Improved Modeling of Cross-Decoder Phone Co-occurrences in SVM-based Phonotactic Language Recognition

Mikel Penagarikano, *Member, IEEE*, Amparo Varona, *Member, IEEE*,
Luis Javier Rodriguez-Fuentes, *Member, IEEE*, and German Bordel, *Member, IEEE*

***Abstract*—Most common approaches to phonotactic language recognition deal with several independent phone decodings. These decodings are processed and scored in a fully uncoupled way, their time alignment (and the information that may be extracted from it) being completely lost. Recently, we have presented two new approaches to phonotactic language recognition which take into account time alignment information, by considering time-synchronous cross-decoder phone co-occurrences. Experiments on the 2007 NIST LRE database demonstrated that using phone co-occurrence statistics could improve the performance of baseline phonotactic recognizers. In this paper, approaches based on time-synchronous cross-decoder phone co-occurrences are further developed and evaluated with regard to a baseline SVM-based phonotactic system, by using: (1) counts of n-grams (up to 4-grams) of phone co-occurrences; and (2) the degree of co-occurrence of phone n-grams (up to 4-grams). To evaluate these approaches, a choice of open software (Brno University of Technology phone decoders, LIBLINEAR and *FoCal*) was used, and experiments were carried out on the 2007 NIST LRE database. The two approaches presented in this paper outperformed the baseline phonotactic system, yielding around 7% relative improvement in terms of C_{LLR}. The fusion of the baseline system with the two proposed approaches yielded 1.83% EER and $C_{LLR} = 0.270$ (meaning 18% relative improvement), the same performance (on the same task) than state-of-the-art phonotactic systems which apply more complex models and techniques, thus supporting the use of cross-decoder dependencies for language recognition[1].**

***Index Terms*—Time-Synchronous Cross-Decoder Phone Co-occurrences, Phonotactic Language Recognition, Support Vector Machines**

I. Introduction

SPOKEN Language Recognition (SLR) refers to the task of recognizing by automatic means the language spoken in an utterance. SLR is needed in many applications, such as multilingual conversational systems [1], spoken language translation [2], multilingual speech recognition [3], spoken document retrieval [4], etc.

The term *spoken language recognition* can be used to describe two possible tasks: (1) *spoken language identification* (SLI), which consists in deciding which language is spoken in an input utterance; and (2) *spoken language verification* (SLV), which consists in deciding whether or not a target language is spoken in an input utterance. In the first case, the task can be further specified as *closed-set identification*, when the language spoken in an input utterance is known to belong to a closed set of languages, or *open-set identification*, when the input utterance may contain any (known or unknown) language. In closed-set identification, the most likely language (according to the available models) is chosen. The same applies for open-set identification, but only if the likelihood of the most likely language exceeds a given threshold; otherwise the language spoken in the input utterance is classified as *unknown*. Finally, language verification can be seen as a special case of open-set identification with just one known class: for each target language, the input utterance is *accepted* only if the likelihood ratio exceeds a given threshold; otherwise, it is *rejected*.

This work focuses on the SLV task as defined for the National Institute of Standards and Technology (NIST) Evaluations [5]: *given a segment of speech and a language of interest (target language), determine whether or not that language is spoken in the segment, based on an automated analysis of the data contained in the segment*. To measure the performance of a spoken language verification system, a set of trials is presented, each trial comprising the following elements: (1) a segment of audio containing speech in a single language; (2) the target language; and (3) the set of non-target languages, that is, those languages that may be spoken in the segment. For each trial, the system must output: (1) a hard decision (yes/no) about whether or not the target language is spoken in the segment; and (2) a score indicating how likely is for the system that the target language is spoken in the segment, the higher the score the greater the confidence that the segment contains the target language.

Most SLR systems can be classified under two main categories [6], [7]: acoustic systems and phonotactic systems. Acoustic systems characterize target languages by means of low-level, usually short-time spectral features, whereas phonotactic systems use higher level features, typically sequences of phones produced by a number of Parallel Phone Recognizers (PPR). Acoustic systems are easy to develop, since they only need speech signals to train acoustic models for the target

Manuscript received August 5, 2010. 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).

M. Penagarikano, A. Varona*, L.J. Rodriguez-Fuentes and G.Bordel are with the Department of Electricity and Electronics, University of the Basque Country, UPV/EHU, 48940, Leioa, Spain. e-mail: mikel.penagarikano@ehu.es, amparo.varona@ehu.es (corresponding author, Phone +34 946012716 Fax +34 946013500), luisjavier.rodriguez@ehu.es, german.bordel@ehu.es.

[1] EDICS: Multilingual Recognition and Identification

languages, but they rely just on acoustic information. On the other hand, phonotactic systems exploit the acoustic, phonetic and phonotactic information contained in phone sequences [8], so they should potentially yield much better performance than acoustic systems. However, they require phonetically transcribed resources to train phone recognizers, and their performance and robustness is highly dependent on the performance and robustness of such recognizers.

Models for target languages in phonotactic systems are built by decoding hundreds or even thousands of training utterances and using the phone-sequence (or phone-lattice) statistics (typically, counts of n-grams) in different ways. Since training data feature a wide range of speakers and diverse linguistic contents, being *language* the common factor, it is expected that phone sequence statistics reflect language-specific characteristics. Additionally, by using parallel phone sequences, which may be providing complementary information, phonotactic systems can potentially exploit such *complementarity*. Note that each phone recognizer handles a different inventory of *sounds* and a different database to train phone models.

The baseline system developed in this work follows one of the most common phonotactic SLR approaches, which uses counts of phone n-grams to build feature vectors that feed a set of discriminative classifiers based on Support Vector Machines (SVM) [9], [10]. In general, N phone decoders are applied in parallel to the input utterance, yielding N phone decodings. The output of each decoder is scored for each target language, by applying a set of SVM models estimated from the outputs of the phone decoder for a training database. This approach (which we call *Phone-SVM*), is reported to perform better [11] than the previously proposed Parallel Phone Recognition followed by Language Modeling (PPRLM) approach (which we call *Phone-LM*) [12].

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. A quite straightforward approach would consist in building a composite feature vector by concatenating the feature vectors corresponding to the N phone decoders, and computing a single score per target language. But this way we would only exploit cross-decoder dependencies among global statistics, time synchronization information being completely lost.

In this work, we start from the hypothesis that using time-synchronous cross-decoder co-occurrences of events (single phones or longer segments spanning several phones) to characterize target languages may improve performance in phonotactic language recognition. Time synchronization information is obtained as a by-product from phone decodings. Storing it explicitly and building models based on that information only represents a slight increase in computational cost compared to the cost of phone decoding.

In a recent work, we have presented a simple approach to phonotactic language recognition which uses statistics of time-synchronous cross-decoder phone co-occurrences at the frame level [13]. In that approach, phone segmentation was extracted as side information from 1-best phone decodings, and allowed us to consider the *simultaneous occurrence* (co-occurrence) of N phone labels (one per decoder) at each frame. This way, a frame-synchronous sequence of multi-phone labels was defined and used for modeling purposes, following either the Phone-LM or the Phone-SVM approaches.

In experiments on the 2007 NIST LRE database, it was shown that fusing baseline phonotactic systems with systems based on time-synchronous cross-decoder phone co-occurrences led to improved performance in all the cases (see [13] for details). However, systems based on time-synchronous cross-decoder phone co-occurrences did not outperform the baseline phonotactic systems. On the other hand, the Phone-LM approach performed better than the Phone-SVM approach, probably due to the fact that only unigram statistics were used in Phone-SVM, whereas up to 4-grams were considered in Phone-LM.

The approach described above was extended in [14], by considering statistics of up to 3-grams (instead of just unigrams) of phone co-occurrences in a SVM classifier. Additionally, a second approach was also introduced in [14], which considered time-synchronous cross-decoder co-occurrences of longer segments, spanning up to three phones (instead of single phones).

In this paper, we present the latest developments attained under both approaches, using statistics of up to 4-grams of phone co-occurrences and statistics of co-occurrences of segments spanning up to 4 phones, respectively, in an SVM-based phonotactic language recognizer. Since the baseline system has been improved with regard to previous works (due to the introduction of SVM weighting), the relative improvements provided by the proposed approaches are smaller than those reported before, but quite remarkable, specially regarding the second approach. As in previous works, systems have been developed by means of open software and evaluated on the 2007 NIST LRE database.

The rest of the paper is organized as follows. Background on spoken language technology, specially that related to phonotactic approaches, including previous work using cross-decoder information to model target languages, is presented in Section II. The baseline system and approaches using statistics of time-synchronous cross-decoder co-occurrences of single phones or segments spanning several phones, are described and formally defined in Section III. Issues regarding the experimental setup (datasets, evaluation measures, phone decoders, etc.) are addressed in Section IV. Section V presents and discusses the results obtained in language recognition experiments on the core task of the 2007 NIST Language Recognition Evaluation, using the baseline system and the two approaches proposed in this work, and compares them with results reported by other authors on the same database. Finally, conclusions and potential lines for future work are outlined in Section VI.

II. Background

The general structure of a SLR system is shown in Figure 1. It involves four stages: (1) extracting features/tokens, (2) applying a classifier which scores feature/token sequences with regard to models of target languages, (3) applying a backend to normalize/calibrate the resulting scores and (4) making a hard decision (which depends on the task).

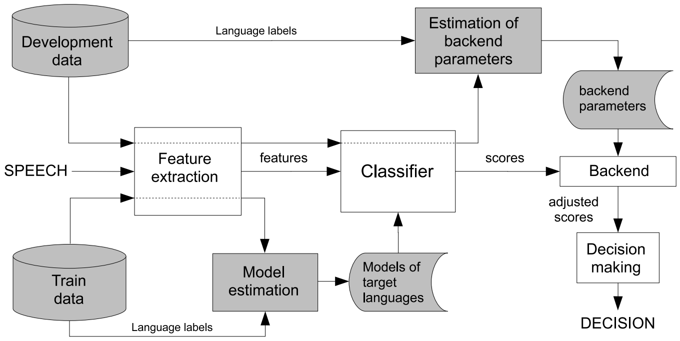

Fig. 1. Structure of a Spoken Language Recognition (SLR) system.

Feature extraction aims to concentrate in few and, as far as possible, independent (that is, uncorrelated) parameters the information relevant to the classification task. Spoken languages can be automatically identified based on features derived from the speech signal at different levels [15]: *sounds* (i.e. short-term spectral patterns), prosodic information, phonotactic information extracted from phone sequences/lattices produced by phone decoders, lexical and syntactic information extracted from word sequences/lattices produced by large vocabulary speech recognizers, etc. In SLR systems based on high-level features, feature extraction involves applying a speech tokenizer (e.g. a phone decoder) which should be trained beforehand. Phonotactic systems assume that phone decoders can deal with acoustic variability, thus phone decodings are assumed to be reliable enough to characterize the spoken language. However, phone decodings may become unreliable if not enough (or unsuitable) data are used to train phone models. It has been shown that using a robust phone decoder is a key issue in the design of high-performance phonotactic SLR systems [16].

Language classifiers capture feature patterns and use them to characterize target languages or to discriminate target languages from each other, depending on the classification approach (generative vs. discriminative).

The backend is introduced to alleviate differences in the volume and type of data used to train language models (which may yield score values in very different ranges). The backend allows to use a single threshold for all the target languages and makes the system work at the desired application point. Finally, if the backend parameters are estimated on development data matching the characteristics of test data, applying the backend may also compensate for a mismatch between train and test conditions (e.g. training on clean speech and using noisy speech to estimate the backend parameters would allow to classify noisy speech).

A. *Spoken language technology: historical perspective*

The availability of high-performance HMM-based phone decoders and n-gram language modeling technology (originally developed for automatic speech recognition applications) allowed the development of the Phone Recognition followed by Language Models (PRLM) approach in mid nineties. PRLM systems used language-specific n-gram models to score the phone sequences produced by a single phone decoder, yielding one score per target language [17], [18]. Parallel PRLM (PPRLM) systems [19], [20] extended the PRLM paradigm by applying several phone decoders (expected to be complementary) and producing one score per target language and phone decoder (a backend was needed in order to get a single score per target language). Around 1996, the PRLM and PPRLM approaches were state-of-the-art SLR technology.

From 1996 to date, the NIST Language Recognition Evaluations (LRE) [21] have provided common data, protocols and performance measures to compare SLR systems from all over the world, supporting and to a large degree leading the development of new methodologies. In 1996, the PRLM and PPRLM approaches yielded the best language recognition performance. Around 2003, systems based on spectral features had reached the performance of phonotactic systems. The GMM-UBM approach, originally developed for speaker recognition applications [22], was applied to language recognition [23] and successfully combined with Shifted Delta Cepstral (SDC) features [24]. Gaussian Mixture Model (GMM) tokenization was proposed as an alternative approach to phone tokenization [25] and Support Vector Machines (SVM) were introduced as classifiers on SDC features using a GLDS kernel [26], [27]. The best single system presented to the 2003 NIST LRE was based on spectral features, and surpassed the performance of PPRLM systems. The best performance was attained by the fusion of one phonotactic and two acoustic systems, using a duration-dependent Gaussian backend [28].

Subsequent evaluations in 2005, 2007 and 2009 dealt with an increasing amount of data and target languages (7, 14 and 23 languages, respectively). Despite this, SLR performance continued to improve due to the use of new and more powerful approaches and the development of fusion and calibration tools, which allowed the easy combination of an arbitrary number of systems (and cross-site collaborations). Also, identifying complementary systems (in other words, systems providing different, uncorrelated information) became an important issue, since they may help global performance through fusion (see e.g. [29]).

Acoustic (spectral-based) and phonotactic (token-based) systems have dominated the SLR technology during the last decade. Acoustic systems have improved due to discriminative GMM training [30], [31], acoustic adaptation (CMLLR) [32], and specially the introduction of GMM supervectors (GSV) as a new means of representing spoken languages for SVM-based discriminative classification [33].

Phonotactic approaches have improved with the use of phone lattices instead of 1-best phone decodings [34]; the use of SVMs to model phonotactics [9], [10] (the same idea had been previously applied to high-level speaker verification [35]); the development of high-quality phone decoders, using large amounts of data [16], more complex structures [36], anti-models [37] and acoustic adaptation [38]; and the efforts to get increasingly less supervised systems, such as ASM tokenization [39][40], where automatically derived universal acoustic units (*Acoustic Segment Models* [41]) were used instead of language-dependent phones.

Finally, increasingly sophisticated fusion and calibration techniques have been applied, including generative Gaussian backends [28], [42] and discriminative logistic regression [43].

B. Current trends

Regarding acoustic systems, authors are exploring the use of universal articulatory features [44], prosodic features [45], speech production traits [46] and other alternative (complementary) features. Also, Joint Factor Analysis (JFA) [47], previously applied to speaker recognition, has been recently applied to spoken language recognition [48].

Regarding token-based approaches, variations on the Phone-SVM approach are being proposed which aim at simplifying, generalizing and reducing supervision in the estimation of models. Recently, Stolcke et al. [49] reported that using a single multilingual phone recognizer (giving universal phonetic coverage) yielded better performance than using various language-specific phone recognizers. Tong has proposed the use of the most discriminant set of phones regarding a language recognition task to build a target-oriented phone tokenizer [50]. Recently, this approach has been refined by extending the front-end with a language model per target language, which takes into account the discriminative ability of phones to define a set of target-aware parallel phone tokenizers [51]. Finally, some efforts are being devoted to deal with high-dimensionality representations in SVM-based phonotactic systems [52], [53], [54].

C. Exploiting cross-decoder information in phonotactic SLR

There is a continued interest in integrating information from various sources at low (feature) and intermediate (model) levels, instead of doing it at the score level (see e.g. [55]). Specifically, the work presented in this paper illustrates a particular way of integrating information from various phone decoders at the feature (token) 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, where two decoupled time and cross-stream dimensions were modelled separately and integrated at the score level [56]. The key idea explored in the JHU 2002 workshop was exploiting high-level features to improve speaker recognition. At that time, the PPRLM methodology had been successfully applied to language recognition, so trying to model speaker-specific pronunciation dynamics by means of token sequences produced by a set of phonetic decoders was almost mandatory. Taking PPRLM as the baseline approach for speaker detection, the hypothesis was made that the statistics of cross-decoder phone co-occurrences may be somehow related to how different speakers realize phonemes. The approach began by aggregating time stamps from all phone sequences into one single segmentation. Then, assuming that the resulting segments were statistically independent, cross-decoder bigram probabilities for all pairs of decoders were computed based on the co-occurrences observed in the alignments for a training corpus. Given an input utterance, decodings were aligned the same way and the resulting phone-pairs were scored by means of previously trained bigram language models (one per target speaker plus and additional background model used for normalization). This approach outperformed the baseline (time dimension) system, yielding less than half the EER in experiments on NIST 2001 SRE Extended Data Task. Moreover, a linear combination of the scores produced by baseline and cross-stream systems further reduced the EER, indicating that they provided complementary information. However, an alternative time-dimension system based on binary decision tree models yielded better results than the cross-stream approach. The same methodology was then applied on the cross-stream dimension (actually time dependencies were also included in the approach), but results were discouraging. According to authors, binary decision trees may be modeling general dependencies across phonetic streams, which strongly contaminate speaker-specific characteristics.

Some years later, cross-stream dependencies were used via multi-string alignments in a language recognition application [11]. Though focused on improving a PPRLM system by applying SVM for both discriminative modeling of phonotactic constraints and discriminative score combination, authors also proposed the use of cross-decoder bigram features, as a way of representing ”*certain sounds that may not be adequately represented by phones in any of the parallel streams*”. So, besides normal *intra-stream* bigrams such as $(a_A(t-1), a_A(t))$, *cross-stream* bigrams of the form $(a_B(t-1), a_A(t))$ were also considered, and the resulting system outperformed the baseline system on most conditions. Note that time-synchronous phone co-occurrences (i.e. simultaneous cross-decoder phone dependencies) were not considered in this work. Since only phone labels were available (without time stamps), the *Clustal W* multiple sequence algorithm [57] was applied to align phone streams, where a similarity weight matrix was used to encode phonetic similarity as defined by experts (taking into account features such as voicing, manner and place of articulation, etc.).

Finally, cross-decoder information has been also exploited to allow cross-lingual phonetic recognition, i.e. applying phone decoders in foreign languages to get phone decodings in a target language. This is accomplished by using context-sensitive probabilistic phone mapping and assuming that the probabilities of observing a symbol and its cross-decoder contexts are independent [58].

III. Improved Modeling of Cross-Decoder Phone Co-occurrences

Figure 2 shows the structure of the baseline phonotactic system used in this work. The input utterance is processed by N parallel phone decoders, which perform all the needed signal processing operations and the computations required to search the 1-best phone hypothesis (according to the available phone models, which are embedded in the decoders), yielding as a by-product the time stamps corresponding to the optimal phone segmentations. Each phone decoder defines an independent Phone-SVM subsystem. The phone sequence produced by each decoder i ($i \in [1, N]$) is scored for each target language j ($j \in [1, L]$), by computing counts of phone n-grams, building a feature vector with them and applying an SVM model $\lambda(i, j)$ estimated from the outputs of the phone decoder i for a training database, taking j as the target language. For ease of presentation, the computation of n-gram counts has been implicitly located inside the SVM modules in Figure 2.

Scores output by each Phone-SVM subsystem are applied a t-norm [59], and calibrated by means of a Gaussian backend. Again, for ease of presentation, both elements have been jointly represented in the backend module in Figure 2. Finally, the resulting calibrated scores are discriminatively fused by means of 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 [43], [60], [61]. More details are given in Section IV.

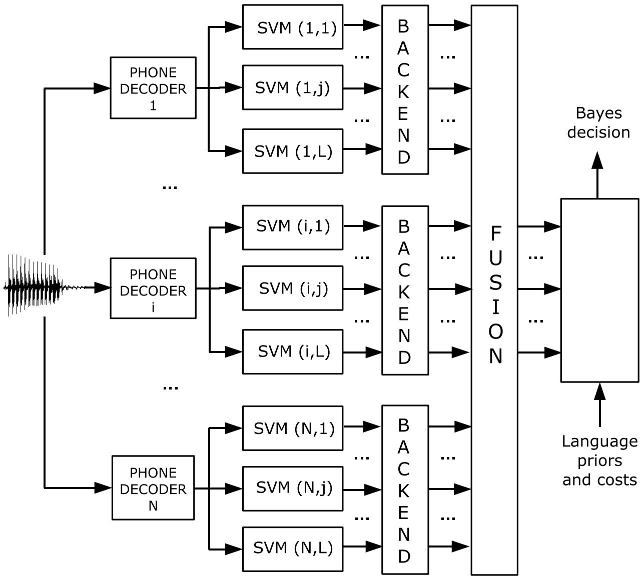

Fig. 2. Baseline SVM-based phonotactic language recognition system.

The two approaches proposed in this paper match the structure of the baseline system described above, except for the way Phone-SVM modules are defined, i.e. the way phone decodings are used to compute features for the SVM. The remaining elements (phone decoders, SVM classification, backend, fusion) are kept unchanged. Let us illustrate this point.

Consider a choice of two decoders A and B from the set of N decoders represented in Figure 2. Time-synchronous cross-decoder phone co-occurrences can be obtained by aligning at the frame level phone sequences produced by decoders A and B. This implicitly yields a joint phone segmentation and (after compacting repeated labels into one single label) the corresponding sequence of two-phone labels. This sequence can be processed and modelled exactly the same way as single-phone sequences in the baseline system (see Figure 3). This configuration can be easily extended to any choice of 2 decoders and a whole co-occurrence system can be built by fusing the scores for all the 2-decoder subsystems. This is how the first approach works.

The above described procedure is similar to that proposed in [56]. However, as we explain in Section III-A, instead of defining two decoupled models in time and cross-decoder dimensions, we define and apply an integrated model which uses n-gram counts of time-synchronous cross-decoder multi-phone labels as features for a SVM-based discriminative classifier. This way, time and cross-decoder dimensions can be jointly modelled.

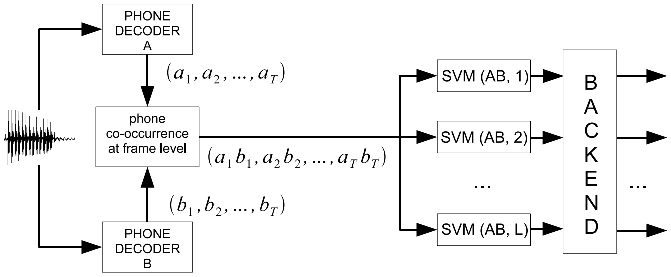

Fig. 3. A 2-decoder phone co-occurrence language recognition subsystem.

The second approach considers longer segments, spanning up to n phones (phone n-grams) in the 1-best phone sequences. This second approach also defines a new 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) from different decoders. This way, we circumvent the border issues (transitional multi-phone labels appearing at phone borders that may distort sequence modeling) observed in the first approach. Details are given in Section III-B.

A. Approach 1: n-gram counts of phone co-occurrences

Let us consider an input utterance X and N phone decoders producing 1-best phone segmentations $S_d(X) = \{s_d(1), \ldots, s_d(T)\}$, $d \in [1, N]$, $s_d(t)$ being the phone label produced by decoder d at frame t, for $t \in [1, T]$. A time-synchronous (frame level) cross-decoder k-phone co-occurrence is defined by the k-tuple $c_\pi(t) = (s_{d_1}(t), s_{d_2}(t), \ldots, s_{d_k}(t))$, $\pi = (d_1, d_2, \ldots, d_k)$ being a choice of k decoders, with $k \in [2, N]$. A sequence of 3-phone co-occurrences (corresponding to 3 decoders) is depicted in Figure 4. Note that a sequence of k-phone co-occurrences $C_\pi = \{c_\pi(1), c_\pi(2), \ldots, c_\pi(T)\}$ includes information from both time and cross-stream dimensions.

We make the assumption that sequences of time-synchronous cross-decoder k-phone co-occurrences are somehow language-specific. So, a language recognition system could be built by counting such events for a training database and estimating SVM-based language models, which should be able to discriminate target languages.

Note that for N decoders, $N!/k!(N-k)!$ of such systems can be defined, applied on an independent way and their scores fused to get a full time-synchronous cross-decoder phone co-occurrence language recognition system. In this work time-synchronous cross-decoder phone co-occurrences are considered only for $k = 2$ and $k = 3$ decoders.

This approach aimed to model cross-decoder segmental (phone-level) dependencies, not cross-decoder frame-level dependencies. The use of frame-level multi-phone labels was motivated just by the need to synchronize phone decodings

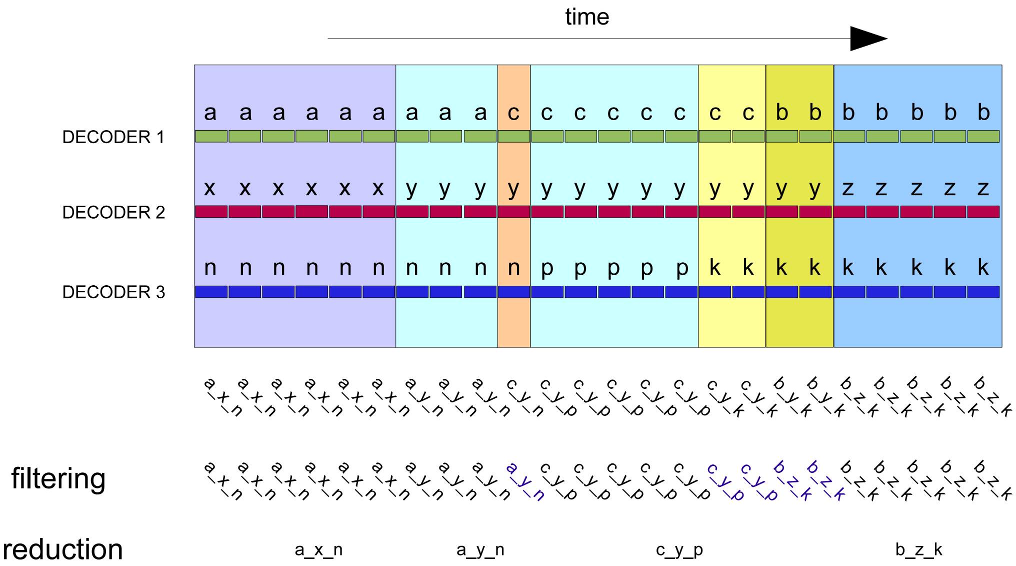
time
DECODER 1
a a a a a a a a a c c c c c c c c b b b b b b b
DECODER 2
x x x x x x y y y y y y y y y y y y y z z z z z
DECODER 3
n n n n n n n n n n p p p p p k k k k k k k k k
a_x_n a_x_n a_x_n a_x_n a_x_n a_x_n a_y_n a_y_n a_y_n c_y_n c_y_p c_y_p c_y_p c_y_p c_y_p c_y_k c_y_k b_y_k b_y_k b_z_k b_z_k b_z_k b_z_k b_z_k
filtering
a_x_n a_x_n a_x_n a_x_n a_x_n a_x_n a_y_n a_y_n a_y_n a_y_n c_y_p c_y_p c_y_p c_y_p c_y_p c_y_p c_y_p b_z_k b_z_k b_z_k b_z_k b_z_k b_z_k b_z_k
reduction
a_x_n
a_y_n
c_y_p
b_z_k

Fig. 4. Approach 1 (3-decoder configuration): (1) time-synchronous cross-decoder phone co-occurrence labels are built by concatenating phone labels from different decoders on a frame-by-frame basis; (2) to handle transitional segments, a mode filter is iteratively applied (until convergence) on a sliding window of 7 frames centered on the analyzed frame; and (3) repeated multi-phone labels are reduced to a single label.

each other. A sort of segmental representation can be recovered by reducing each sequence of repeated multi-phone labels to a single label. However, when analyzing frame-level sequences, two types of segments can be identified: (1) *stationary segments*, corresponding to relatively long portions of speech for which decoders keep the same labels; and (2) *transitional segments*, appearing at phone borders, resulting from the fact that each decoder detects phone transitions at different points (see an example in Figure 4). We hypothesized that cross-decoder phone co-occurrences corresponding to transitional segments reflected random variations in the way each decoder determined phone borders and could greatly distort language models. So, before reducing repeated labels in stationary segments to a single label, transitional segments were filtered out. Details are given in Section IV-D.

After filtering transitional segments and reducing stationary segments, the resulting sequences of multi-phone labels (representing time-synchronous cross-decoder phone co-occurrences) were used to compute n-gram statistics and build feature vectors, which were applied either to estimate SVM parameters or to score an input signal with regard to SVM-based language models (exactly the same way as for sequences of single-phone labels in the baseline system).

B. Approach 2: degree of co-occurrence of phone n-grams

The development of a second approach was motivated by the border issues described above. In Approach 1, co-occurrence information and sequence information were extracted in first and second place, respectively. In between, transitional segments were filtered out, since they were assumed to introduce noise. However, segments considered transitional may actually convey important (discriminative) information. In Approach 2, sequence information is extracted in first place, by considering segments spanning up to n phones (phone n-grams) in the 1-best phone decodings, and co-occurrence information is extracted in second place, by computing the degree of time-synchronous cross-decoder co-occurrence for such segments.

To compute the degree of co-occurrence for any combination of k segments (each coming from a different decoder), we add the counts corresponding to frames in the input utterance where those segments actually overlap. The count assigned to each frame will depend on the length of the segments and on the number of different combinations of k segments overlapping at that frame. We consider time-synchronous cross-decoder co-occurrences only for segments spanning the same number of phones. Co-occurrence information for segments of different length (unigrams co-occurring with bigrams, bigrams with trigrams, etc.) is not used in this work. Note that for each decoder, up to n phone n-grams overlap at each frame t, which means that up to n^k combinations of phone n-grams can co-occur at each frame for a choice of k decoders.

The key points of count computation are: (1) each phone n-gram is counted once for each decoder, so its count is distributed among all the frames it spans; and (2) the contribution corresponding to a given phone n-gram at a given frame for a given decoder is distributed among all the combinations of phone n-grams at that frame for the remaining decoders.

In order to give a formal specification of the computation of the degree of co-occurrence, we first provide some definitions. We recommend to check the example in Figure 5, which is briefly analyzed at the end of this section, to better understand these definitions.

Let $\Gamma_d^{(n)}(t)$ be the set of n-grams (segments spanning n phones) overlapping at frame t in decoding d. Let $w_d^{(n)} \in \Gamma_d^{(n)}(t)$ be one of such n-grams and $len(w_d^{(n)}, t)$ the number of frames it spans. Note that $|\Gamma_d^{(n)}(t)| = n$ for all t except for a number of frames at both ends, where $1 \leq |\Gamma_d^{(n)}(t)| < n$.

Let $G_\pi^{(n)}(t)$ be the set of time-synchronous cross-decoder

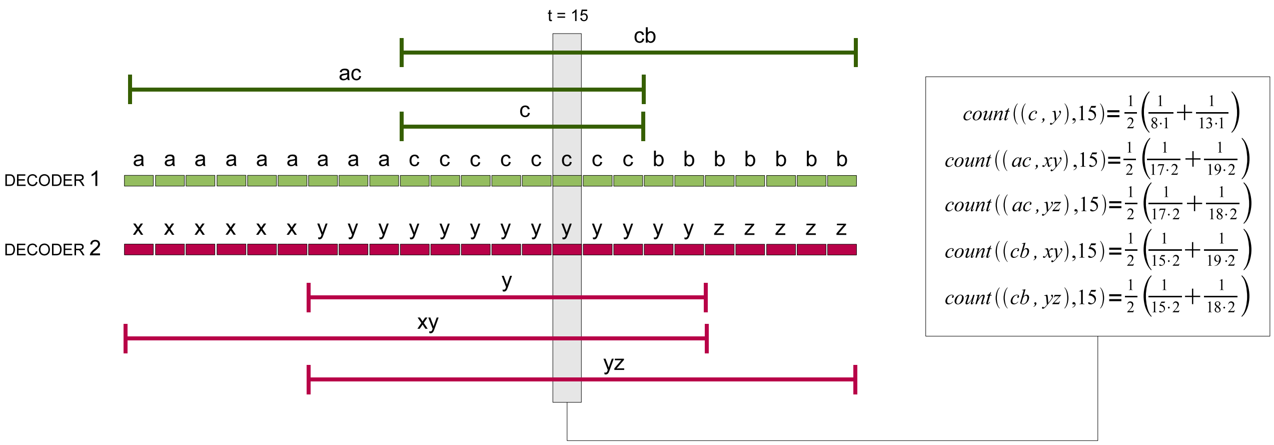
t = 15
cb
ac
c
a a a a a a a a a c c c c c c c c b b b b b b b
DECODER 1
x x x x x x y y y y y y y y y y y y y z z z z z
DECODER 2
y
xy
yz
count((c , y),15)= 1/2 (1/(8·1) + 1/(13·1))
count((ac , xy),15)= 1/2 (1/(17·2) + 1/(19·2))
count((ac , yz),15)= 1/2 (1/(17·2) + 1/(18·2))
count((cb , xy),15)= 1/2 (1/(15·2) + 1/(19·2))
count((cb , yz),15)= 1/2 (1/(15·2) + 1/(18·2))

Fig. 5. Approach 2 (2-decoder configuration, up to bigrams): (1) each phone n-gram is counted once for each decoder, so its count is distributed among all the frames it spans; (2) the contribution corresponding to a given phone n-gram at a given frame for a given decoder is distributed among all the combinations of phone n-grams appearing at that frame for the remaining decoders; and (3) the count corresponding to a cross-decoder co-occurrence of phone n-grams at a given frame is computed as the average contribution of the phone n-grams appearing in the co-occurrence (one per decoder).

co-occurrences of k phone n-grams at frame t, for a choice of decoders $\pi = (d_1, d_2, \ldots, d_k)$, and let $c_\pi^{(n)} = (w_{d_1}^{(n)}, \ldots, w_{d_k}^{(n)}) \in G_\pi^{(n)}(t)$ be one of such co-occurrences. The contribution of the phone n-gram $w_{d_j}^{(n)} \in c_\pi^{(n)}$ to the count of $c_\pi^{(n)}$ at frame t is defined as follows:

$$count(w_{d_j}^{(n)}, t) = \frac{1}{len(w_{d_j}^{(n)}, t) \cdot \prod_{\substack{l=1 \\ l \neq j}}^{k} |\Gamma_{d_l}^{(n)}(t)|} \tag{1}$$

The count corresponding to any co-occurrence $c_\pi^{(n)}$ at frame t is computed as the average contribution of the phone n-grams included in $c_\pi^{(n)}$, only if $c_\pi^{(n)}$ actually appears at frame t:

$$count(c_\pi^{(n)}, t) = \begin{cases} \frac{1}{k} \sum_{j=1}^{k} count(w_{d_j}^{(n)}, t) & \text{if} \quad c_\pi^{(n)} \in G_\pi^{(n)}(t) \\ \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Finally, the *degree of co-occurrence* corresponding to any choice of k cross-decoder phone n-grams $c_\pi^{(n)} = (w_{d_1}^{(n)}, \ldots, w_{d_k}^{(n)})$ is computed by adding the counts for all frames:

$$dc(c_\pi^{(n)}) = \sum_{t=1}^{T} count(c_\pi^{(n)}, t) \tag{3}$$

In practice, the degree of co-occurrence is computed in two passes. The first pass computes and stores $|\Gamma_d^{(n)}(t)|$ and $len(w_d^{(n)}, t)$ for each decoder d and each frame t. Starting from these values, the second pass accumulates the counts of cross-decoder co-occurrences on a frame-by-frame basis, by applying equation 2 for each combination of phone n-grams appearing at frame t.

Let us consider the example in Figure 5, which shows co-occurrences of phone unigrams and phone bigrams for a choice of two decoders. The sets of n-grams appearing at frame $t = 15$ are:

$$\Gamma_1^{(1)}(15) = \{c\} \qquad \Gamma_1^{(2)}(15) = \{ac, cb\}$$
$$\Gamma_2^{(1)}(15) = \{y\} \qquad \Gamma_2^{(2)}(15) = \{xy, yz\}$$

and their lengths:

$$\begin{array}{ll} len(c, 15) = 8 & len(ac, 15) = 17 \\ len(y, 15) = 13 & len(cb, 15) = 15 \\ & len(xy, 15) = 19 \\ & len(yz, 15) = 18 \end{array}$$

Starting from these values and according to equation 2, the counts of co-occurrences of phone n-grams at frame $t = 15$ are computed as follows:

$$\begin{aligned} count((c, y), 15) &= \frac{1}{2} \cdot \left(\frac{1}{8 \cdot 1} + \frac{1}{13 \cdot 1} \right) \\ count((ac, xy), 15) &= \frac{1}{2} \cdot \left(\frac{1}{17 \cdot 2} + \frac{1}{19 \cdot 2} \right) \\ count((ac, yz), 15) &= \frac{1}{2} \cdot \left(\frac{1}{17 \cdot 2} + \frac{1}{18 \cdot 2} \right) \\ count((cb, xy), 15) &= \frac{1}{2} \cdot \left(\frac{1}{15 \cdot 2} + \frac{1}{19 \cdot 2} \right) \\ count((cb, yz), 15) &= \frac{1}{2} \cdot \left(\frac{1}{15 \cdot 2} + \frac{1}{18 \cdot 2} \right) \end{aligned}$$

Note that, in this approach, SVM feature vectors do not contain the statistics (e.g. n-gram counts) of a sequence of labels (as in Approach 1), but a joint distribution of time-synchronous co-occurrence counts for cross-decoder combinations of phone unigrams, phone bigrams, phone trigrams, etc. Since storing and using this information for all the possible cross-decoder combinations of phone n-grams is computationally unfeasible, only those combinations yielding the highest counts on a training database are used to estimate the SVM and to score input utterances (see Section IV-E for details).

IV. Experimental Setup

In this Section we provide details about the task, datasets and measures used to evaluate the proposed approaches, and about the implementation of SLR systems (all following the Phone-SVM modeling approach): phone decoders used as front-end, how the lack of synchronization in phone decodings was handled to get time-synchronous cross-decoder features, SVM feature representation and modeling decisions, SVM feature weighting and score calibration and fusion. In some cases, parameters were tuned heuristically, by choosing those values yielding best results in preliminary experiments on the development dataset (which are presented too).

A. Train, development and evaluation datasets

Experiments have been carried out on the 2007 NIST Language Recognition Evaluation (LRE) database [5]. The 2007 NIST LRE defined a spoken language recognition task for conversational speech across telephone channels, involving 14 target languages: Arabic, Bengali, Chinese (Cantonese, Mainland, Taiwan, Min and Wu), English, (American and Indian), Hindustani (Hindi and Urdu), Spanish (Caribbean and non-Caribbean), Farsi, German, Japanese, Korean, Russian, Tamil, Thai and Vietnamese. Some languages featured various dialects or accents (shown above in parentheses). The test set was split into three subsets, each including 2158 segments, according to their nominal duration: 30, 10 and 3 seconds, respectively. Results reported in this paper have been computed, except where noted, on the subset of 30-second speech segments for the closed-set condition, which was the primary task in the 2007 NIST LRE.

Train and development data were limited to those distributed by NIST to all 2007 LRE participants: (1) the Call-Friend Corpus[2]; (2) the OHSU Corpus provided by NIST for the 2005 LRE[3]; and (3) the development corpus provided by NIST for the 2007 LRE[4]. Table I summarizes the languages/dialects included in these corpora. For development purposes, 10 conversations per language were randomly selected, the remaining conversations being used for training. Development conversations were further divided into segments, each containing 30 seconds of speech (see Table II for more details).

TABLE I
LANGUAGES/DIALECTS IN THE TRAINING AND DEVELOPMENT DATASETS FOR THE 2007 NIST LRE.

Data	Languages
CallFriend	English, (Southern, non-Southern), Mandarin (Mainland, Taiwan), Korean, Japanese, Vietnamese, Hindi, French, Arabic, Farsi, German, Tamil, Spanish (Caribbean, non-Caribbean)
OHSU 2005	English, (American, Indian),Hindi, Japanese, Korean, Mandarin (Mainland, Taiwan), Tamil, Spanish, German
LRE07 Dev	Arabic, Bengali, Chinese (Min, Wu, Cantonese), Russian, Thai, Urdu

TABLE II
2007 NIST LRE CORE CONDITION: TRAINING DATA (HOURS), DEVELOPMENT AND EVALUATION DATA (NUMBER OF 30-SECOND SPEECH SEGMENTS), DISAGGREGATED FOR TARGET LANGUAGES.

Language	Training (hours)	Development (#segments)	Evaluation (#segments)
Arabic	2894	179	80
Bengali	277	76	80
Chinese	9149	567	398
English	7909	288	240
Farsi	2544	225	80
German	3139	173	80
Industani	3543	243	240
Japanese	4354	141	80
Korean	4010	150	80
Russian	277	66	160
Spanish	6460	531	240
Tamil	3202	165	160
Thai	277	64	80
Vietnamese	2570	205	160
TOTAL	50605	3073	2158

B. Evaluation measures

In spoken language verification tasks two types of errors are considered: (1) *misses*, those for which the correct answer is *Accept* but the system says *Reject*; and (2) *false alarms*, those for which the correct answer is *Reject* but the system says *Accept*. Therefore, for any test condition the corresponding error rates can be computed as the fraction of target trials that are rejected (*miss error rate*, P_{miss}) and the fraction of impostor trials that are accepted (*false alarm error rate*, P_{fa}), and suitable cost functions can be defined as combinations of these basic error rates.

1) Graphical evaluation: Detection Error Tradeoff (DET) curves [62] provide a straightforward way of comparing global performance of different systems for a given test condition. A DET curve is generated by computing P_{miss} and P_{fa} for a wide range of operation points (thresholds), based on the scores yielded by the analyzed system for a given test set. DET curves are used in NIST evaluations to support system performance comparisons. In this work, DET curves were generated by means of NIST software.

2) Equal Error Rates: The most common performance measure is the Equal Error Rate (EER), which reports system performance when at the operation point for which the false alarm error rate (P_{fa}) is equal to the miss error rate (P_{miss}). EER is a very simple measure, useful in many contexts, but it does not allow to compare the global performance of two systems.

3) Log-Likelihood Ratio Average Cost (C_{LLR}): When scores represent (or can be interpreted) as log-likelihood ratios, it is possible to evaluate systems also in terms of the so called C_{LLR} [63], which is used as an alternative performance measure 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). Let us now recall how C_{LLR} is computed.

Let $LR(X, i)$ be the *likelihood ratio* corresponding to

[2] See http://www.ldc.upenn.edu/.

[3] OHSU Corpora, http://www.ohsu.edu/.

[4] See http://www.nist.gov/speech/tests/lre/2007/.

segment X and target language i. The likelihood ratio can be expressed in terms of the conditional probabilities of X with regard to the alternative target and non-target hypotheses, as follows:

$$LR(X,i) = \frac{prob(X|i)}{prob(X|\neg i)} \tag{4}$$

Let E be an evaluation set, consisting of the union of L disjoint subsets: E_j ($j \in [1,L]$) containing segments with speech in the target language j. Pairwise costs $C_{LLR}(i,j)$, for $i,j \in [1,L]$, are defined as follows:

$$C_{LLR}(i,j) = \begin{cases} \frac{1}{|E_i|} \sum\limits_{X \in E_i} \log_2(1 + LR(X,i)^{-1}) & j = i \\ \frac{1}{|E_j|} \sum\limits_{X \in E_j} \log_2(1 + LR(X,i)) & j \neq i \end{cases} \tag{5}$$

Finally, the average cost C_{LLR} is computed by adding the pairwise costs for all the combinations of target and non-target languages, as follows:

$$C_{LLR} = \frac{1}{L}\sum_{i=1}^{L}\{P_t \cdot C_{LLR}(i,i) + \sum_{\substack{j=1\\ j\neq i}}^{L} P_{nt} \cdot C_{LLR}(i,j)\} \tag{6}$$

where P_t is the prior probability of target languages and $P_{nt} = (1 - P_t)/(L-1)$ is the prior probability of non-target languages.

The cost function C_{LLR} returns an unbounded non-negative value which can be interpreted as information bits, with lower values representing better performance, the value 0 corresponding to a perfect system and the value $\log_2(L)$ corresponding to a system which just relies on (uniform) priors, thus providing no information to decide a trial. To compute C_{LLR}, the *FoCal* toolkit can be used [64]. Further details about the reasons for using and the interpretation of C_{LLR} can be found in [63], [43].

C. Phone decoders

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) [65] are the core elements of all the systems developed in this work. BUT decoders have been previously used by other groups (besides BUT [66], the MIT Lincoln Laboratory [6]) 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* (silence). After mapping, we get phone inventories of size 43 for Czech, 59 for Hungarian and 50 for Russian. Also, before doing phone tokenization, an energy-based voice activity detector is applied to split and remove non-speech segments from the signals. Since each BUT decoder runs an acoustic front-end, it can be seen as a black box which takes a speech signal as input and gives the 1-best phone 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, whose main features are:

- Czech Decoder (CZ) - 8 kHz, trained on the Czech SpeechDat(E) Database, containing 12 hours of speech from 1052 Czech speakers (526 males, 526 females), recorded over the Czech fixed telephone network.
- Hungarian Decoder (HU) - 8 kHz, trained on the Hungarian SpeechDat(E) Database, containing 10 hours of speech from 1000 Hungarian speakers (511 males, 489 females), recorded over the Hungarian fixed telephone network.
- Russian Decoder (RU) - 8 kHz, trained on the Russian SpeechDat(E) Database, containing 18 hours of speech from 2500 Russian speakers (511 males, 489 females), recorded over the Russian fixed telephone network.

D. Filtering cross-decoder synchronization noise

As pointed above, different decoders determine different phone boundaries, which generate what we call *cross-decoder synchronization noise*. The effect of this noise differs for the two approaches proposed in this work, which make use of co-occurrence information.

In Approach 1 (n-gram counts of phone co-occurrences), the synchronization noise generates short transitional co-occurrence segments (see Figure 4). These short segments must be filtered out, and then feature counts can be computed from the reduced (collapsed) sequence of multi-phone (time-synchronous cross-decoder co-occurrence) labels. In this work, filtering is performed by replacing the multi-phone label at each frame by the mode (the label with the largest number of observations) computed on a window of size w around it (applied iteratively until convergence). Table III shows language recognition performance attained with Approach 1 by applying a mode filter for different window sizes. Best results were obtained with a window of size 7, which roughly makes sequences of length shorter than 3 to be *absorbed* by the surrounding sequences (see an example in Figure 4).

TABLE III
NOISE REDUCTION IN APPROACH 1 USING A MODE FILTER: EER AND C_{LLR} FOR DIFFERENT VALUES OF THE WINDOW SIZE (w).

	w	%Unit reduction	%EER	C_{LLR}
$k=2$	1 (none)	0.00	2.58	0.364
	5	27.97	2.16	0.338
	7	38.28	2.23	0.312
	9	42.64	2.25	0.339
	11	50.40	2.74	0.398
$k=3$	1 (none)	0.00	4.43	0.629
	5	40.31	4.50	0.625
	7	46.65	3.98	0.608
	9	54.41	4.43	0.647
	11	60.42	5.15	0.730

In Approach 2 (degree of co-occurrence of phone n-grams), the cross-decoder synchronization noise also generates short transitional co-occurrences of phone n-grams (see Figure 5). The main difference with respect to Approach 1 is that all the observed combinations of segments receive counts, including short transitional (maybe noisy) combinations. By keeping transitional segments, we circumvent the possible lack of information related to deleting such segments. However, short transitional segments get low counts, and their distribution is not expected to depend on the language, whereas stationary segments (corresponding to relatively long portions of speech for which decoders keep the same labels) get higher counts which dominate the feature vectors used to model target languages. Moreover, as we explain in Section IV-E, only the most frequent features (those with the highest counts) will be used for SVM modeling. Therefore, since both the method used to compute feature counts and the way features are used in SVMs are expected to minimize the effect of transitional segments, we did not apply any filtering approach in this case.

E. SVM modeling

All the SLR systems developed in this work follow the Phone-SVM phonotactic approach described above. SVM vectors consist of counts of features representing the phonotactics of an input utterance: phone n-grams (baseline), n-grams of phone co-occurrences (Approach 1) or the degree of co-occurrence of phone n-grams (Approach 2). Given an input utterance X and a feature f, the probability $p(f|X)$ is computed as follows:

$$p(f|X) = \frac{count(f,X)}{\sum_{\forall f'} count(f',X)} \tag{7}$$

where $count(f,X)$ is the count of feature f for the input utterance X. These probabilities are then used to fill a sparse vector with entries $D(f) \cdot p(f|X)$, where $D(f)$ is a weighting function [67] (details are given in Section IV-F).

In all approaches, 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, $(59 * 50)^4$ and $(43 * 59 * 50)^4$ for the baseline system, a system based on n-grams of two-decoder phone co-occurrences and a system based on n-grams of three-decoder phone co-occurrences, respectively. In this work, a sparse representation was used instead, which stored counts only for the most frequent features. That is, instead of using a full space representation, features were ranked according to their counts on a training dataset, and only those with the M highest counts were considered. On a previous work [14], using up to 3-grams on the baseline system, the total number of features with non-null counts in the training dataset was below 100000 (though the number of possible 3-grams could be up to $59^3 = 205379$). Taking that result into account, in this work the parameter M (i.e. the number of features used to perform language recognition) has been heuristically fixed to 200000. However, given an input utterance, most of them have null counts and are not explicitly included in the representation, so the actual size of the SVM feature vector is far less than 200000. Table IV shows the average size of the SVM feature vectors under a sparse representation, computed on the development set, for the baseline and the two proposed approaches. Note that for Approach 1 (n-grams of phone co-occurrences), the average size of the feature vector is very similar to that found for the baseline system, whereas for Approach 2 (co-occurrences of phone n-grams) more dense vectors are obtained (maybe meaning more reliable and informative features). On the other hand, training and decoding times depend linearly on the actual size of SVM feature vectors. So, whereas the computational cost of Approach 1 is similar to that of the baseline system, Approach 2 has twice the cost of the baseline system.

SVM modeling was performed using LIBLINEAR [68], an open source SVM software library for large-scale linear classification that shares many features with the popular SVM library LIBSVM [69]. A Crammer and Singer solver for multiclass SVMs was applied [70], and some minor changes were made to the source code of LIBLINEAR in order to obtain regression values instead of class labels.

TABLE IV
AVERAGE SIZE OF SVM FEATURE VECTORS UNDER A SPARSE REPRESENTATION, COMPUTED ON THE DEVELOPMENT SET, FOR THE BASELINE SYSTEM AND SYSTEMS BASED ON APPROACH 1 AND APPROACH 2.

	CZ	HU	RU
Baseline	726	786	835
	CZ-HU	CZ-RU	HU-RU
Approach 1 (k=2)	731	790	809
Approach 2 (k=2)	1380	1488	1563
	CZ-HU-RU		
Approach 1 (k=3)	744		
Approach 2 (k=3)	1099		

F. SVM Weighting

As noted in [67], a suitable selection of the weight $D(f)$ is critical for good system performance. A typical choice has the following form:

$$D(f) = \min\left(C, \sqrt{\frac{1}{p(f|\mathcal{S})}}\right) \tag{8}$$

where C is a constant and $\mathcal{S}$ is a set of *background* utterances (including utterances for all the classes considered in the application task) used to estimate what could be seen as an *average* feature probability. Note that if $C = 1$ then $D(f) = 1$ and raw feature probabilities are used (without weigthing). On the other hand, if $C = \infty$, then feature probabilities are weighted by the inverse of the square root of the *average* feature probability (as in [67], [52] and [53]).

The constant C can be heuristically optimized by choosing that value yielding the best performance on a development set. Table V shows EER and C_{LLR} using the baseline language recognition system on the development and evaluation sets for different values of C (*maxWeight*). Results confirm that the choice of C is critical to obtain best performance, and that the development set defined for these experiments matches quite well the evaluation set, since the optimal C values for both datasets are very close each other, being slightly lower for evaluation than for development. The optimal C values found for the development dataset were taken as reference to choose slightly lower C values for evaluation. The C values applied for each approach are shown in Table VI.

G. Calibration and fusion

Each Phone-SVM subsystem generates L scores for each input utterance. Each score is then applied a t-norm [59] that is

TABLE V
EER AND C_{LLR} PERFORMANCE OF THE BASELINE SYSTEM FOR DIFFERENT VALUES OF THE MAXIMUM ALLOWED FEATURE WEIGHT, IN LANGUAGE RECOGNITION EXPERIMENTS ON THE DEVELOPMENT AND EVALUATION SETS.

	%EER		C_{LLR}	
maxWeight	devel	eval	devel	eval
50	1.19	3.03	0.171	0.483
100	0.83	2.50	0.127	0.395
200	0.77	2.29	0.111	0.349
300	0.72	2.19	0.108	0.334
400	0.69	2.21	0.106	0.332
500	0.65	2.22	0.105	0.337
700	0.67	2.30	0.106	0.349
1000	0.73	2.45	0.110	0.361
1500	0.77	2.59	0.115	0.378
2000	0.78	2.79	0.119	0.392
3000	0.77	2.73	0.122	0.397

estimated from the other $L-1$ scores. The resulting scores are calibrated by means of a generative Gaussian backend trained on the development data, and the final scores are obtained by fusing the scores of the calibrated SVM-based phonotactic subsystems. Fusion is based on discriminative linear logistic regression, its parameters being estimated on the development dataset too. The *FoCal* toolkit has been used for calibration and fusion (see [43] and [60] for details).

Calibration and fusion optimize the information delivered to the user by the fused system and offer application-independent scores. Well-calibrated and fused scores can be interpreted as proper log-likelihoods and, therefore, be used to make cost-effective Bayes decisions according to application-dependent language priors and costs.

V. RESULTS

Table VI shows EER and C_{LLR} performance in language recognition experiments on the 2007 NIST LRE database using the baseline phonotactic system and the time-synchronous cross-decoder co-occurrence approaches proposed in this work. First of all, note that we call *systems* either to those that, for a given approach, are obtained by fusing subsystems of one or two decoders, or to those working on the whole set of three decoders. For the sake of completeness, the performance of single subsystems and partial fusions is also shown in Table VI, the fusion operation being represented by means of the symbol '+'. Rows corresponding to complete fusions have been shaded. Scores from single subsystems have been also calibrated in order to get comparable performance measures.

Under the baseline approach, SVMs were also trained on mixed sets of features, by concatenating the n-gram counts from two or three decoders in a single vector, thus obtaining four different concatenations: (CZ,HU), (CZ,RU), (HU,RU) and (CZ,HU,RU). Results under this approach were similar to or slightly worse than those obtained by fusing single subsystems, except when using the counts of all subsystems, which yielded slightly lower EER (2.17%) than fusing the corresponding subsystems (2.21%). By inspecting these results, it may seem that no performance gain can be extracted from cross-decoder dependencies in SVM-based phonotactic language recognition. However, gathering feature frequencies from different decoders into a single representation does not provide feature synchronization (i.e. time alignment) information. This is just what time-synchronous cross-decoder co-occurrences provide, and that information will significantly help to improve performance, as is shown below.

TABLE VI
EER AND C_{LLR} PERFORMANCE OF THE BASELINE PHONOTACTIC SYSTEM AND SYSTEMS BASED ON THE TIME-SYNCHRONOUS CROSS-DECODER CO-OCCURRENCE APPROACHES PROPOSED IN THIS WORK (C: MAXIMUM WEIGHT IN SVMS, k: NUMBER OF DECODERS).

	System	%EER	C_{LLR}
Baseline (C=400)	CZ	5.07	0.724
	HU	4.62	0.659
	RU	4.64	0.691
	CZ + HU	2.85	0.417
	CZ + RU	2.91	0.467
	HU + RU	2.53	0.381
	CZ + HU + RU	2.21	0.332
	(CZ,HU)	2.85	0.423
	(CZ,RU)	3.22	0.488
	(HU,RU)	2.56	0.392
	(CZ,HU,RU)	2.17	0.350
Approach 1 (k=2, C=500)	CZ-HU	3.65	0.520
	CZ-RU	3.81	0.588
	HU-RU	3.36	0.484
	CZ-HU + CZ-RU + HU-RU	2.23	0.312
Approach 1 (k=3, C=400)	CZ-HU-RU	3.98	0.608
Approach 2 (k=2, C=1000)	CZ-HU	3.09	0.424
	CZ-RU	3.50	0.514
	HU-RU	2.70	0.399
	CZ-HU + CZ-RU + HU-RU	2.09	0.308
Approach 2 (k=3, C=700)	CZ-HU-RU	3.59	0.510

When considering complete fusions for $k = 2$ decoders, the two approaches proposed in this work outperformed the baseline system in terms of C_{LLR}. The approach 2 yielded better results (2.09% EER, C_{LLR} = 0.308) than the approach 1 (2.23% EER, C_{LLR} = 0.312), the improvement provided by the former being around 5% relative in terms of EER and 7% relative in terms of C_{LLR} with regard to the baseline system.

Going further in analyses, 2-decoder co-occurrence subsystems performed consistently better than single-decoder subsystems; in particular, those based on Approach 2 performed better than those based on Approach 1. This result is quite interesting, since it indicates that co-occurrence information is actually helping language recognition (specially in the way it is conveyed by Approach 2). However, fusions of two single-decoder subsystems performed better than the corresponding 2-decoder co-occurrence subsystems. For instance, the fusion of the HU and RU baseline subsystems yielded 2.53% EER and C_{LLR} = 0.381, whereas the HU-RU co-occurrence subsystem based on Approach 2 yielded 2.70% EER and C_{LLR} = 0.399. This result seems contradictory, since baseline subsystems are based on token sequences in time and co-occurrence subsystems hypothetically convey both time and cross-decoder information. However, the comparison is not fair, because fusion parameters are optimized for class discrimination. In fact, the fusion of the three 2-decoder co-occurrence subsystems performs better than the fusion of the three single-decoder baseline subsystems. This supports the hypothesis that time-synchronous cross-decoder co-occurrences convey useful information for language discrimination.

Under 3-decoder configurations, both approaches showed a poor performance compared to the baseline system (see Figures 6 and 7). We knew that robustness issues could arise from the huge amount of co-occurrences that are theoretically possible when dealing with $k \geq 3$ decoders.

In Approach 1, the number of transitional segments may explode as the number of decoders increases, thus producing noisy sequences of phone co-occurrences. We tried to avoid short segments by means of a mode filter (see Section IV-D), but attending to system performance (3.98% EER, C_{LLR} = 0.608, worse than those of 2-decoder subsystems for the Approach 1), it seems that such a filtering was not enough or not suitable.

In Approach 2, a huge number of combinations of cross-decoder phone n-grams could appear, specially in the case of 3-grams and 4-grams. As noted in Section IV-E, the SVM feature vector may include at most 200000 elements, corresponding to co-occurrences with the highest counts on a training database. This way, we expected to overcome robustness issues. However, attending to system performance (3.59% EER, C_{LLR} = 0.510, similar or worse than that of 2-decoder subsystems for the Approach 2), we conclude that Approach 2 under a 3-decoder configuration is also remarkably affected by synchronization noise, maybe because too many short transitional segments are being used to characterize input utterances.

A lesson learned is that co-occurrence information can be effectively extracted in 2-decoder configurations (less sensitive to robustness issues) and recovered by means of fusion. In any case, we still hope (and will keep trying) to find an exit to the combinatorial dead end intrinsic to cross-decoder approaches, and future work will be partly devoted to that task.

A. Fusing baseline and cross-decoder co-occurrence systems

Attending to results, one may conclude either that cross-decoder information can only provide small performance improvements in SVM-based language recognition, or (more probably) that issues related to cross-decoder co-occurrence sparseness (unreliable estimations, lack of coverage) strongly limit the discriminative power of the proposed approaches. In any case, further improvements can be expected from fusing the baseline system (which focuses on time sequences) and the proposed cross-decoder co-occurrence systems (which convey complementary cross-decoder information).

Table VII shows EER and C_{LLR} performance of several system fusions, including confidence intervals for a confidence level of 99% under a t-test. The t-test was computed by selecting 1000 random segments in the test set for 100 different experiments. This way, reasonably different subsets were used in the experiments and, at the same time, EER and C_{LLR} were estimated with enough accuracy. Note that results shown above in Table VI are different, since they were computed in a single experiment on the whole test set. In any case, they fall within the confidence intervals shown in Table VII.

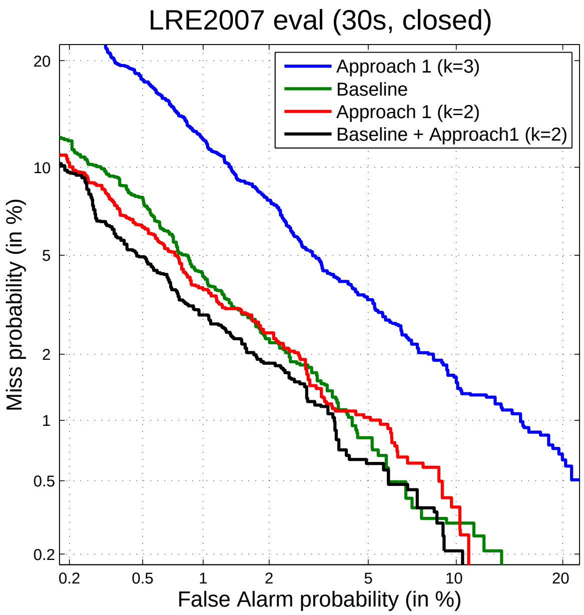

Fig. 6. Pooled DET curves for the baseline phonotactic language recognition system, two systems based on Approach 1 (n-gram counts of cross-decoder phone co-occurrences, for $k = 2$ and $k = 3$ decoders) and the fusion *Baseline + Approach 1* ($k = 2$).

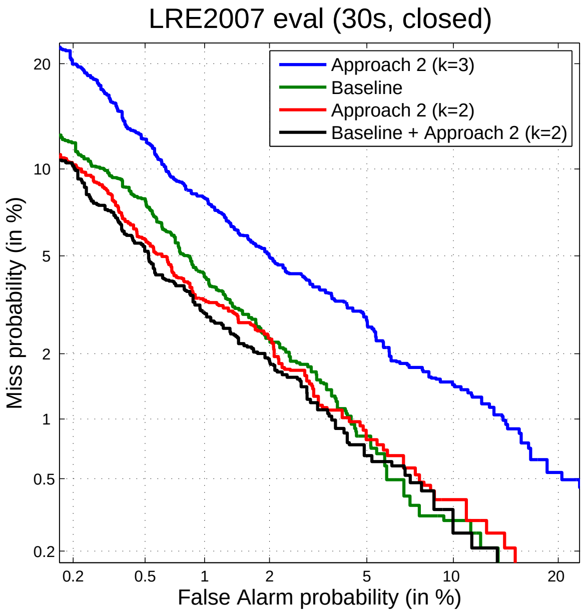

Fig. 7. Pooled DET curves for the baseline phonotactic language recognition system, two systems based on Approach 2 (degree of cross-decoder co-occurrence of phone n-grams, for $k = 2$ and $k = 3$ decoders) and the fusion *Baseline + Approach 2* ($k = 2$).

First, the system (CZ,HU,RU), which uses the concatenation of n-gram counts for the three BUT decoders as SVM representation, slightly improved EER performance with regard to the baseline system, but fusing it with the baseline system did not lead to significantly better results. As noted above, this suggests that using just cross-decoder dependencies between n-gram counts, without time-alignment information, do not help language recognition.

Systems based on 3-decoder co-occurrences did only slightly improve (or nothing at all) the performance of systems based on 2-decoder co-occurrences. This means that they basically model the same cross-decoder information and do not complement each other in any way. This argument is supported by the fact that when fused with the baseline system, 3-decoder co-occurrence systems provided remarkable improvements, leading to 2.05% EER and 2.02% EER (8.48% and 9.82% relative improvement) for approaches 1 and 2, respectively. But adding a 3-decoder co-occurrence system to the fusion of the baseline system and a 2-decoder co-occurrence system did not improve performance for Approach 2, and did slightly improve performance in terms of EER for Approach 1.

TABLE VII

AVERAGE EER AND C_{LLR} PERFORMANCE AND CONFIDENCE INTERVALS (FOR A CONFIDENCE LEVEL OF 99% IN A T-TEST) FOR SEVERAL FUSED SYSTEMS, INVOLVING THE BASELINE SYSTEM AND SYSTEMS BASED ON APPROACH 1 (A1) AND APPROACH 2 (A2).

Single and Fused Systems	%EER	C_{LLR}
Baseline	2.24 (± 0.07)	0.338 (± 0.009)
(CZ,HU,RU)	2.13 (± 0.06)	0.354 (± 0.010)
Baseline + (CZ,HU,RU)	2.21 (± 0.06)	0.338 (± 0.010)
A1 (k=2)	2.24 (± 0.07)	0.308 (± 0.009)
A1 (k=3)	3.99 (± 0.10)	0.608 (± 0.011)
A2 (k=2)	2.11 (± 0.05)	0.308 (± 0.009)
A2 (k=3)	3.56 (± 0.09)	0.510 (± 0.011)
A1 (k=2) + A1 (k=3)	2.20 (± 0.06)	0.306 (± 0.008)
A2 (k=2) + A2 (k=3)	2.09 (± 0.05)	0.301 (± 0.008)
A1 (k=2) + A2 (k=2)	2.02 (± 0.06)	0.289 (± 0.009)
Baseline + A1 (k=2)	1.91(± 0.06)	0.276 (± 0.008)
Baseline + A2 (k=2)	1.94 (± 0.06)	0.287 (± 0.008)
Baseline + A1 (k=3)	2.05 (± 0.07)	0.301 (± 0.009)
Baseline + A2 (k=3)	2.02 (± 0.08)	0.312 (± 0.009)
Baseline + A1 (k=2) + A1 (k=3)	1.88 (± 0.06)	0.272 (± 0.008)
Baseline + A2 (k=2) + A2 (k=3)	1.97 (± 0.07)	0.291 (± 0.010)
Baseline + A1 (k=2) + A2 (k=2)	1.83 (± 0.06)	0.270 (± 0.008)

Separate fusions of the baseline system with systems based on 2-decoder co-occurrences yielded very competitive performances: 1.91% and 1.94% EER (14.73% and 13.39% relative improvement) for approaches 1 and 2, respectively (two first shaded rows in Table VII). Fusing 2-decoder co-occurrence systems for approaches 1 and 2 improved performance by around 4% with regard to the 2-decoder system based on Approach 2. Finally, the best performance (1.83% EER and C_{LLR} = 0.270) was attained by fusing the baseline system with 2-decoder co-occurrence systems for approaches 1 and 2 (last shaded row in Table VII), meaning around 18% relative improvement. This result reveals that time-synchronous cross-decoder co-occurrences convey useful (complementary) information that can effectively help (through discriminative fusion) to improve state-of-the-art phonotactic language recognition. It is worth noting that cross-decoder time alignment information is already there and no additional computations are needed, but only building and applying models based on it.

Additional experiments were carried out to check how the proposed approaches helped state-of-the-art phonotactic language recognition when dealing with shorter (10- and 3-second) speech segments, which is a very interesting task in practical applications. As shown in Table VIII, the performance of Approach 1 was almost identical to that of the baseline system in all conditions, whereas the performance of Approach 2, which was the best for 30-second speech segments, degraded for 10- and 3-second speech segments. This may be due to the fact that Approach 2 is based on frequencies of longer units (phone n-grams), which are scarce and less predictable as less speech is available. Best performance for 10- and 3-second speech segments was found when fusing the baseline system and Approach 1, leading to 12.9% and 6.6% relative improvements, respectively.

TABLE VIII

AVERAGE EER PERFORMANCE AND CONFIDENCE INTERVALS (FOR A CONFIDENCE LEVEL OF 99% IN A T-TEST) ON THE SUBSETS OF 30-, 10- AND 3-SECOND SPEECH SEGMENTS, FOR SEVERAL SINGLE AND FUSED SYSTEMS.

	%EER		
Systems	30s	10s	3s
Baseline	2.24 (± 0.07)	8.22(± 0.14)	20.30 (± 0.16)
A1 (k=2)	2.24 (± 0.07)	8.23 (± 0.12)	20.16 (± 0.17)
A2 (k=2)	2.11 (± 0.05)	10.01 (± 0.12)	22.93 (± 0.17)
Baseline + A1	1.91 (± 0.06)	7.16 (± 0.12)	18.96 (± 0.14)
Baseline + A2	1.94 (± 0.06)	7.48 (± 0.12)	19.36 (± 0.16)
Baseline + A1+ A2	1.83 (± 0.06)	7.21 (± 0.13)	18.97 (± 0.16)

B. Overall Performance Comparison

For the primary task of the 2007 NIST LRE (30-second speech segments, closed-set condition) many results have been published in the literature. Best performance has been reported when fusing several subsystems, specially when acoustic and phonotactic subsystems were fused [6] [7] [66]. Table IX shows the best performance attained in the NIST 2007 LRE [5], and results reported by the Massachusetts Institute of Technology (MIT) [6] and the Brno University of Technology (BUT) [66].

TABLE IX

BENCHMARK ON THE PRIMARY TASK OF THE 2007 NIST LRE (30-SECOND SPEECH SEGMENTS, CLOSED-SET CONDITION) FOR THE FUSION OF ACOUSTIC AND PHONOTACTIC SYSTEMS.

Fused systems	%EER	$100 \cdot C_{avg}$
2007 NIST LRE [5]	–	1.00
seven subsystems [6]	0.93	0.97
three subsystems [66]	–	1.28

Results shown in Table IX were obtained by fusing several acoustic and phonotactic subsystems. However, the approaches presented in this paper are purely phonotactic. Table X shows the most significant references and the best results reported to date using phonotactic approaches on the primary task of the 2007 NIST LRE, corresponding to systems developed by MIT [52], BUT [71] [72], the Institute for Infocomm Research

(IIR) [50][54] and the Laboratoire d'Informatique pour la Mecanique et les Sciences de l'Ingenieur (LIMSI) [73].

TABLE X
BENCHMARK ON THE PRIMARY TASK OF THE 2007 NIST LRE (30-SECOND SPEECH SEGMENTS, CLOSED-SET CONDITION) USING PHONOTACTIC SYSTEMS

Classifier	Model	%EER	$100 \cdot C_{avg}$
LM	4-gram, 1-best [73]	4.9	–
	3-gram, lattices [71], [72]	–	5.54
	4-gram, lattices [73]	2.8	–
Binary Tree	3-gram, lattices [71], [72]	–	4.52
SVM	3-gram, 1-best [50]	4.64	–
	3-gram, lattices [50]	3.54	–
	3-gram, lattices [52]	2.20	–
	4-gram, lattices [54]	1.84	–
	4-gram, lattices [52]	1.80	–

Results in Table X first suggest that applying SVM-based scoring leads to better results than using either n-gram or binary tree-based scoring. Second, that using phone lattices to compute n-gram statistics leads to better results than using 1-best phone sequences. And third, that using up to 4-gram statistics is useful, despite robustness issues related to the high dimension of the feature space. In particular, best results (around 1.80% EER) were attained by systems based on the Phone-SVM approach, using phone lattices and up to 4-gram statistics, and applying discriminative feature selection based on SVM weights and a wrapper/filter method [52][54]. Finally, it must be noted that the phonotactic approaches proposed in this paper, based on 1-best phone sequence statistics and using up to 4-grams but without discriminative feature selection, yielded the same performance than state-of-the-art phonotactic systems. On the other hand, time-synchronous cross-decoder co-occurrence information can be easily extracted in most PPR-based phonotactic systems. So, improved performance at almost no cost can be attained in phonotactic SLR by using features with both time (sequence) and cross-decoder (time-synchronous co-occurrence) information.

C. Discussion

There is an important difference between approaches 1 and 2, which regards how cross-stream and time dimensions are processed. Approach 1 first concentrates on the cross-decoder dimension and then considers the time dimension, but phone sequence modeling is somehow lost in the way. Approach 2 runs the opposite route: it can be seen as a phonotactic system (whose factory equipment includes phone sequence modeling) enhanced with additional modeling of cross-decoder co-occurrences of phone n-grams. This may explain why, when the available amount of speech was large enough (30-second speech segments), the latter provided the best performance among single systems (specially in terms of C_{LLR}), its DET curve being close to that of the optimal fusion (see Figure 7). However, since the baseline system provides phone sequence information partly lost in Approach 1, they complement each other well, and may explain why Approach 2 did not outperform Approach 1 when fused with the baseline system.

VI. CONCLUSIONS AND FUTURE WORK

Two approaches using time-synchronous cross-decoder co-occurrence information in SVM-based phonotactic language recognition have been defined and evaluated (for combinations of $k = 2$ and $k = 3$ decoders): Approach 1 (n-gram counts of phone co-occurrences) and Approach 2 (degree of co-occurrence of phone n-grams). Both approaches rely on the assumption that time-synchronous cross-decoder co-occurrence information is somehow specific to each target language. They do not involve significant additional computation with regard to a baseline phonotactic system, and represent just a means to extract more information from existing decodings.

Systems based on 2-decoder co-occurrences outperformed the baseline system in language recognition experiments on the primary task of the 2007 NIST LRE. The system based on Approach 2 using 2-decoder phone n-grams yielded the best performance among all single systems, with 2.11% EER (above 5% relative improvement with regard to baseline 2.24% EER) and C_{LLR} = 0.308 (above 8% relative improvement with regard to baseline C_{LLR} = 0.338). However, when using 3-decoder configurations, both approaches showed a poor performance compared to the baseline system. This may reveal robustness issues related to: (1) significant differences in the detection of phone boundaries (Approach 1) which make transitional segments to be dominant, thus producing noisy sequences of phone co-occurrences; and (2) a huge number of phone n-gram combinations (Approach 2), whose statistics cannot be robustly estimated.

When considering fusions, best results were attained when combining the baseline system with systems based on 2-decoder co-occurrences, with no significant differences between approaches 1 and 2. The best fused system (Baseline + Approach 1 ($k = 2$) + Approach 2 ($k = 2$)) yielded 1.83% EER and C_{LLR} = 0.270, meaning around 18% and 20% relative improvement, respectively, with regard to the baseline system. Finally, using time-synchronous cross-decoder co-occurrences led to improved performance (by fusing the baseline system with Approach 1) also when applied to short (10- a 3-second) speech segments.

We are currently working on various co-occurrence selection schemes, with the aim to improve performance by using more discriminant features, and on replacing 1-best phone sequences by phone lattices, with the aim to increase the robustness of co-occurrence statistics. Future work will focus on increasing the robustness of phonotactic approaches that integrate time and cross-stream dependencies, specially when using $k \geq 3$ decoders.

REFERENCES

[1] V. W. Zue and J. R. Glass, "Conversational interfaces: Advances and challenges," *Proceedings of the IEEE, Special Issue on Spoken Language Processing*, vol. 88, no. 8, pp. 1166–1180, August 2000.

[2] A. Waibel, P. Geutner, L. M. Tomoyiko, T. Schultz, and M. Woszczyna, "Multilinguality in speech and spoken language systems," *Proceedings of the IEEE, Special Issue on Spoken Language Processing*, vol. 88, no. 8, pp. 1181–1190, August 2000.

[3] B. Ma, C. Guan, H. Li, and C.-H. Lee, "Multilingual speech recognition with language identification," in *Proceedings of ICSLP (Interspeech)*, 2002, pp. 505–508.

[4] N. Bertoldi and M. Federico, "Cross-language spoken document retrieval on the TREC SDR collection," in *Advances in Cross-Language Information Retrieval, Lecture Notes in Computer Science, Volume 2785/2003*. Springer, 2003, pp. 476–481.

[5] A. F. Martin and A. N. Lee, "NIST 2007 Language Recognition Evaluation," in *Proceedings of Odyssey 2008: The Speaker and Language Recognition Workshop*, 2008.

[6] 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.

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

[8] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *The Journal of the Acoustical Society of America*, vol. 62, no. 3, pp. 708–713, September 1977.

[9] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, June 2005, pp. 515–522.

[10] W. M. Campbell, T. Gleason, J. Navratil, D. Reynolds, W. Shen, E. Singer, and P. Torres-Carrasquillo, "Advanced language recognition using cepstra and phonotactics: MITLL system performance on the NIST 2005 language recognition evaluation," in *Proceedings of Odyssey 2006 - The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006.

[11] C. White, I. Shafran, and J.-L. Gauvain, "Discriminative classifiers for language recognition," in *Proc. of IEEE ICASSP*, 2006, pp. 213–216.

[12] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, January 1996.

[13] M. Penagarikano, A. Varona, L. Rodriguez-Fuentes, and G. Bordel, "Using cross-decoder phone coocurrences in phonotactic language recognition," in *Proceedings of IEEE ICASSP*, Dallas, Texas (USA), 2010, pp. 5034–5037.

[14] M. Penagarikano, A. Varona, L. Rodríguez-Fuentes, and G. Bordel, "Improved modeling of cross-decoder phone co-occurrences in SVM-based phonotactic language recognition," in *Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop*, Brno, Czech Republic, 2010.

[15] M. P. Harper and M. Maxwell, *Spoken Language Characterization*. Benesty, Sondhi, Huang (Eds), Springer, 2008, ch. Springer Handbook of Speech Processing, Chapter 40, pp. 797–809.

[16] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Proceedings of Interspeech*, Lisboa, Portugal, September 2005, pp. 2237–2241.

[17] T. J. Hazen and V. W. Zue, "Automatic language identification using a segment-based approach," in *Proceedings of Eurospeech*, Berlin, Germany, September 1993, pp. 1303–1306.

[18] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling," in *Proceedings of IEEE ICASSP*, April 1994, pp. 305–308.

[19] M. A. Zissman, "Language identification using phoneme recognition and phonotactic language modeling," in *Proceedings of IEEE ICASSP*, May 1995, pp. 3503–3506.

[20] Y. Yan and E. Barnard, "An approach to automatic language identification based on language-dependent phone recognition," in *Proceedings of IEEE ICASSP*, May 1995, pp. 3511–3514.

[21] NIST LRE, http://www.itl.nist.gov/iad/mig/tests/lre/.

[22] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[23] E. Wong and S. Sridharan, "Methods to improve Gaussian mixture model based language identification system," in *Proceedings of ICSLP (Interspeech)*, 2002, pp. 93–96.

[24] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller, "Approaches to language identification using Gaussian mixture models and Shifted Delta Cepstral features," in *Proceedings of ICSLP*, 2002, pp. 89–92.

[25] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. R. Deller, "Language identification using Gaussian mixture model tokenization," in *Proceedings of IEEE ICASSP*, vol. I, 2002, pp. 757–760.

[26] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of IEEE ICASSP*, vol. I, 2002, pp. 161–164.

[27] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, "Language recognition with support vector machines," in *Proceedings of Odyssey 2004 - The Speaker and Language Recognition Workshop*, Toledo, Spain, May-June 2004, pp. 41–44.

[28] 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 *Proceedings of Eurospeech (Interspeech)*, Geneva, Switzerland, 2003, pp. 1345–1348.

[29] A. G. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proceedings of Eurospeech (Interspeech)*, Geneva, Switzerland, 2003, pp. 841–844.

[30] Q. Dan and W. Bingxi, "Discriminative training of GMM for language identification," in *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR)*, Tokyo, Japan, April 2003, pp. 67–70 (paper MAP8).

[31] L. Burget, P. Matejka, and J. Cernocky, "Discriminative training techniques for acoustic language identification," in *Proceedings of IEEE ICASSP*, vol. I, Toulouse, France, May 2006, pp. 209–212.

[32] W. Shen and D. Reynolds, "Improved GMM-based language recognition using constrained MLLR transforms," in *Proceedings of IEEE ICASSP*, March 2008, pp. 4149–4152.

[33] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Acoustic language identification using fast discriminative training," in *Proceedings of Interspeech*, Antwerp, Belgium, August 2007, pp. 346–349.

[34] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proceedings of ICSLP*, 2004, pp. 1283–1286.

[35] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "High-level speaker verification with support vector machines," in *Proceedings of IEEE ICASSP*, vol. I, Montreal, Quebec, Canada, May 2004, pp. 73–76.

[36] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Proceedings of IEEE ICASSP*, vol. I, Toulouse, France, May 2006, pp. 325–328.

[37] P. Matejka, P. Schwarz, L. Burget, and J. Cernocky, "Use of anti-models to further improve state-of-the-art PRLM language recognition system," in *Proceedings of IEEE ICASSP*, vol. I, Toulouse, France, May 2006, pp. 197–200.

[38] W. Shen and D. Reynolds, "Improving phonotactic language recognition with acoustic adaptation," in *Proceedings of Interspeech*, Antwerp, Belgium, August 2007, pp. 358–361.

[39] B. Ma, H. Li, and C.-H. Lee, "An acoustic segment modeling approach to automatic language identification," in *Proceedings of Interspeech*, Lisboa, Portugal, September 2005, pp. 2829–2832.

[40] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 271–284, January 2007.

[41] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proceedings of IEEE ICASSP*, vol. I, April 1988, pp. 501–504.

[42] M. F. BenZeghiba, J. L. Gauvain, and L. Lamel, "Language score calibration using adapted Gaussian back-end," in *Proceedings of Interspeech 2009*, Brighton, UK, September 2009, pp. 2191–2194.

[43] 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.

[44] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Proceedings of Interspeech*, Brighton, UK, 2009, pp. 168–171.

[45] R. W. M. Ng, C.-C. Leung, T. Lee, B. Ma, and H. Li, "Prosodic attribute model for spoken language identification," in *Proceedings of IEEE ICASSP*, Dallas, Texas, USA, 2010, pp. 5022–5025.

[46] A. Sangwan, M. Mehrabani, and J. H. L. Hansen, "Automatic language analysis and identification based on speech production knowledge," in *Proc. of IEEE ICASSP*, Dallas, Texas, USA, 2010, pp. 5006–5009.

[47] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Tech. Rep. Technical Report CRIM-06/08-13, 2005, [Online]. Available: http://www.crim.ca/perso/patrick.kenny/.

[48] F. Castaldo, S. Cumani, P. Laface, and D. Colibro, "Language recognition using language factors," in *Proceedings of Interspeech*, Brighton, UK, September 2009, pp. 176–179.

[49] A. Stolcke, M. Akbacak, L. Ferrer, S. Kajarekar, C. Richey, N. Scheffer, and E. Shriberg, "Improving language recognition with multilingual phone recognition and speaker adaptation transforms," in *Proceedings*

of Odyssey 2010 - The Speaker and Language Recognition Workshop, Brno, Czech Republic, 28 June - 1 July 2010, pp. 256–262.

[50] R. Tong, B. Ma, H. Li, and E. S. Chng, "A target-oriented phonotactic front-end for spoken language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 7, pp. 1335–1347, September 2009.

[51] R. Tong, B. Ma, H. Li, E. S. Chng, and K.-A. Lee, "Target-aware language models for spoken language recognition," in *Proceedings of Interspeech*, Brighton, UK, 2009, pp. 200–203.

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

[53] F. Richardson, W. Campbell, and P. A. Torres-Carrasquillo, "Discriminative n-gram selection for dialect recognition," in *Proceedings of Interspeech*, 2009, pp. 192–195.

[54] 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.

[55] R. Tong, B. Ma, D. Zhu, H. Li, and E. S. Chng, "Integrating acoustic, prosodic and phonotactic features for spoken language identification," in *Proceedings of IEEE ICASSP*, vol. I, 2006, pp. 205–208.

[56] Q. Jin, J. Navratil, D. Reynolds, J. Campbell, W. Andrews, and J. Abramson, "Combining cross-stream and time dimensions in phonetic speaker recognition," in *Proc. of IEEE ICASSP*, vol. IV, 2003, pp. 800–803.

[57] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acid Research*, vol. 22, no. 22, pp. 4673–4680, November 1994.

[58] K. C. Sim and H. Li, "Stream-based context-sensitive phone mapping for cross-lingual speech recognition," in *Proceedings of Interspeech*, Brighton, UK, 2009, pp. 3019–3022.

[59] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, January 2000.

[60] N. Brümmer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[61] N. Brümmer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D. dissertation, Department of Electrical and Electronic Engineering, University of Stellenbosch, Private Bag X1, 7602 Matieland, South Africa, 2010.

[62] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," in *Proceedings of Eurospeech*, 1997, pp. 1985–1988.

[63] 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.

[64] FoCal, *Toolkit for Evaluation, Fusion and Calibration of statistical pattern recognizers*, 2008, http://sites.google.com/site/nikobrummer /focal.

[65] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Faculty of Information Technology, Brno University of Technology, http://www.fit.vutbr.cz/, Brno, Czech Republic, 2008.

[66] 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, US, 2007, pp. 1–5.

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

[68] 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/∼cjlin/liblinear.

[69] C. Chang and C. Lin, *LIBSVM: a library for support vector machines*, 2001, http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[70] 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.

[71] O. Glembek, P. Matějka, L. Burget, and T. Mikolov, "Advances in phonotactic language recognition," in *Proceedings of Interspeech 2008*, Brisbane, AU, 2008, pp. 743–746.

[72] P. Matějka, "Phonotactic and acoustic language recognition," Ph.D. dissertation, Faculty of Electrical Engineering and Communication, Department of Radio Electronics, Brno University of Technology, 2008.

[73] M. F. BenZeghiba, J. L. Gauvain, and L. Lamel, "Improved n-gram phonotactic models for language recognition," in *Proceedings of Interspeech*, Makuhari, Japan, 2010, pp. 2710–2713.

Mikel Penagarikano was born in Zumarraga, Spain, in 1973. He received the M.Sc. degree in Physics from the University of the Basque Country (UPV/EHU), Leioa, Spain, in 1996. He is currently pursuing the Ph.D. degree at the same university.

From 1997 to 2000 he was at the Department of Electricity and Electronics, UPV/EHU, under a research grant. Since 2000, he has been Assistant Professor of Computer Science in the same department. His research interest focuses on developing efficient software architectures for speech processing applications, such as ASR, language recognition, speaker recognition, etc.

Amparo Varona was born in Barakaldo, Spain, in 1970. She received the M.Sc. and Ph.D. degrees in Physics from the University of the Basque Country (UPV/EHU) in 1993 and 2000, respectively.

From 1994 to 1996 she was at the Department of Electricity and Electronics, UPV/EHU, under a research grant. From 1996 to 2003 she was Assistant Professor and since 2003 she has been Associate Professor of Computer Science in the same department. Her past research activities include language modelling and efficient search for ASR. Her current research interests include spoken document retrieval, language recognition and speaker recognition.

Luis Javier Rodriguez-Fuentes was born in Bilbao in 1968. He received the M.Sc. and Ph.D. degrees in Physics from the University of the Basque Country (UPV/EHU) in 1991 and 2004, respectively.

From 1993 to 1996 he was at the Department of Electricity and Electronics, UPV/EHU, under a research grant. Since 1996, he has been Assistant Professor of Computer Science in the same department. His past research activities include acoustic modelling, spontaneous speech modelling and speaker adaptation for ASR. His current research interests include spoken document retrieval, language recognition and speaker recognition.

German Bordel was born in Bilbao in 1961. He received the M.Sc. and Ph.D. degrees in Physics from the University of the Basque Country (UPV/EHU) in 1985 and 1996, respectively.

In 1988, after two years working on microprocessor systems for control, he joined the Department of Electricity and Electronics, UPV/EHU, as Assistant Professor of Computer Science, where since 2008 he is Associate Professor. His interests include software architectures, web-based applications, spoken document retrieval and speaker recognition.