A DYNAMIC APPROACH TO THE SELECTION OF HIGH ORDER N-GRAMS IN PHONOTACTIC LANGUAGE RECOGNITION

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

GTTS, Department of Electricity and Electronics
University of the Basque Country, Spain

mikel.penagarikano@ehu.es

ABSTRACT

Due to computational bounds, most SVM-based phonotactic language recognition systems consider only low-order n-grams (up to $n = 3$), thus limiting the potential performance of this approach. The huge amount of n-grams for $n \geq 4$ makes it computationally unfeasible even selecting the most frequent n-grams. In this paper, we demonstrate the feasibility and usefulness of using high-order n-grams for $n = 4, 5, 6, 7$ in SVM-based phonotactic language recognition, thanks to a dynamic n-gram selection algorithm. The most frequent n-grams are selected, but computational issues (those regarding memory requirements) are prevented, since counts are periodically updated and only those units with the highest counts are retained for subsequent processing. Systems were built by means of open software (Brno University of Technology phone decoders, *HTK*, *LIBLINEAR* and *FoCal*) and experiments were carried out on the NIST LRE2007 database. Applying the proposed approach, a 1.36% EER was achieved when using up to 4-grams, 1.32% EER when using up to 5-grams (11.2% improvement with regard to using up to 3-grams) and 1.34% EER when using up to 6-grams or 7-grams.

Index Terms— Phonotactic Language Recognition, SVM, high-order n-grams, Feature Selection

1. INTRODUCTION

For Language Recognition (LR) tasks, two main complementary approaches are typically used [1]: *low level* acoustic modeling and *high level* phonotactic modeling. To model the target language, *low level* acoustic systems take information from the spectral characteristics of the audio signal, whereas *high level* phonotactic systems use sequences of phones produced by Parallel Phone Recognizers (PPR).

In this paper, we focus on the currently most common phonotactic approach: counts of phone n-grams are used to build feature vectors which feed a discriminative classifier based on Support Vector Machines (SVM) [2]. In general, N phone decoders are applied to the input utterance, yielding N phone decodings. The output of each decoder i ($i \in [1, N]$) is scored for each target language j ($j \in [1, L]$), by applying the SVM model $\lambda(i, j)$ (estimated using the outputs of the phone decoder i for a training database, taking j as the target language). Scores for the subsystem i are calibrated, typically by means of a Gaussian backend. Finally, $N \times L$ calibrated scores are fused applying linear logistic regression, to get L final scores for which a minimum expected cost Bayes decision is taken, according to application-dependent language priors and costs (see [3] for details).

The performance of each phone recognizer can be increased significantly by computing the statistics from phone lattices instead of 1-best phone strings [4], since lattices provide richer and more robust information. Another way to increase system performance is the use of high-order n-gram counts, which are expected to contain more discriminant (more language-specific) information. However, the number of n-grams grows exponentially as n increases, and SVM parameters need to be estimated on huge amounts of data, or alternatively most vector components would not be robustly estimated (most of them being zero). This leads to issues regarding either the availability of computational resources (specially memory) or the robustness of SVM parameters.

Dimensionality reduction techniques can be applied to get SVM vectors of a reasonable size, such as feature transformation methods (PCA, LDA, etc.) [5] and feature selection methods [6]. Among the latter, two techniques have been successfully applied: (1) feature selection based on frequency (low frequency n-grams are discarded); and (2) discriminative feature selection, which takes into account the rank of feature weights in the SVM vectors (least discriminant n-grams are discarded). In both cases, selection requires building *complete* vectors (i.e. vectors containing all the components). Again, the huge amount of n-grams for $n \geq 4$ makes it computationally unfeasible a brute-force approach to selecting n-grams. To solve this, a kind of suboptimal expansion has been proposed [6]: starting from a relatively small trigram SVM system, a 4-gram SVM system is built

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).

by using an alternating wrapper/filter method. In the wrapper step, the most discriminant/frequent trigrams are selected. Then, in the filter step, the subset of 4-grams is generated by appending/prepending each phone in the phone set to each selected trigram. In [6] the most discriminant features were selected according to their weights in the SVM. The resulting SVMs (including up to 4-grams) yielded a relative improvement of 18% with regard to the baseline SVMs (including up to trigrams). Following the same method, SVMs including counts of up to 5-grams were estimated, but their performance degraded significantly. Recently in [7], high order n-grams (actually, 4-grams) were used (and performance improvements were reported), by applying the alternating wrapper/filter method on the weights of the SVM and the Chi-squared measure. However, as far as we know, no performance improvements have been reported when using 5-grams (or higher order n-grams) in SVM-based phonotactic language recognition.

In this paper, we propose a new n-gram selection algorithm that allows the use of high-order n-grams (for $n = 4, 5, 6, 7$) to improve the performance of a baseline system based on trigram SVMs. The algorithm requires one single parameter: M, the desired number of features, and works by dynamically updating a ranked list of the most frequent units (from unigrams to n-grams), retaining only those units whose counts are higher than a given threshold. Finally, after processing all the training data, the M most frequent units are output.

The rest of the paper is organized as follows. Section 2 presents the main features of the baseline phonotactic language recognition system used in this work. Section 3 describes the proposed dynamic feature selection method. 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 are summarized in Section 6.

2. BASELINE SVM-BASED PHONOTACTIC LANGUAGE RECOGNIZER

In this work a SVM-based phonotactic language recognizer is used as baseline system, and the NIST LRE2007 database is used for development and evaluation. An energy-based voice activity detector is applied in first place, which splits and removes long-duration non-speech segments from the signals. 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) [8], are applied to perform phone tokenization. BUT decoders have been previously used by other groups (besides BUT [9], the MIT Lincoln Laboratory [10]) as the core elements of their phonotactic language recognizers, with high-accuracy results. Before processing phone sequences, non-phonetic units: *int* (intermittent noise), *pau* (short pause) and *spk* (non-speech speaker noise) are mapped to *sil* (silent pause). After mapping, the number of units is 43 for Czech, 59 for Hungarian and 49 for Russian.

BUT recognizers are used along with HTK [11] to produce phone lattices. Lattices encode multiple hypotheses with acoustic likelihoods and are used to produce expected counts of phone n-grams. Each BUT decoder runs its own acoustic front-end, which takes a speech signal as input and gives the lattice decoding as output. Regarding channel compensation, noise reduction, etc. all the systems presented in this paper rely on the acoustic front-end provided by BUT decoders.

In the baseline system, phone lattices are modeled by means of SVM. SVM vectors consist of counts of phone n-grams (up to 3-grams), weighted as proposed in [6]. A Crammer and Singer solver for multiclass SVMs with linear kernels has been applied, by means of LIBLINEAR [12], which has been modified by adding some lines of code to compute regression values. Finally, the baseline system is built by fusing the scores of the three calibrated SVM-based phonotactic subsystems. The *FoCal* toolkit is used for calibration and fusion (see [3] for details).

3. DYNAMIC FEATURE SELECTION

As noted in Section 1, when high-order n-grams are considered, the number of n-grams grows exponentially, leading to huge computational costs and making the baseline SVM approach impracticable. To reduce the dimensionality of the SVM feature vector, feature selection can be applied, but an exhaustive search of the optimal feature set is computationally unfeasible. The wrapper/filter method [6] tries to select the most discriminant $(n-1)$-grams and expands them to get a *suboptimal* subset of n-grams. However, this method has proven useful only for $n = 4$.

In this work, we propose a new feature selection method with the following characteristics:

- Selection is performed in the target feature space, using an estimate of the feature frequency as criterion.
- The algorithm works by periodically updating a ranked list of the most frequent units, so it doesn't need to index all the possible n-grams but just a relatively small subset of them.
- A single parameter is required: M, the total number of units (unigrams + bigrams + ... + n-grams).
- The process involves accumulating counts until their sum is higher than K and updating the ranked list of units by retaining only those counts higher than a given threshold τ. Note that the algorithm depends on two heuristics: K, the updating period (sum of accumulated counts that must be attained before performing a new update); and τ, the threshold applied to retain counts.
- At each update, all the counts lower than τ are implicitly set to zero; this means that the selection process is *suboptimal*, since many counts are discarded.

- The algorithm outputs the M leading items of the ranked list; note that K and τ must be tuned so that enough number of alive counts (at least, M) are kept at each update.

A hash table is used to rank features according to their counts (the table is indexed by features and stores counts). Though counts increase monotonically, the size of the hash table increases at a slower pace as more updates are performed, since less likely units should rarely accumulate counts higher than τ in an updating period K. In any case, K and τ must be tuned with an eye put on trying the size of the hash table not to grow too much. In practice, suitable values for K and τ produce hash tables with final sizes around $10 \times M$, M being the highest size considered for an n-gram order. Given a set of training samples Ω, the dynamic feature selection algorithm can be summarized as follows:

Dynamic feature selection algorithm

table $\leftarrow \emptyset$
t $\leftarrow$ 0
for $X \in \Omega$ **do**
  accumulate_counts(table,X)
  $t \leftarrow t$ + total_counts(X)
  if $t > K$ **then**
    t $\leftarrow$ 0
    update(table,τ)
truncate(table,M)

4. EXPERIMENTAL SETUP

4.1. Train, development and evaluation 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 Equal Error Rate (EER), which, along with DET curves, is the most common way of comparing the performance of language recognition systems, but also in terms of the so called C_{LLR} [13], an alternative performance measure used in NIST evaluations. We internally consider C_{LLR} as the most relevant performance indicator, for three reasons: (1) C_{LLR} allows us to evaluate system performance globally by means of a single numerical value, which is somehow related to the area below the DET curve, provided that scores can be interpreted as log-likelihood ratios; (2) C_{LLR} does not depend on application costs; instead, it depends on the calibration of scores, an important feature of detection systems; and (3) 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).

5. RESULTS

The phonotactic system described in Section 2, based on phone lattices, has been developed and evaluated on the NIST LRE2007 database. Note that we call *system* to the fusion of three subsystems, each corresponding to a TRAPS/NN phone decoder. When a 4-gram SVM system was considered, the total number of features was about 2000000 for each decoder (1895778 for CZ, 2920755 for HU and 2300064 for RU). But, obviously, not all of them appeared in the SVM vectors. We studied the average size of the SVM vector, and found that it was about 70000. This system yielded 1.32% EER and $C_{LLR} = 0.22508$, meaning a relative improvement of 11% and 6%, respectively, compared to the trigram SVM system.

Table 1. EER and C_{LLR} on the NIST LRE07 closed-set evaluation subset of 30-second speech segments, for various 4-gram SVM systems working on feature sets obtained with the *dynamic selection algorithm*. The number of features (M) and the average SVM vector size are shown too. All systems are fusions of three subsystems corresponding to CZ, HU and RU decoders.

n-gram order	M	Average vector size	%EER	$\mathbf{C_{LLR}}$
3	96416	13634	1.4932	0.23949
4	2000000	68627	1.3274	0.22508
	1000000	66871	1.3269	0.22534
	500000	61963	1.3189	0.21874
	200000	49614	1.3417	0.22123
	100000	37635	1.3747	0.21861
	90000	35793	1.4229	0.22048
	80000	33754	1.4334	0.21997
	70000	31477	1.3768	0.22335
	60000	28931	1.3862	0.22197
	50000	26028	1.3536	0.22613
	40000	22689	1.3838	0.22834
	30000	18778	1.3676	0.22810
	20000	14076	1.4109	0.23404
	10000	8178	1.6077	0.25932
	5000	4507	1.6981	0.28028

Taking the 4-gram SVM system as baseline, we applied the proposed *dynamic selection algorithm* for M ranging from 2000000 down to 5000. In this case, we heuristically fixed $K = 10^6$ and $\tau = 10^{-5}$ to ensure that more than 2000000 features were kept at the end of each iteration. Table 1 shows the EER and C_{LLR} performance attained with

SVM systems based on the selected features. Note that, due to local effects around the EER region, the EER shows some *oscillations*. On the other hand, the C_{LLR}, which allows us to evaluate systems globally, reflects no significant loss of performance for $M = 30000$ and higher values. In particular, for $M = 30000$, the average vector size was reduced from 68637 to 18888, still yielding 1.36% EER and $C_{LLR} = 0.2281$ (a relative improvement of 8.5% and 4.6%, respectively, compared to the trigram SVM system).

Two reference M values have been selected taking into account results for $n = 4$ in Table 1: $M = 100000$ provided better performance than the trigram SVM system, with a similar number of features (though 3 times larger vectors on average); on the other hand, $M = 30000$ provided better performance than the trigram SVM system with similar vector sizes on average. Finally, the proposed *dynamic selection algorithm* has been also applied for $n = 5, 6, 7$, using the two reference values of M. Results are shown in Table 2. Note that best performance was obtaind for $n = 5$: 1.3267% EER ($C_{LLR} = 0.2230$) for $M = 100000$ and 1.3576% EER ($C_{LLR} = 0.2261$) for $M = 30000$. Moreover, performance does not degrade when increasing the n-gram order, as it was the case of other selection approaches in the literature. These results prove that the feature selection algorithm proposed in this work is providing suitable and robust sets of features.

Table 2. EER and C_{LLR} on the NIST LRE07 closed-set evaluation subset of 30-second speech segments, for different n-gram SVM systems working on feature sets obtained with the *dynamic selection algorithm* using $M = 100000$ and $M = 30000$. All systems are fusions of three subsystems corresponding to CZ, HU and RU decoders.

n-gram order	M	Average vector size	%EER	C_{LLR}
3	96416	13634	1.4932	0.23949
4	100000	37635	1.3747	0.21861
	30000	18778	1.3676	0.22810
5	100000	41161	1.3267	0.22300
	30000	19195	1.3576	0.22613
6	100000	40823	1.3415	0.22366
	30000	19187	1.3671	0.23007
7	100000	39357	1.3451	0.22152
	30000	19119	1.3987	0.22973

6. CONCLUSIONS

A dynamic feature selection method has been proposed which allows to perform phonotactic SVM-based language recognition with high-order n-grams. Performance improvements with regard to a baseline trigram SVM system have been reported in experiments on the NIST LRE2007 database when applying the proposed algorithm to select the most frequent units up to 4-grams, 5-grams, 6-grams and 7-grams. The best performance was obtaind when selecting the 100000 most frequent units up to 5-grams, which yielded 1.3267% EER (11.2% improvement with regard to using up to 3-grams). We are currently working on the evaluation of smarter selection criteria under this approach.

7. REFERENCES

[1] P.A. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D.A. Reynolds, F. Richardson, and D.E. Sturim, "The MITLL NIST LRE 2009 language recognition system," in *Proc. of ICASSP 2010*, 2010, pp. 4994–4997.

[2] 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.

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

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

[5] Tomas Mikolov, Oldrich Plchot, Ondrej Glembek, Pavel Matejka, Lukas Burget, and Jan Cernocky, "PCA-based feature extraction for phonotactic language recognition," in *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, 2010, pp. 251–255.

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

[7] Rong Tong, Bin Ma, Haizhou Li, and Eng Siong Chng, "Selecting phonotactic features for language recognition," in *Proceedings of Interspeech*, September 2010, pp. 737–740.

[8] Petr Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Faculty of Information Technology BUT, http://www.fit.vutbr.cz, Brno, CZ, 2008.

[9] 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*, 2007, pp. 1–5.

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

[11] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Lui, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge, UK, 2006.

[12] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen 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/∼cjlin/liblinear.

[13] Niko Brümmer and Johan A. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2–3, pp. 230–275, 2006.