Automatic Language Identification using Phoneme and Automatically Derived Unit Strings

Pavel Matějka[1,2], Igor Szöke[2,3], Petr Schwarz[2], and Jan Černocký[2]

[1] Brno University of Technology, Faculty of Elec. Eng. and Communication,
[2] Brno University of Technology, Faculty of Information Technology,
[3] ESIEE Paris, Dpt. Signal et Télécommunications
matejkap|szoke|schwarzp|cernocky@fit.vutbr.cz

Abstract. Language identification (LID) based on phono-tactic modeling is presented in this paper. Approaches using phoneme strings and strings of units automatically derived by an Ergodic HMM (EHMM) are compared. The phoneme recognizers were trained on 6 languages from OGI multi-language-corpus and Czech SpeechDat-E. The LID results are obtained on 4 languages. The results show superiority of Czech phoneme recognizer while used in LID and promising trends using the EHMM-derived units.

1 Introduction

The goal for Language Identification is to determine the language a particular speech segment was spoken. This work concentrates on phono-tactic approach to language identification. The speech signal is first converted into a sequence of discrete sub-word (tokens) units that can characterize the language (Figure 1). In our case, these units are phonemes detected by a phoneme recognizer or automatically derived units obtained by an Ergodic Hidden Markov model (EHMM).

In the training, the tokens are obtained for all languages which we want to recognize. For all languages, phono-tactic models are estimated using N-gram modeling based on strings of derived tokens. Universal background model (UBM) is estimated from all languages together. UBM normalizes all language-dependent models.

In testing, scores are evaluated using language-dependent and UBM models. Based on the normalized scores, the test segment is attributed to one of the languages or rejected as unknown.

2 The LID system

2.1 Phonemes derived using TRAP

Phonemes are the usual choice for LID systems based on phono-tactic models. Our previous work has shown great efficiency of phoneme recognition based

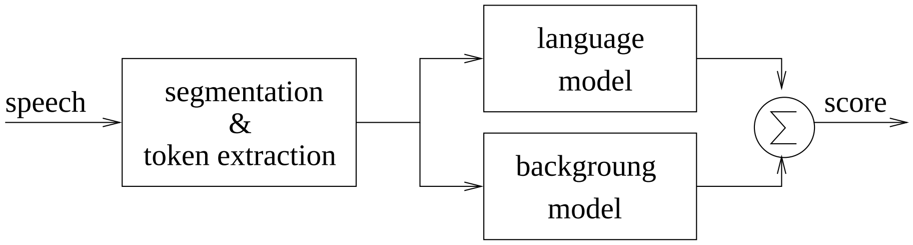

Fig. 1. Language identification system

on Temporal Patterns (TRAPs) and Neural Networks [1]. In this technique, frequency-localized posterior probabilities of sub-word units (phonemes) are estimated from temporal evolution of critical band spectral densities within a single critical band. Such estimates are then used in another class-posterior estimator which estimates the overall phoneme probability from the probabilities in the individual critical bands.

Base TRAPs and Merger-only system The pre-processing for TRAPs is very similar to conventional feature extraction: Speech signal is divided into 25 ms long frames with 10 ms shift. The Mel filter-bank is emulated by triangular weighting of FFT-derived short-term spectrum to obtain short-term critical-band logarithmic spectral densities.

TRAP feature vector describes a segment of temporal evolution of such critical band spectral densities within a single critical band. The usual size of TRAP feature vector is 101 points [3]. The central point is actual frame and there are 50 frames in past and 50 in future. That results in 1 second long time context. The mean and variance normalization can be applied to such temporal vector. Finally, the vector is weighted by Hamming window.

In "classical" works dealing with TRAPs [2, 3], this vector forms an input to a classifier. Outputs of the classifier are posterior probabilities of classes which we want to distinguish among (context-independent phonemes). Such classifier is applied in each critical band. The merger is another classifier and its function is to combine band classifier outputs into one. The described techniques yields phoneme probabilities for the center frame. Both band classifiers and merger are neural nets.

The system described above is however quite complex, and we have suggested a simpler variant: *Merger only* system. There are no band classifiers and all TRAP vectors are going directly to one classifier - merger. Discrete Cosine Transform (DCT), Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) is used in this case, because we are not able to deal with highly dimensional vector created by concatenation of TRAP vectors. This system is shown in Figure 2. The advantage is more then real time processing with the same accuracy as previous system [1, 4]. Based on our previous investigation we have also reduced the context to 31 points (300 ms) — in [1] we have shown

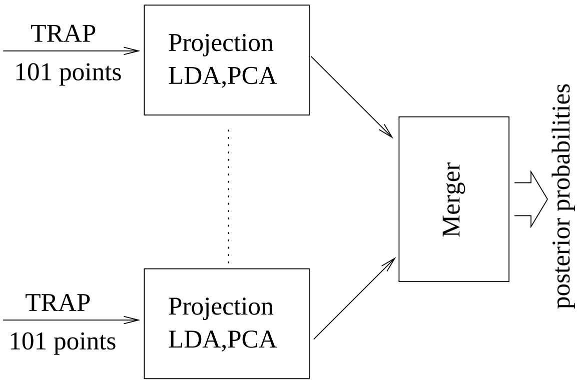

Fig. 2. Merger system only

that this size was optimal for phoneme recognition. This implies further saving of computational power.

2.2 Speech units derived automatically using Ergodic HMM

Phoneme recognizers have a major drawback – they have to be trained on a phonetically labeled database. As LID is one of applications, where we do not need to decode the lexical information, we should be able to use any acoustically coherent speech units. Our group has been active in very low bit-rate coding using such units [6]. In this work, the units are derived using a simple Ergodic Hidden Markov model (EHMM).

A comparison of EHMM with widely used left-right HMM is shown in 3. EHMM is fully connected and (unlike standard HMMs which need labeled data) it is trained on the entire speech database using standard Baum-Welch algorithm. Crucial part of EHMM training is the initialization of its states (see next paragraph).

In recognition mode, the recognizer outputs a stream of states. Each state represents a part of speech (automatically generated unit).

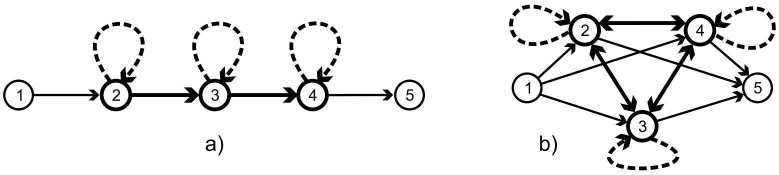

Fig. 3. Hidden Markov models: a) standard, b) ergodic

Initialization The initialization is important for EHMM's successful usage [5]. Correctly trained EHMM should assign only one state to a set of acoustically similar segments.

The simplest way to initialize EHMM is to put *constant or random values* to vector means and variances in states. This is whoever the worst approach. Constant values will theoretically lead to all states representing the same unit. Random values are not good too. The vector state space is smaller than all possible values generating by random generator.

Slightly better results are obtained using *random values over training database*, where each vector mean value is set to a random vector in database. All variances are set to global variance of the database. The disadvantage of this approach is that we initialize states to most widely represented sounds in the database (for example 30% of states represent silence if database contains 30% of silence).

The best results were obtained with *iterative state splitting*. One selected state is split into two states. Means are set to be little bit different (adding and subtracting fraction of global mean variance) and EHMM is trained again. We tested the following methods to select the candidate state to be split:

data – Splitting states according amount of data belonging to states. Split the state which have the biggest amount of data.

likelihood – Splitting states according decoded log–likelihood. State which have the worst log–likelihood is split to two states.

norm likelihood – Splitting states according decoded log–likelihood normalized by number of consequent frames belonging to one state. State which have the worst log–likelihood is split to two states.

2.3 Phono-tactic modeling

Both phoneme recognizer and EHMM produce strings of discrete units. Statistical models are used to model their sequencing in different languages. We have used modeling three consequent units (tri-grams) which is very popular also in language models (LM). A simple smoothing was implemented using a fixed threshold for minimum tri-gram probability.

3 Experiments

All experiments were evaluated on database with 4 languages (2 Slavic, 1 Germanic, 1 exotic) recorded using telephone line. The Database includes also separate portions of silence, technical noises (like modem) and dialing tones. The amounts of data are shown in Table 1.

The results were evaluated in terms of Equal Error Rate (EER) (averaging the point with equal probability of false acceptation and false rejection for all target languages) and correctness (CORR), where a threshold had to be set and decision taken for each test segment.

Prior to phoneme/EHMM recognition, the data were pre-processed in the following way:

Language	Amount of Data
exotic	18.94 hours
Germanic	25.23 hours
Slavic # 1	17.33 hours
Slavic # 2	19.80 hours
Technical noise	28.89 min
Silence	44.03 min
Dialing tone	7.93 min

Table 1. Amount of data per language

- Long parts of silence have been removed, because the final step of language identification does not need to deal with all the data and whole system is much faster.
- Algorithms for detecting technical noises and dialing tones were used for cutting out these parts or rejecting whole sentence. These algorithms were based on constant energy for a long time, equally spread spectrum of signal over all frequencies and ratios between highest peaks in spectrum.

3.1 LID using phoneme recognition

Seven different phoneme recognizers were tested for tokenization of speech signal. Each recognizer is trained on different language. There are six languages from OGI multi-language corpus [9] (English, German, Hindi, Japanese, Mandarin and Spanish) and the seventh language is Czech from SpeechDat-E corpus. Amounts of training data are shown in Table 2. All recognizers were designed according to Figure 2. Time trajectory of 310 ms reduced with 15 coefficients of DCT transform were extracted from 15 mel-scaled frequency banks. The recognizers were trained on the training parts of respective databases. The increase of classification error on the cross-validation part during training was used as a stopping criterion to avoid over-training. There is one ad hoc parameter in the system, the word (phoneme) insertion penalty, which has to be set. Generally, for phoneme recognition, this constant is tuned to the equal number of inserted and deleted phonemes on the cross-validation part of the database. In our case this constant was tuned to the best language identification score.

Phoneme error rates and number of tokens (phonemes) are shown in Table 2. It is without any doubt that the error rate heavily depends on the amount of data — we have presented 8% improvement on TIMIT database between for 0.5 versus 3 hours of training data [4]. Therefore it is not surprising why the first 6 recognizers are about 10% worse on phoneme error rate against the Czech one. But different recognizers output different segmentations with different tokens and modeling of them brings different hypotheses for language identification.

Accuracy (%)	**English**	**German**	**Hindi**	**Japanese**	**Mandarin**	**Spanish**	**Czech**
Training Data [h]	2.20	1.05	0.72	0.67	0.64	1.12	10.98
Phonemes	39	43	46	29	44	39	46
Accuracy	38.11	41.88	50.14	51.51	40.24	53.11	62.80

Table 2. Accuracy of phoneme recognition on different languages

	EER [%]	**CORR [%]**
English	15.89	77.08
German	16.67	75.52
Hindi	14.36	77.60
Japanese	14.84	77.34
Mandarin	16.97	70.57
Spanish	13.54	79.95
Czech	8.07	88.02

Table 3. Equal error rate and correctness of LID for several phoneme recognizers trained on different languages

All 7 recognizers were used to tokenize the LID target data. Statistical modeling of context 3 (tri-grams) were used to capture context dependencies of tokens. Results of identification for individual recognizers are shown in Table 3.

Channel adaptation We were aware that the phoneme recognizers used were trained on data from channels different from target LID data. Therefore we used Czech phoneme recognizer to label all the LID data and using these labels we trained another phono-tactic model that should be closer to the target data. We improved the system more than 1%. Complete results are in the Table 4.

3.2 LID using units derived automatically by EHMM

In case of EHMM, it is difficult to assess the correctness of the segmentation, as we can not compare with any reference labels. To check the coherence of units, we can use some *visualization tool.* We can plot speech signal, spectra and generated state alignment. Coherency can be easily seen. Visualization starts however to be difficult for larger EHMMs. Another test is *subjective listening* units belonging to one state. By this way, we can listen whether units of one state sound coherently. But this approach is quite time consuming, is subjective and can hardly detect different states covering similar type of units. Therefore, the ultimate number for the EHMM approach is the reached EER and correctness on the target data.

EHMMs with up to 32 states (automatically derived units) were trained on Czech part of SpeechDat database. Classical features were used in EHMM

	EER [%]	**CORR** [%]
Czech	8.07	88.02
Adapted	7.05	90.38

Table 4. Equal error rate and correctness of original system and adapted one.

EHMM	states	whole database		cut database	
system		**EER** [%]	**CORR** [%]	**EER** [%]	**CORR** [%]
data	8	28.21	60.90	29.49	54.49
data	16	12.18	76.28	26.92	58.97
data	32	**9.62**	**83.33**	23.72	60.26
likelihood	8	28.85	60.90	29.49	53.21
likelihood	16	28.21	54.49	25.00	57.05
likelihood	32	25.64	58.33	**8.33**	**85.90**
norm likelihood	8	30.77	52.56	27.56	55.13
norm likelihood	16	25.00	57.69	26.28	57.05
norm likelihood	32	15.38	74.36	25.00	58.97

Table 5. Equal error rate and correctness of LID with EHMM units.

experiments: 12 MFCC coefficients plus zeroth cepstral coefficient appended with deltas and double deltas. Two approaches were tested for the training of EHMM:

– using whole database including silence and other noises.
– only clean speech (cut database).

Table 5 shows results with three methods of splitting states in EHMM, described in 2.2.

4 Conclusions

The results of LID based on phoneme recognition proved that the quality of phoneme recognizer is crucial for good LID performances. Not surprisingly, the amount of data available for training of phoneme recognizer is the main factor influencing directly the final LID error rates. The gain obtained using a simple channel adaptation is promising and we will investigate more elaborate techniques. We will also explore the possibility to use more states in phoneme models and to merge the results of LID using different number of states. Construction of universal phoneme recognizer using all available training data is yet another possibility.

The use of EHMM-derived units is a novel method and the experiments are still not complete. From the preliminary results presented in the paper, we can conclude that the performances of automatically derived units in LID approach those of traditional phoneme models. Further experiments need however to be conducted to verify, that the EHMM-units do not over-represent the channel and non-speech parts and that they generalize also for other sources of data.

Finally, the phono-tactic modeling could be done by more complicated technique based on decision trees. These trees would look for typical sequences of phonemes much longer than 3-grams, but be able to back-off to shorter ones.

5 Acknowledgments

This research has been partially supported by Grant Agency of Czech Republic under project No. 102/02/0124 and by a grant from Ministry of Defense of Czech Republic. Pavel Matějka was supported by doctoral grant of Grant Agency of Czech Republic No. 102/03/H105 and Jan Černocky by post-doctoral grant of Grant Agency of Czech Republic No. GA102/02/D108.

References

1. Schwarz, P., Matějka, P., Černocký, J. "Recognition of Phoneme Strings using TRAP Technique". *Proc. Eurospeech'03*, pp. 825-828, September 2003.
2. S. Sharma, D. Ellis, S. Karajekar, P. Jain and H. Hermansky, "Feature extraction using non-linear transformation for robust speech recognition on the Aurora database", *Proc. ICASSP 2000*, Turkey, 2000.
3. H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech", *Proc. ICASSP'99*, Phoenix, Arizona, USA, Mar, 1999
4. Schwarz, P., Matějka, P., Černocký, J. "Towards Lower Error Rates in Phoneme Recognition". *Submitted to TSD 2004.*
5. Szöke, I., Černocký, J. "Speech Units Automatically Generated by Ergodic Hidden Markov Model". *Submitted to EEICT 2004.*
6. Černocký, J., Baudoin, G., Chollet, G. "Segmental vocoder - going beyond the phonetic approach", *Proc. IEEE ICASSP 98*, Seattle, May 1998, pp. 605–608.
7. The SPRACHcore software packages, *www.icsi.berkeley.edu/∼dpwe/projects/sprach*
8. HTK toolkit, *htk.eng.cam.ac.uk*
9. OGI MultiLanguage Telephone Speech. *www.cslu.ogi.edu/corpora/mlts/*, January 2004.