IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

Exploring Transformer-based Language Recognition using Phonotactic Information

David Romero[1], *Luis Fernando D'Haro*[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
dromeromog@gmail.com, lfdharo@die.upm.es, csalamea@ups.edu.ec

Abstract

This paper describes an encoder-only approach based on the "Transformer architecture" applied to the language recognition (LRE) task using phonotactic information. Due to the use of one global set of phonemes to recognize all languages, the proposed system needs to overcome difficulties due to the overlapping and high co-occurrences of similar phone sequences across languages. To mitigate this issue, we propose a single transformer-based encoder trained for classification, where the attention mechanism and its capability of handling large sequences of phonemes help to find discriminative sequences of phonotactic units that contribute to correctly identify the language for short, mid and long audio segments. The proposed approach provides significant improvements, outperforming phonotactic-based RNNs and Glove-based i-Vectors architectures, getting a relative improvement of 5.5% and 38.5% respectively. Our experiments were carried out using phoneme sequences obtained by the "Allosaurus phoneme recognizer" applied to the Kalaka-3 Database. This dataset is challenging since the languages to identify are mostly similar (i.e. Iberian languages, e.g. Spanish, Galician, Catalan). We provide results using the C_{avg} metric proposed for NIST evaluations.

Index Terms: language recognition, phonotactic information, self-attention mechanism.

1. Introduction

Phonotactic-based LRE attempts to recognize the language that is spoken in a speech sample using as input the sequence of phonemes obtained by a phoneme recognizer, which uses a global set of phonemes to transcribe each audio file in the corpus with independency of the languages to be recognized. This global set contributes to make the system easier, in contrast with traditional approaches like Parallel Phone Recognition Language Modeling (PPRLM) [1], but introduces difficulties due to the overlapping and high co-occurrences of similar phonemes between languages. Therefore, finding discriminative combinations of these units is one of the main goals to differentiate transcriptions generated by similar languages.

Recent approaches to phonotactic LRE have used Recurrent Neural Networks (RNN's) [2][3] to take advantage of its recurrent ability for processing sequences using past information. However, if we consider that these phonetic sequences are usually long, this type of neural network could suffer from vanishing gradient problems [4] during training, which lead to performance degradation. Other approaches [5] have used i-Vector-based frameworks that take as input phonotactic-based embeddings learned through Skip-Gram [6] and Glove [7] models. While powerful, these approaches use small context windows limiting the use of longer context information that could be useful to find discriminative combinations.

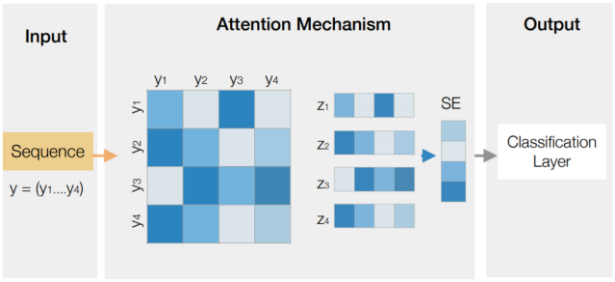

Figure 1: *High level overview of our approach. The attention mechanism is used to find discriminative combinations of phonetic units.*

In this work, we consider the use of an encoder-only approach based on the Transformer architecture [8] for the LRE task using phonotactic information. Models based on Transformers have shown their versatility and robustness [9] to perform various tasks such as seq2seq translation [8], sentiment analysis [10], or language understanding [11], commonly achieving state-of-the-art results outperforming architectures like Long Short-Term Memory (LSTM's) [12]. We attempt to take advantage of its self-attention mechanism that allows the network to use and capture contextual information from long sequences, helping to find discriminative combinations of the global phonetic units.

Figure 1 shows an overview of our approach. We use as input an embedding sequence $(y_1,..,y_L)$, where each embedding represents a n-gram phonetic unit of a given input sequence. Then, contextual vector representations of these units are learned together with the self-attention weights, allowing each unit to attend to the other units in the sequence capturing contextual information and helping to find the most relevant tokens to make each final representation $(z_1,\dots,z_L)$. Finally, we use average pooling to obtain the final sentence embedding (SE) which is the input to a classification layer. Our experimental results show that the proposed architecture outperforms RNN's and i-Vector based systems for modeling phonotactic information only.

10.21437/IberSPEECH.2021-53

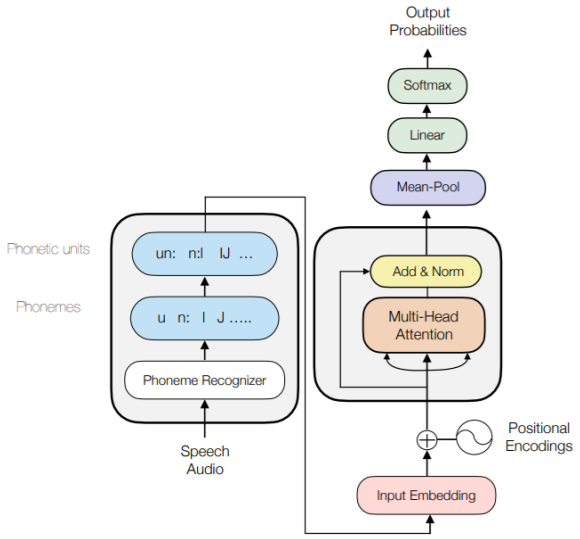

Figure 2*: An overview of the Transformer-based Encoder applied for LRE using phonotactic information.*

This paper is organized as follows: in Section 2 describes the proposed architecture. Section 3 explains the experimental setup and describes the database and metrics used in our experiments. Section 4 discusses the obtained results. And, finally, Section 5 shows the conclusions and future work.

2. Model Description

In this section we describe the proposed model in detail. Figure 2 shows the different components of the architecture with special emphasis on the simplified transformer-based encoder used in this paper. Below, we provide detailed information for each component in the figure.

2.1. Phoneme recognizer

The first step is the recognition of phoneme sequences for each audio file in the corpus, using the "Allosaurus Phoneme recognizer" [13]. This recognizer incorporates some phonology knowledge in its model through an allophone layer, which associates a universal narrow phone set with the phonemes that appear in each language, making it possible to perform universal phone recognition. This allows the model to incorporate individual recognizers for over 2000 languages and a global recognizer called "IPA" that use a vocabulary of around 230 phonemes using the inventory of all the languages supported in the recognizer, which includes the languages of the corpus that is used in this work.

For this task we use a different approach than other works based on RNN's [2,3] and i-Vectors [5], that use a phoneme recognizer developed by the Brno University of Technology (BUT) [14], which has only 3 recognizers for Hungarian, Russian and Czech with 61, 46 and 52 different phonemes respectively. These approaches use only one of these recognizers to make phoneme representations for all the languages in the corpus. In our case, we use the Allosaurus phoneme recognizer using the global vocabulary, with which we obtain better performance; on the one hand, due to the better recognition using a global set of phonemes that summarizes all the languages in the corpus rather than using a single model that incorporates information of only one of them; on the other hand, it has a larger phoneme vocabulary set than the Brno Recognizer, allowing more variability in the transcriptions (this will be explained in more detail in the next subsection) which ended showing benefits to the transformer-based encoder. We believe this variability helps the model to find unique tokens and therefore discriminative combinations that benefit the LRE task.

2.2. Phonetic units

To incorporate more context information as input to the architecture, we used n-gram phonetic units. The use of these units has shown better performances [4] than just using phonemes thanks to the incorporation of local information and diversity when learning the vector embeddings for each unit. Let $x = (x_1, x_2, \dots, x_L)$ be a sequence of recognized phonemes of length L; we form phonetic units by concatenating two or more phonemes into one new n-gram unit, generating the new sequence $x_n = (x_{1:2}, x_{2:3}, \dots, x_{L-1:L})$, the start of sentence (SOS) and end of sentence (EOS) tokens have been added at the beginning and end of this sequence that will be used as input to the transformer-based encoder. We experimented with different sizes for these phonetic units (n-gram order) which will be presented in the next section. The creation of these units is shown in the top left block of Figure 2.

2.3. Transformer-based Encoder

The original transformer [8] consists of stacked encoder and decoder layers. In this work we have used only an encoder-based approach trained for classification. In this subsection we describe the encoder layer and how it was applied for our work, please refer to Figure 2 for this discussion.

2.3.1. Input

The input of the encoder layer is a sequence of phonetic units $x = (x_{1:2}, x_{2:3}, \dots, x_{L-1:L})$, each unit in this sequence is tokenized using a "Word tokenizer". Once this tokenization is done, each token is represented by an embedding that is learned during training, this input sequence is represented as $y = (y_{1:2}, y_{2:3}, \dots, y_{L-1:L})$ where $y_i \in \mathbb{R}^{d_{model}}$.

2.3.2. Positional Encodings

Since the transformer contains no recurrence to make use of the order of the sequence, it uses "Positional encodings" that are added to the input embeddings at the bottom of the layer. The positional encodings have the same dimension d_{model} as the input embeddings and are represented as $p = (p_1, \dots, p_{L-1})$ where $p_j \in \mathbb{R}^{d_{model}}$.

2.3.3. Encoder

The encoder layer used in this work is a reduced version of the original transformer encoder [8]. Our architecture (see Figure 2) uses only the multi-head self-attention mechanism along with a residual connection and a layer normalization, which allows to jointly attend to information from different representations subspaces at different positions, performing the attention function in parallel. The reason we do not implement the second sublayer of the original transformer encoder, which is a fully connected feed-forward network, is to reduce the number of model parameters and to avoid the overfitting produced by the small size of the database used in our experiments (see section 3.2). Therefore, when using the multi-head self-attention mechanism each attention head operates on

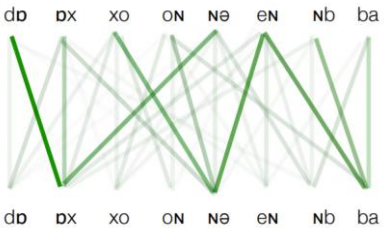

Figure 3: *Behavior of the attention mechanism given a phonetic sequence.*

the input sequence y and computes a new sequence $z = (z_{1:2}, z_{2:3}, \dots, z_{L-1:L})$ where $z_i \in \mathbb{R}^{d_{model}}$.

This final sequence is obtained using the attention-mechanism allowing each token in the input sequence to attend to the other tokens to find those phonetic units that are most important to learn each representation, allowing modeling short and long-distance dependencies between them. We attempt to take advantage of this ability to attend to the most discriminative phonotactic units that could be representative for each language. An example of the attention mechanism applied to a phonetic sequence is shown in Figure 3.

2.3.4. Output

Finally, to get a single representation from the final sequence of embeddings z we choose the average-pooling operation. We apply this operation due to its benefits for tasks with long input sequences [15]. Here we experimented with other techniques such as max-pooling, max attention and the use of the classification token (CLS). However, we obtained better results using average pooling. Next, we use a classification layer with a SoftMax activation given the generated sentence embedding.

3. Database and Experimental Setup

In this section we describe the database used in our experiments, the model parameters and the metric used to measure our performance.

3.1. Database

For our experiments, we used the Kalaka-3 database [16]. This database contains clean and noisy audio recordings of 6 different languages in the closed-set condition (i.e. Basque, Catalan, English, Galician, Portuguese, Spanish), including 108 hours of speech in total. Some relevant statistics of the train, test and validation sets are shown in Table 1.

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
Structure	Nº Files	4656	458	941
	Nº of clean files	3060	-	-
	Nº of noisy files	1596	-	-
	Length <= 30s	2855	121	267
	Length 120s	1801	337	674

As explained before we apply a phoneme recognizer to the audio files to get phoneme sequences, which are later used to form phonetic units. To train our model, we set the number of phonetic units in each sequence to 512 tokens. If any sequence has more, we split it in multiple parts in order to make more training data. We perform this operation only for training, during evaluation we just use the first 512 tokens.

3.2. Experimental Setup

To train our model we attempted to reduce the number of parameters used by the architecture as much as possible due to the small size of the database. In all our experiments, we used a single encoder layer that uses an embedding of size 32 throughout all the operations in the network and a multi-head attention with 2 heads. The phonetic embeddings are then passed to the average-pooling operation to get the final sentence embedding. Finally, this sentence embedding is passed to a final classification layer to obtain per-class probability distributions. We train our model using the Adam Optimizer [17] with a custom learning rate scheduler according to the formula proposed in the original transformer architecture, using as parameters $\beta_1 = 0.9, B_2 = 0.98, \varepsilon = 10^{-9}$. The batch size is set to 64 for all experiments. We train for 25 epochs and select the best model according to the best C_{avg} value. We select this metric to evaluate our model and it is described in the 3.3 subsection [18].

3.3. Average Detection Cost Function

The performance of our experiments is evaluated using accuracy and the average detection cost function (C_{avg}). The latter, is an evaluation metric proposed by NIST that weights the number of false acceptances and false alarms generated by the network, representing them as a detection cost function

$$C_{avg} = \frac{1}{N_L} \cdot \sum_{L_T} \begin{cases} C_{miss} \cdot P_{Target} \cdot P_{miss}(L_T) \\ + \sum_{L_N} C_{FA} \cdot P_{Non-Target} \cdot P_{FA}(L_T, L_N) \\ + C_{FA} \cdot P_{Out-of-Set} \cdot P_{FA}(L_T, L_O) \end{cases} \quad (1)$$

Where:

N_L is the number of languages in the (closed-set) test, L_T is the target language, L_N is the non-target language, L_O is the Out-of-Set "language", $C_{miss} = C_{FA} = 1$, $P_{Target} = 0.5$,

$P_{Out-of-Set} = 0.0$ $(for\ the\ plenty\ closed\ condition)$,

$$P_{NONtarget} = \frac{1 - P_{Target} - P_{out-of-set}}{(N_L - 1)} \quad (2)$$

4. Results

4.1. Experimental Results

The results are presented in Table 2. Here, we show the performance of our model for the development and evaluation sets using different sizes for the phonetic units (i.e. n-gram order). Unfortunately, increasing the n-gram order increases the number of combinations and therefore the number of phonetic units in the corpus; using 2-grams and 3-grams we get a vocabulary of around 10,000 and 200,000 respectively. For this reason, we fixed the vocabulary size used for the word tokenizer. The table shows the best performance we got by using 3-gram phonetic units and setting the vocabulary size to the 30k most common units. We think that this best

performance is due to a better modeling of the local context and that the model is capable of finding more discriminative combinations. On the other hand, we found that when using higher order n-grams there were scattering issues, the training time was increased, and we did not get better performance.

Table 2: *Results*

Size	#Tokens	Dev		Eval	
		Accuracy	Cavg	Accuracy	Cavg
2-gram	5000	85.8	8.6	82.1	10.6
3-gram	30000	**86.0**	**8.4**	**82.7**	**10.2**

In our initial tests, we found that using the original set of BUT Hungarian phonemes performed worse than using the Allosaurus phoneme set. Therefore, all our experiments use the latter. In future work, we will carry additional experiments to better evaluate the reason for this and the possibility of combining information from both sets.

4.2. Baselines

In this subsection we describe the baseline systems which are used to compare the performance of our approach.

4.2.1. RNNs-based architecture using neural phone embeddings

This approach [2] uses phonetic sequences obtained by the Brno recognizer using a PPLRM system, which is followed by a Recurrent neural network that is used to learn phonetic-based embeddings. These phonetic-representations are then used to compare with trained RNN Language Models to obtain entropy values for each language which are used as input to a multi-class logistic classifier. This architecture also uses vocabulary reduction techniques in order to replace rare phonetic units by others that are more discriminative for each language. The best result given in this work uses a size of 3-grams for the phonetic units and k-means clustering as the technique used to reduce the vocabulary.

4.2.2. I-Vectors-Based architectures using Skip-Gram and Glove Models

This work [5] proposed an i-Vector based architecture that uses as input phonetic-based embeddings learned using Skip-Gram and Glove Models. This architecture uses a Multiple Vector Embedding approach (MVE), which consists of using phone-based embeddings trained for each language individually (Language Phone-Based Embeddings) to obtain i-Vectors for each language. Then, these i-Vectors are used as input to a multi-class logistic regression classifier to perform LRE. Finally, the scores provided by each individual language-dependent system are fused to obtain a better performance.

The phoneme sequences used in this approach were obtained by the Brno Recognizer using the Hungarian vocabulary, and its best results are given using a size of 2-grams for the phonetic units

4.3. Comparison to Baseline Algorithms

The comparison of our performance with the best results of our baselines is shown in Table 3. Our approach outperforms all the baselines given an improvement of 45% and 38.5% to i-Vectors based architectures that use the Skip-Gram and Glove Models respectively, and an improvement of 5.5% to the RNN-based architecture.

Table 3: *Comparison of Performance for eval set*

Systems	Cavg	Improvement
Transformer-Based Encoder	**10.2**	(Ours)
i-Vector approach using Skip-gram	18.7	45%
i-Vector approach using Glove	16.7	38.5%
RNN-based architecture	10.8	5.5%

5. Conclusions and Future Work

In this work we show how using a reduced version of the transformer architecture can be beneficial for the LRE task when using phonotactic information. We took advantage of the attention-mechanism used by the transformer-based encoder to find discriminative combinations of phonetic units that helps the model to recognize a language. The model tackles the main hurdle of this task which is the overlapping and high-cooccurrences of phonetic units between languages by learning phonotactic embeddings through attention, attending to the most important tokens in the sequence to find these useful short and long-distance combinations. Our approach showed that it can outperform RNN's and i-Vector based architectures that employ Skip-gram and Glove models. For future work, we plan to apply this model to larger databases. We will also evaluate our approach using longer sequences as input to the model by using different attention windows (as the one used by recent approaches such as the longformer [19] or bigbird [9]) and to use sub-word tokenizers to take advantage of all the vocabulary in the corpus.

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

[2] 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.

[3] 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", *in Odyssey*, pp. 117-118.

[4] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. A Field Guide to Dynamical Recurrent Neural Networks", *IEEE*, pp. 237–243, 2001.

[5] 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" in *IBERSPEECH 2018 – November 21-23, Barcelona, Spain, Proceedings*, 2018, pp. 55–59.

[6] D. Guthrie, B. Allison, W. Liu, L. Guthrie and Y. Wilks, "A closer look at Skip-Gram modelling", *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pp. 1–4, 2006.

[7] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representations", *Proceedings of conference on empirical methods in natural language processing*, pp. 1532–1543, 2014.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need",

in Advances in Neural Information processing systems, pp. 5998-6008, 2017.

[9] 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", *in NIPS – 2020*.

[10] C. Sun, L. Huan, and X. Qiu, "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computer Linguistics: Human Language Technologies*, vol.1., pp. 380-385, 2019.

[11] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of bidirectional transformers for language understanding", *in NAACL-HLT*, arXiv: 1810.04805.

[12] S. Hochreiter, and J. Schmidhuber, "Long short-term memory", *in Neural computation*, 9(8), pp. 1735-1780, 1997.

[13] 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.

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

[15] P. Maini, K. Kolluru, D. Pruti, Mausam, "Why and when should you pool? Analyzing Pooling in Recurrent Architectures", *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4568-4586.

[16] 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*, pp. 221-243, 2016.

[17] P. Diederik, Kingma and Jimmy Ba, "Adam: A method for stochastic optimization", 2017. arXiv: 1412.6980

[18] A. Martin and C. Greenberg, "The 2009 NIST Language Recognition Evaluation", *in Speaker and Language Recognition Workshop, IEEE Odyssey*, pp. 165-171,2010.

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