Advances in Phonotactic Language Recognition

Ondřej Glembek, Pavel Matějka, Lukáš Burget, Tomáš Mikolov

Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic

Abstract

This paper summarizes recent advances in PRLM language recognition within the context of the NIST 2007 LR evaluations (LRE). We present a comparison of binary decision tree (BT) vs. N-gram models when adaptation from a universal (background) model (UBM) is used, we introduce *multi-models*—anchor-model-like approach to scoring, and we adopt the framework of intersession variation using factor analysis.

Index Terms: language recognition, phonotactic, binary decision tree

1. Introduction

Phonetic recognition followed by language model (PRLM) is an essential part of most state-of-the-art language recognition systems. Its principle lies in estimating a language model (LM) on top of 1-best phoneme sequences generated by a phoneme recognizer from the training data and applying this model to the sequences generated from the test data.

Traditionally, N-gram LM's have been used in PRLM. They compute the conditional probability of a coming phoneme given the history of $N-1$ previous phonemes. Therefore the LM for a given language can be built by collecting N-gram statistics (counts) from 1-best phoneme strings and computing their conditional probabilities. Furthermore, Gauvain [1] showed, that N-gram statistics can be computed from the N-gram posterior probabilities taken from the phoneme lattices—as well generated by the phoneme recognizer. This way, alternative phoneme recognition paths are taken into account, giving useful information and improving the performance. We will implicitly present our results on these statistics.

Navratil [2, 3] has shown that clustering the N-gram history by using binary decision trees (BT) improves the performance. Growing the tree is based on finding questions about the history, following the maximum entropy reduction (or likelihood increase) criterion. Each of these questions clusters the data into two subsets. The conditional probabilities are then stored in the leaves and are estimated from the clustered data. Two approaches to BT estimation are proposed—building the whole tree for each class in one case, and adapting from a UBM in the other case. We have adopted the latter framework and used it in conjunction with other techniques.

From this perspective, the N-gram LM is a special case of BT where each N-gram is a cluster itself. We will show, that when applying the BT adaptation scheme from the previous paragraph to N-gram LM, the BT clustering effect is suppressed and both models give similar results. We will also present the application of BT smoothing (as used in [3]) to N-gram LMs.

Note, that N-gram probabilities can be used when referring to the BT leaf probabilities further in the text.

Using linear backends to calibrate the model scores has been studied recently. We will study the effect of linear Gaussian backend (LDA, [4]) and linear logistic regression backend (LLR, [5]). In case of LDA, for each class, a single full-covariance Gaussian (with the covariance matrix shared among all classes) is trained using the vector of scores from all models. This corresponds to an affine transformation of the vector of scores). LLR is somewhat constrained (one multiplier per class is used), however it is trained discriminatively, therefore we used both backends in cascade (LDA followed by LLR).

Being inspired by the anchor models [6], we present the technique of multi-models. Here, each class is represented by multiple models trained on different data sets. In our case, instead of training one model on all databases of that language, we cluster these databases according to dialects, and we train separate models on these clusters. The LDA backend is then used to reduce the number of scores to the desired number of classes.

We try to compensate for the inter-session variation by incorporating the factor analysis framework (see [7, 8]). Treating the leaf log-likelihoods as the model parameter space, we look for its subspace, which is language invariant and whose variability is caused by "parasite" factors (e.g. channel). When scoring an utterance, we let the model adapt in this subspace only.

Section 2 gives theoretical background. We begin by explaining the way that N-gram models can be adapted using the BT framework in Section 2.1, we address the issue of model smoothing in Section 2.2, Section 2.3 describes our inter-session variation compensation technique, and *multi-models* are presented in Section 2.4. Experimental setup is described in Section 3 We conclude the paper in Section 5

2. Theoretical background

2.1. Adaptation

When little data is available for building the (sub-)tree, adaptation scheme has been proposed [3]. A UBM is build on separate training set, where large amount of data is available. When adapting a new tree, the UBM structure is copied and the leaf distributions are estimated in the maximum a-posteriori (MAP) framework.

So far, the N-gram LM's have been estimated on training data only, using the maximum likelihood criterion. However, the BT adaptation scheme can be easily adopted to the N-gram LMs:

For each leaf (cluster, N-gram history) l and symbol s, compute the new conditional probability $\hat{P}'(s|l)$ as follows:

$$\hat{P}'(s|l) = \left[b_{s,l} \frac{\#(s|l)}{|l|} + (1 - b_{s,l}) \hat{P}(s|l) \right] / D \tag{1}$$

where

$$b_{s,l} = \frac{\#(s|l)}{\#(s|l) + r} \tag{2}$$

where $\hat{P}(s|l)$ is the original UBM probability of symbol s given leaf l, $\#(s|l)$ stands for counts of symbols s in leaf l, D normalizes the values to probabilities, and r is an empirical value controlling the strength of the update.

2.2. Smoothing

When traversing the BT, each node splits tha data into two subsets, causing data sparsity. Each node in the BT can hold the phoneme distribution before the split. Navratil [3] proposes a smoothing technique, where the smoothed probability of a symbol s in node l $\hat{P}'_{sm}(s|l)$ is given as:

$$\hat{P}'_{sm}(s|l) = b_{s,l}\hat{P}(s|l) + (1 - b_{s,l})\hat{P}_{sm}(s|parent(l)) \quad (3)$$

where $b_{s,l}$ is as in Eq. 2, with r being the smoothing factor. The equation is applied recursively until $parent(l) = root$, when $\hat{P}_{sm}(s|root) = \hat{P}(s|root)$

To be able to use this technique with N-gram LM, we would have to treat it as a $|\mathcal{A}|$-ary tree with depth $N-1$, where questions at level n ask about the nth predictor. $\mathcal{A}$ is a set

In the case of BT's, this approach is beneficial, having the smoothing constant r set to 2. However, when applied to N-gram LM's, the performance generaly degrades.

2.3. Factor Analysis

Inspired by the inter-session variation techniques in acoustic language recognition [7, 8], we have adopted the framework to the phonotactic LR. With the leaf log-likelihoods defining the model parameter space, we search for their subspace which best describes the inter-session variability. In the testing phase, we then let the model adapt to the test utterance in this subspace.

Let us denote the concatenation of leaf log-probabilities as a column super-vector $\mathbf{d}$, and let $\mathbf{n}_a$ be the concatenation of clustered N-gram statistics of the inspected utterance a. The standard way to evaluate the tree score for utterance a is to compute the inner product of $\mathbf{d}$ and $\mathbf{n}_a$:

$$S_a = \mathbf{n}_a^{\mathrm{T}}\mathbf{d} \quad (4)$$

In FA, we define a transform matrix $\mathbf{V}$ with the same number of rows as is the dimensionality of $\mathbf{d}$ and the number of columns corresponding to the desired number of vectors of factors. The vectors of factors are weighted by column vector $\mathbf{x}$, and then added to the original model parameter supervector $\mathbf{d}$. The FA objective function and output score for utterance a are computed as:

$$S_a = \sum_i n_a^i \log \frac{e^{l_a^i}}{\sum_{j \in cluster(i)} e^{l_a^j}}, \quad (5)$$

with

$$l_a^i = d^i + \mathbf{v}^i\mathbf{x}_a \quad (6)$$

where d^i is the ith element of $\mathbf{d}$, $\mathbf{v}^i$ is i-th row of matrix $\mathbf{V}$, $cluster(i)$ corresponds to the leaf to which i belongs, and $\mathbf{x}_a$ is a column vector of weights estimated for each utterance a.

Matrix $\mathbf{V}$ and vector $\mathbf{x}_a$ are estimated numerically using the R-prop algorithm [9] as an alternative to the traditional gradient descend method, where gradient sign is used instead of the gradient value.

We begin by estimating the vector $\mathbf{x}_a$. The gradient is computed as:

$$\frac{\partial}{\partial \mathbf{x}_a} S_a = \sum_i n_a^i \mathbf{v}^i - \sum_i n_a^i \frac{\sum_j \mathbf{v}^j e^{l_a^j}}{\sum_j e^{l_a^j}} \quad (7)$$

$\mathbf{V}$ is estimated by maximizing Eq. 5 summed over a selected sub-set of training utterances A balanced among languages. The gradient for each row i of matrix $\mathbf{V}$ is computed as:

$$\frac{\partial}{\partial \mathbf{v}^i} S = \sum_{a \in \mathrm{A}} n_a^i \mathbf{x}_a^{\mathrm{T}} \left[1 - \frac{e^{l_a^i}}{\sum_j e^{l_a^i}}\right] \quad (8)$$

The scheme for one training iteration of matrix $\mathbf{V}$ comprised 10 sub-iterations of estimating the vector $\mathbf{x}_a$ for each utterance a from training set A, followed by 10 sub-iterations of re-estimating matrix $\mathbf{V}$. After 5 iterations, the recognition performance converged.

When scoring a test utterance $\hat{a}$, vector $\mathbf{x}$ was estimated in 10 iterations, and the score was obtained using Eq. 5.

The critical issue was the initialization of $\mathbf{V}$. We observed that simple principal component analysis framework, as in [8], gave best convergence speed. The weights $\mathbf{x}$ are initialized by a zeros vector.

Empirically, the number of factors for the evaluation was set 4.

2.4. Multi-models

Inspired by the principle of anchor models [6], we model language classes by a combination of "other" language models. Instead of merging all resources (databases) of one language together for UBM adaptation, those resources with large amount of data were "hand-clustered", and a single LM was created for each of these clusters (e.g. 7 LMs for English). Such hand-clustering reflected some specifics such as foreign-accented English, different dialects, etc. A linear backend is used to post-process these individual outputs to come up with one score per language.

Multi-models address the question of tweaking the weights for multiple training sources. The application of LDA backend is nothing but a linear combination of multiple model scores, which is equivalent to giving different weights to different data sources in the 1-model-per-language situation.

3. Experimental setup

The results are reported for the NIST LRE 2007 primary condition, for three tasks reflecting the nominal length of the testing utterances—30, 10, and 3 seconds. As the metrics, the $100 \times C_{avg}$ (see [10] for formulas) is used.

All results are presented for calibrated systems using linear backend (LDA) followed by linear logistic regression [5] (LRR). Tab. 1 shows the impact of using different calibration schemes on one phonotactic system. The backend for each condition was trained on appropriate dev set (see Sec. 3.1).

3.1. Data

The training data for each language were taken from the packages distributed by LDC and ELRA, and the amount for each language ranged from 1.5 to 228 hours. The exact numbers are in [11].

The development data for this evaluation were defined by MIT Lincoln Labs. The data have nominal duration of 3, 10 and 30 seconds and they are based on segments from previous

Tab. 1: *Effect of calibration for the English GMM/HMM phonotactic system on LRE 2007 data with BTs (*$100 \times C_{avg}$*)*

	30	10	3
No backend	9.02	14.21	24.37
LLR	3.96	10.83	22.97
LDA	3.85	10.55	22.58
LDA+LLR	3.54	10.68	22.66

evaluations plus additional segments extracted from longer files from training databases (which were not included in the training set).

As mentioned before, the test set was the 2007 evaluation data.

3.2. Phone recognizers

The phonotactic systems were based on 3 phoneme recognizers: two left-context/right-context hybrid ANN/HMM [12, 13], and one based on GMM/HMM context dependent models. See Tab. 2 for the comparison.

Tab. 2: *Different phoneme recognizers as tokenizers for phonotactic approach on LRE 2007 data with BTs — English GMM/HMM, Hungarian Hybrid, Russian Hybrid (*$100 \times C_{avg}$*)*

	30	10	3
EN_Tree	3.54	10.68	22.66
HU_Tree	5.58	11.54	23.45
RU_Tree	6.31	12.99	24.51

3.2.1. Hybrid phoneme recognizers

The phoneme recognizer is based on hybrid ANN/HMM approach, where artificial neural networks (ANN) are used to estimate posterior probabilities of phonemes from Mel filter bank log energies using the context of 310ms around the current frame. Hybrid recognizers were trained for Hungarian and Russian on the SpeechDat-E databases. For more details see [12, 13].

3.2.2. GMM/HMM phoneme recognizer

The third phoneme recognizer was based on GMM/HMM context dependent state clustered triphone models, which are trained in similar way as the models used in AMI/AMIDA LVCSR [14]. The models were trained using 2000 hours of English telephone conversational speech data from Fisher, Switchboard and CallHome databases. The features are 13 PLP coefficients augmented with their first, second and third derivatives projected into 39 dimensional space using HLDA transformation. The models are trained discriminatively using MPE criterion [15]. VTLN and MLLR adaptation is used for both training and recognition in SAT fashion. The triphones were used for phoneme recognition with a bi-gram phonotactic model trained on English-only data.

4. Results

4.1. N-gram LM vs. Binary Tree

In our PRLM systems, both the N-gram LMs and BTs were used. In both cases, trigram lattice counts were used. See Tab. 3 for the comparison of these approaches.

The setup for the BT training was setting the minimum data mass criterion to 450, the minimum entropy reduction was set to 0.001, and both the adaptation and smoothing constants r were set to 2 (see [3] for details on these parameters).

Our strategy for scoring the test utterances in the case of N-gram models in the previous years was to choose those N-grams, that appeared in the training data (of any language) certain amount of times to avoid scoring unseen data. Furthermore, the Witten-Bell smoothing was applied (see lines 2 and 3 in Tab. 3). This would however mean, that the set of suitable N-grams for the 2007 evaluations would be very limited as for some languages, very limited data was available. On the other hand, using unseen N-grams would cause severe data sparsity. The adaptation (as described in Sec. 2.1) turned to be a good approach. See Tab. 3 for results. We expected that smoothing the N-grams in the fashion described in Sec. 2.2 would also bring some gain, however no improvement was observed.

Tab. 3: *BT versus different N-gram LM's – Witten Bell smoothed, no smoothing, LM adapted from UBM (*$100 \times C_{avg}$*)*

LRE 2007	30	10	3
HU_Tree	5.58	11.54	23.45
HU_LM Witten Bell (BUT 2005)	5.85	12.63	23.81
HU_LM no smoothing (MIT 2005)	6.30	12.72	25.06
HU_LM MAP adapt from UBM	5.54	11.75	23.54

4.2. Multi-models

We found, that it is beneficial to train multiple models per language, if there is sufficient amount of data for that language. We have chosen those languages, for which large training databases are available. The abbreviation A3E7M5S3G2 denotes the number of models per particular language (e.g. A3 stands for 3 models for Arabic). Better description with exact database division is in Tab. 4. These models could represent different dialects, group of speakers, databases, etc. In our case, the clustering was simply empirical. We end up by producing several scores per language in the scoring phase. Producing final one-score-per-language is again done using LDA backend. Results are presented in Tab. 5.

4.3. Latent Factor Analysis - LFA

Tab. 6 describes influence of LFA to the phonotactic system with Binary decision trees. It mainly helps for 30 second condition. We observed little or no improvement in case of 10 and 3 seconds tasks, where little data for model adaptation was available.

The results with multi-models are similar to the one in Tab. 5, but the system with LFA is more complementary to our other acoustic and phonotactic systems [11].

Tab. 4: *Multiple models distribution with abbreviation A3E7M5S3G2*

Arabic	CallFriend, Fisher, other
English	Foreign accented Eng., Fisher, Callhome, OGI 22, OGI multilang, SRE 2005 - native, SRE 2005 - foreign, CallFriend - south, CallFriend - north
Mandarin	Fisher (HKUST), SRE 2006, CallFriend - mandarin, CallFriend - Taiwan, other
Spanish	CallFriend Caribbean, CallFriend non-Caribbean, other
German	CallFriend, other

Tab. 5: *Multimodels for binary decision tree on LRE 2007 data (*$100 \times C_{avg}$*)*

LRE 2007	30	10	3
HU_Tree	5.58	11.54	23.45
HU_TREE_A3E7M5S3G2	4.54	10.96	23.34

5. Conclusions

In this paper, we have shown, that the data sparsity problem of the N-gram LMs can be solved by using the binary-tree adaptation scheme. Our experiments show, that it is the adaptation from UBM that solves the problem, rather than the BT structural context clustering. We proposed the technique of multi-models and we have shown that it is beneficial to split the training data of each class to several subsets, train separate models on these subsets, and have the backend do their linear combination. We have presented a concept of factor analysis in PRLM and we have shown its gain in LRE.

6. Acknowledgements

This work was partly supported by European projects AMIDA (FP6-033812), Caretaker (FP6-027231) and MOBIO (FP7-214324), by Grant Agency of Czech Republic under project No. 102/08/0707 and by Czech Ministry of Education under project No. MSM0021630528. The hardware used in this work was partially provided by CESNET under project No. 201/2006. Lukáš Burget was supported by Grant Agency of Czech Republic under project No. GP102/06/383.

7. References

[1] J.L. Gauvain, A. Messaoudi, and H. Schwenk. Language recognition using phone lattices. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, 2004.

[2] J. Navrátil. Spoken language recognition - a step towards multilinguality in speech processing. *IEEE Trans. Audio and Speech Processing*, 9(6):678–85, September 2001.

[3] J. Navrátil, Q. Jin, W. Andrews, and J.P. Campbell. Phonetic speaker recognition using maximum-likelihood binary-decision tree models. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, April 2003.

Tab. 6: *Binary decision trees with LFA (*$100 \times C_{avg}$*)*

LRE 2007	30	10	3
HU_Tree	5.58	11.54	23.45
HU_Tree_LFA	5.01	11.45	23.83
HU_TREE_A3E7M5S3G2_LFA	4.52	10.35	23.66

[4] Wade Shen, William Campbell, Terry Gleason, Doug Reynolds, and Elliot Singer. Experiments with lattice-based PPRLM language identification. In *Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop*, pages 1–6, 2006.

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

[6] D. E. Sturim, D. A. Reynolds, E. Singer, and J. P. Campbell. Speaker indexing in large audio databases using anchor models.

[7] P. Kenny. Joint factor analysis of speaker and session variability : Theory and algorithms - technical report CRIM-06/08-13. Montreal, CRIM, 2005, 2005.

[8] L. Burget, P. Matejka, P. Schwarz, O. Glembek, and J. Cernocky. Analysis of feature extraction and channel compensation in GMM speaker recognition system. *IEEE Transactions on Audio, Speech and Language Processing*, 15(7):1979–1986, September 2007.

[9] M. Riedmiller and H. Braun. RPROP – a fast adaptive learning algorithm, 1992.

[10] The 2007 NIST language recognition evaluation plan (lre07): http://www.nist.gov/speech/tests/lang/2007/lre07evalplan-v8b.pdf.

[11] P. Matejka and et al. BUT language recognition system for NIST 2007 evaluation. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, Submited to ICSLP 2008.

[12] P. Schwarz, P. Matějka, and J. Černocký. Hierarchical structures of neural networks for phoneme recognition. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 325–328, Toulouse, France, May 2006.

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

[14] T. Hain, V. Wan, L. Burget, M. Karafiát, J. Dines, J. Vepa, G. Garau, and M. Lincoln. The AMI system for the transcription of speech in meetings. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 357–400, Honolulu, Hawaii, USA, October 2007.

[15] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, July 2004.