ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746459

PHONOTACTIC LANGUAGE RECOGNITION USING A UNIVERSAL PHONEME RECOGNIZER AND A TRANSFORMER ARCHITECTURE

David Romero[1], *Luis Fernando D'Haro*[2], *Marcos Estecha-Garitagoitia*[2], *Christian Salamea*[1,2]

[1] Interaction, Robotics and Automation Research Group, Universidad Politécnica Salesiana, Calle Vieja 12-30 y Elia Liut, Cuenca, Ecuador.

[2] Speech Technology Group, Information and Telecommunication Center, Universidad Politécnica de Madrid, Ciudad Universitaria Avda. Complutense, 30, 28040, Madrid.

ABSTRACT

In this paper, we describe a phonotactic language recognition model that effectively manages long and short n-gram input sequences to learn contextual phonotactic-based vector embeddings. Our approach uses a transformer-based encoder that integrates a sliding window attention to attempt finding discriminative short and long cooccurrences of language dependent n-gram phonetic units. We then evaluate and compare the use of different phoneme recognizers (Brno and Allosaurus) and sub-unit tokenizers to help select the more discriminative n-grams. The proposed architecture is evaluated using the Kalaka-3 database that contains clean and noisy audio recordings for very similar languages (i.e. Iberian languages, e.g., Spanish, Galician, Catalan). We provide results using the Cavg and accuracy metrics used in NIST evaluations. The experimental results show that our proposed approach outperforms by 21% of relative improvement to the best system presented in the Albayzin LR competition.

Index Terms— Language recognition, phonotactic information, transformers, acoustic systems.

1. INTRODUCTION

Most current systems for spoken language recognition (LR) are based on using acoustic information due to their high accuracy and performance. However, it is well known that the fusion of these systems with others systems based on high-level information such as modeling phonotactic sequences provides a higher accuracy.

Phonotactic-based LR systems model sequences of phonetic units and learn the probabilistic distribution of phoneme units for the different languages to recognize. However, focusing on methodologies to find discriminative phonetic units to give them more weight or to pay them more attention has also been found to be important [1]. Traditional approaches focused on the Parallel Phoneme Recognition followed by a Language Model (PPRLM) framework [2], where phonetic sequences are obtained by a single or multiple phoneme recognizers trained to generate a finite set of units (vocabulary) that may belong to the same languages to recognize or not (producing a mismatch that reduces performance). On the other hand, phonotactic systems [3] need to deal with overlapping problems due to using the same phoneme set to represent different languages in the evaluation corpus. Moreover, the recognized phoneme sequences are usually very long, containing hundreds of phonetic units, which brings another difficulty to find discriminative units.

To mitigate some of these problems, [4][5] proposed using a single phoneme recognizer to get the phonetic sequences that are then used as input to a Skip-gram, Glove or FastText model to learn phonotactic based continuous vector-embeddings using contextual information. Alternatively, [6,7,8,9,10] used one or multiple phoneme recognizers (PPRLM) to get the phoneme sequences that are then used by a recurrent neural network (RNN) that learns the representations of the phonetic units using past information. [11][3] proposed a single phoneme recognizer to generate the phoneme sequences, then estimating trigram posteriorgram counts to create the representation of the phonetic units. In all these systems, the representations of the phonetic units are used to generate i-vectors that are then combined with acoustic-based i-vector models [12]. Finally, [13] describes a different approach using latent semantic analysis in order to capture salient phonotactics units, as well as [14] that uses phoneme variability factors and language-dependent discriminative information. Different from the previous works, in [15] we proposed the creation of a phonotactic system based on the transformer encoder architecture [16] and a language classifier on top. Besides, we also used a universal phone recognizer that is called Allosaurus [17] which provided some marginal improvements w.r.t. the RNN-based i-vector system [4].

In this work we propose several improvements over previous works. First, through the integration of a sliding attention window [18][19] to handle longer input sequences; secondly, we use larger n-gram phonetic units and different sub-unit tokenizers to limit the vocabulary size of the model and to select the most discriminative units.

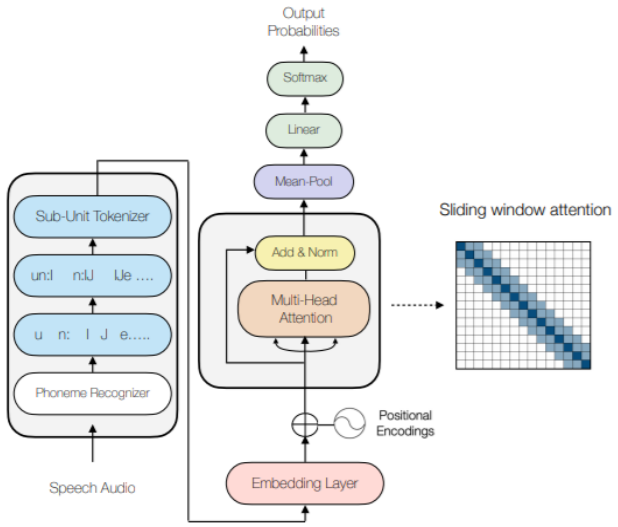

Figure 1*: Proposed architecture for phonotactic language recognition*

Finally, we compare and evaluate the use of the Allosaurus phoneme recognizer [17] w.r.t. the popular SotA Brno phoneme recognizers [20]. This paper is organized as follows: in section 2, we describe the proposed architecture. In section 3, we describe the experimental setup and the data used in this work. Then, in section 4 we present our general and ablation results. Finally, section 5 presents our conclusions and future work.

2. MODEL DESCRIPTION

2.1. Phoneme Recognizer

The first step consists of getting the phoneme sequences from the input speech file. For this task, we compare the use of the universal Allosaurus phoneme recognizer and the Brno phoneme recognizers. The Allosaurus recognizer includes individuals' recognizers for over 2000 languages and a global vocabulary of ~230 IPA phonemes allowing incorporating knowledge for multiple languages. The Brno recognizers are hybrid HMM-NN based phone decoders trained for 3 different languages (Hungarian, Russian and Czech). By using these recognizers, we want to find the one that provides the best performance and compare their performance against previously reported results.

2.2. Phonetic Units

After getting the sequences of phonemes, we create n-gram phonetic units by concatenating contextual phonemes. The usage of high order n-gram increases the vocabulary size and produces scattering issues. To mitigate this problem, we try different sub-unit tokenizers to limit the vocabulary retaining the most common n-gram units and allow the handling of unseen n-gram sequences (OOV). We evaluate the WordPiece tokenizer [29] used in BERT [21] and the Byte-Pair Encoding (BPE) tokenizer [22]. These tokenizers automatically select the most representative phone units, based on maximizing the likelihood of the training data or the frequency of the units, respectively.

2.3. Transformer-based Encoder

The tokenized units are fed to train an embedding layer and positional encodings to obtain the initial representations of the phonetic units; these representations are used by the encoder layer, which consists of a multi-head attention mechanism, a residual connection and a normalization layer. The attention mechanism is intended to learn which units are the most important for the classification problem, and to model short and long-distance dependencies between them. Finally, we apply a mean pooling operation to get the final sentence embedding that is used as input to a linear layer and SoftMax activation function.

2.3.1. Sliding Attention Window

Due to the usage of high order n-gram phonetic units and given the length of the audio files, it is very difficult to handle such long sequences to correctly model short and long dependencies, as well as training the system to pay attention to the discriminative units. Unfortunately, transformer-based models have difficulties processing long sequences due to the self-attention operation, which scales quadratically with the sequence length. To mitigate this limitation, we rely on recent architecture improvements proposed in [18][19], where the attention is focalized, and the computational process reduced, by using sliding windows. The sliding attention window employs a fixed-sized window w that surrounds each token attending half of the tokens on each side (see Figure 1). The advantage is that the computation complexity of this pattern scales linearly with the input sequence length (i.e., non-quadratic) and makes the processing of long input sequences easier.

2.4. MFCC-SDC Acoustic System

Due to the best performance of acoustic-based LR systems, in this work we fuse the scores produced by an i-vector based system using MFCC-SDC parameters with the scores obtained by the transformer encoder. The acoustic system is the optimal configuration reported in [11]. For each speech audio we extract 12 MFCC coefficients including C0 for each frame (25 ms with 10 ms overlap). Silence and noise segments are removed using an energy-based Voice Activity Detector (VAD). To reduce the noise perturbation, a RASTA filter is used together with a cepstral mean and variance normalization (CMNV). Then, we generate a 56-dimensional feature vector created from the concatenation of SDC features using the 7-1-3-7 configuration [23]. Finally, the SDC feature vectors are used to train the UBM

model with 512 Gaussians and the total variability matrix, which are both used to extract 400 dimensional i-vectors.

2.5. Transformer Encoder without attention windows

The baseline system [15] used in this work consists of a phonotactic system that uses the Allosaurus phoneme recognizer, and a transformer encoder with a language classifier on top to perform LR. This architecture achieves the best performance using as input 512 length trigram phone sequences. The main differences between the baseline architecture and the one presented in this work is the incorporation of the sliding attention mechanism, allowing us to manage longer input sequences than the baseline architecture, as well as the use of a sub-unit tokenizer to limit the vocabulary size of the phonetic units and to select the most discriminative units, avoiding scattering issues produced by the large size of the vocabulary.

3. DATABASE AND EXPERIMENTAL SETUP

3.1. Database

For our experiments, we used the Kalaka-3 database [24]. This database contains clean and noisy audio recordings of 6 different languages in the closed-set condition (see Table 1). This dataset is challenging as Catalan, Galician, Portuguese, and Spanish languages are highly similar in their set of used phonetic units.

Table 1: *Statistics of the Kalaka-3 Database*

		Train	Dev	Eval
Languages	Basque	794	70	150
	Catalan	649	79	158
	English	587	81	156
	Galician	975	67	160
	Portuguese	853	84	163
	Spanish	798	77	154
Overall	Nº Files	4656	458	941
	Nº of clean files	3060	-	-
	Nº of noisy files	1596	-	-

3.2. Experimental Setup

To train our model, we experimented with different lengths for the input phonetic sequences to the transformer, getting the best performance using a length of 1024 units and 3-gram as the size of the phonetic units. To avoid truncation and to increase the number of training sequences, speech audios with longer sequences were split into multiple files. This operation is only applied to the training data; for evaluation, we use the first 1024 tokens. In all our experiments, we use a single encoder layer using embedding vectors of size 32 throughout all the operations in the network, a multi-head attention layer with 2 heads, and a sliding attention window of size 128. Finally, we use 6 neurons for the classification layer. We train our model using the Adam Optimizer [25] with a custom learning rate scheduler following the same approach as for the original transformer architecture. The batch size is set to 32 and train for 25 epochs in all experiments. Finally, the performance of our experiments is evaluated using accuracy and the average detection cost function (C_{avg}) [26].

4. RESULTS

4.1. Comparison of phoneme recognizer

4.1.1. Experimental Setup

First, we evaluate and compare the LR results using the Allosaurus and the three Brno phoneme recognizers for Hungarian, Russian and Czech. Since, each recognizer handles a different vocabulary set of phonemes, they generate in some cases longer sequences as well as more training data after we split files with more than 1024 units. For a fairer comparison, we equalize the amount of training data by keeping constant the number of resulting splits for all recognizers. Resulting in 6160 speech training files (the original number was 4656 as reported in Table 1), we also limit and equalize the vocabulary size along all recognizers, to do this we use the Byte-Pair-encoding sub-unit tokenizer due to this tokenizer selects the final units based on their frequency of occurrence. For this, we set the min occurrences to be 5 resulting in a vocabulary size of around 16800 units for all the recognizers.

4.1.2. Results

Table 2 shows the averaged LR results (C_{avg} and Accuracy) comparing the recognizers, together with the standard deviations (by using 5 different initialization seeds for the transformer architecture). Since the Brno recognizers use phoneme sets that are not related to the languages to be recognized (which usually reduce performance), we test the results of the Allosaurus phone set using the full universal phoneme set (IPA) and the corresponding (i.e., similar) one for the Brno recognizers (for a fairer comparison).

The results show that the Allosaurus phoneme recognizer achieves best performance in all cases for Hungarian, Czech, and Russian. Being the Allosaurus Russian recognizer the one that obtains the best performance. On the other hand, the use of the full phoneme set (IPA) in the Allosaurus recognizer shows significant improvements in comparison with the performance of the Russian recognizer; specifically, 5.1% relative improvement in terms of accuracy and 22.6% in terms of C_{avg}. This could be due that using the full set allows the recognizer to incorporate knowledge from the different trained languages, as well as closer matching to the one that naturally will appear in the languages of the Kalaka-3 database. For the rest of the experiments, we use the IPA Allosarus phoneme recognizer.

Table 2: *Averaged LR test results on Kalaka-3 dataset comparing the Brno and Allosaurus phoneme recognizers*

	Brno		Allosaurus	
	Accuracy	Cavg	Accuracy	Cavg
Hung	72.4 ± 0.43	16.5 ± 0.26	78.5 ± 0.39	12.9 ± 0.24
Czech	66.9 ± 0.63	19.8 ± 0.36	78.0 ± 0.39	13.1 ± 0.24
Russian	69.7 ± 0.68	18.0 ± 0.39	*80.7 ± 0.59*	*11.5 ± 0.36*
IPA	-	-	**85.0 ± 0.51**	**8.9 ± 0.29**

4.2. Comparison of sub-unit tokenizer

4.2.1. Experimental Setup

Due to the large vocabulary produced by the creation of high order n-gram units and the sparsity problems that it generates, we decided to use two different sub-tokenization strategies. The use of phoneme sub-tokenization, allows us to retain relevant units from all orders, keep the vocabulary size fixed, and to handle OOV units. We compare the Byte Pair Encoding (BPE) and the Word Piece subword tokenizers, setting the resulting vocabulary size to 30.000 tokens for both; here we did not have to constraint the number of training files as in 4.1.1.

4.2.2. Results

Table 3 shows the results comparing both subword tokenizers. The WordPiece tokenizer obtains a statistically significant relative improvement of 0.9% in Accuracy and 7.2% in terms of C_{avg} in comparison with the performance obtained by the BPE, being this the best performance obtained by our architecture.

Table 3: A*veraged LR test results on Kalaka-3 comparing different tokenizers*

Systems	Accuracy	C_{avg}
Byte Pair Encoding	86.1 ± 0.26	8.3± 0.13
Word Piece	**86.9 ± 0.28**	**7.7± 0.16**

4.3. Phonetic Models and Fusion

In table 4, we show the performance of our proposed system w.r.t. our previous transformer architecture [15], when using only the sliding window, and then the tokenizer. Finally, a comparison with a more complex phonotactic i-Vector system [27]. On the other hand, in table 5 we show the fusion operation with the acoustic system and the comparison with other fused systems and the best system presented in [27] that is the fusion of 6 different models.

Table 4: Individual *Performances (C_{avg})*

Systems	Devel	Eval
Phonotactic i-Vector system [27]	6.94	9.85
Acoustic MFCC [27]	**6.50**	**6.95**
Transformer baseline [15]	8.42	10.21
+ sliding window (this work)	7.45	9.03
+ sliding window & tokenizer (this work)	**6.85**	**7.78**

Table 5: *Fusion Performances on the test set*

Systems	C_{avg}	Imp (%)
Best fusion of 2 models [27]	5.03	-
2 acoustics + Phonotactic i-vector [27]	4.48	10.93
Acoustic + Transformer (this work)	3.62	47.91
Best fusion with 6 models [27]	3.52	49.35

As we can see, in table 4, the phonotactic system presented in this work outperforms the previous transformer architecture presented in [15] as well as a phonotactic i-Vector system [27]. On the other hand, in table 5 we show that the fusion of our new system with an acoustic model provides complementary information that is best than any of the two or three fused models reported in [27], and almost reaches the same performance with the best reported result (which is a combination of 6 different models: 5 acoustic + 1 phonotactic) but in this case with only two systems which make it suitable for real-time or low resource applications.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have described a new phonotactic-based architecture for LR based on using transformer models. The proposed model makes use of local contextual information thanks to the use of high order n-gram units, and long-range information thanks to a sliding window attention mechanism. The paper reports experiments using two different state of the art phoneme recognizers, finding that the universal Allosaurus phoneme recognizer provides the best performance w.r.t. all the Brno recognizers. Besides, the paper describes the use of two subword tokenizers as a mean to reduce the size of the vocabulary used by the transformer and to reduce the sparsity problems that are highly frequent. In this case, the use of the WordPiece tokenizer improves the system performance in ~1% absolute. As future work, we plan to explore other continuous embedding approaches (e.g. X-Vectors), we also want to try our method with the large Nist database.

6. ACKNOWLEDGEMENTS

This work has been supported by the Spanish projects AMIC (MINECO, TIN2017-85854-C4-4-R) and CAVIAR (MINECO, TEC2017-84593-C2-1-R) projects partially funded by the European Union. We also gratefully acknowledge the support of the Universidad Politécnica Salesiana.

7. REFERENCES

[1] R. Cordoba, L.F. D'Haro, F. Fernandez-Martinez, J. Macias-Guarasa and J. Ferreiros, "Language Identification based on n-gram Frequency Ranking", *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[2] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech", *IEEE Translations on Speech and Audio Processing*, pp. 31-35, 1996.

[3] L.F. D'Haro, O. Glembek, O.Plchot, P. Matějka, M. Soufifar, R. Cordoba and J. Černocký, "Phonotactic Language Recognition using i-vectors and Phoneme Posteriogram Counts*", Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[4] C. Salamea, R. de Córdoba, L.F. D'Haro, R.S. Segundo, and J. Ferreiros, "On the use of Phone-based Embeddings for Language Recognition", *IBERSPEECH 2018*, Barcelona, Spain, pp. 55–59, 2018.

[5] D.Romero, C.Salamea, "On the use of Phonotactic Vector Representations with FastText for Language Identification", *Conversational Dialogue Systems for the next decade. Lectures Notes in Electrical Engineering*, vol 704. Springer, Singapore, 2021.

[6] C.Salamea, R.Cordoba, L.F. D'Haro, "Incorporation of Language Discriminative Information into Recurrent Neural Networks Models to LID tasks", *Smart Technologies, Systems and Applications*, Communications in Computer and Information Science, vol 1154, 2019.

[7] C. Salamea, L.F. D'Haro, and R. de Córdoba, "Language Recognition Using Neural Phone Embeddings and RNNLMs", *IEEE Latin America Transactions*, vol. 16, no. 7, pp. 2033–2039, 2018.

[8] C. Salamea, L.F D'Haro, R. de Córdoba, and R.S. Segundo, "On the use of phone-gram units in recurrent neural networks for language identification", *Odyssey*, pp. 117-118.

[9] C. Mayer, M. Nelson, "Phonotactic learning with neural language models". *Proceedings of the Society for Computation in Linguistics*, vol 3, Article 16.

[10] B. Mojan, H. Vydana, A. Vuppala, and M. Shrivastava, "Significance of neural phonotactic models for large-scale spoken language identification", *International Joint Conference on Neural Networks*. pp. 2144-2151, 2017.

[11] L.F. D'Haro, "The GTH-LID System for the Albayzin LRE12 Evaluation", *IBERSPEECH*, Madrid, Spain, pp. 528-539, 2012.

[12] D. Martinez, O.Plchot, L.Burget, O.Glembek and P. Matějka, "Language recognition in i-vectors space".*Twelfth annual conference of the international speech communication association*. 2011.

[13] H. Li, B.Ma, "A Phonotactic Language Model for Spoken Language Identification", *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 512-522, 2005.

[14] X. Wang, Y. Wan, L. Yang, R. Zhou, Y. Yan, "Phonotactic language recognition using dynamic pronunciation and language branch discriminative information", *Speech Communication Journal, Science Direct*, pp. 50-61, 2015.

[15] D. Romero, L.F. D'Haro, C.Salamea, "Exploring Transformer-based Language Recognition using Phonotactic Information", *IBERSPEECH*, Valladolid, Spain, pp. 250-254, 2021.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", *Advances in Neural Information processing systems*, pp. 5998- 6008, 2017.

[17] X. Li, S. Dalmia, J. Li, M. Lee, P. Littell, J. Yao, A. Anastasopoulos et al, "Universal Phone Recognition with a Multilingual Allophone System*", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 8249–8253, 2020.

[18] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang and A. Ahmed, "Big Bird: Transformers for Longer Sequences", *NIPS,* 2020.

[19] I. Beltagy, M.E. Petters and A. Cohan. Longformer: The long-document transformer, *arXiv preprint arXiv:2004.05150*, 2020.

[20] P. Schwarz, "Phoneme Recognition based on Long Temporal Context, PhD Thesis", Brno University of Technology, 2009.

[21] J. Devlin, M.W. Chang, K.Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *arXiv:1810.04805*, 2018.

[22] R. Sennrich, B. Haddow, A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *arXiv:1508.07909*, 2016.

[23] P. 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", *Seventh international conference on spoken language processing*, 2002.

[24] L.J. Rodríguez-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". *Language Resources and Evaluation*, 50(2), pp.221-243. 2016.

[25] P. Diederik, K. Ba, J. Ba, "Adam: A method for stochastic optimization", *arXiv:1412.6980,* 2014.

[26] A. Martin. and C.S. Greenberg, "The 2009 NIST Language Recognition Evaluation*", Speaker and Language Recognition Workshop*, IEEE Odyssey (Vol. 30). 2010.

[27] D'Haro LF, Cordoba R, Salamea C, Echeverry JD. "Extended Phone Log-Likelihood Ratio Features and Acoustic-Based i-Vectors por Language Recognition", *ICASSP IEEE International Conference on Acoustics Speech and Signal Processing*, Italy.

[28] Snyder D, Garcia-Romero D, McCree A, Sell G, Povey D, Khudanpur S. Spoken language recognition using x-vectors. In *Odyssey 2018* Jun (pp. 105-111).

[29] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint* arXiv:1609.08144.