On the Use of Lattices of Time-Synchronous Cross-Decoder Phone Co-occurrences in a SVM-Phonotactic Language Recognition System

Amparo Varona, Mikel Penagarikano, Luis Javier Rodriguez-Fuentes, German Bordel

GTTS, Department of Electricity and Electronics
University of the Basque Country, Spain
amparo.varona@ehu.es

Abstract

This paper presents a simple approach to phonotactic language recognition which uses Lattices of Time-Synchronous Cross-Decoder Phone Co-occurrences at the frame level. In previous works we have successfully applied cross-decoder information, but using statistics of n-grams extracted from 1-best phone strings. In this work, the method to build and properly use lattices of cross-decoder phone co-occurrences is fully explained and developed. For evaluating the approach, a choice of open software (Brno University of Technology phone decoders, HTK, SRILM, LIBLINEAR and *FoCal*) was used, and experiments were carried out on the 2007 NIST LRE database. The proposed approach outperformed the baseline phonotactic systems both considering n-grams up to n=3 (yielding around 13% relative improvement) and up to n=4 (yielding around 7% relative improvement). In both cases, best results were obtained by considering the m=400 most likely cross-decoder coocurrences: 1.29% EER and $C_{LLR} = 0.203$. The fusion of the baseline system with the proposed approach yielded 1.22% EER and $C_{LLR} = 0.203$ (meaning 18% and 15% relative improvements, respectively) for n=3, and 1.17% EER and $C_{LLR} = 0.197$ (meaning 15% and 10% relative improvements, respectively) for n=4, outperforming state-of-the-art phonotactic systems on the same task.

Index Terms: Phonotactic Language Recognition, Support Vector Machines, Phone Lattices, Cross-Decoder Co-occurrences

1. Introduction

Nowadays, the most common phonotactic approach for Spoken Language Recognition (SLR) tasks uses counts of phone n-grams to build a feature vector which feeds a classifier based on Support Vector Machines (SVM) [1]. Typically, N phone decoders are applied in parallel to the input utterance, yielding N phone decodings. The output of the phone decoder i ($i \in [1, N]$) is scored for each target language j ($j \in [1, L]$), by applying the model $\lambda(i, j)$ (estimated using the outputs of the phone decoder i for a training database, taking j as the target language). System performance can be improved with the use of phone lattices instead of 1-best phone strings [2], since lattices provide richer and more robust information. However, the above described structure defines N independent data processing channels, and no cross-decoder dependencies are exploited for language modeling, information being fused only at the score level. As far as we know, the idea of using phonetic information in the cross-stream (cross-decoder) dimension was first applied for speaker recognition in the Johns Hopkins University (JHU) 2002 Workshop [3], where two decoupled time and cross-stream dimensions were modelled separately and integrated at the score level. Some years later, cross-stream dependencies were also used via multi-string alignments in a language recognition application [4].

In previous works [5][6] we have presented two simple approaches to phonotactic language recognition which, starting from 1-best phone strings, use statistics of cross-decoder phone co-occurrences at the frame level. Let us consider a choice of two decoders A and B. In the first approach, time-synchronous cross-decoder phone co-occurrences are obtained by aligning at the frame level 1-best phone sequences produced by decoders A and B. This implicitly yields a joint phone segmentation and the corresponding sequence of two-phone labels. The second approach considers longer segments, spanning up to n phones (phone n-grams) in the 1-best phone sequences and applies a different way of computing co-occurrence statistics which does not rely on discrete counts, but on a continuous measure of the degree of co-occurrence of segments (phone n-grams).

In this paper, we present the latest developments attained under the two approaches described above but using lattices instead of 1-best decodings to compute statistics of n-grams (up to n=3 and up to n=4) of phone co-occurrences. Systems have been developed by means of open software (BUT phone decoders, HTK, SRILM, LIBLINEAR and *FoCal*) and evaluated on a relevant database (NIST LRE2007).

The rest of the paper is organized as follows. Section 2 presents the main features of the lattice-based phonotactic SLR system used as baseline in this work. Section 3 describes the proposed approach, based on the use of lattices to compute statistics of time-synchronous cross-decoder phone co-occurrences. The experimental setup is briefly described in Section 4. Results obtained in language recognition experiments on the NIST LRE2007 database (pooled for all the target languages) are presented in Section 5. Finally, conclusions and future work are outlined in Section 6.

2. Baseline Phonotactic SLR System

A phonotactic language recognizer based on phone lattices and SVM scoring is used as baseline system. An energy-based voice activity detector is applied in first place, which splits and removes long-duration non-speech segments from the signals.

This work has been supported by the University of the Basque Country under grant GIU10/18, by the Government of the Basque Country under program SAIOTEK (project S-PE10UN87) and by the Spanish MICINN under Plan Nacional de I+D+i (project TIN2009-07446, partially financed by FEDER funds).

Then, the Temporal Patterns Neural Network (TRAPs/NN) phone decoders, developed by the Brno University of Technology (BUT) for Czech (CZ), Hungarian (HU) and Russian (RU) [7], are applied to compute phone lattices. Regarding channel compensation, noise reduction, etc. all the systems presented in this paper rely on the acoustic front-end provided by BUT decoders. BUT decoders have been previously used by other groups (besides BUT [8], the MIT Lincoln Laboratory [9]) as the core elements of their phonotactic language recognizers, with high-accuracy results.

BUT decoders do not generate phone lattices but phoneme posterior probabilities, which are stored and later processed with the HVite decoder from HTK [10]. The *config* file of each decoder must be modified in the following way: *softering_func=gmm_bypass 0 0 0.* BUT decoders take into account three non-phonetic units: *int* (intermittent noise), *pau* (short pause) and *spk* (non-speech speaker noise) along with 42, 58 and 49 phonetic units of Czech, Hungarian and Russian respectively. For each unit, a three-state model is used, so three posterior probabilities per frame are calculated and stored.

Before generating phone lattices, non-phonetic units *int, pau* and *spk* are integrated into a single 9-state model (which hereafter we will call *pau*). After that, the number of units is 43 for Czech, 59 for Hungarian and 50 for Russian. Then, posterior probabilities are used as input to the HVite decoder from HTK to produce phone lattices, which encode multiple hypotheses with acoustic likelihoods. Finally, the *lattice-tool* from *SRILM* [11] is used to produce the expected counts of phone n-grams.

3. Modeling Cross-Decoder Phone Co-occurrences Using Lattices

Let us consider a choice of two decoders A and B from the set of 3 possible decoders (CZ, HU and RU). Lattices of time-synchronous cross-decoder phone co-occurrences can be obtained by composing the posterior probabilities of two phone models i, j (i from decoder A and j from decoder B) at each frame t, thus creating a single unit on the lattice which represents the time-synchronous co-occurrence of both phones.

BUT decoders produce a sequence of numbers representing the posterior probabilities $p_{i,s}^{t}$ for each one of the three states s of each phone i at each frame t, encoded in the following way:

$$x(p_{i,s}^{t}) = \sqrt{-2\log p_{i,s}^{t}} \qquad (1)$$

To combine posterior probabilities of two phones, i from decoder A and j from decoder B, at each state s and each frame t, the following expression can be applied:

$$\begin{aligned} x(p_{ij,s}^{t}) = x(p_{i,s}^{t}, p_{j,s}^{t}) &= \sqrt{-2\log\left(p_{i,s}^{t} p_{j,s}^{t}\right)} \\ &= \sqrt{-2(\log p_{i,s}^{t} + \log p_{j,s}^{t})} \\ &= \sqrt{x^2(p_{i,s}^{t}) + x^2(p_{j,s}^{t})} \qquad (2) \end{aligned}$$

Therefore, the new composite model ij, corresponding to the time-synchronous co-occurrence of phones i and j, has the same number of states than models i and j.

In this work, all the phonetic units of decoder A are combined with all phonetic units of decoder B, resulting composite models of three states. However, the *pau* model, which is always present in both decoders for any choice of A and B, is treated in an special way: a unique composite model pau, pau of nine states is considered. Therefore, sets of posterior probabilities for (42*49)+1= 2059, (42*58)+1= 2437 and (58*49)+1= 2843 units are calculated, for the choices of decoders CZ-HU, CZ-RU and HU-RU, respectively.

Once the new (quite large) sequence of posterior probabilities representing the phone co-occurrences is obtained, the HVite decoder from HTK can be used to get the (quite large too) composite lattice. The symbols associated to the arcs of the lattice represent the co-occurrence of two units, but the computational treatment is exactly the same as in the baseline system.

The generation of such a large lattice is computationally very expensive. However, it can be drastically reduced by taking into account that many of the 2-phone combinations are, in practice, unlikely and can be skipped. To determine which 2-phone combinations could be skipped, the sum of posterior probabilities for each composite unit ij was calculated on the entire training set of LRE 2007 database (see Subsection 4.1 for details) in the following way:

$$p_{ij} = \sum_{s=1}^{S}\sum_{t=1}^{T} p_{ij,s}^{t} \qquad (3)$$

where S is the number of states corresponding to each composite unit ij and T is the total number of frames in the training database. Once p_{ij} has been calculated for all the combinations, a ranked list is created by sorting the values of p_{ij} from highest to lowest. Percentages of accumulated posterior probabilities for the first m units are shown in Table 1. The unit with the highest posterior probability (m=1) is *pau pau* in all cases.

Table 1: Accumulated posterior probabilities p_{ij} (in %) for the m composite units (co-occurrences) with the highest probabilities, for several values of m (including 1 and all).

m	CZ_HU	CZ_RU	HU_RU
1	26.34%	29.37%	21.87%
50	73.86%	73.79%	68.92%
100	82.89%	82.85%	78.95%
200	90.39%	90.26%	87.50%
400	95.81%	95.92%	94.21%
800	98.88%	99.01%	98.33%
All	100%	100%	100%

The 400 most likely units accumulate 95% of the probability, whereas the 800 most likely units accumulate almost 99% of the probability. So, it does not seem necessary to take into account larger sets of units.

4. Experimental Setup

4.1. Train, development and test datasets

Train and development data were limited to those distributed by NIST to all 2007 LRE participants: (1) the Call-Friend Corpus; (2) the OHSU Corpus provided by NIST for LRE05; and (3) the development corpus provided by NIST for the 2007 LRE. For development purposes, 10 conversations per language were randomly selected, the remaining conversations being used for training. Each development conversation was further split in segments containing 30 seconds of speech. Evaluation was carried out on the 2007 LRE evaluation corpus, specifically on the 30-second, closed-set condition (primary evaluation task).

4.2. Evaluation measures

In this work, systems will be compared in terms of: (1) Equal Error Rate (EER), (2) average cost performance C_{avg} as defined by NIST and (3) the so called C_{LLR} [12], an alternative performance measure used in NIST evaluations. We internally consider C_{LLR} as the most relevant performance indicator, because allows us to evaluate system performance globally by means of an application independent single numerical value. C_{LLR} does not depend on application costs; instead, it depends on the calibration of scores, an important feature of detection systems. C_{LLR} has higher statistical significance than EER, since it is computed starting from verification scores (in contrast to EER, which depends only on Accept/Reject decisions).

4.3. SVM modeling

All SLR systems developed in this work follow a SVM phonotactic approach. SVM vectors consist of counts of features representing the phonotactics of an input utterance: phone n-grams (baseline) and n-grams of phone co-occurrences (co-oc), both weighted as in [13]. A Crammer and Singer solver for multiclass SVMs with linear kernels has been applied [14], by means of LIBLINEAR [15], which has been modified by adding some lines of code to compute regression values. Finally, systems are built by fusing the scores of three calibrated SVM-based phonotactic subsystems. The *FoCal* toolkit is used for calibration and fusion (see [16] for details).

In this work, up to 4-grams have been considered. Therefore, using the raw SVM feature space became unfeasible, due to its huge dimension: the number of possible 4-grams could be up to 59^4 and $(59 * 50)^4$ for the phone lattice system and the proposed phone co-occurrence lattice systems, respectively. A sparse representation was used instead, which involved only the most frequent features. That is, instead of using a full space representation, features were ranked according to their counts on the training dataset using a feature selection algorithm based on frequency [17], and only those with the M highest counts were considered. In this work M=100000. However, given an input utterance, most features have null counts and are not explicitly included in the representation, so the actual size of the SVM feature vector is far less than 100000 (see Table 3 for details).

5. Lattice-based Experimental Results

Phonotactic systems described in Sections 2 and 3 were developed and evaluated on the NIST LRE2007 database. Table 2 shows EER, 100* C_{avg} and C_{LLR} performance in language recognition experiments applying the baseline lattice system and the proposed phone co-occurrence lattice systems (co-oc) using different sets of the most likely cross-decoder co-occurrences (from m=50 to 800). In both approaches, performances using up to 3-grams and up to 4-grams are presented. Note that we call *system* to the fusion of three subsystems, each corresponding to one phone decoder in the baseline system (CZ, HU, RU), and to a 2-decoder choice in the co-ocurrence systems (CZ-HU, CZ-RU, HU-RU).

When up to 3-grams were considered, the baseline performance (1.49% EER and 0.230 C_{LLR}) was surpassed by the use of the m=100 most likely co-occurrences. Best performance was obtained by using m=400, which yielded 1.30% EER and 0.209 C_{LLR} meaning around 13% and 11% relative improvements in terms of EER and C_{LLR}, respectively. Fusions of the baseline and different co-ocurrence systems is also presented in Table 2 . Fusions led to better performances even when using only the m=50 most likely co-occurrences (3% relative improvements). Again, best performance was achieved for m=400: 1.22% EER and $C_{LLR} = 0.203$ (meaning 18% an 15% relative improvements, respectively).

When up to 4-grams were considered, the baseline performance improved: 1.37% EER and 0.219 C_{LLR}, meaning 8% and 8.7% relative improvements with regard to using up to 3-grams. Best results with the proposed phone co-occurrence approach were also obtained for m=400: 1.29% EER and 0.203 C_{LLR}, almost the same result as using up to 3-grams. The best fusion yielded 1.17% EER and $C_{LLR} = 0.197$ (meaning 15% an 10% relative improvements), better performance (on the same task) than that reported by state-of-the-art phonotactic systems [13][18], thus supporting the use of cross-decoder dependencies for language recognition.

Table 2: Performance (EER, $100 * C_{avg}$ and C_{LLR}) for: (1) the baseline lattice system; (2, 3, 4, 5, 6) the proposed phone co-occurrence lattice systems (co-oc) when the m most likely co-ocurrences (m from 50 to 800) were considered; and the fusion of the baseline and the co-occurrence systems. Results when using up to 3-grams and up to 4-grams are shown.

n	System	EER	$100 * C_{avg}$	C_{LLR}
3	(1) Baseline	1.49 %	1.54	0.239
	(2) 50 co-oc	1.71%	1.68	0.288
	(3) 100 co-oc	1.28%	1.48	0.241
	(4) 200 co-oc	1.31%	1.44	0.224
	(5) 400 co-oc	1.30%	1.37	0.209
	(6) 800 co-oc	1.45%	1.35	0.221
	(1) + (2)	1.46%	1.46	0.232
	(1) + (3)	1.34%	1.43	0.224
	(1) + (4)	1.21%	1.23	0.210
	(1) + (5)	1.22%	1.34	0,203
	(1) + (6)	1.30%	1.35	0.210
4	(1') Baseline	1.37%	1.46	0.219
	(2') 50 co-oc	1.61%	1.56	0.262
	(3') 100 co-oc	1.41%	1.39	0.239
	(4') 200 co-oc	1.27%	1.45	0.221
	(5') 400 co-oc	1.29%	1.30	0.203
	(6') 800 co-oc	1.43%	1.31	0.219
	(1') + (2')	1.29%	1.25	0.212
	(1') + (3')	1.15%	1.42	0.210
	(1') + (4')	1.17%	1.27	0,204
	(1') + (5')	1.17%	1.15	0,197
	(1') + (6')	1.25%	1.16	0.203

For the sake of completeness, the performance of subsystems (1-decoder configurations for the baseline system and 2-decoder configurations for the m=400 co-occurrence system) is shown in Table 3. Regarding subsystems, note that 2-decoder subsystems performed consistently better than 1-decoder subsystems. Table 3 also shows the average size of the SVM feature vectors under the sparse representation (see Subsection 4.3), computed on the training and test sets for the baseline and the proposed co-occurrence systems.

Note that training and decoding times depend linearly on the actual size of the SVM feature vectors. When up to 3-grams are used, vectors in the training set are 30% more dense for the proposed approach than for the baseline system. However, when up to 4-grams are considered, vectors in the training set are 47% less dense for the proposed approach than for the baseline system. In both cases, vectors in the test set are smaller for the proposed approach, which means that decoding times are also smaller than for the baseline system (specially when using up to 4-grams).

Table 3: Performance (EER and C_{LLR}) and average vector size on the training and test sets for the baseline and the proposed co-occurrence systems (co-oc) when the m=400 most likely co-occurrences are considered

					Average vector size	
n	Subsystem		EER	C_{LLR}	train	test
3	baseline	CZ	3.44 %	0.541	12155	2610
		HU	2.71%	0.403	14166	2830
		RU	3.11%	0.464	13045	2719
	400 cooc	CZ-HU	1.72%	0.273	18222	2340
		CZ-RU	2.19%	0.334	18991	2474
		HU-RU	1.82%	0.277	18873	2432
4	baseline	CZ	2.95%	0.464	35641	5533
		HU	2.02%	0.322	35983	5571
		RU	2.55%	0.408	36919	6068
	400 cooc	CZ-HU	1.97%	0.283	19534	2467
		CZ-RU	2.05%	0.329	20605	2647
		HU-RU	1.84%	0.273	20067	2555

6. Conclusions

In this paper, the latest developments under an approach using lattices of cross-decoder co-occurrences of phone n-grams in SVM-based phonotactic language recognition have been presented and evaluated. The proposed approach relies on the assumption that cross-decoder co-occurrence information is somehow specific to each target language. The approach does not involve significant additional computation with regard to a baseline phonotactic system. It represents just a means to extract more information from existing decodings.

If all the possible co-occurrences were considered, computational costs would make the approach unfeasible. But the number of phone co-occurrences can be drastically reduced by taking into account that many of the 2-phone combinations in lattices are, in practice, unlikely and can be skipped. An exhaustive study considering different number of the m most likely cross-decoder co-occurrences has been made. Best results were obtained considering m=400 (about 16% of all the possible co-occurrences). The proposed phone co-occurrence lattice system outperformed the baseline phone lattice system both considering n-grams up to n=3 (yielding around 13% relative improvement) and up to n=4 (yielding around 7% relative improvement). In both cases, best results were obtained by considering the m=400 most likely cross-decoder coocurrences: 1.29% EER and $C_{LLR} = 0.203$. The fusion of the baseline system with the proposed approach yielded 1.22% EER and $C_{LLR} = 0.203$ (meaning 18% and 15% relative improvements, respectively) for n=3, and 1.17% EER and $C_{LLR} = 0.197$ (meaning 15% and 10% relative improvements, respectively) for n=4. Better performance (on the same task) than that reported for state-of-the-art phonotactic systems, thus supporting the use of cross-decoder dependencies for language recognition.

Future work will focus on increasing the robustness of phonotactic approaches that integrate time and cross-stream dependencies using time-synchronous co-occurrences of phone n-grams for 3-decoder configurations.

7. References

[1] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, “Support vector machines for speaker and language recognition,” *Computer Speech and Language*, vol. 20, no. 2–3, pp. 210–229, 2006.

[2] W. M. Campbell, F. Richardson, and D. A. Reynolds, “Language recognition with word lattices and support vector machines,” in *ICASSP*, Honolulu, 2007, pp. 15–20.

[3] Q. Jin, J. Navratil, D. Reynolds, J. Campbell, W. Andrews, and J. Abramson, “Combining cross-stream and time dimensions in phonetic speaker recognition,” in *Proceedings of ICASSP*, vol. 4, 2003, pp. 800–803.

[4] C. White, I. Shafran, and J.-L. Gauvain, “Discriminative classifiers for language recognition,” in *Proceedings of ICASSP*, 2006, pp. 213–216.

[5] M. Penagarikano, A. Varona, L. Rodriguez-Fuentes, and G. Bordel, “Using cross-decoder co-ocurrences of phone n-grams in svm-phonotactic language recognition,” in *Proceedings of Interspeech*, Japan, 2010, pp. 745–748.

[6] M. Penagarikano, A. Varona, L. J. Rodriguez-Fuentes, and G. Bordel, “Improved modeling of cross-decoder phone co-occurrences in SVM-based phonotactic language recognition,” *IEEE Transactions on Speech and Audio Processing*, p. 16, 2011, (In Press).

[7] P. Schwarz, “Phoneme recognition based on long temporal context,” Ph.D. dissertation, Faculty of Information Technology BUT, http://www.fit.vutbr.cz, Brno, CZ, 2008.

[8] P. Matejka, L. Burget, O. Glembek, P. Schwarz, V. Hubeika, M. Fapso, T. Mikolov, and O. Plchot, “BUT system description for NIST LRE 2007,” in *Proceedings of the 2007 NIST Language Recognition Evaluation Workshop*, Orlando, USA, 2007, pp. 1–5.

[9] P. Torres-Carrasquillo, E. Singer, W. Campbell, T. Gleason, A. McCree, D. Reynolds, F. Richardson, W. Shen, and D. Sturim, “The MITLL NIST LRE 2007 language recognition system,” in *Proceedings of Interspeech*, 2008, pp. 719–722.

[10] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Lui, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge, UK, 2006.

[11] A. Stolcke, “SRILM - an extensible language modeling toolkit,” in *Proceedings of ICSLP*, November 2002, pp. 257–286.

[12] N. Brümmer and J. A. du Preez, “Application-independent evaluation of speaker detection,” *Computer Speech & Language*, vol. 20, no. 2–3, pp. 230–275, 2006.

[13] F. Richardson and W. Campbell, “Language recognition with discriminative keyword selection,” in *Proceedings of ICASSP*, 2008, pp. 4145–4148.

[14] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2002.

[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008, software available at http://www.csie.ntu.edu.tw/c̃jlin/liblinear.

[16] N. Brümmer and D. van Leeuwen, “On calibration of language recognition scores,” in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[17] M. Penagarikano, A. Varona, L. Rodríguez-Fuentes, and G. Bordel, “A dynamic approach to the selection of high-order n-grams in phonotactic language recognition,” in *Proceedings of ICASSP*, Prage, Czech Republic, 2011.

[18] R. Tong, B. Ma, H. Li, and E. S. Chng, “Selecting phonotactic features for language recognition,” in *Proceedings of Interspeech*, September 2010, pp. 737–740.