On the use of Phone-based Embeddings for Language Recognition

Christian Salamea[1,2], *Ricardo de Córdoba*[2], *Luis Fernando D'Haro*[2], *Rubén San Segundo*[2], *Javier Ferreiros*[2]

[1]Interaction, Robotics and Automation Research Group, Universidad Politecnica Salesiana, Calle Vieja 12-30 y Elia Liut, Cuenca, Ecuador.
[2]Speech Technology Group, Information and Telecomunications Center, Universidad Politecnica de Madrid, Ciudad Universitaria Av. Complutense, 30, 28040, Madrid

csalamea@ups.edu.ec, cordoba@die.upm.es, lfdharo@die.upm.es, lapiz@die.upm.es, jfl@die.upm.es

Abstract

Language Identification (LID) can be defined as the process of automatically identifying the language of a given spoken utterance. We have focused in a phonotactic approach in which the system input is the phoneme sequence generated by a speech recognizer (ASR), but instead of phonemes, we have used phonetic units that contain context information, the so-called "phone-gram sequences". In this context, we propose the use of Neural Embeddings (NEs) as features for those phone-grams sequences, which are used as entries in a classical i-Vector framework to train a multi class logistic classifier. These NEs incorporate information from the neighbouring phone-grams in the sequence and model implicitly longer-context information. The NEs have been trained using both a Skip-Gram and a Glove Model. Experiments have been carried out on the KALAKA-3 database and we have used Cavg as metric to compare the systems. We propose as baseline the Cavg obtained using the NEs as features in the LID task, 24,7%. Our strategy to incorporate information from the neighbouring phone-grams to define the final sequences contributes to obtain up to 24,3% relative improvement over the baseline using Skip-Gram model and up to 32,4% using Glove model. Finally, the fusion of our best system with a MFCC-based acoustic i-Vector system provides up to 34,1% improvement over the acoustic system alone.

Index Terms: language identification, phonotactic, neural embeddings

1. Introduction

Automatic spoken language identification (LID) is the process of identifying the actual language of a sample of speech using a known set of trained language models [1]. There are currently two main ways of achieving this goal: the first one uses acoustic features extracted from the speech signal in which the spectral information is used to distinguish between languages, while the second method uses the phonetic sequences obtained using an automatic phonetic recognizer (ASR) as features.

In general, the best results for the LID task are achieved using acoustic-based systems. However their fusion with phonetic/phonotactic-based systems provides a higher accuracy [2]. This paper focuses on the study of phonotactic techniques, but we will also show that it contributes to an improvement in the overall system thanks to the fusion [3] of both techniques.

Neural Embeddings (NEs) are vector representations [4] of the phonetic units and have been used in speech recognition tasks [5]. These vector representations are extracted from either the hidden layer in an Neural Network (NN) [6] or from the occurrence matrix [7] of phonetic units in an unlabeled corpus . NEs has been successfully used in speech recognition systems at a word level [5], because of their ability to model the probability of one word appearing in a context close to another one, however their suitability in phonotactic LID systems has not yet been considered, probably due to some difficulties at the phoneme level that do not appear at the word level.

For example, NEs are normally created to reflect the relation at the semantic and syntactic level between words [8], characteristics that do not exist at a phonetic level. Our proposal is to process NEs similarly to the i-Vector framework, as continuous vectors that contain the most representative information about a language in a low dimension [9]. Our expectation is that NEs in a language will be projected into some particular direction being this projection discriminative in comparison to the other languages.

To overcome the limitations in NEs at the phoneme level, we propose the use of phonetic units that incorporate context information (phone-grams). However, both the high-order phone-grams and the embedding vectors size could make it difficult the training of the i-Vectors because of the large size of the matrix needed for this implementation. We will show two alternatives for dealing with the feature vectors.

Context information has also been incorporated in those feature vectors which we have called "Context Neural Embedding Sequences", which we have used to generate the i-Vectors. These i-Vectors are used as features of a multiclass logistic classifier that obtain the detection cost values (Cavg) for the language trained.

Finally, all of the systems are fused obtaining the final global Cavg for all of the languages trained.

This paper is organized as follows. In section 2, we describe the different techniques, as well as the acoustic system used in the fusion with the phonotactic system. In section 3, we present the experiments and the final results. Finally, in section 4, the conclusions and future work are presented.

10.21437/IberSPEECH.2018-12

2. System Description

2.1. Phone-gram definition

Phonotactic systems use context information to improve the performance of LID. In this regard, we propose to use phonetic units that implicitly incorporate context information as features (phone-grams). They can be defined as the grouping of two or more phonemes in a new unit (Figure 1). In this work we have used 2grams only because of the scattering observed in higher order.

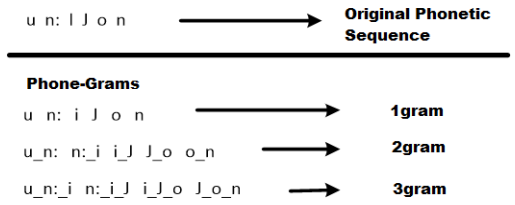

Figure 1: *Concept of phone-gram.*

2.2. Database

The KALAKA-3 database was created for the Albayzin 2012 LRE [10]. It is designed to recognize up to 6 languages in the closed set condition (i.e. Basque, Catalan, English, Galician, Portuguese, and Spanish) using noisy and clean files with an average duration of 120s and includes 108 hours in total. This database contains training, development, testing, and evaluation examples distributed as shown in Table 1:

Table 1: *KALAKA-3 database.*

	Train	Dev	Test	Eval
Nº Files	4656	458	459	941
Nº of clean files	3060	-	-	-
Nº of noisy files	1596	-	-	-
Length<=30s	2855	121	113	267
Length 120s	1801	337	346	674

To compare the systems, we have considered the Cavg metric, that weights the number of the false acceptances and false alarms generated by the recognizer, representing them as a detection cost function [11]. Using this metric, lower values correspond to better systems.

2.3. Phoneme Recognizers

The phoneme sequences of the utterances used to obtain the "phone-grams" have been generated from a phoneme recognizer. In our system, it is based on the system designed by the Brno University (BUT) [12], which uses monophone three state HMMs. There are 3 sets of HMMs (for Hungarian, Russian, and Czech) with 61, 46, and 52 different phonemes respectively.

2.4. Phonetic Vector Representation

In our approach, we have called SG-Emb to thevector representations extracted from the hidden layer of an NN and Gl-Emb to the vector representations extracted from the co-occurrence of elements matrix obtained from the global corpus. In the first case, the objective is to predict the phonetic unit that is going to appear next according to the context in which the unit is included. In the second case, the objective is to normalize the counts and then smooth them trying to obtain a vector representation with homogeneous values.

The model definition normally used to train both is focused at the word-level [13] but we work at the phone level. The objective is to find the co-occurrence of phonemes and phoneme sequences that tend to appear in similar contexts for a specific language. Hence, we expect to improve the results compared with the system based on uniphone sequences. Our study focuses on phone-grams, and their use in the continuous space has been called Phone-based Embeddings (Ph-Emb)..

2.5. Skip-Gram Model

In relation to the SK-Emb obtained from a NN, they are obtained with the following procedure: we consider a NN with one input layer, one hidden layer and one output layer. In the input layer we have the phone-gram with 1-of-N coding, in the state layer we obtain the vector representation of the phone-gram, and the SK-Emb are generated by applying a modelling technique on the vector representation of the input phone-gram together with its context. These SK-Emb will be the feature vectors that we will use in our LID system. From several models that have been proposed for that purpose, two of them are the most used: Skip-Gram [14] and C-Bow [15]. We have selected Skip-Gram as we obtained better results in initial experiments.

The Skip-Gram model is a classic NN, where the activation functions are removed and hierarchical Soft-max [16] is used instead of soft-max normalization. The training objective of the Skip-Gram model is to predict the context of the input phone-gram in the same sentence [17].

2.6. GloVe Model

On the other hand, the Gl-Emb from the co-occurrence matrix have been obtained as follows: the matrix contains the counts of a phonetic unit appearing close to a possible context. Rows are the phonetic units of the vocabulary while columns are the possible context of those phonetic units. The least squares model is used in the training process (GloVe model) [7].

GloVe models capture the statistics of the global corpus to learn the vector representations of words. It is normally used at a word level, but in our case, we are going to evaluate them at a phonetic level using phone-grams. GloVe models are similar to Skip-Gram models except for the context windows. GloVe uses the co-occurrence of elements, capturing the global statistics of the corpus in a matrix and analyzing the entire possible co-occurrence probability rates using test phonetic units. This way, it is possible to distinguish the relevant and non-relevant phonemes in a sentence [7].

2.7. i-Vector system based on Phone-based Embeddings applied to LID

We propose to use these Phone-based Embeddings as feature vectors for the input of a LID system based on the i-Vector framework [18]. In Figure 2, we present the global system architecture. Our system has two components. The first one called "Front-End", is where the acoustic signal processing is carried out followed by the phoneme recognizer,

[12] that produces the phonetic sequences corresponding to the utterances.

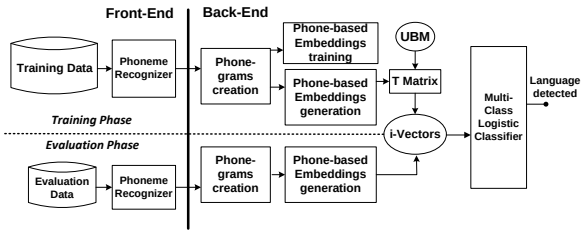

Figure 2: *Global System Architecture*.

The second component of the system is the "Back-End". Firstly, we obtain the phone-gram sequences from the phoneme sequences of each language. The sequences obtained have been used to train the Phone-based Embeddings . To model the Phone-based Embeddings we have used both alternatives described above, Skip-Gram and GloVe modeling. After that, we have replaced every phone-gram by its respective Phone-based Embedding to use it as input feature vector to the i-Vector system. All these vectors are used to train the T matrix and the UBM model needed to obtain the i-Vectors. Finally we have used these i-Vectors as features to train a multiclass logistic classifier to define the detected language [19], [20].

As we have one different Phone-based Embedding for each language to be recognized, we considered two alternatives to manage the set of vectors for each phone-gram. We have called them: "Single vector embedding" and "Multiple vector embeddings".

2.8. Single Vector Embedding (SVE)

We organize the phone-gram sequence in a column, replacing each phone-gram by its corresponding Phone-based Embedding of a specific language and repeat this process for all the other languages. So, the first column will contain the Phone-based Embeddings sequences trained with data from language 1, the second one will contain the Phone-based Embeddings sequences trained with data from language 2, and so on. Finally, we obtain a matrix that includes the Phone-based Embeddings trained with all the languages to be recognized (Figure 3).

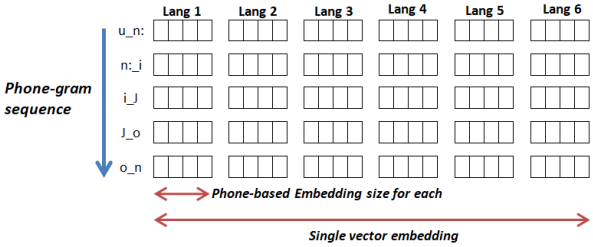

Figure 3: *Single Vector Embedding*.

2.9. Multiple Vector Embedding (MVE)

As SVE could easily have a problem of excessive dimensionality, we considered a system where we use the Phone-based Embeddings trained for each language individually (Language Phone-based Embeddings) to obtain different i-Vectors for each language. Our proposal is to fuse the scores provided by the individual language-dependent systems expecting a better performance and a lower computational cost.

2.10. Acoustic System using MFCCs

We have fused the scores of the proposed techniques with the scores obtained from an acoustic system to check if they provide complementary information. The acoustic system has been generated as follows: from each speech utterance, 12 MFCC coefficients including C0 [21] are extracted for each frame. The silence and noise segments of the acoustic signal have been removed using a Voice Activity Detector. To reduce the noise perturbation, a RASTA filter has been used together with a cepstral mean and variance normalization (CMNV). We have a feature vector of dimension 56, generated from the concatenation of the SDC parameters using the 7-1-3-7 configuration. Feature vectors are used to train the total variability matrix, from which the i-vectors of dimension 400 with 512 Gaussians are extracted (optimal configuration).

3. Results

We have to define the Phone-based Embeddings optimal training parameters for 2grams: vector size, window size, number of training iterations and negative sampling factor. Negative sampling is an optimization method used to improve the NEs robustness applying logistic regression. It reduces the computational complexity and increases the vector estimated efficiency.

The window size corresponds to the number of phonetic phone-gram units considered to the left and to the right of the current phonetic unit and it is considered as contextual information. The vector size is the vector embedding size. In all cases, the results in the tables represent the fusion of the three phonetic recognizers.

3.1. Single Vector Embedding (SVE)

As we described in Section 2.8 we have generated a sequence of phone-grams for each language. We have tested several options for the feature vector size, obtaining an optimum for size 40, being 240 the final vector, considering the 6 languages to be recognized. In relation to the number of Gaussians in the i-Vectors system we have obtained an optimum for 512. The best result was 24.69% of Cavg.

3.2. MVE using the Skip-Gram model

When we considered a single Phone-based Embedding feature vector for each language as in SVE we obtained 29,15% of Cavg using only the model for the Basque language. After fusing all the languages we obtained 19,73% of Cavg, which is a relative improvement of 20.1% over SVE. So, we decide to use MVE in all remaining experiments.

3.3. Inclusion of Context in the vector embeddings

Based on the hypothesis that the vector representations obtained from NN have some linguistic regularities as the additive composition [13], we propose to use *Contextual information*, including information from the neighboring phone-grams in the final Phone-based Embedding.

Our proposal is to use a weighted sum of neighboring phone-grams. We considered two options for that, the first one

uses a three phone-gram context window and the second one a five phone-gram context window with the following weights:

A) 3 phone-grams context: Final Ph-Emb = Left Ph-Emb * 0.25 + Central Ph-Emb * 0.50 + Right Ph-Emb * 0.25

B) 5 phone-grams context: Final Ph-Emb = Second Left Ph-Emb * 0.10 + Left Ph-Emb * 0.15 + Central Ph-Emb * 0.50 + Right Ph-Emb * 0.15 + Second Right Ph-Emb * 0.10

The objective is to assign more weight to the current phone-gram but taking into account information from the neighboring units. Using option A (Cavg: 24,38) we obtained a 14,4% relative improvement over option B (Cavg: 28,49) using the model for the Basque language and the Skip-Gram technique as reference. The optimum vector size for SG-Emb is 80, with 512 Gaussians in the iVectors system, 10 iterations and a window size of 8. All the optimization has been obtained using the data development set. After fusing all the languages, we obtained 18.70% of C_{avg} with MVE, which is a relative improvement of 24.3% over SVE.

3.4. GloVe model for the MVE

We have also evaluated our approach using the GloVe model (Section 2.6) instead of the Skip-Gram model for our best system with contextual information, because it incorporates information of the co-occurrence of phone-grams in all the training data set. The optimal configuration parameters are: vector size of 80, window size of 4, and 30 iterations. The optimum number of Gaussians for the iVectors system has been 512 (the same as Skip-Gram). Fusing all the languages as before, we obtain a 16,70% of C_{avg}, which is a 10.7% of relative improvement over MVE based on the Skip-Gram model.

3.5. Summary of results

In Table 2 we present the summary of results obtained with the techniques proposed in this paper. As we can see, the final system using the GloVe model provides the best results.

Table 2: *Summary of results.*

System	Cavg	Improvement %
SVE	24.69	
MVE context and Skip-Gram	18.70	24.3
MVE context and GloVe	16.70	32.4

3.6. Fusion with the acoustic model

The objective of this technique is to improve an existing LID system, which is based on acoustic information (section 2.10). So, we present the results of fusing the existing acoustic LID system with our two best systems, based on Phone-based Embeddings obtained with the Skip-Gram and Glove models (Table 3).

Table 3: *Phone-based Embeddings systems fused with an acoustic system.*

System	Cavg	Improvement %
Acoustic system	7.60	
Fusion with SG-Emb	5.40	28.9
Fusion with Gl-Emb	5.01	34.1

4. Conclusions

We have demonstrated that the use of Phone-based Embeddings as feature vectors provides improvements in an LID task. We have used as a baseline a first system that uses Phone-based Embeddings as feature vectors with rather poor results. However, using the new approaches proposed in this paper results improved, and the fusion of our best configuration with our acoustic system provides significant improvements.

Our baseline system uses the "SVE" technique to obtain 24.69% of Cavg. Considering this poor result, we decided to change the approach and use an individual matrix for each language, fusing the scores from all individual systems at the back-end, and we obtained 19,73% of Cavg using the Skip-Gram modelling.

Then, we proposed the inclusion of context information in the Phone-based Embeddings including the two or four nearest neighbours. After fusing all the language models we obtained 18.7% of Cavg using the Skip-Gram modelling. Finally, using the GloVe modelling we obtain 16.7% of Cavg with a 10.7% relative improvement over Skip-Gram modelling and a 32.4% compared to the baseline system.

Also, the fusion with the acoustic based system provides a 34.1% relative improvement, which demonstrates that both systems provide complementary information for the LID task.

As future research lines, we propose to study the effects of higher order units using a larger database. We will also evaluate other types of language models for the neural embeddings. Also, we expect to use models with a high number of layers (char-RNN) and use its combination with convolutional DNNs to get better local context characteristics.

5. Acknowledgements

The work leading to these results has been supported by AMIC (MINECO, TIN2017-85854-C4-4-R), and CAVIAR (MINECO, TEC2017-84593-C2-1-R) projects. Authors also thank Mark Hallet for the English revision of this paper and all the other members of Speech Technology Group for the continuous and fruitful discussion on these topics. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

6. References

[1] Y. Muthusamy, E. Barnard and A. Cole, "Reviewing automatic language identification," in *Signal Processing Magazine, IEEE* 1994, pp. 33–41.

[2] L. D'Haro, R. Cordoba, C. Salamea and J. Echeverry "Extended phone-likelihood ratio features and acoustic-based i-vectors for language recognition," in *Proceedings in Acoustics, Speech and Signal Processing, ICASSP*, 2014, pp. 5342–5346.

[3] N. Brummer and D. Van Leeuwen, "On calibration of language recognition scores," in *Speaker and Language Recognition Workshop, IEEE Odyssey 2006*, pp. 1-8.

[4] J. Turian, L. Ratinov, and Y. Bengio, "Word Representations: A simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 384–394.

[5] S. Bengio and G. Heigold, "Word Embeddings for Speech Recognition," in *INTERSPEECH 2014 – 15th Annual Conference of the International Speech Communication Association, September 14-18, Singapore, Proceedings*, 2014, pp. 1053–1057.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
[7] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representations," in *Proceedings of the conference on empirical methods in natural language processing*, 2014, pp. 1532–1543.
[8] P. Wang, B. Xu, J. Xu, G. Tian, C. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," , 2016, pp. 806–814.
[9] L. D'Haro, R. Cordoba, M. Caraballo and J. Pardo, "Low-resource language recognition using a fusion of phoneme posteriorgram counts, acoustic and glottal-based i-vectors," in *Proceedings in Acoustics, Speech and Signal Processing ICASSP*, 2013, pp. 6852–6856.
[10] L. Rodriguez-Fuentes, M. Penagarikano, A. Varona, M. Diez and G. Bordel, "KALAKA-3: a database for the assessment of spoken language recognition technology on YouTube audios," in *Language Resources and Evaluation*, 2016, pp. 221–243.
[11] A. Martin and C. Greenberg, "The 2009 NIST Language Recognition Evaluation," in *Speaker and Language Recognition Workshop, IEEE Odyssey* 2010, pp. 165–171.
[12] P. Ace, P. Schwarz and V. Ace, "Phoneme recognition based on long temporal context," PhD. Thesis, Brno University of Technology, Faculty of Information Technology, 2009.
[13] T. Mikolov, K. Cheng, G. Corrado and J. Dean, "Efficient estimation of word representation in vector space," in *Proceedings of Workshop at ICLR*, pp. 1–12.
[14] D. Guthrie, B. Allison, W. Liu, L. Guthrie and Y. Wilks, "A closer look at Skip-Gram modelling," in *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006, pp. 1–4.
[15] Y. Yuang, L. He, L. Peng and Z. Huang, "A new study based on word2vec and cluster for document categorization," in *Journal of Computational Information Systems.*, 2014, pp. 9301–9308
[16] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Aistats*, 2005, pp. 246–252.
[17] S. Yujing, X. Yeming, X. Ji, P. JieLin and Y. Yonghong, "Recurrent neural network language model with vector-space word representations," in *the 21th International Congress on Sound and Vibration*, 2014.
[18] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-end factor analysis for speaker verification. in *Audio, Speech and Language Processing*, 2011, pp. 788–798.
[19] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Kara, 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" in *Audio, Speech and Language Processing.*, 2007, pp. 2072–2084.
[20] M. BenZeghiba, J. Gauvain and L. Lamel, "Language score calibration using adapted Gaussian back-end," in *INTERSPEECH 2009 -- 10th Annual Conference of the International Speech Communication Association*, 2019, pp. 2191–2194.
[21] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *Acoustics, Speech and Signal Processing.* 1980, pp. 357–366.