Acoustic Event Recognition for Low Cost Language Identification

Danilo Spada, Ignacio Lopez, Doroteo T. Toledano, Joaquín González-Rodríguez

Biometric Recognition Group – ATVS
Escuela Politécnica Superior
Universidad Autónoma de Madrid, Spain

{danilo.spada, doroteo.torre, ignacio.lopez, joaquin.gonzalez}@uam.es

Abstract

One of the most popular approaches to Automatic Language Identification (LID) is Parallel Phone Recognition followed by Language Modeling (Parallel PRLM or PPRLM). This approach has proved to be very successful in LID. However, it has two mayor drawbacks: its high computational cost due to the need to run several phone recognizers on the same test segment; and the need to train the phone recognizers on manually transcribed data that may not match closely the type of speech on which the system will work. In this paper we present a novel approach for LID that tries to solve these two problems. It is based on substituting the phonetic recognizers by an Acoustic Event Recognizer (AER) that can be trained on untranscribed data and is much faster than the phone recognizers. Results show that this method, which we call AERLM, can be much faster than PRLM, although at the cost of reduced LID precision, and therefore suitable for low-cost LID.

1. Introduction

Automatic Language Identification is the task of recognizing the language spoken in a sample of speech. This automation can be very useful in multicultural environments like airports, congresses or international meetings. It can act as integrating part of all those services, both fully automated or not, that are able to act in different languages, adapting themselves to the user's spoken language.

Nowadays it is possible to distinguish two main groups of techniques for automatic language recognition: a high level approach (which uses acoustic features and linguistic units) and an acoustic approach (the algorithms which only use acoustic features). We can classify the most popular systems as the following:

acoustic level techniques:
- GMM, Gaussian Mixture Model classification;
- SVM-GLDS, Support Vector Machines with General Lineal Discriminant Sequence kernel;

high level techniques:
- PPR, parallel phone recognition;
- PRLM, phone recognition followed by language modeling;
- PPRLM, parallel PRLM;
- Improvements on PPRLM (lattices and SVM).

Most commonly used high level techniques can be grouped together as phonotactic techniques because they try to recognize languages based on the phones and sequences of most frequent phones in a language. This approach has two major drawbacks. Firstly, all these approaches are based on the concept of phoneme, a knowledge-based linguistic concept that is language-dependent and in many cases difficult to deal with in speech processing. An example of this difficulty is that in order to train a phoneme recognizer it is considered a requirement to have a manually phonetically transcribed database. Secondly, all these phonotactic approaches rely on phoneme recognizers that are costly in terms of computation, particularly when several recognizers in different languages are run in parallel as in PPRLM.

The first drawback is becoming less important with the increasingly number of transcribed speech corpora in different languages, as well as with techniques such as PRLM that don't require transcribed speech from a language in order to recognize it. However, the second drawback is becoming more and more important, particularly when emphasis is starting to be put not only in obtaining low-error systems, but also in obtaining low-cost systems [1].

In this paper, we present a modification of PRLM systems in which the Phonetic Recognizer (PR) is substituted by a data-driven Acoustic Event Recognizer (AER) to create what we call an AERLM system (Acoustic Event Recognizer followed by Language Modeling). In this way, we eliminate the need for phonetically transcribed data for training, which allows training the AER on data as close as possible to the testing (or working) data. Perhaps more importantly, AER is much faster than a phonetic recognizer, which makes it a good alternative to PRLM for limited-resource systems like embedded systems, as well as for a fast-matching stage prior to a more detailed matching.

The rest of the paper is organized as follows. Section 2 gives a panoramic view of our AERLM system. Section 3 gives more details about the Acoustic Event Segmentation. Section 4 describes our Acoustic Event Clustering. In section 5 we explain the language modeling and in section 6 our experiments. Finally, section 7 presents some conclusions as well as future work.

2. AERLM system

The AERLM technique is based on the same idea of PRLM: modeling and identifying languages based upon token sequences detected by a tokenizer. In AERLM, however, the tokenizer is not a phone recognizer. Our idea is to obtain transcriptions by an Acoustic Event Segmentation followed by an Acoustic Event Clustering.

The Acoustic Event Segmentation tries to approximate a phonetic segmentation. Segments are obtained using the variations in the spectrogram of speech, while the silence segments are removed. A similar technique has been used by Glass and Zue [2] for speech recognition, and also by Chollet for language independent speaker recognition [3]. This last technique was successfully applied in the context of speaker recognition [4]. This work tries to apply a similar (but even simpler) technique to the domain of automatic language identification.

The Acoustic Events obtained are parameterized using 13 MFCC and each segment is represented by a vector obtained averaging all the vectors of the segment. The parameterized segments are then used to train a clustering algorithm and to obtain the transcriptions, given by the sequences of recognized cluster numbers. In a second step, we model and recognize each language using statistical information about the obtained transcriptions, like in the PRLM architecture. In the literature, we found that Heck and Sankar built a cluster for speech segmentation [5], and, more recently, clustering has been used in language recognition to improve the modeling of co-articulation behavior [6].

3. Acoustic Event Segmentation

The first step in the process is the segmentation of the utterance into acoustically stable segments that intend to represent phonemes or stable parts of phonemes. This segmentation is based on a Spectral Variation Function (SVF) based on the Euclidean distance between the static MFCCs to the left and right of the current frame. After this SVF has been computed the utterance is segmented initially into segments divided by the maxima of the SVF. After this initial segmentation is applied, a Voice Activity Detector (VAD) working on a segmental basis is applied. This VAD is based on the average energy on the segment and maximum and minimum durations of speech and silence pulses. After the VAD is applied all contiguous silence segments are unified into a single segment and are not considered for the rest of the processing. For the segments corresponding to speech we compute the average of the static MFCCs as well as deltas and double deltas within each segment. Given that the segmentation procedure is intended to produce spectrally stable segments it makes sense to summarize the spectral content of the segment by a single vector (although more testing would be required to determine whether this option is the best or it would be better to take the central vector, for instance). In this way, we expect to reduce the computational complexity of the system without compromising its discriminative power.

4. Acoustic Event Clustering

Clusters have been largely used in pattern analysis to compress data [7]. By using clusters, the feature space is divided into subspaces, each one represented by a centroid. The clustering algorithm goal is to substitute each data vector for its nearest centroid. Data compression is achieved by then replacing the centroid by its associated token.

Somehow in a similar way, phonetic transcriptors replace a sequence of data vectors by its associated phoneme. This work proposes a modification of the phonotactic language identification problem by replacing the HMM phoneme models of a phonetic trancriptor with an acoustic event segmentation followed by a clustering of the segments.

The use of clusters in modeling acoustic events has several advantages: (i) Reduced computational cost; (ii) Phonetically labeled data is not needed to train the cluster; and (iii) It is a complementary approach to the phonotactic language identification approach, therefore fusion of standard systems and our system is expected to improve performance.

Popular algorithms to solve this problem are k-means or binary splitting. Nevertheless, during the last years many other have been introduced [8].

In this paper we propose a data driven algorithm based on GMM modeling. After the acoustic event segmentation and the averaging of the MFCCs over each segment, the resulting average MFCCs from the training corpus are modeled using a GMM. This GMM is then used to cluster together all the average MFCC vectors that produce the maximum likelihood for the same Gaussian of the GMM. In this way, the whole segment producing the average MFCC vector is substituted by the Gaussian number. With this approach the number of tokens produced is the same as the number of Gaussians in the GMM. For our experiments we used 64, 128 and 512 Gaussians/clusters. .

Depending on the GMM training data, the cluster can be used to model different acoustic events. We can train the cluster on a single language (single-language clustering) or train it using data from different languages (multi-language clustering).

5. Language modeling

Independent language models are created by obtaining a transcription of a different training set by means of a nearest-neighbor with the previously trained cluster. The stored language model is formed by the frequencies of all the unigrams, bigrams and trigrams for the given codebook. A Universal Background Model (UBM) with information of several languages is also trained following the same algorithm.

The independent language models are adapted from the UBM in order to obtain adapted language models by means of the following formula:

$$P(c \mid \lambda A) = \alpha P(c \mid \lambda L) + (1-\alpha)(c \mid \lambda UBM) \quad \textit{(1)}$$

where c is a given n-gram, λA is the adapted language model, λL the independent language model, λUBM the Universal Model, and α a given constant in the range [0-1].

Finally, test scores are computed for all the modeled languages using the adapted language model and the UBM .

6. Experiments

In this section we present language recognition results using as training material the LDC CallFriend database [9] and as testing database the evaluation data from the 2005 NIST Language Recognition Evaluation [10]. It is worth noting that we haven't made use of any phonetically transcribed data for these experiments.

This section is organized into 4 sections. The first one presents the experimental set-up, the second one presents results with a single clustering system trained on a single language. Then we build a system similar to a PPRLM system in which we train a clustering system for each language and then fuse all of them. In section 6.4 we introduce results obtained training the cluster on more than one language and finally, section 6.5 compares our results with those obtained by our group in NIST 2005 Language Recognition Evaluation [10], where a more standard PPRLM system with 6 or 12 phonetic decoders were used.

6.1. Experimental set-up

Our base system has a front-end module that outputs 13 MFCC, Δ and ΔΔ parameters (39), but we only used the first 13 (static) MFCCs in the experiments described in this paper. Three different GMM base clusters, with 64, 128 and 512 mixtures have been trained for each of the following ten languages: english, mandarin, spanish, arabic, farsi, german, hindi, japanese, korean, tamil. For this task we used the complete LDC CallFriend database [9]. For each clustering, we processed this corpus to obtain transcriptions of all 12 languages. In a second step we used the transcriptions to train the UBM and to adapt the independent language models. We used unigrams, bigrams and trigrams. The testing material consisted of the samples of 30 seconds of the NIST LRE 2005 test corpus [10]. The selected target languages are: English, Hindi, Japanese, Korean, Mandarin, Spanish and Tamil. English, Mandarin and Spanish appear with two dialects. We applied TNorm to all experiments presented.

6.2. Results for a single AERLM system

For the systems using single language trained clustering, we obtained an EER around 36%.

In Figure 1 we present the results obtained processing the target languages with one of the system that seems to work better: the one using the clustering trained on Japanese.

Is it possible to imagine that the average computed during the acustic event segmentation and clustering reduces the amount of information available for discriminating among languages. An interesting aspect to explore in the future consists in removing this limitation by not averaging features over an acoustic event segment.

6.3. Parallel AERLM

The distribution of phonemes across languages may be different depending of the languages involved. In some cases, phonemes which are typical in one language may be rare in another. Therefore a clustering trained on a single language can introduce a loss of information.

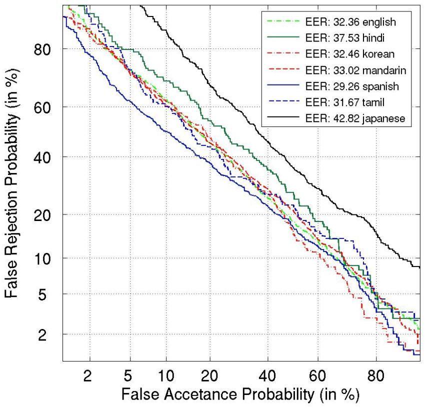

Figure 1: *Results on NIST LRE 2005 per Language for an AERLM system with AER trained on Japanese*.

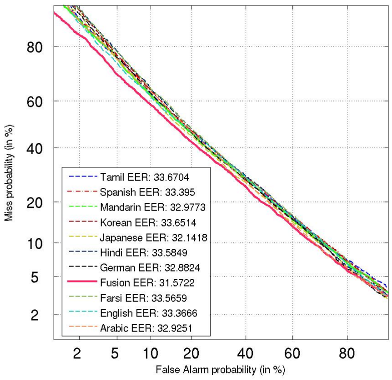

Figure 2: *Results on NIST LRE 2005, all languages. Parallel AERLM with AERs using 128 clusters trained for 10 languages.*

For PRLM language recognition systems Hazen proposed to train a phoneme recognizer on more than one language [11]. An alternative possibility is to run concurrently many systems and then fuse results: this is the base idea of PPRLM. The fusion of the different PRLM systems is typically obtained by using first a score normalization algorithm, for example T-Normalization [12], due to the intrinsic difference between the subsystems.

In our approach we explored both ideas: running many systems and fusing results (Parallel-AERLM) and training a cluster on many languages (Multi-language-Cluster-AERLM).The results obtained with Parallel AERLM are presented in Figure 2.

By fusing all AERLM systems we have obtained an absolute performance improvement of 1,5 points in EER relative to the single AERLM global behaviour.

6.4. Multi-Language AERLM system

A multi-language cluster, which models acoustic events from all languages, can be trained by using a language independent GMM. The main advantage of Multi-Language AERLM systems is that we do not need several language-dependent AERLM systems, as in the Parallel AERLM approach. So, we would obtain a computational cost of a tenth of the P-AERLM.

We explored the possibility to train a clusterig on more than one language using 64, 128 and 512 clusters. Results shows that the multilanguage system performance is similar to any single language AERLM system. Probably because, although the tokenization is richer, tokens are not discriminative enough.

6.5. Comparison with PPRLM systems

The main goals of the AERLM system were to improve the PPRLM system in two ways: firstly by avoiding the need to use phonetically transcribed speech to train the phonetic decoders, and secondly, by making language recognition much faster. In this sense the AERLM system achieved these goals since we don't need phonetically transcribed material any more and the computational cost for performing the NIST LRE 2005 test is only 80 hours with the 10 AERLMs in parallel, much less than the 500 hours that would require a system with 10 PRLM systems in parallel. Moreover, performance for any single-language AERLM system or the multi-language AERLM system is only slightly worse than the 10 AERLM systems in parallel, and they consume 10 times less processing time. This last option is particularly adequate for resource-limite systems and also for a fast-matching module prior to a more detailed matching. In all cases experiments were run on a Pentium IV system at 2.4 GHz. with 1Gb RAM.

But of course the goal was to do this without significant degradation in performance. Unfortunately this last condition has not yet been met. Figure 4 presents results achieved by our group in 2005 NIST Language Recognition Evaluation (NIST LRE 2005). In there ATVS1 was a PPRLM system using 12 phonetic decoders trained on OGI Multi-Language Telephone Speech Corpus and ATVS2 was similar but with only 6 phonetic decoders. For 30s test segments results were between 20 and 22% in terms of EER. Our best AERLM system so far is still far from this result (31.5% EER), as presented in Figure 3. For future work we need to fine tune some parameters of the AERLM systems to try to get better performance while keeping the computational cost low.

7. Conclusions

We have analyzed the substitution of the phone recognizers in a PPRLM system by Acoustic Event Recognizers (AER) that are much faster and can be trained on untranscribed data. With this substitution, we can build AERLM and Parallel AERLM systems that are much faster than the corresponding PRLM and PPRLM systems and have the additional advantage that they can be trained on untranscribed data, thus increasing the amount of available training data and allowing for a better fit between the characteristics of the training and test (or working) data.

Given the reduced computational cost of the systems proposed, particularly for a single multi-language AERLM system, which performs almost as well as the 10-langauge Parallel AERLM system with 10 times less computational cost, we can envisage these type of systems as a very useful alternative to more computational complex systems for embedded devices or even as a fast-matching stage prior to a more detailed (and complex) matching where required.

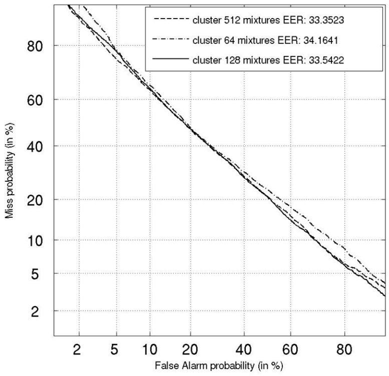

Figure 3: *Results on NIST LRE 2005, all languages. Multi-Language Cluster AER systems with 64, 128 and 512 clusters.*

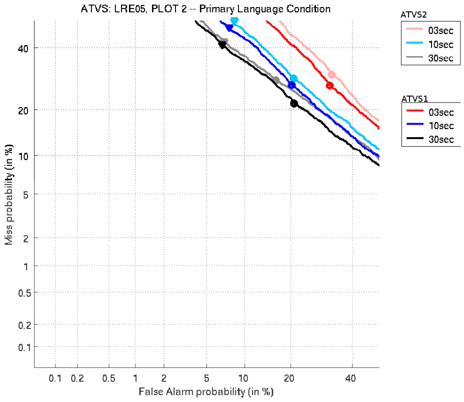

Figure 4: *Results achieved by ATVS on NIST 2005 LRE data using a standard PPRLM system trained on OGI Multi-Language Telephone Speech data.*

8. References

[1] W. M. Campbell, D. E. Sturim, W. Shen, D. A. Reynolds, J. Navratil, “The MIT-LL/IBM 2006 Speaker Recognition System:High-Performance Reduced-Complexity Recognition”, in IEEE ICASSP 2007, pp. 217-220.

[2] J.R Glass, V.W.Zue, “Multi-level acoustic segmentation of continuous speech”, in Acoustics, Speech, and Signal Processing, ICASSP-88, New York, USA, 1988.

[3] G. Chollet, J. ˇ Cernock´y, A. Constantinescu, S. Deligne, and F. Bimbot, “Towards ALISP: a proposal for Automatic Language Independent Speech Processing,” In Keith Ponting, editor, NATO ASI: Computational models of speech pattern processing Springer Verlag, 1999.

[4] Asmaa El Hannani1, Doroteo T. Toledano, Dijana Petrovska-Delacretaz, Alberto Montero-Asenjo and Jean Hennebert, “Using Data-driven and Phonetic Units for Speaker Verification”, In Proceedings IEEE Odyssey 2006, San Juan, Puerto Rico.

[5] L Heck, A Sankar “Acoustic Clustering and Adaptation for Robust Speech Recognition”- Proceedings of EUROSPEECH, Rhodes, Greece 1997

[6] Chien-Lin Huang, Chung-Hsien Wu, “Phone set generation based on acoustic and contextual analysis for multilingual speech recognition” Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ‘07, Hawaii 2007

[7] S. Theodoridis, and K. Koutroumbas., “Pattern Recognition” Second ed. Amsterdam, Elsevier Academic Press, 2003

[8] RO Duda, PE Hart, DG Stork “Pattern Classification”, New York, Wiley-Interscience, 2000

[9] Linguistic Data Consortium, http://www.ldc.upenn.edu/

[10] “Speaker recognition evaluations”, http://www.nist.gov/speech/

[11] TJ Hazen, VW Zue, “Recent improvements in an approach to segment-based automatic language identification”, Proc. ICSLP '94, Yokohama, Japan, pp 1883-1886, 1994

[12] R Auckenthaler, M Carey, H Lloyd-Thomas, “Score Normalization for Text-Independent Speaker verification system”, Digital Signal Processing, vol 10 2000, pp 42-54, 2000