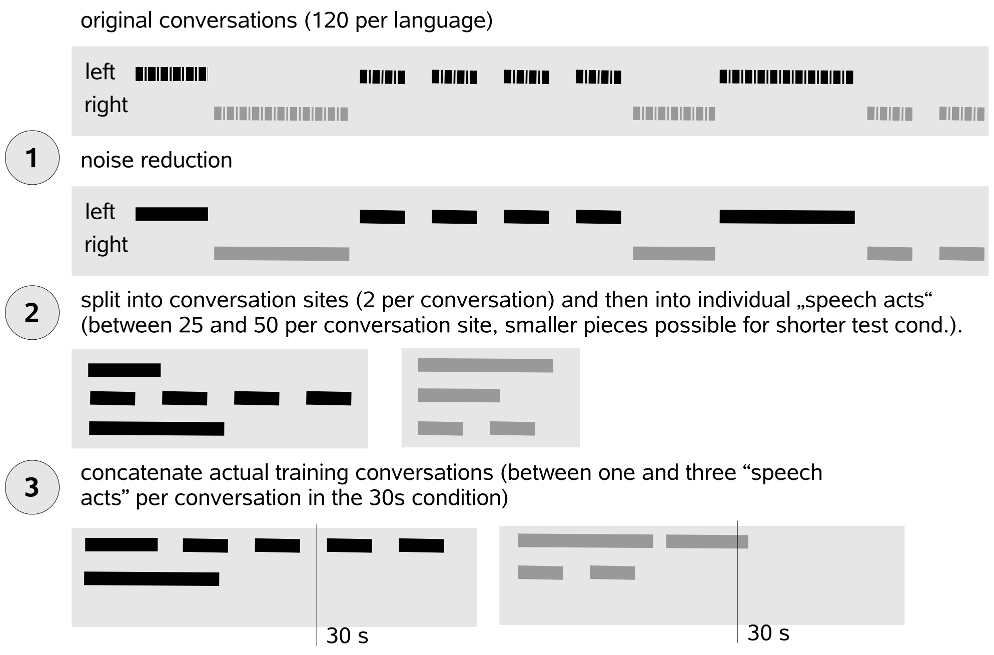
The remainder of this paper is organized as follows: section 2 describes the training data we used as well as the preprocessing method; section 3 provides a general overview over the system components; section 3.1 describes the various phone recognizer frontends; section 3.2 provides a description of the rank normalization procedure; the training of the support vector machines is detailed in section 3.3; section 3.4 provides the processing speed measures; in section 4, results on the 2005 NIST language recognition evaluation (LRE05) as well as our development test set (an excerpt of the training data) are presented.

2. Training data

We used the training data provided by NIST for this years language recognition evaluation[1]. It comprises fourteen languages: Arabic (ARA), Bengali (BEN), Chinese (CHIN), English (ENG), Farsi (FAR), German (GER), Hindustani (HIN), Japanese (JAP), Korean (KOR), Russian (RUS), Spanish (SPA), Tamil (TAM), Thai (THA), and Vietnamese (VIE).

The data preprocessing scheme is depicted in Figure 1. In the first step, Wiener filtering was applied to the original conversations to reduce the amount of noise. Hereafter, the files were split into individual conversation sides containing the left and the right channel, respectively. An intensity-based silence detector was applied to both parts, splitting them into individual dialog turns and removing the silence in-between. The detector's parameters include an intensity threshold which was set to 20 dB (everything below that threshold is considered as silence) as well as a threshold for the minimum length of silence. The latter was set to one second to avoid splitting at pauses that possess a linguistic purpose rather than marking the end of a turn.

[1] see http://www.nist.gov/speech/tests/lang/2007/



original conversations (120 per language)
left
right
1
noise reduction
left
right
2
split into conversation sites (2 per conversation) and then into individual „speech acts" (between 25 and 50 per conversation site, smaller pieces possible for shorter test cond.).
3
concatenate actual training conversations (between one and three "speech acts" per conversation in the 30s condition)
30 s
30 s

Figure 1: *Data preprocessing.*

The turns varied in length between one and 80 seconds. Accordingly, the original conversations typically consisted of between 25 and 50 turns. In the third preprocessing step, turns were concatenated with one or more successors until the desired length is reached.

In the experiments presented here, we only created 30 seconds long conversations. However, it is also possible to generate different data sets for the three and ten seconds conditions. With adapting the intensity and pause duration thresholds used in step two, smaller pieces can be obtained which facilitates the creation of shorter training conversations.

In the case of the test data, the original conversations were recovered after the silence removal. Table 1 summarizes the statistics for the training, background, and test data sets. The samples for the development test set were selected randomly from the data set of each language but not used for training.

data set	median length	total
Arabic	32.0 s	40.0 h
Bengali	29.8 s	2.9 h
Chinese	31.9 s	67.9 h
English	31.0 s	84.0 h
Hindustani	32.5 s	43.6 h
Spanish	32.9 s	83.3 h
Farsi	33.0 s	12.5 h
German	33.6 s	44.5 h
Japanese	33.7 s	26.3 h
Korean	32.6 s	37.6 h
Russian	29.7 s	2.9 h
Tamil	33.9 s	37.5 h
Thai	29.9 s	2.9 h
Vietnamese	33.4 s	40.3 h
background	32.1 s	453.6 h
devtest (from training data)	18.6 s	9.2 h
LRE05	29.4 s	30.3 h

Table 1: Length of conversations sides and total amount of audio after preprocessing for training, background, and test sets. The background data set is comprised of all training data sets plus a small amount of data not used for training.

3. System description

Figure 2 provides an overview over the system. After preprocessing, the data was conveyed to multiple *frontends* consisting of a unit recognition and an n-gram counting component. After normalization, the relative n-gram counts were concatenated to a single large feature vector and fed into a support vector machine (SVM).

3.1. Phone recognition frontends

We were using four different phone recognizer frontends: (1) The English open-loop DECIPHER recognizer [6], developed by SRI. Our version of DECIPHER uses gender-dependent, 3-state hidden Markov models for open-loop phone recognition. The Markov models were trained using mel-frequency cepstral coefficient features of order 13 plus first and second order deriva-

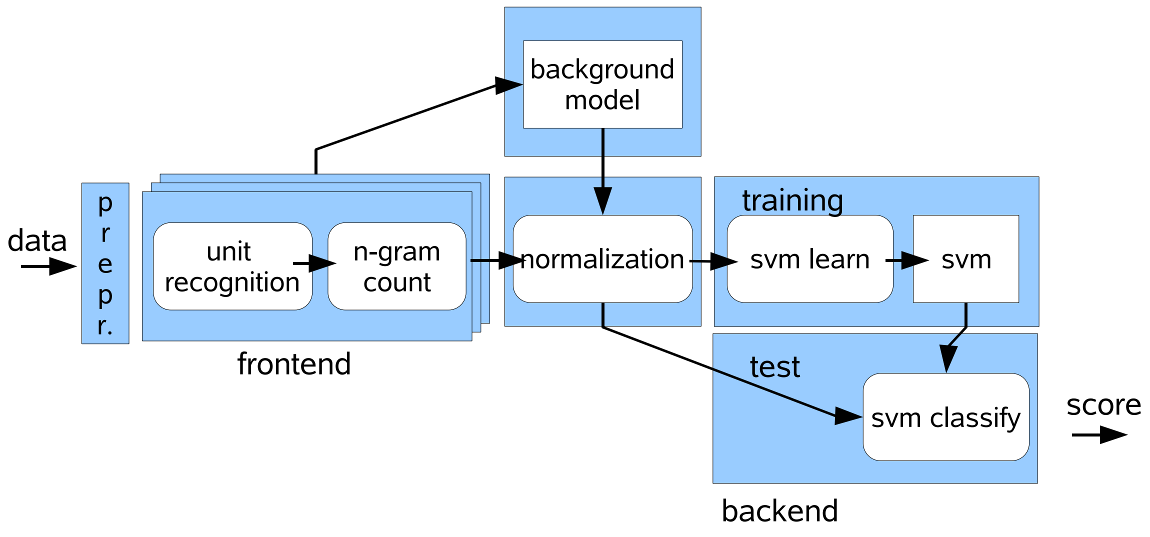


Figure 2: *Schematic diagram of the ICSI 2007 language recognition system.*

tives, with overall dimensionality of 39, on the Switchboard I and II corpora [7]; (2-4) Multi-layer perceptron (MLP) phone classifiers were built for English, Arabic, and Mandarin Chinese languages. The inputs comprised PLP features plus first and second derivatives, with a fast GMM-based estimate for vocal tract length normalization, over a local context of 9 consecutive frames. A feed-forward network structure fully connects the inputs to a large hidden layer, which is connected to output units corresponding to the phones of a language. For each frame of a test utterance, a phone label is determined as the network's output unit with the maximal activation.

For English, gender-specific MLPs with 20800 hidden units were trained on 2000 hours of 8kHz conversational telephone speech, classifying 46 phones; gender was detected with GMM likelihoods. For Arabic, a gender-independent MLP with 10000 hidden units was trained on 465 hours of 16kHz broadcast news, classifying 36 phones. For Mandarin, a gender-independent MLP with 15000 hidden units was trained on 870 hours of 16kHz broadcast news, classifying 71 phones; to aid discrimination of tonal vowels, the PLP inputs were augmented with pitch-based features. For the latter two, the 8kHz samples are up-sampled to 16kHz prior to the recognition.

The MLP-based phone recognizers generated frame-by-frame phone labels. In this vein, the length of the phones is taken into account by the relative counts of homogenous trigrams (e.g. 67_67_67).

In an earlier version of the system the phonotactic difference between languages were learned on the basis of counts of abstract phone-like sub-word units [8] that are generated by models trained on the actual LRE training data. Note, that "real" phone recognizers cannot be trained with that data because the transcriptions or phonetic annotations are not provided. An advantage of this approach was that the "phone set" – i.e. the number of abstract classes – could be chosen according what might be appropriate in a multilingual context. Ideally, one sub-word unit recognizer could be built for each language in the test. However, despite some reasonable results on the LRE03 evaluation set, this variant performed one order of magnitude worse than the one described here on the LRE05 test (the NIST cost value has been at the order of 0.20). This was presumably due to channel issues and requires further investigation.

3.2. Normalization

The relative n-gram counts were first rank-normalized [13] in order to obtain comparable ranges for all features and to map the n-gram frequencies to a uniform distribution. An ordered list of values was created for each feature using the background data. The rank of a given value then corresponds to the position in the list divided by the total number of occurrences of the respective feature in the background data (see section 2). Zero values corresponding to instances in which a particular unit has not occurred in a given sample, were mapped to zero. The ranks lie in the closed interval from 0 to 1 and were used as the normalized value. The feature-value-rank triples were stored in a lookup-table. In testing, linear interpolation was applied if a given triple did not exist. To save memory and processing time, only the most frequent triples were loaded at startup. However, to be able to experiment with the number of features actually used in model training, the less frequent features were normalized as well (the respective triples were loaded when they occurred for the first time). With rank normalization, the difference between two normalized feature values corresponds to the percentage of background samples that fall between the two values. Accordingly, differences were emphasized in high density regions and compressed in low density regions. In the pre-tests we performed, rank normalization outperformed mean-variance normalization.

The normalized features were assigned to a unique (continuous) feature number because the actual trigram name (e.g. 128096003 for coding the trigram 128-096-003) suggests a much larger number of features than actually exist. When combining multiple frontends, a respective index was added to the feature number.

3.3. Model training

The maximum likelihood classification, which represents the backend of the classic PPRLM approach, was replaced by a discriminative training with support vector machines (SVMs). The ability of SVMs to handle very large feature vectors enabled us to use uni-, bi-, and trigrams simultaneously and combine multiple frontends. We used the SVM LITE implementation [14].

The number of components of the feature vectors was reduced by only choosing 30 % of the trigrams (unigrams and bigrams do no contribute as much to the total number of features). This lead to a total number of 136591 features.

For each each language l_i, a one-against-all SVM with a second order polynomial kernel was trained, using the training examples of l_i as positive examples and the training examples all other languages as negative examples. The model for Arabic, for example, was trained on Arabic as positive examples and all other languages test as negative examples. The bias which results from a larger number of negative examples was compensated by choosing an appropriate "j-parameter" – a cost-factor by which training errors on positive examples outweigh errors on negative ones. For the time being, we haven't experimented with the "c-parameter", the trade-off between the training error and the margin.

T-norm score normalization was applied to the scores. With t-norm, scores for a test utterances were generated against the impostor models in order to estimate the impostor score distribution [15]. The mean and variance of the distribution was used to normalize the score of the target model:

$$S_{TN}(X) = \frac{S(X) - \mu_{impostor}(X)}{\sigma_{impostor}(X)} \tag{1}$$

where $S_{TN}(X)$ is the normalized score, $S(X)$ is the original score, and $\mu_{impostor}(X)$ and $\sigma_{impostor}(X)$ are the mean and standard deviation of the distribution of scores for test utterance X against the set of impostor speaker models.

The decision threshold was obtained by testing the models using the development test set. The threshold we applied was the one that generates the equal error rate (EER) rather than the one that minimizes the NIST cost function (see below). Given that the costs for misses and false alarm are equally weighted, we believed that the EER threshold exhibits a better generalization across different test sets.

3.4. Processing speed

We performed a processing speed test on a single 64 bit dual core AMD^{TM}Opteron^{TM}processor (operated in 32 bit mode) running on a Linux 2.6.9 operating system. The data was an excerpt of the LRE 2005 (30 seconds) test corpus and was preprocessed using the scheme described in section 2 (the time for preprocessing was not included in the measure). The processing speed was calculated as the total amount of speech processed (30 hours) divided by the total amount of CPU time. The result was 0.935. Note that the system supports parallel processing as the various frontends can be applied at the same time. In that case the processing speed would be 2.44.

4. Experimental Results

We performed experiments on the development test set as well as on the 2005 NIST language recognition evaluation (LRE05) set (see table 2). The test was designed as a detection test: Each language was consecutively used as target language. A decision threshold was applied to the score of the respective model to decide whether or not a given segment corresponded to the target language. The decision error was expressed in terms of the probability of false alarms (Pfa), the number of trials for which the decision of the system was 'yes' but the segment language was not the target language relative to the number of occurrences of the target language in the data set, and the probability of false rejects (Pfr), the number of trials for which the decision of the system was 'no' but the segment language was in fact the target language relative to the number of non-target languages.

The upper part of Table 4 presents the probabilities of false alarms (Pfa) for a given pair of target and non-target languages. For the target Chinese, Asian non-targets (Japanese and Korean) produced more false alarms than the others which indicates the similarity of the languages. According to Table 4, other pairs of similar languages are English and German, Hindi and Tamil, as well as Japanese and Korean. The results on the development test set also indicate similarities between Arabic and both Farsi, Bengali and Russian, and Tamil and Vietnamese.

In the lower part of Table 4 the probabilities of false alarms (Pfa) and misses (Pmiss) are combined into a single number that represents the cost performance of a system as used by NIST for the language recognition evaluations. The cost value represents an application-motivated cost model and is defined as:

$$C(L_T, L_N) = C_{Miss} * P_{Target} * P_{Miss}(L_T) + C_{FA} * (1 - P_{Target}) * P_{FA}(L_T, L_N)$$

where L_T and L_N are the target and non-target languages, and C_Miss, C_{FA}, and P_{Target} are application model parameters. Here, the application parameters are $C_{Miss} = C_{FA} = 1$, and $P_{Target} = 0.5$

The average cost of the system presented here was 0.0886. The decision-error-tradeoff (DET) curve is presented in Figure 3. Table 4 as well as Figure 3 were created using the scoring software provided by NIST for the 2007 [2].

5. Acknowledgments

The authors would like to thank Andreas Stolcke (SRI International) for providing the English DECIPHER open loop phone recognizer and Arlo Faria (ICSI) for providing the English, Arabic, and Mandarin phone classifiers.

6. References

[1] M.A. Zissman, "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, 1996.

[2] evaluation http://www.nist.gov/speech/tests/lang/2007/

ERROR RATES: Pfa(Lt,Ln) on LRE05

	ARA	BEN	CHI	ENG	FAR	GER	HIN	JAP	KOR	RUS*	SPA	TAM	THA	VIE
CHI			–	.0974		.0124	.0197	.0187	.0425		.0093	.0021		
ENG			.1274	–		.0563	.0551	.0057	.0138		.0333	.0138		
GER			.2195	.2561		–	.0488	.0000	.0122		.0244	.0000		
HIN			.0775	.1479		.0000	–	.0141	.0423		.0845	.0423		
JAP			.3581	.0854		.0000	.0854	–	.1708		.1157	.0220		
KOR			.3581	.1484		.0032	.0677	.1452	–		.0419	.0226		
SPA			.0430	.0826		.0083	.0942	.0281	.0165		–	.0446		
TAM			.0447	.0950		.0000	.1229	.0223	.0447		.0838	–		
Pmiss			.0135	.0563		.1951	.1127	.1433	.1677		.0694	.1117		
Avg Pfa			.1755	.1304		.0115	.0705	.0334	.0490		.0561	.0211		
Avg Pmiss = .1087			Avg Pfa = .0684											

ERROR RATES: Pfa(Lt,Ln) on development test set

	ARA	BEN	CHI	ENG	FAR	GER	HIN	JAP	KOR	RUS	SPA	TAM	THA	VIE
ARA	–	.1733	.0800	.1067	.0867	.1000	.1200	.0400	.0267	.0467	.0867	.0800	.0533	.0067
BEN	.1500	–	.0500	.0333	.0667	.0167	.2333	.1333	.0333	.1667	.1000	.0667	.0500	.0167
CHI	.0920	.1264	–	.1264	.0230	.0460	.0862	.1322	.1322	.0747	.0920	.0402	.1954	.1149
ENG	.1111	.0370	.1376	–	.0476	.0741	.1005	.0423	.0688	.0265	.0952	.0476	.0370	.0582
FAR	.2174	.0217	.1196	.0761	–	.1304	.0978	.0326	.0435	.0109	.0870	.0217	.0000	.0217
GER	.1024	.0236	.0472	.1339	.0472	–	.1181	.0472	.0315	.0472	.0315	.0394	.0000	.0236
HIN	.1005	.1376	.1164	.1429	.0635	.0317	–	.0212	.0635	.0423	.1217	.1111	.0317	.0106
JAP	.0841	.0187	.1495	.0374	.0374	.0467	.0374	–	.3084	.0187	.1121	.0841	.0000	.0093
KOR	.1349	.0397	.1667	.0714	.0476	.0476	.0873	.1349	–	.0079	.0794	.0556	.0159	.0317
RUS	.1803	.1639	.1639	.1475	.0820	.0820	.2623	.0656	.0328	–	.1475	.0000	.0164	.0000
SPA	.1006	.0566	.0881	.1132	.0314	.0314	.1572	.0440	.0377	.0440	–	.0943	.0126	.0566
TAM	.0877	.0175	.0175	.0526	.0088	.0351	.2368	.0175	.0877	.0263	.1404	–	.0351	.0351
THA	.0889	.1333	.4000	.0444	.0000	.0667	.1778	.0000	.0667	.0222	.0444	.0667	–	.2000
VIE	.0833	.0000	.1083	.0833	.0333	.0500	.1250	.0000	.0333	.0167	.0583	.0417	.0417	–
Pmiss	.1133	.0667	.1034	.0952	.0435	.0630	.1217	.0561	.0794	.0492	.1006	.0702	.0667	.0500
Avg Pfa	.1179	.0730	.1265	.0899	.0442	.0583	.1415	.0547	.0743	.0424	.0920	.0576	.0376	.0450
Avg Pmiss = .0771			Avg Pfa = .0754											

COSTS: C(Lt,Ln) on LRE05

	ARA	BEN	CHI	ENG	FAR	GER	HIN	JAP	KOR	RUS	SPA	TAM	THA	VIE
CHI			–	.0768		.1038	.0662	.0810	.1051	.0394	.0569			
ENG			.0705	–		.1257	.0839	.0745	.0908	.0514	.0628			
GER			.1165	.1562		–	.0807	.0716	.0900	.0469	.0559			
HIN			.0455	.1021		.0976	–	.0787	.1050	.0770	.0770			
JAP			.1858	.0708		.0976	.0990	–	.1693	.0926	.0669			
KOR			.1858	.1023		.0992	.0902	.1442	–	.0557	.0672			
SPA			.0282	.0695		.1017	.1034	.0857	.0921	–	.0782			
TAM			.0291	.0756		.0976	.1178	.0828	.1062	.0766	–			
Avg Cost			.0945	.0933		.1033	.0916	.0883	.1084	.0628	.0664			
Avg Cost = .0886														

COSTS: C(Lt,Ln) on development test set

	ARA	BEN	CHI	ENG	FAR	GER	HIN	JAP	KOR	RUS	SPA	TAM	THA	VIE
ARA	–	.1200	.0917	.1010	.0651	.0815	.1208	.0480	.0530	.0479	.0936	.0751	.0600	.0283
BEN	.1317	–	.0767	.0643	.0551	.0398	.1775	.0947	.0563	.1079	.1003	.0684	.0583	.0333
CHI	.1026	.0966	–	.1108	.0332	.0545	.1040	.0941	.1058	.0619	.0963	.0552	.1310	.0825
ENG	.1122	.0519	.1205	–	.0455	.0685	.1111	.0492	.0741	.0378	.0979	.0589	.0519	.0541
FAR	.1654	.0442	.1115	.0857	–	.0967	.1098	.0443	.0614	.0300	.0938	.0460	.0333	.0359
GER	.1078	.0451	.0753	.1145	.0454	–	.1199	.0517	.0554	.0482	.0661	.0548	.0333	.0368
HIN	.1069	.1021	.1099	.1190	.0535	.0474	–	.0386	.0714	.0458	.1112	.0906	.0492	.0303
JAP	.0987	.0427	.1265	.0663	.0404	.0549	.0795	–	.1939	.0339	.1064	.0771	.0333	.0297
KOR	.1241	.0532	.1351	.0833	.0455	.0553	.1045	.0955	–	.0286	.0900	.0629	.0413	.0409
RUS	.1468	.1153	.1337	.1214	.0627	.0725	.1920	.0608	.0561	–	.1241	.0351	.0415	.0250
SPA	.1070	.0616	.0957	.1042	.0375	.0472	.1395	.0500	.0586	.0466	–	.0823	.0396	.0533
TAM	.1005	.0421	.0605	.0739	.0261	.0490	.1793	.0368	.0835	.0377	.1205	–	.0509	.0425
THA	.1011	.1000	.2517	.0698	.0217	.0648	.1497	.0280	.0730	.0357	.0725	.0684	–	.1250
VIE	.0983	.0333	.1059	.0893	.0384	.0565	.1233	.0280	.0563	.0329	.0795	.0559	.0542	–
Avg Cost	.1156	.0699	.1150	.0926	.0439	.0607	.1316	.0554	.0768	.0458	.0963	.0639	.0521	.0475
Avg Cost = .0762														

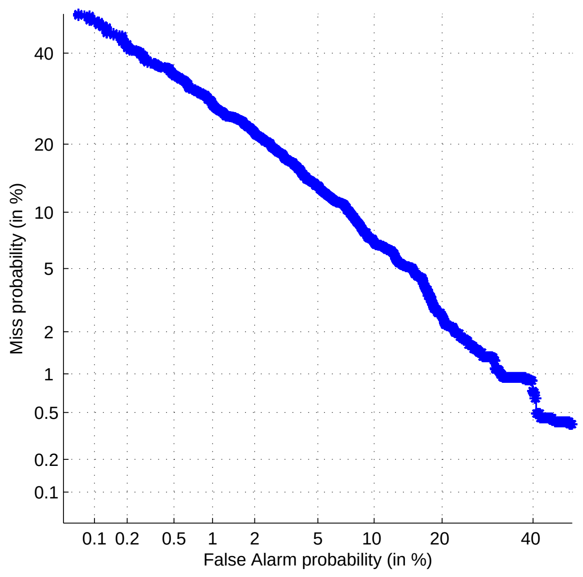


Figure 3: *Decision Error Tradeoff (DET) curve on the LRE05 evaluation set.*

[2] Zhai L., Siu M., X. Yang, and H. Gish, "Discriminatively trained Language Models Using Support Vector Machines for Language Identification," in *Proceedings of the 2006 Odyssey Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, 2006, IEEE.

[3] Bin Ma and Haizhou Li, "A Comparative Study of Four Language Identification Systems," *Computational Linguistics and Chinese Language Processing*, vol. 11, no. 2, 2006.

[4] Christopher White, Izhak Shafran, and Jean-Luc Gauvain, "Discriminative Classifiers for Language Recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 06)*, 2006.

[5] W. M. Campbell, E. Singer, Torres-Carrasquillo, and D. A. P. A., Reynolds, "Language Recognition with Support Vector Machines," in *In Proceedings of the Odyssey Speaker and Language Recognition Workshop*, Toledo, Spain, 2004, pp. 41–44.

[6] S. Kajarekar, L. Ferrer, A. Venkataraman, K. Sonmez, E. Shriberg, A. Stolcke, and R.R. Gadde, "Speaker Recognition using prosodic and lexical features," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2003, pp. 19–24.

[7] David Graff and Steven Bird, "Many uses, many annotations for large speech corpora: Switchboard and TDT as case studies," in *Proceedings of the 2nd Language Resources and Evaluation Conference (LREC 2000)*, Athens, Greece, 2000.

[8] Alberto Montero, "Exploring PPRLM performance for NIST 2005 Language Recognition Evaluation," in *Proceeedings of the IEEE Odyssey 2006 Workshop on Speaker and Language Recognition*, San Juan, Puerto Rico, 2006.

[9] J. Wilpon, B. Juang, and L. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '87)*, Dallas,TX, 1987, vol. Volume 12, pp. 821–824.

[10] K.K. Paliwal, "Lexicon-building methods for an acoustic sub-word based speechrecognizer," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '90)*, Albuquerque, NM, 1990, vol. Volume 2, pp. 729–732.

[11] A.K.V.S. Jayram, V. Ramasubramanian, and T.V. Sreenivas, "Automatic Language identification using parallel sub-word recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, 2003, vol. 1, pp. 32–35.

[12] T Nagarajan and H.A. Murthy, "Language identification using parallel syllable-like unit recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, Montreal, Canada, 2004, pp. 401–404.

[13] E. Shriberg, S. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling Prosodic Feature Sequences for Speaker Recognition," *Speech Communication*, vol. 46, pp. 455–472, 2005.

[14] T. Joachims, "Making large-scale svm learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C Burges, and A. Smola, Eds. MIT-Press, 1999.

[15] R . Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.