



Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal

Lukas Mateju, Petr Cerva, Jindrich Zdansky, Radek Safarik

Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic

{lukas.mateju, petr.cerva, jindrich.zdanský, radek.safarik}@tul.cz

Abstract

This paper investigates the use of deep neural networks (DNNs) for the task of spoken language identification. Various feed-forward fully connected, convolutional and recurrent DNN architectures are adopted and compared against a baseline i-vector based system. Moreover, DNNs are also utilized for extraction of bottleneck features from the input signal. The dataset used for experimental evaluation contains utterances belonging to languages that are all related to each other and sometimes hard to distinguish even for human listeners: it is compiled from recordings of the 11 most widespread Slavic languages. We also released this Slavic dataset to the general public, because a similar collection is not publicly available through any other source. The best results were yielded by a bidirectional recurrent DNN with gated recurrent units that was fed by bottleneck features. In this case, the baseline ER was reduced from 4.2% to 1.2% and C_{avg} from 2.3% to 0.6%.

Index Terms: language identification, Slavic languages, deep neural networks, convolutional neural networks, recurrent neural networks

1. Introduction

Spoken language identification (LID) is the task of correctly determining the language spoken in a speech utterance. In recent years, many scientific efforts have been dedicated to this task, and nowadays, LID modules form an integral part of many speech processing applications including, e.g., systems for multilingual speech recognition or spoken language translation. LID systems are also used for spoken document retrieval, emergency call-routing or in dialog systems. Although the accuracy of all these systems is constantly improving, it is still not perfect. For example, one of the significant bottlenecks of LID systems is to distinguish between closely related languages.

Most of the state-of-the-art LID systems utilize various advanced acoustic modeling techniques.

One of the most popular techniques relies on the total variability factor analysis, and it is known as an i-vector framework [1, 2]. I-vector is a fixed length representation of an utterance, and it jointly contains information about the speaker, language, etc. (e.g., LDA might be applied to obtain discriminative features). To extract i-vector features, hand-crafted shifted delta cepstral features (SDC) derived from mel-frequency cepstral coefficients (MFCCs) [3] and phone log-likelihood ratios (PLLRs) [4] are most commonly used as inputs. The i-vector extraction is usually followed by a classification stage, where multiclass logistic regression, cosine scoring or Gaussian models are utilized. The major drawback of the i-vector approach is the decreasing performance on shorter test utterances [5].

Over the past few years, deep neural networks have had an upsurge in popularity in LID systems thanks to their outstanding

performance in many other speech processing applications (e.g., speech recognition [6]). Both direct and indirect approaches exist for utilizing deep learning for LID.

In the former case, so-called bottleneck features (BTNs) are widely used in many systems [7, 8, 9] due to their superior performance. Usually, these features are extracted from a DNN trained to discriminate individual physical states of a tied-state triphone model at first, and then used as inputs to an i-vector based system [10, 11].

In the latter case, various end-to-end systems based on different DNN architectures are trained to identify the language in the input utterance. In 2014, feed-forward DNN yielded excellent results on short utterances (less than 3 seconds) [5]. Since then, other more advanced architectures, such as attention based DNNs [12], convolutional neural networks (CNNs) [13, 14, 15], time delay neural networks (TDNNs) [16, 17] or sequence summarizing neural networks (SSNNs) [18] have also been successfully used. The most recent direct approaches take advantage of recurrent neural networks (RNNs) and their context modeling ability. Gated recurrent unit (GRU) RNNs [19], long short-term memory (LSTM) RNNs [20, 21, 22, 23, 24] and bidirectional LSTM RNNs [25, 26] all yield the state-of-the-art performance.

In this paper, various state-of-the-art LID methods are investigated. We adopt feed-forward DNNs at first, then CNNs, and finally also unidirectional as well as bidirectional RNNs with both previously mentioned types of units. We also combine these direct methods with the indirect approach: we feed the networks with bottleneck features. To the best of our knowledge, results of some of these approaches and their comparison on one dataset have not yet been published for LID.

The experimental evaluation is performed on a dataset consisting of the 11 most widespread Slavic languages. These were selected for two main reasons.

The first is that most of these languages are related to each other which makes our dataset more challenging. This is especially true for those pairs of languages that belong to the same language branch. For example, it is difficult to distinguish between Croatian and Serbian (South Slavic branch), even for native speakers.

Secondly, only results obtained for several (pairs) of Slavic languages have been published so far (e.g., [27]). For example, Polish and Russian formed one cluster of related languages within the last Language Recognition Evaluation (LRE) challenge in 2017 [28]. On the contrary, a detailed analysis for all evaluated Slavic languages using a confusion matrix is presented in this work.

Finally, note that our dataset of Slavic languages is available for download to the general public¹.

¹<https://owncloud.cesnet.cz/index.php/s/gXHkFs9UDEqe34G>

2. Dataset of Slavic languages

Slavic languages are spoken by approximately 320 million people throughout Eurasia mostly in Central, Eastern, and Southern Europe. There are at least ten languages with over a million native speakers (e.g., Russian (~150 million speakers), Polish (~55 million speakers), Czech (~11 million speakers), etc.).

Slavic languages can be divided on the basis of geographical and genealogical principle into three main branches: East, South and West Slavic languages. Most of the Slavic languages belonging to the same branch are somehow close to each other. However, while some languages may have similar phonetics (e.g., Croatian and Serbian are practically identical in their phonetics), some languages may have different phonetics (e.g., Polish or Bulgarian are phonetically somewhat more similar to East Slavic languages than to languages in their branch). Every Slavic language has its unique phonetic inventory which distinguishes it from other languages (except for the previously mentioned Croatian and Serbian). It can help with language identification. Moreover, rich morphology, a high degree of inflection, and more or less free word order result in a large linguistic complexity of all these languages.

Due to the lack of an extensive audio dataset for all Slavic languages, we had to create a new one. It is compiled from recordings belonging to the 11 most widespread Slavic languages so that it covers all three branches:

- East Slavic languages - Belarusian, Russian, Ukrainian,
- South Slavic languages - Bulgarian, Croatian, Macedonian, Serbian, Slovene,
- West Slavic languages - Czech, Polish, Slovak.

The source of data for individual languages varies. A majority of the data originates in TV and radio broadcasts, and it was retrieved as described in detail in [29]. It formerly served for acoustic model training for speech recognition, and thus it contains mostly clean speech. The rest of the dataset is formed by microphone recordings.

The data for each language is compiled from recordings belonging to multiple speakers (with both genders represented). It is divided into two non-overlapping subsets: 20 hours of recordings are available for training and 500 utterances for evaluation. The focus is on short recordings (similar to [5]), the maximal duration of an evaluation recording is 5 seconds.

3. Evaluation metrics

Within the scope of this paper, two different performance metrics (namely error rate (ER) and C_{avg}) were utilized to evaluate the performance of LID approaches.

The first metric, error rate, is defined as:

$$ER[\%] = \frac{M_{utt}}{N_{utt}} * 100, \quad (1)$$

where M_{utt} is the number of misclassified speech utterances, and N_{utt} is the total number of evaluated speech utterances.

The second metric is the official metric of the 2015 NIST Language Recognition Evaluation, C_{avg} . Detailed information about this closed set multi-language cost function and its definition can be found in the 2015 LRE Plan [30].

4. Investigated approaches and results

4.1. Acoustic features used

Three different types of 39-dimensional feature vectors were extracted within all of the following experiments: MFCCs (13-dimensional with Δ and $\Delta\Delta$ coefficients), filter bank coefficients (FBCs) and bottleneck features. Both MFCCs and FBCs were computed using 25 ms frames of the signal with frame shifts of 10 ms. As suggested, e.g., in [7, 8, 9], bottleneck features were extracted from DNN trained to discriminate physical states (senones) of a Czech tied-state triphone acoustic model. This DNN was trained on 270 hours of speech recordings belonging to the Czech language and using hyper-parameters as follows: 5 hidden layers with the third one being the bottleneck layer, 1024 neurons per hidden layer (39 for the bottleneck layer), ReLU activation function (sigmoid for the bottleneck layer), a learning rate of 0.08 and 50 epochs of training.

The input for DNNs consists of 11 consecutive FBC vectors, 5 preceding and five following the current frame. Normalization of these vectors was performed within a 1-second long window.

4.2. Baseline i-vector approach

To set a baseline performance, an i-vector system was trained using a full covariance GMM-UBM based system and logistic regression model. Within this training, MFCCs filtered by voice activity detection were employed, and the final extracted i-vectors were 600-dimensional. Note that this baseline approach follows the Ire07 recipe as present in Kaldi ASR².

The results yielded by this system are presented in the first row of Table 1. They provide a decent baseline (i.e., ER of 4.2% and C_{avg} of 2.3%) for further experimental work.

4.3. Feed-forward fully connected DNN architecture

The first adopted deep learning architecture was a feed-forward fully connected DNN. Its output was formed by a softmax layer with 11 neurons (this value corresponds to the number of languages). This DNN was trained using Torch framework³ to directly distinguish between languages (i.e., direct method). The hyper-parameters used for training were similar to those for extraction of the bottleneck features (i.e., 5 hidden layers, 1024 neurons per hidden layer, ReLU activation function, a learning rate of 0.08 and 20 training epochs).

During the classification phase, a probability vector was obtained for each frame of given utterance (i.e., by doing a forward pass). These vectors were then averaged, and the language with maximum average probability was selected as an output.

The obtained results for all three types of considered feature vectors are summarized in Table 1. They show that MFCCs slightly outperformed FBCs, but the difference was rather small. It is also evident that the direct approaches (using both MFCCs and FBCs) did not exceed the baseline i-vector based system. On the contrary, the baseline system was outperformed significantly by bottleneck features (see the fourth row of Table 1). The improvement was over 2 % in ER (from 4.2% to 2.0%) and over 1% in C_{avg} (from 2.3% to 1.1%).

Note that we also performed several experiments (not presented in this paper) with the size of the bottleneck layer, but no further reduction in error rate was obtained.

²<http://kaldi-asr.org/>

³<http://torch.ch/>

Table 1: Results of feed-forward fully connected DNN for different types of features in comparison to baseline i-vector system.

approach	ER [%]	C_{avg} [%]
LR + i-vectors	4.2	2.3
DNN + MFCCs	5.7	3.1
DNN + FBCs	5.9	3.3
DNN + BTNs	2.0	1.1

4.3.1. The influence of context window size

The next experiment was focused on the importance of the size of input feature context window. The reason is that additional context information may be beneficial for reduction of the error rate of the system. On the contrary, broader context slows down the training and evaluation phases. Several DNNs with a variety of context window sizes from 5-1-5 (i.e., 0.1 seconds long window) up to 50-1-50 (i.e., 1 second) were trained using bottleneck features and evaluated.

The reached results are summarized in Table 2. They show that our initial context window size was too short and degraded the performance. The ideal context window size seems to be longer and around 15-1-15 (i.e., 0.3 seconds). In this case, ER decreased from 2.0% to 1.2% and C_{avg} from 1.1% to 0.7%.

Table 2: Results for different context window size in the system with bottleneck features and feed-forward fully connected DNN.

context window size	ER [%]	C_{avg} [%]
5-1-5	2.0	1.1
10-1-10	1.3	0.7
15-1-15	1.2	0.7
20-1-20	1.2	0.7
25-1-25	1.5	0.8
50-1-50	6.6	3.6

4.4. CNN architectures

The next type of DNN networks we focused on were convolutional networks. In contrast to [13], we also tried to utilize the bottleneck features.

The employed CNNs were composed of two convolutional layers and three fully connected layers (each with 1024 neurons). The inputs consisted of 31 feature maps (i.e., context window size of 15-1-15), each 39×1 in size. Our experiments were performed with FBCs and BTn features. The first convolutional layer was comprised of 105 feature maps 39×1 in size followed with a 3:1 max-pooling layer. The second consisted of 157 feature maps 13×1 in size. The rest of the hyper-parameters was set as stated in Sect. 4.3, and the CNNs were also trained using Torch framework.

To explore deeper configurations of CNNs, an additional max-pooling and third convolutional layer (209 feature maps 13×1 in size) were added, and the CNNs were trained.

The achieved results are depicted in Table 3. As expected, the BTn features outperformed FBCs by a large margin (by almost 4%). The more interesting fact is that the difference in performance of DNNs and CNNs was practically negligible. There was no gain in using more complex architecture. The deeper configuration of CNN only worsened the results.

Table 3: Results of architectures based on CNNs.

approach	conv. layers	ER [%]	C_{avg} [%]
CNN + FBCs	2	4.9	2.7
CNN + BTNs	2	1.3	0.7
CNN + FBCs	3	6.4	3.5
CNN + BTNs	3	1.7	0.9

4.5. RNN architectures

The last DNN architecture we explored was the recurrent neural network. At first, we focused on long short-term memory RNN architecture (e.g., [20, 31]), but unlike these cited papers, we also investigated the use of bottleneck coefficients. After that, we examined the gated recurrent unit RNNs [19]. Finally, we also explored the possibilities of bidirectional RNNs. We studied slightly different configurations of bi-LSTM RNNs as in [26].

The RNNs were comprised of two recurrent layers (each with 1024 neurons) and two fully connected layers (1024 neurons per layer). The inputs were once again FBCs, and BTn features with a context window size of 15-1-15. The rest of the hyper-parameters remained the same as in Sect. 4.3. The RNNs were trained using the ADAM optimizer in PyTorch⁴. Note that for unidirectional models, the final language for each utterance was obtained by averaging only last 10% of frame probabilities, as suggested in [31], to exploit the learning capabilities of RNNs.

The results are summarized in Table 4. First, as in previous experiments, BTn features outperformed the FBCs. However, the difference in performance was distinctly smaller than that of CNN & DNN architectures. Recurrent neural networks can thus extract more information about the target language from standard acoustic features. Secondly, GRU RNNs exceeded the LSTM RNNs in performance (e.g., 1.4% vs. 1.2% ER). Next, the bidirectional RNNs performance was mixed. Although the bi-LSTM RNNs performed slightly worse than the unidirectional equivalent, the bi-GRU RNNs outperformed its counterpart. However, the gain in performance was rather small (less than 0.1% in ER and 0.1% in C_{avg}). Furthermore, the difference in results between feed-forward fully connected DNN and bi-GRU RNN was rather low as well (0.1% in C_{avg}). The bi-GRU RNN yielded slightly better results but at the cost of more complex architecture to train and evaluate. Finally, the performance of the bidirectional GRU RNN was the best throughout this paper.

Table 4: Performance of systems based on RNNs.

approach	ER [%]	C_{avg} [%]
LSTM + FBCs	3.0	1.7
LSTM + BTNs	1.4	0.7
GRU + FBCs	2.5	1.4
GRU + BTNs	1.2	0.7
bi-LSTM + FBCs	3.0	1.6
bi-LSTM + BTNs	1.5	0.8
bi-GRU + FBCs	2.4	1.3
bi-GRU + BTNs	1.2	0.6

⁴<http://pytorch.org/>

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	472	10	4	3	1	0	0	0	9	1	0
SK	5	483	5	0	0	0	3	1	0	3	0
PL	9	5	478	5	0	0	0	0	1	1	1
RU	5	3	9	470	1	2	0	0	8	1	1
SI	0	1	0	0	479	4	11	0	2	1	2
UA	1	0	1	1	0	481	0	0	3	9	4
RS	0	3	0	0	10	2	477	1	5	0	2
MK	0	1	0	0	0	2	4	489	1	0	3
HR	8	1	2	5	1	0	6	0	476	0	1
BY	0	1	1	0	4	15	0	2	2	471	4
BG	0	0	0	0	3	0	2	2	0	0	493

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	497	3	0	0	0	0	0	0	0	0	0
SK	3	489	4	0	1	0	0	0	2	1	0
PL	1	3	493	1	0	0	0	0	1	0	1
RU	3	1	5	490	0	0	0	0	1	0	0
SI	0	0	0	0	495	1	3	1	0	0	0
UA	0	0	0	0	0	495	0	1	0	4	0
RS	0	2	0	0	5	1	484	0	8	0	0
MK	0	1	0	0	0	1	0	493	1	0	4
HR	2	0	0	3	0	0	4	0	491	0	0
BY	0	0	0	0	0	11	1	0	0	487	1
BG	0	1	0	0	0	0	0	1	0	1	497

Figure 1: Comparison of confusion matrices produced by baseline i-vector system (left) and the best bi-GRU RNN system (right). (CZ - Czech; SK - Slovak; PL - Polish; RU - Russian; SI - Slovene; UA - Ukrainian; RS - Serbian; MK - Macedonian; HR - Croatian; BY - Belarusian; BG - Bulgarian)

4.6. Error analysis and confusion matrices

More detailed results obtained in the form of confusion matrices are depicted in Figure 1. They show that most of the errors are confusions between related languages. These are mostly caused by a common phonetic inventory but also because the languages have some common words in vocabulary and similar phonotactics, as a wider context is used for identification.

For example, the highest value of errors is between Belarusian and Ukrainian. These languages are more similar to each other than to Russian as they phonetically differ only in a few phonemes and they have similar vocabularies. A comparable case is between Croatian, Serbian and Slovene since the first two have the same phonetic inventory and Slovene differs from both of them only in few phonemes. Vocabularies of these languages are also very similar. Also all west Slavic languages are confused with each other. However, their phonetic inventories are not so close, and the source of confusion may be similar vocabularies and phonotactics.

On the other hand, in some other cases, the confusions are harder to explain and may lay in the nature of recordings (e.g., the source of recordings, acoustic conditions, speaker characteristics) rather than in closeness of the languages. A good example is errors between Russian and Polish as Russian is much closer to other East Slavic languages. Note that some of these confusions, e.g., Croatian with Czech and Russian occurring for the baseline i-vector system, diminished with the use of bi-GRU RNN system.

5. Conclusions

From all the above-stated results, the following conclusions can be drawn: 1) Bottleneck features are beneficial for all investigated DNN architectures, namely for fully connected networks, and yielded the lowest error rates in all scenarios. 2) Without the use of these features, the baseline i-vector based system was able to outperform systems with fully connected DNNs as well as CNNs. 3) The best results were obtained by using bidirectional RNNs with GRU units; however, the relative improvement over the same, but unidirectional system, was small. 4) The evaluation set consisted of recordings no longer than 5 sec-

onds so that the resulting configuration may be utilized even for short recordings.

The more detailed analysis of results in the form of confusion matrices further showed that: 1) According to assumptions, the worst results were in most cases reached for pairs of languages that are related to each other and belong to the same branch of Slavic languages (i.e., they are also difficult to distinguish for humans). 2) The most challenging pair for identification is Belarusian and Ukrainian (East branch). 3) Other more difficult groups of languages to distinguish are Czech, Slovak and Polish (West branch) and Serbian, Croatian and Slovene (from South branch). 4) The resulting RNN-based system was able to reduce mistakes for pairs of languages with low as well as high baseline error rates (i.e., throughout the whole confusion matrix).

6. Acknowledgements

This work was supported by the Technology Agency of the Czech Republic (Project No. TH03010018), and by the Student Grant Scheme 2018 of the Technical University in Liberec.

7. References

- [1] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 857–860.
- [2] D. M. González, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 861–864.
- [3] E. Singer, P. A. Torres-Carrasquillo, D. A. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. E. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Odyssey 2012: The Speaker and Language Recognition Workshop, Singapore, June 25-28, 2012*. ISCA, 2012, pp. 209–215.
- [4] L. F. D'Haro, R. de Córdoba, C. S. Palacios, and J. D. Echeverry, "Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*

- 2014, Florence, Italy, May 4-9, 2014. IEEE, 2014, pp. 5342–5346.
- [5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. J. Moreno, “Automatic language identification using deep neural networks,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 2014, pp. 5337–5341.
 - [6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
 - [7] M. McLaren, L. Ferrer, and A. Lawson, “Exploring the role of phonetic bottleneck features for speaker and language recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5575–5579.
 - [8] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, “Study of senone-based deep neural network approaches for spoken language recognition,” *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 1, pp. 105–116, 2016.
 - [9] F. Richardson, D. A. Reynolds, and N. Dehak, “A unified deep neural network for speaker and language recognition,” in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 1146–1150.
 - [10] Y. Song, X. Hong, B. Jiang, R. Cui, I. V. McLoughlin, and L. Dai, “Deep bottleneck network based i-vector representation for language identification,” in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 398–402.
 - [11] R. Fér, P. Matejka, F. Grézl, O. Plchot, and J. Cernocký, “Multilingual bottleneck features for language recognition,” in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 389–393.
 - [12] M. K. V., S. Achanta, L. H. R., S. V. Gangashetty, and A. K. Vuppala, “An investigation of deep neural network architectures for language recognition in indian languages,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 2930–2933.
 - [13] A. Lozano-Diez, R. Zazo-Candil, J. Gonzalez-Dominguez, D. T. Toledano, and J. González-Rodríguez, “An end-to-end approach to language identification in short utterances using convolutional neural networks,” in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 403–407.
 - [14] M. Jin, Y. Song, I. McLoughlin, L. Dai, and Z. Ye, “Lid-senone extraction via deep neural networks for end-to-end language identification,” in *Odyssey 2016: Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*. ISCA, 2016, pp. 210–216.
 - [15] M. Jin, Y. Song, I. V. McLoughlin, W. Guo, and L. Dai, “End-to-end language identification using high-order utterance representation with bilinear pooling,” in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2571–2575.
 - [16] D. Garcia-Romero and A. McCree, “Stacked long-term TDNN for spoken language recognition,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3226–3230.
 - [17] M. Tkachenko, A. Yamshinin, N. Lyubimov, M. Kotov, and M. Nastasenko, “Language identification using time delay neural network d-vector on short utterances,” in *Speech and Computer - 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016, Proceedings*. Springer, 2016, pp. 443–449.
 - [18] J. Pesán, L. Burget, and J. Cernocký, “Sequence summarizing neural networks for spoken language recognition,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3285–3288.
 - [19] W. Geng, Y. Zhao, W. Wang, X. Cai, and B. Xu, “Gating recurrent enhanced memory neural networks on language identification,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3280–3284.
 - [20] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, “Automatic language identification using long short-term memory recurrent neural networks,” in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, 2014, pp. 2155–2159.
 - [21] W. Geng, W. Wang, Y. Zhao, X. Cai, and B. Xu, “End-to-end language identification using attention-based recurrent neural networks,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 2944–2948.
 - [22] G. Gelly and J. Gauvain, “Spoken language identification using lstm-based angular proximity,” in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2566–2570.
 - [23] R. Masumura, T. Asami, H. Masataki, and Y. Aono, “Parallel phonetically aware dnns and LSTM-RNNS for frame-by-frame discriminative modeling of spoken language identification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. IEEE, 2017, pp. 5260–5264.
 - [24] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, “Phonetic temporal neural model for language identification,” *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 26, no. 1, pp. 134–144, 2018.
 - [25] G. Gelly, J. Gauvain, V. B. Le, and A. Messaoudi, “A divide-and-conquer approach for language identification based on recurrent neural networks,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3231–3235.
 - [26] S. Fernando, V. Sethu, E. Ambikairajah, and J. Epps, “Bidirectional modelling for short duration language identification,” in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2809–2813.
 - [27] H. Zhao, D. Banský, G. R. Doddington, C. S. Greenberg, J. M. Howard, J. Hernandez-Cordero, L. P. Mason, A. F. Martin, D. A. Reynolds, E. Singer, and A. Tong, “Results of the 2015 NIST language recognition evaluation,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3206–3210.
 - [28] “NIST 2017 language recognition evaluation plan.”
 - [29] J. Nouza, R. Safarik, and P. Cerva, “ASR for south slavic languages developed in almost automated way,” in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3868–3872.
 - [30] “The 2015 NIST language recognition evaluation plan (LRE15).”
 - [31] R. Zazo, A. Lozano-Diez, and J. Gonzalez-Rodriguez, “Evaluation of an lstm-rnn system in different nist language recognition frameworks,” in *Odyssey 2016: Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*. ISCA, 2016, pp. 210–216.