Multi-Head Self-Attention Networks for Language Identification

by

Ravi Kumar Vuddagiri, Tirusha Mandava, Hari Vydana, Anil Kumar Vuppala

in

12th International Conference on Contemporary Computing (IC3)
(*IC3-2019*)

Noida, India

Report No: IIIT/TR/2019/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2019

Multi-Head Self-Attention Networks for Language Identification

Ravi Kumar Vuddagiri
Speech Processing Laboratory, LTRC
International Institute of Information Technology
Hyderabad, India
ravikumar.v@research.iiit.ac.in

Tirusha Mandava
Speech Processing Laboratory, LTRC
International Institute of Information Technology
Hyderabad, India
mandava.tirusha@research.iiit.ac.in

Hari Krishna Vydana
Speech Processing Laboratory, LTRC
International Institute of Information Technology
Hyderabad, India
hari.vydana@research.iiit.ac.in

Anil Kumar Vuppala
Speech Processing Laboratory, LTRC
International Institute of Information Technology
Hyderabad, India
anil.vuppala@iiit.ac.in

***Abstract*—Self-attention networks are being popularly employed in sequence classification and sequence summarization tasks. State-of-the-art models use sequential models to capture the high-level information, but these models are sensitive to length of utterance and do not equally generalize over variable length utterances. This work explores to study the efficiency of recent advancements in self-attentive networks for improving the performance of the LID system. In self-attentive network, variable length input sequence is converted to fixed dimensional vector which represents the whole sequence. The weighted mean of input sequence is considered as utterance level representation. Along with the mean, a standard deviation is employed to represent the whole input sequence. Experiments are performed using AP17-OLR database. Use of mean with standard deviation has reduced the equal error rate (EER) with an 8% relative improvement. A multi-head attention mechanism is introduced in self-attention networks with an assumption that each head captures the distinct information to discriminate languages. Use of multi-head self-attention has further reduced the EER with a 13% relative improvement. Best performance is achieved with multi-head self-attention network with residual connections. Shifted delta cepstral features (SDC) and stacked SDC features are used for developing LID systems.**

***Index Terms*—Language identification system, Deep neural network, Self-attention mechanism, Multi-head, Equal error rate**

I. Introduction

Language identification (LID) refers to the task of detecting the language from a spoken utterance. Most of the speech technology applications use LID as a front-end module to switch between multiple languages [1]. Some of them are multilingual automatic speech recognizers (ASR), multilingual dialogue systems, and voice service applications such as call routing systems. Spoken utterance contains information about the message, language, emotion, age, and gender, etc. LID system has to be invariant all the other information in addition to noise.

LID systems are majorly categorized into two types i.e., explicit and implicit LID systems. In an explicit LID system, an acoustic sequence is initially tokenized to an intermediate representation such as senones, phones or tokens and the sequential information in the intermediate representation is modeled for developing LID systems. In an implicit LID system, the acoustic sequence is directly modeled to detect the language of the spoken utterance. In this work, implicit systems are studied for developing LID systems.

In the early stages of the LID Gaussian mixture models (GMM), Gaussian mixture models with universal background models (GMM-UBM) are explored for developing LID systems [2]–[4]. Deep neural networks (DNNs) are used for developing LID systems and these systems shown better performance with the vailability of larger sized datasets [5], [6]. Though DNNs have performed better, the probabilities computed for each frame are to be averaged over the utterance to obtain the language ID. LID systems which can capture long temporal information form the accoustic sequence can perform better [7]–[9]. i-vectors convert the variable length sequence to fixed dimension continuous representations, i-vectors are used as features for training LID systems [10]. Multilingual bottleneck features are explored for developing LID systems [11]. The performance of the LID system can also influence varying background and mobile environments are explored in [12], [13].

Recently recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been explored for developing LID systems [14], [15]. Bottleneck features along with bidirectional recurrent DNN with gated recurrent neural units have been used for developing LID systems [16]. Though the sequential models like LSTMs can process the whole input sequence they cannot be parallelized. Self-attentive networks have been recently explored for developing LID systems [17], [18]. In [19], [20] this long-term temporal context is achived

978-1-7281-3591-5/19/$31.00

by appending successive shifted delta cepstral (SDC) features i.e stacking SDC. LID systems developed by stacking SDC features have shown significant improvement compared to the system trained with SDC features. Attention mechanism shown significant improvement in machnetranslation,image captioning, and ASR. Self attention mechanism is used to convert variable length sequence to a fixed dimension vector and fixed representaions is used to discriminate languages. Self-attention mechanism aggregates the frame level features by selecting important frames and the selection is done by scaling the frames through a parametrized attention layer. Mean of attentively weighted frame level representations is considered as the representation of the whole utterance. The whole network can be trained as a single-framework through back-propagation.The study explores the recent advancements in self-attention networks along with residual connections for the task of language identification.

In a self-attention network, along with the mean standard deviation of the hidden representations is also considered for representing utterance. Multi-head attention mechanism is introduced in a self-attention network to further improve the performance. Multiple heads in a self-attention network can capture various discriminative features [21]–[23]. Multi-head self-attentive networks have been recently studied for developing text-independent speaker verification and for developing speaker embeddings [21]–[25]. Residual connections are added to this network. Residual networks are act as feature re-estimators [26]. Features which can model the long temporal context have performed better in LID systems [19]. SDC and stacked SDC features are used for developing LID systems.

The remaining paper is organized as follows: Section II explains the details of multi-head self-attention architecture. The experimental setup and results are presented in Section III. Conclusions and future scope are presented in Section IV.

II. Multi-head self-attentive network

The basic architectural aspects of multi-head self-attention networks are briefly described in this section. A self-attention network is a feed-forward network with an attention mechanism. The architecture can be explained [24], in five different blocks, i.e.,

- Frame-level feature extractor.
- Self-attention layer.
- Pooling layer.
- Utterance-level feature extractor.
- Multi-head attention.

as shown in Fig. 1.

The first block is frame-level feature extractor, which takes the sequence of speech features as input and produce the hidden activations as output. The following neural networks are applied for the frame level feature extractor such as DNN, residual neural networks (ResNet). As shown in Fig 1, $I_1, I_2, ..I_i$ are number of hidden layers depend on requirements. Let $S = [s_0, s_1, ...s_L]$ be a sequence of acoustic vectors and $H = [h_1, h_2, h_3...., h_t, ...h_L]$ be the corresponding hidden

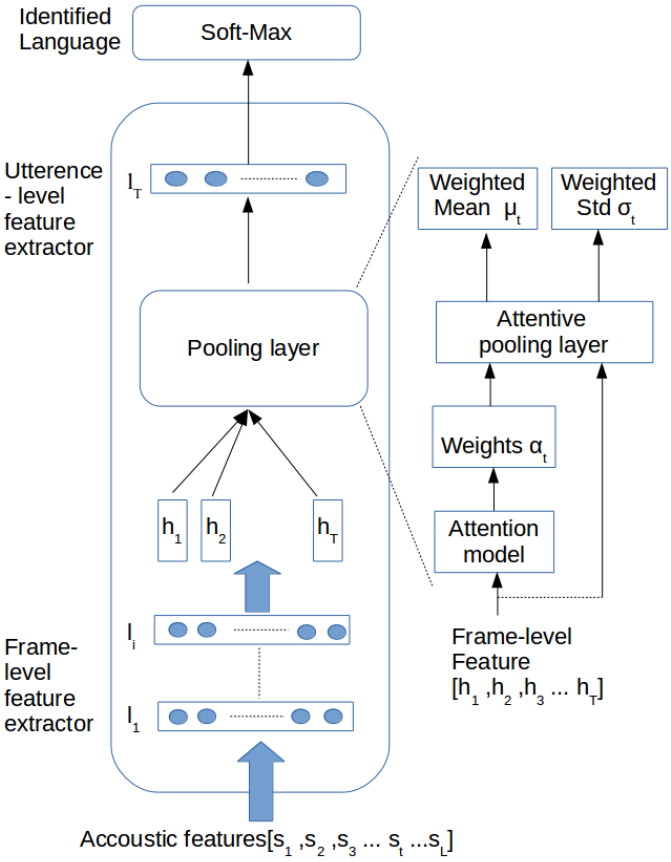

Fig. 1: Block diagram explaining various aspects of a multi-head self-attention network. [24]

activations after the frame level feature extractor network. Where L is the length of input acoustic sequence.

$$H = f(W_{ih}S + b_{ih}) \tag{1}$$

Where f is the nonlinear activation function. W_{ih}, b_{ih} are hidden layer parameters, d_h is the dimensionality of the hidden representations. In this study, ReLU functions are used as nonlinearities

These hidden activations are used by the self-attention layer which is second block in the network to produce a scalar value for each frame and these scalars are considered to represent the relative importance of each frame.

$$e_t = tanh(W_a H^T) \tag{2}$$

The dimension of e_t is $1 \times L$. The values of e_t are normalized using a softmax activation.

$$\alpha_t = \frac{exp(e_t)}{\sum_{t=1}^{L} exp(e_t)} \tag{3}$$

Here α_t are the scalar values that specify the relative importance of each frame.

The pooling layer is the third block, which takes variable-length frame-level features as input and converts to fixed-dimensional vector. The most standard type of pooling layer obtains the average of all frame-level features. In this network, self-attention layer is embedded into pooling layer as shown in Fig. 1. This layer provides weights for each frame to compute weighted hidden representations. The pooling layer compute mean and standard deviation of these weighted hidden representations.

$$\mu_t = \sum_{t=1}^{L} \alpha_t h_t \tag{4}$$

$$\sigma_t = \sqrt{\sum_{t=1}^{L} \alpha_t h_t \odot h_t - \mu_t \odot \mu_t} \tag{5}$$

Where $\odot$ represents Hadamard product. Both weighted mean and standard deviation vectors are concatenated, which represent the whole utterance.

The utterance level representation is fourth block, further passed through a utterance level feature extractor which is a single feed-forward layer as shown in Fig. 1. Output dimension of this feed-forward layer is the number of language classes. A softmax activation function is employed to convert these activations to the language probabilities.

Finaly, Multi-head attention networks have multiple self-attention modules to compute various utterance level representations. These representations are concatenated and given as input to the utterance level feature extractors. Multi-head can be implemented with ease by modifying the dimension of the attention layer, i.e., W_a is $heads \times d_h$. Multi-head attention is expected to capture different aspects that can discriminate languages. In the case of multi-head, a penalty term is added to the objective function to ensure each head capturing dissimilar information.

$$penalty = ||W_a W_a^T - I||_F^2 \tag{6}$$

Where $||.||_F$ represents the Frobenius norm of the matrix. Implementation details of this architecture have explained in next section.

III. Experimental setup and Results

A. Database

In this paper, experiments have been conducted using AP17-Oriental Language Recognition (OLR) challenge database. The Database has collected under the National Natural Science Foundation of China (NSFC) project by Speech ocean and Multilingual Mino-lingual Automatic Speech Recognition (M2ASR). It consists, data of 10 languages Russian, Korean, Cantonese, Mandarin, Japanese, Indonesian, Vietnamese, Kazakh, Tibet, and Uyghur. Each language has around 10 hours of data. From each language, 1800 utterances have used for development set, and remaining have used for training. Test data consists of full-length(5-sec) utterance. All the results have reported in this paper are tested on full-length utterances in terms of the equal error rate (EER) in percentage.

B. Shifted delta cepstral features

SDC features capture long-term temporal variations by spanning information over multiple frames. Here SDC features are computed using MFCC features. From every 20 ms frame with an overlap of 10 ms, MFCC features are extracted. Let x(t) be a frame level acoustic vector (MFCC) delta-cepstra are computed using equation 7

$$\delta_d(t,i) = x(t+ip+d) - x(t+ip-d) \tag{7}$$

for i=0,1,2,..., k-1. Where d is the delta distance between acoustic vectors, p is the distance between blocks, N is a dimension of the acoustic vector, and k is the no of blocks compute SDC feature. SDC features are obtained by concatenating acoustic vector to the shifted delta-cepstra over multiple blocks.

$$SDC(t) = [\ x(t)\ \delta_d(t,i)\ \delta_d(t,1)..... \delta_d(t,k-1)\ ] \tag{8}$$

Here commonly used configuration 7-1-3-7 [N-d-p-k] is employed to extract shifted delta cepstra.

C. Multi-head self-attention network with SDC features

The frame-level feature extractor comprises of an input layer with two hidden layers, and each hidden layer comprises of 1024 units followed by Relu activations. Attention and utterance level feature extractors are single feed-forward layers. Adam is used as optimizer with a learning rate of 0.0001. Learning rate is halved upon observing an increase in validation cost. The training is halted upon encountering an increase in validation accuracy over three successive epochs. The entire network is trained with a cross-entropy objective function.

To better model the temporal context, successive SDC features are stacked with a temporal context. The temporal context in the features can be increased by stacking successive SDC features with left and right context. But by appending the successive feature vectors increases the redundancy in the feature representation which may over-fit the network quite easily. In [19] adapt different optimizers have helped the networks to converge to a better solution. LID systems using stacked SDC features have performed better compared to SDC features in [19].

The performance of base line system shown in Table. I. It presents EER for different baseline LID systems provided by AP17-OLR challenge [27]. Row 1, Row 2, and Row 3 contains EER(%) for i-vector, TDNN-LID and LSTM-LID respectively.

TABLE I: Performances of baseline LID systems. The results are taken from [27] for comparison.

System	Full-length
i-vector	6.22
TDNN-LID	14.65
LSTM-LID	16.03

The work is revisited in the context of self-attention networks, and the results are tabulated in Table. II. Column 1 specifies the temporal context used for stacking SDC features. Column 2 is the dimension of stacked SDC features. Column 3, 4, 5 is the performances of self-attention networks by increasing the depth of the network.

TABLE II: LID systems developed using stacked SDC features.

Temporal context	Feature dimension	2H	3H	4H	5H
1	56	9.16	9.65	11.48	10.45
1-1-1	168	8.54	9.26	10.21	10.66
2-1-2	280	**8.41**	8.58	9.17	10.07

From Table. II, it can be observed stacking the successive SDC features has improved the performance of LID systems. Stacking the features with a temporal context of $\pm$ 2 has produced the optimal performance and further increasing the temporal context has not improved the performance. Further, in this work, stacked SDC features with $\pm$ 2 context (280-Dimensions) have been used for developing LID systems.

Self-attention networks compute the utterance level representation by considering the mean of weighted hidden representations. Along with the mean in this work standard deviation of weighted representations is also used. The results obtained are listed in Table. III. Column 1 of Table. III is the number of hidden layers and column 2, 3 is the performance of LID systems trained by considering mean and standard deviation to obtain utterance level representations respectively. Column 4 is the performance of LID systems obtained by considering both mean and the standard deviation to represent the utterance level feature. Rows 2,3,4 are the performance of the LID systems obtained by varying the depth of the network.

TABLE III: LID systems trained using self-attention networks by varying the depth of the network. The networks have used mean and standard deviation of hidden representations to obtain utterance vector.

Hidden Layers	Mean	Standard deviation	Mean and Standard deviation
2H	**8.41**	**7.55**	**7.80**
3H	8.58	9.02	8.62
4H	9.17	9.37	8.38

From Table. III, it can be observed that the depth of 2 hidden layers (2H) is optimal for the present dataset. Adding the standard deviation imporoved the discriminative capability among the languages. During the studies, we have observed that using both mean and standard deviation to represent the utterance level vector has consistently improved the performance of LID systems. This approach has reduced the EER from 8.41 to 7.80. This work explores the use of multi-head attention mechanism in a self-attention network and the results are presented in Table. IV. Column 1 of Table. IV is the number of heads used and column 2,3 is the EER of the corresponding LID system.

TABLE IV: Performance of LID systems trained using multi-head self-attentive networks and multi-head self-attentive networks with residual connections

No of heads	Multi-head attention	Multi-head attention with residual connections
1-head	7.80	6.21
2-heads	7.17	5.79
3-heads	**6.44**	**5.65**
4-heads	7.98	6.38

From Table. IV, it can be observed using multi-head attention has improved the performance of LID systems. To encourage various attention heads to capture different aspects, a penalty term specified in equation 6 is added to the loss function while optimizing the network. The use of this constraint has improved the consistency in the networks. In this study, using 3-head attention has reduced the EER from 7.80 to 6.44. Further, adding residual connections in multi-head self-attention network reduced the EER from 6.44 to 5.65. Use of residual connections for multi-head self-attention network has helped the models to converge better, it can be noted that the performance of resnet is superior to DNN multi-head self-attention networks , and the margin of improvement is higher for state-of-the-art system i-vector as shown in Table. I.

IV. Conclusion and Future scope

This work explores the use of self-attention networks for developing LID systems. Self-attending networks convert the variable length sequence to fixed dimensional utterance vector. The mean of attentively scaled hidden representations is used as utterance vector. This work has studied the use of standard deviation along with the mean to represent the utterance vector. This works studied the efficiency of multi-head attention mechanism in self-attention networks. Stacking the successive SDC features with a temporal context of $\pm$ 2 have reduced the EER from 9.16 to 8.41. Using standard deviation along the mean to represent the utterance vector has reduced the EER from 8.41 to 7.80. Mutli-head attention mechanism in self attention networks have reduced the EER from 7.80 to 6.44. Furhter, residual connections reduced the EER from 6.44 to 5.65.

The consistency of self-attention networks are to be studied for shorter utterances. The performance of these networks are to be studied in the presence of the noise and the mismatched conditions. Exploring the use of utterance vectors in a self-attention networks are used as embeddings for developing LID systems.

V. Acknowledgements

The authors would like to thank Science & Engineering Research Board (SERB) for funding Language Identification in Practical Environments (YSS/2014/000933) project.

REFERENCES

[1] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[2] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. R. Deller, "Language identification using Gaussian mixture model tokenization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1. IEEE, 2002, pp. I–757.

[3] E. Wong and S. Sridharan, "Methods to improve Gaussian mixture model based language identification system," in *Seventh International Conference on Spoken Language Processing*, 2002.

[4] V. R. Kumar, H. K. Vydana, and A. K. Vuppala, "Significance of GMM-UBM based modelling for Indian language identification," *Procedia Computer Science*, vol. 54, pp. 231–236, 2015.

[5] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

[6] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. IEEE, 2014, pp. 5337–5341.

[7] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[8] M. A. Zissman, "Automatic language identification using Gaussian mixture and hidden Markov models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2. IEEE, 1993, pp. 399–402.

[9] B. M. L. Srivastava, H. Vydana, A. K. Vuppala, and M. Shrivastava, "Significance of neural phonotactic models for large-scale spoken language identification," in *Int. Joint Conf. Neural Networks*. IEEE, 2017, pp. 2144–2151.

[10] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the international speech communication association*, 2011.

[11] W. Geng, J. Li, S. Zhang, X. Cai, and B. Xu, "Multilingual tandem bottleneck feature for language identification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[12] V. Ravi Kumar, H. K. Vydana, J. V. Bhupathiraju, S. V. Gangashetty, and A. K. Vuppala, "Improved language identification in presence of speech coding," in *International Conference on Mining Intelligence and Knowledge Exploration*. Springer, 2015, pp. 312–322.

[13] V. Ravi Kumar, H. K. Vydana, and A. K. Vuppala, "Curriculum learning based approach for noise robust language identification using DNN with attention," *Expert Systems with Applications*, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.06.004

[14] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," *Proc. Odyssey-14, Joensuu, Finland*, 2014.

[15] W. Geng, W. Wang, Y. Zhao, X. Cai, B. Xu *et al.*, "End-to-end language identification using attention-based recurrent neural networks." in *Proc. INTERSPEECH*, 2016, pp. 2944–2948.

[16] L. Mateju, P. Cerva, J. Zdansky, and R. Safarik, "Using deep neural networks for identification of slavic languages from acoustic signal," *Proc. INTERSPEECH*, pp. 1803–1807, 2018.

[17] K. Mounika, S. Achanta, H. Lakshmi, S. V. Gangashetty, and A. K. Vuppala, "An investigation of deep neural network architectures for language recognition in Indian languages." in *Proc. INTERSPEECH*, 2016, pp. 2930–2933.

[18] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[19] R. K. Vuddagiri, H. K. Vydana, and A. K. Vuppala, "Improved language identification using stacked SDC features and residual neural network," pp. 205–209.

[20] V. Mounika Kamsali, V. Ravi Kumar, S. V. Gangashetty, and V. Anil Kumar, "Combining evidences from excitation source and vocal tract system features for Indian language identification using deep neural networks," *International Journal of Speech Technology*, pp. 1–8, 2017.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[22] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2018.

[23] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. INTERSPEECH*, pp. 3573–3577, 2018.

[24] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[25] Q. Wang, K. Okabe, K. A. Lee, H. Yamamoto, and T. Koshinaka, "Attention mechanism in speaker recognition: What does it learn in deep speaker embedding?" *arXiv preprint arXiv:1809.09311*, 2018.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[27] Z. Tang, D. Wang, Y. Chen, and Q. Chen, "AP17-OLR challenge: Data, plan, and baseline," *arXiv preprint arXiv:1706.09742*, 2017.