USE OF ANTI-MODELS TO FURTHER IMPROVE STATE-OF-THE-ART PRLM LANGUAGE RECOGNITION SYSTEM

Pavel Matějka, Petr Schwarz, Lukáš Burget and Jan Černocký

Speech@FIT group, Brno University of Technology, Czech Republic
{matejkap,schwarzp,burget,cernocky}@fit.vutbr.cz

ABSTRACT

This paper concentrates on PRLM (phoneme recognizer followed by language model) approach to language recognition. It elaborates on our prior work concerning the quality of phoneme recognition and amounts of training data for phoneme recognizer training. It reports improvements brought to our PRLM system by better phoneme recognition and Witten-Bell discounting in LM-modeling. The paper then concentrates on the use of phoneme lattices and anti-models. Training and scoring on phoneme lattices brought significant improvement in language recognition accuracy. The anti-models are simple, yet powerful technique to improve the discrimination between target and non-target languages. All results are reported on standard NIST 2003 data; comparison with other published results is favorable to our system.

1. INTRODUCTION

Automatic language identification (LID) has increasing importance among speech processing applications. It can be used to route calls to human operators (commerce, emergency), pre-select suitable speech recognition system (information systems) and has many uses in security applications.

The goal for Language Identification is to determine the language a particular speech segment was spoken. The algorithms for LID can be roughly divided (see for example [1]) into two groups. In *phonotactic modeling*, a tokenizer transcribes the input speech into phonemes and the scoring is performed on phoneme strings or lattices. This approach is mostly referred to as PRLM (Phoneme recognizer followed by language model) or PPRLM (Parallel PRLM). In *acoustic modeling*, the input features are modeled directly by Gaussian mixture models (GMM), artificial neural networks, support vector machines, or other techniques [2].

This paper concentrates on the phonotactic approach. In [3] we have claimed that the quality of PRLM and PPRLM heavily depended on the quality of phoneme recognizer and on the amount of available training data. We use high-quality phoneme recognizer based on so called LC-RC FeatureNet approach and in [3], we have presented phoneme recognizers trained on 4 languages from SpeechDat-East database. Although none of these languages is equivalent to any of the target languages in NIST 2003 LID data, the simple fact that these databases contain $10\times$ more data than OGI-Stories (usually used to train tokenizers in LID) greatly improves the LID accuracy.

Here, we first report further improvement of this system by slight changes in the phoneme recognizer. The main focus is however on the following points:

- using phoneme lattice rather than strings for both training and scoring by phonotactic models. This approach was pioneered by LIMSI [4] with good results, and our results with phoneme lattices (though our approach was simpler) were also very satisfactory.
- use of anti-models: phonotactic models trained on misrecognized segments that should help to discriminate between target and non-target language. Similar approach was used by SRI in large vocabulary continuous speech recognition (LVCSR) [5] to compensate for hypothesis that are acoustically confusable with the correct transcriptions, we have however not seen any use of such technique in LID.

The paper is organized as follows: section 2 reviews the architecture of our PRLM system. The following section 3 concentrates on the experimental data and baseline results. Section 4 reports the results obtained with phoneme lattices and section 5 contains the core of the paper - investigation into anti-models. The paper is concluded in section 6.

2. SYSTEM DESCRIPTION

2.1. Phoneme recognizer - LCRC FeatureNet

Good phoneme recognizer is the most important part of an accurate PRLM LID system. We use a hybrid system based on Neural Networks (NN). The feature extraction makes use of long temporal context, known as TRAPs (temporal patterns) [7]. First, Mel filter bank energies are obtained in conventional way. After sentence mean normalization in each band, temporal evolution of critical band spectral densities are taken around each frame. Based on our previous work in phoneme recognition [8], the context of 31 frames (310 ms) around the current frame was selected. This context is split into 2 halves: Left and Right Contexts (hence the name "LCRC"). This allows for more precise modeling of the whole trajectory while limiting the size of the model (number of weights in the NN) and reducing the amount of necessary training data [8]. Both parts are processed by discrete cosine transform to de-correlate and reduce dimensionality. Two NNs are trained to produce phoneme posterior probabilities for both context parts. Third NN functions as a merger and produces final set of phoneme-state posterior probabilities (Figure 1).[1]

A simple Viterbi decoder[2] without any language model constraints processes output of the merger and produces string of

This work was partially supported by EC project Augmented Multiparty Interaction (AMI), No. 506811, Grant Agency of Czech Republic under project No. 102/05/0278 and by industrial grant from CAMEA Ltd., Brno.

[1] All nets are trained using QuickNet from ICSI http://www.icsi.berkeley.edu/Speech/qn.html

[2] SVite, which is part of STK-toolkit developed at Brno University of Technology: http://www.fi t.vutbr.cz/speech/sw/stk.html

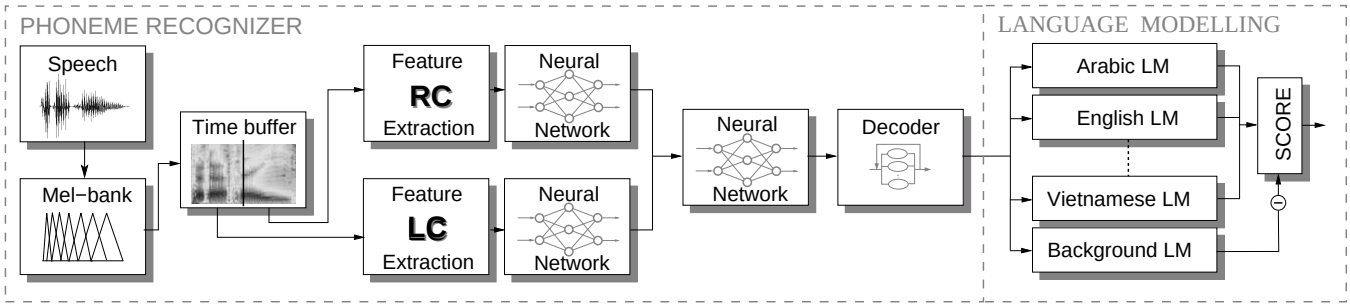

Fig. 1. PRLM system based on phoneme recognizer with split temporal context

Arabic (Egyptian)	Japanese	Farsi
French (Canadian French)	German	Hindi
English (American)	Korean	Mandarin
Spanish (Latin American)	Tamil	Vietnamese

Table 1. The twelve target languages

phonemes. In [8], we have shown that this system outperforms phoneme recognizers with GMM/HMM modeling.

2.2. Phonotactic model - trigrams

Smoothed trigram language model was used to capture phonotactic statistics of each language. It was created by passing training speech of all target languages through phoneme recognizer and counting trigrams for each language separately. Phoneme insertion penalty (PIP) in the decoder was tuned on 1996 NIST data with the best LID performance as criterion. Previously, the problem of unseen trigrams was solved by replacing them by a constant, which had to be experimentally tuned. The current version uses standard Witten-Bell discounting [10].[3]

2.3. Recognition

During recognition, the test sentence is passed through the phoneme recognizer. The resulting string of phonemes is processed by all phonotactic models, and for each, the likelihoods of all trigrams are multiplied. Likelihoods are normalized over all languages by dividing the score by sum of all. Finally we have scores for all target languages. Our previous paper [3] discusses also the merging of scores of several PRLM recognizers into one (PPRLM), here, we concentrate on the use of just one phoneme recognizer.

2.4. Evaluation

The evaluation is done according to NIST [12] per-language, considering each system is a language *detector* rather than recognizer. A standard detection error trade-off (DET) curve is evaluated as a plot of probability of false alarms against the probability of misses with the detection threshold as parameter and equal priors for target and non-target languages. Equal error rate (EER) is the point where these probabilities are equal. The total EER of the whole LID system is the average of language-dependent EERs.

[3]implemented in SRI LM toolkit [9]
http://www.speech.sri.com/projects/srilm/

3. DATA AND BASELINE RESULTS

3.1. Databases

All data used for experiments were recorded over telephone line.

The **phoneme recognizer** used throughout this paper was trained on Hungarian SpeechDat-East database [11] which generated the best individual PRLM results in our previous work [3]. Only phonetically balanced items are used for the training of phoneme recognizer, the sizes of training, test and cross-validation (CV) sets are 7.86, 1.97 and 0.77 hours respectively. Hungarian phonetic alphabet contains 62 phonemes.

Phonotactic models were trained on the CallFriend Corpus [13]. Each of 12 target languages (Table 1) contains 20 complete half-hour conversations.

Test Data comes from NIST 2003 LID evaluation [12]. This data set consists of 80 segments with durations of 3, 10 and 30 second in each of 12 target languages (Table 1). All results in this paper (except for the final Table 4) are reported for 30s segments. This data comes from conversations collected for the CallFriend Corpus but not included in its publicly released version. In addition, there are four additional sets of 80 segments of each duration selected from other LDC conversational speech sources, namely Russian, Japanese, English and cellular English.

3.2. Baseline results

Table 2 summarizes the baseline results. First, the result obtained with Hungarian phoneme recognizer in [3] was reproduced. Then, the obvious shortcoming in our previous work — use of hard constant to replace unseen trigrams — was fixed by Witten-Bell discounting. This improved slightly the resulting EER.

More improvement was obtained from optimizing the phoneme recognizer. The first change consisted in increasing the size of hidden layer from 500 to 1500 neurons. As the next step, the scheduler for neural network learning rate was changed to halve the learning rate learning if the decrease in the frame error-rate (FER) on the *training* (rather than on the CV) set is less than 0.5 %. The number of training epochs was fixed to 20. Both changes lead to more than 0.5% absolute improvement in EER. HU-PRLM-new+Witten-Bell was used as the new baseline, it also served to generate the lattices and anti-models described in the following sections.

4. PHONEME LATTICES

The following work concentrated on estimating LM statistics from phoneme lattices rather than from strings and also using lattices to

system	PER [%]	EER [%]
HU-PRLM from [3]	35.9	4.4
HU-PRLM from [3]+Witten-Bell	35.9	3.7
HU-PRLM-new+Witten-Bell	33.3	3.1

Table 2. Baseline results with Hungarian phoneme recognizer and phonotactic model on phoneme strings. For information, phoneme error rates are in the first column.

	training on string	training on lattice
scoring string	3.1	3.1
scoring lattice	25.5	2.3

Table 3. Experiments with phoneme strings and lattices.

score. PRLM is based on tokenizing speech first. We have shown that it is not important when the tokens (phonemes) come from a different language. We have however to take into account that the tokenizer, as all speech recognition techniques, is not 100% accurate. Common practice in LVCSR, acoustic information retrieval, etc. is to use richer structure at the end of decoder: lattices instead of strings.

In LID, this approach was tested by Gauvain at al. [4] with good results. Gauvain at al. find estimates of N-gram statistics iteratively by EM algorithm. First, they compute phoneme posterior probabilities in the lattice. New N-gram estimates are then weighted by these posteriors. In several iterations, they 1) expand old lattice by the new N-gram probabilities, 2) recompute phoneme posteriors (now with phonotactic scores), and 3) make new N-gram estimates.

We have used simpler approach: we generated phoneme lattices only from acoustic scores without introducing any phonotactic constraints. Then, all four combinations of LM-estimation and scoring (see Table 3) were tested. When lattice was used in training or scoring, the N-grams on given path were weighted by the posterior probability of this path.

In Table 3 we see that training on lattice and scoring raw strings does not bring any improvement and training on strings and scoring lattices causes complete breakdown of the system. The most intuitive lattice-lattice setup performs the best bringing almost 1% absolute improvement in EER.

5. ANTI-MODELS

Anti-model training works in the following way: We will denote all utterances belonging to language L as set S_L^+ and all utterances not belonging to language L as set S_L^-. First, the training of phonotactic model LM_L^+ of each language L is done in standard way using only the set S_L^+. Then, all training utterances are scored by all phonotactic models and posteriors of utterances are derived:

$$\mathcal{P}(\mathcal{O}_r|L) = \frac{\mathcal{L}(\mathcal{O}_r|LM_L^+)}{\sum_{\forall L}\mathcal{L}(\mathcal{O}_r|LM_L^+)} \quad (1)$$

where $\mathcal{O}_r$ is the r-th training utterance and $\mathcal{L}(\mathcal{O}_r|LM_L^+)$ is the likelihood provided by phonotactic model LM_L^+.

For language L, the parameters of anti-model LM_L^- should be trained on all segments from S_L^- mis-recognized as L. We can however use *all* utterances $\mathcal{O}_r \in S_L^-$ and weight their trigram counts by the posteriors $\mathcal{P}(\mathcal{O}_r|L)$. Obviously, an utterance from S_L^- with high probability to be mis-recognized as L will contribute more to the anti-model than an utterance correctly recognized as language G where $G \neq L$.

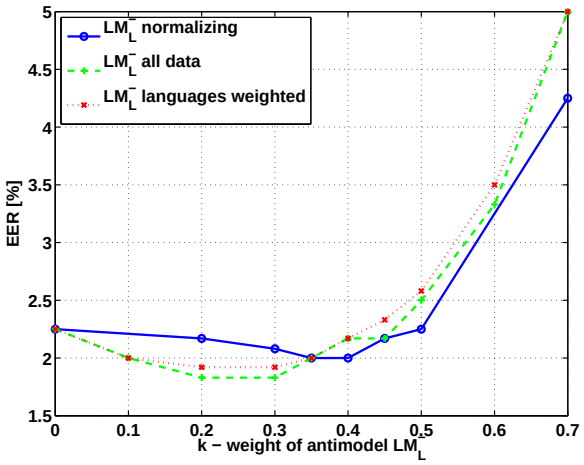

Fig. 2. Results with different anti-models.

We have tested three flavors of anti-model training:

1. LM_L^- is estimated from segments of S_L^- but also from S_L^+. We could call this model "normalizing model" rather than anti-model.
2. LM_L^- is estimated only from segments of S_L^- with pure posterior (Eq. 1) weighting of trigram counts.
3. LM_L^- is estimated only from segments of S_L^-, but in addition to posterior weighting, the trigram counts are also inversely weighted by the priors of different languages in S_L^-. For example, when we train anti-model for Arabic and we see 90% English and 10% of Tamil in S_L^-, the counts of English segments are divided by 0.9 and these of Tamil by 0.1.

We have also experimented with weighting the trigram counts by posteriors (Eq. 1) while training the *target* model LM_L^+ but abandoned this idea – doing this, we are effectively decreasing the amount of data available to train LM_L^+; several tests showed much worse results with this training than while training LM_L^+ with all data from S_L^+ without any weighting.

In all three cases, the *score* of utterance $\mathcal{O}_r$ is obtained by subtracting the weighted likelihood of anti-model from the target model:

$$\log \mathcal{S}(\mathcal{O}_r|L) = \log \mathcal{L}(\mathcal{O}_r|LM_L^+) - k \log \mathcal{L}(\mathcal{O}_r|LM_L^-),$$

where the constant k needs to be tuned experimentally.

In all anti-model experiments, language models were trained and evaluated on lattices and Witten-Bell discounting was used. Figure 2 presents the resulting EERs of the system for different settings of k. For $k = 0$ (no anti-model), all results are equal to EER=2.25% reported already in Table 3. We see that all three anti-models improve the results. The normalizing LM_L^- is the worst, and the position of its minimum EER is very sensitive on optimal tuning of k. On the other hand, "pure" anti-models do well with a stable minimum at $k = 0.3$. The anti-model using *all data* from S_L^- is preferred. The results were verified also with other test segment durations (10s and 3s), another phoneme recognizer (Czech) and different target data (NIST 1996), with the same stable peak at $k = 0.3$.

system	**30s**	**10s**	**3s**
MIT-FUSE	2.8	7.8	20.3
LIMSI-NN	2.7	7.9	**18.3**
BUT-SPDAT	2.4	8.1	19.1
BUT-PRLM-2005	**1.8**	**6.6**	18.8

Table 4. Comparison of systems on NIST 2003 data

6. CONCLUSIONS

Table 4 compares our system to the best published results on NIST 2003 data. MIT system [2] labeled MIT-FUSE was based on merging of output of PPRLM (6 languages from OGI Stories), Gaussian Mixture Model and Support Vector Machine trained on acoustic features. LIMSI-NN system [4] is a PPRLM trained on 3 languages (CallHome – Arabic, SchwitchBoard – English and CallHome – Spanish); it uses phoneme lattices to train and score phonotactic models and neural-net based merging of individual scores. Our system BUT-SPDAT [3] is a PPRLM trained on 4 languages from SpeechDat-East with linear merging of individual scores.

The system described in this paper: BUT-PRLM-2005 includes **one** PRLM system with all contributions from this paper: improved Hungarian phoneme recognizer, back-off with Witten-Bell discounting, lattice training and scoring and anti-models.

We see, that for durations 30s and 10s, our system based on a single phoneme recognizer outperforms sophisticated systems using parallel combination of phonotactic models and acoustic scoring. The system is currently being improved by training other phoneme recognizers (again from our favorite SpeechDat-E database). The resulting PPRLM system, together with our acoustic system reaching also EER of 1.8% [14], will compete in 2005 NIST language recognition evaluations.

7. REFERENCES

[1] M.A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech.," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[2] E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, and D.A. Reynolds, "Acoustic,phonetic,and discriminative approaches to automatic language identification," in *Proc. Eurospeech*, Sept. 2003, pp. 1345–1348.

[3] Pavel Matějka, Petr Schwarz, Jan Černocký, Pavel Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition" in *Proc. Eurospeech*, Sept. 2005.

[4] J.L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phoneme lattices," in *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Sept. 2004, pp. 1283–1286.

[5] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauché, C. Richey, E. Shriberg, K. Sönmez, F. Weng, and J. Zheng, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proceedings NIST Speech Transcription Workshop*, College Park, MD, May 2000.

[6] P. Schwarz, P. Matějka, and J. Černocký, "Recognition of phoneme strings using TRAP technique," in *Proc. Eurospeech*, Geneva, Switzerland, Sept. 2003, pp. 825–828.

[7] H. Hermansky and S. Sharma, "Temporal patterns (TRAPs) in ASR of noisy speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Phoenix, Arizona, Mar. 1999, pp. 2427–2431.

[8] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Proc. International Conference on Text, Speech and Dialogue*, Brno, Czech Republic, Sept. 2004, pp. 465–472.

[9] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit", in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002.

[10] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression", *IEEE Trans. Inform. Th.*, 37(4):1085 1094, 1991.

[11] "SpeechDat-East project," http://www.fee.vutbr.cz/SPEECHDAT-E.

[12] A.F. Martin and M.A. Przybocki, "NIST 2003 language recognition evaluation," in *Proc. Eurospeech*, Sept. 2003, pp. 1341–1344.

[13] "Callfriend corpus, telephone speech of 15 different languages or dialects," http://www.ldc.upenn.edu/Catalog/.

[14] L. Burget, P. Matějka, and J. Černocký, "Discriminative training techniques for acoustic language identification", submitted to ICASSP 2006, Toulouse, France, 2006.