MEMORY LAYERS WITH MULTI-HEAD ATTENTION MECHANISMS FOR TEXT-DEPENDENT SPEAKER VERIFICATION

Victoria Mingote, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{vmingote, amiguel, ortega, lleida}@unizar.es

ABSTRACT

In this paper, we explore an approach based on memory layers and multi-head attention mechanisms to improve in an efficient way the performance of text-dependent speaker verification (SV) systems. The most extended SV systems based on Deep Neural Networks (DNN) extract the embedding of the utterance from the average pooling of the temporal dimension after processing. Unlike previous works, we can exploit the phonetic knowledge needed for text-dependent SV systems by combining the temporal attention of multiple parallel heads with the phonetic embeddings extracted from a phonetic classification network, which helps to guide to the attention mechanism with the role of the positional embedding. The addition of a memory layer to a text-dependent SV system was tested on the RSR2015-part II and DeepMine-part I databases, where, in both cases outperformed the baseline result and the reference system based on the same transformer network without the memory layer.

***Index Terms*—** Memory Layer, Multi-head Attention, Temporal Attention, Text-Dependent Speaker Verification

1. INTRODUCTION

Speaker verification (SV) field involves the process to determine correctly whether an utterance belongs to a claimed identity or not. Depending on the constraints of the lexicon content of the utterances, SV is usually divided into two categories: text-independent and text-dependent SV. In the former, there are no restrictions on the uttered phrase pronounced which can produce a large variability in the duration of the utterances, while in the latter requires the same constraints in the lexicon content.

In the context of text-independent tasks, most of the current SV systems are based on Deep Neural Networks (DNN) where the output vectors extracted from these DNN systems are known as x-vectors [1]. This kind of approaches has been successful thanks to the availability of large databases. Nevertheless, the application of the same deep learning techniques for text-dependent SV systems has lead to mixed results. In cases with large databases, these systems have outperformed the traditional approaches [2]. Unlike text-independent SV, those large databases existent for text-dependent tasks are not publicly available. When we desire to develop a system in closed conditions for a task similar to RSR2015 database [3], the use of DNN models based on x-vectors may lead to problems due to overfitting. To address this issue, the recently multipurpose DeepMine dataset has been released [4]. This dataset was created mainly to provide a large-scale database for text-dependent SV purposes where the phonetic variability on short-duration utterances can be analyzed.

As we showed in our previous work [5], keeping the order of the phonetic information is important for text-dependent tasks due to the lexical content, since this information is part of the identity. DNN models using standard average pooling mechanisms to transform the processed utterance information to an embedding vector can have problems for this task. In this paper, we explore a different type of processing for the temporal information in the DNN which is provided by temporal attention mechanisms enhanced by a memory layer, which provides an efficient access to knowledge stored in the training phase. Architectures with attention mechanisms have become an effective approach in a wide variety of application areas [6, 7] to focus the processing of the DNNs on certain areas of the feature maps or certain temporal slots, including the scenario of SV [8, 9]. The success achieved with this approach in SV may be due to this kind of mechanism allows models to learn the frame-level representations, which are more precise to represent the speaker characteristics. Recently, multi-head attention proposed in [7] for Transformer architecture is a powerful attention mechanism which allows using several single attention mechanisms to extract diverse information over different parts of the network [10, 11]. Furthermore, to improve the model capacity while computational efficiency is kept similar, memory layers [12] have been introduced in combination with the multi-head attention mechanism with success for language modelling tasks. However, this technique has not yet been applied to SV tasks.

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by the Government of Aragon (Reference Group T36_20R) and co-financed with Feder 2014-2020 "Building Europe from Aragon", and by Nuance Communications, Inc. The Titan V used for this research was donated by the NVIDIA Corporation.

In this paper, we present an architecture based on Residual Networks (RN) [13] combined with multi-head attention and memory layers [12] for text-dependent SV tasks with small and large-scale databases. This memory layer is a product-key attention mechanism which allows storing the knowledge learned by the DNN during the training process. Moreover, we have added more phonetic information to the different architectures with the use of phonetic embeddings extracted from a phonetic classification network [10, 14]. These phonetic embeddings are used as a complement to the feature extractor and can be used by the dot attention mechanism to locate phoneme similarities which can play the role of the positional embedding that is not used in our architecture.

The remainder of this paper is laid out as follows. Section 2 provides a review of the Multi-head Attention mechanism. In Section 3, we describe the Memory Layer. Section 4 provides a description of system proposed. In Section 5, the experimental setup is described. Finally, Section 6 presents and discusses results and Section 7 concludes the paper.

2. MULTI-HEAD ATTENTION MECHANISM

Multi-head attention mechanism is the core mechanism of the transformer architectures [7]. Instead of performing a single attention function, this mechanism consists of multiple dot-product attention which allows to do attention in parallel. Dot-product attention can be calculated for each head as,

$$H_h = softmax_t(\frac{QW_h^Q \cdot (KW_h^K)^T}{\sqrt{d_k}}) \cdot VW_h^V, \tag{1}$$

where Q is the query, and K, V are the key-value pairs which are the input for the attention layer, W_h^Q, W_h^K, W_h^V are learnable weight matrices to make the linear projections, and d_k is the number of dimensions of the query/key vector, and the softmax operation is performed over the temporal axis, which allows each head to focus on certain frames of the input sequence for each output. In this work, we use only the encoder part of the transformer, so the input to the attention mechanism is the same for the query, key and value signals (Q, K, V). Thus, the multi-head attention is defined as the concatenation of the outputs from each head:

$$MHA(X) = [H_1, H_2 \dots H_{d^{head}}] \cdot W^{head}, \tag{2}$$

where X is the input to the attention layer, H_h are the output for the $h-th$ attention layer, W^{head} is a learnable weight matrix to make a final linear projection, and d^{head} is the number of attention heads in this layer. When the architecture is trained with a speaker classification objective, the multi-head attention mechanism learns to calculate a set of weights for each head that focus on different positions of the sequence and provide more relevance to the most important frames to discriminate better among speakers and utterances.

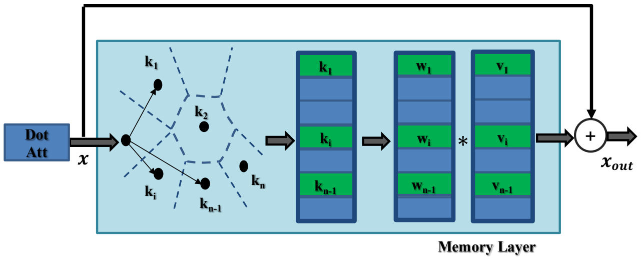

Fig. 1. Memory layer which uses the output of the dot attention to select the closest stored values and produce a vector to add extra information to concatenate with the input.

3. MEMORY LAYER

The idea of using an external memory with a neural network was introduced in [15, 16]. A simplified version, which acts as a read-only memory at inference time, was used in [12] to improve the model capacity of a transformer architecture with a large external memory. This module is called memory layer and has an insignificant computational overhead while providing significant improvement of performance. In this work, we also use read-only memory layers that are able to store the knowledge obtained for the network during the training process. Since the information has to be stored while training, as any other parameter of the network, we need to use a differentiable mechanism to address the location. Thus, a product key-attention is used [15, 16, 12] where the closest keys to the signal enable the output of the learnable memory slots, and the output is composed by the weighted sum of the corresponding memory values of the k nearest keys.

Using this layer, as Fig.1 depicts, the input data is compared with all the keys, and the scores obtained are used to select the keys with the highest scores and compute the associated weight vectors with the following expression:

$$w = softmax_n(x \cdot U^K), \tag{3}$$

where x is the input to the memory layer, U^K is the keys matrix, and the softmax is computed over the memory index axis, which allows to focus on certain contents of the memory that will be used to provide the output. After that, these weights are combined with the memory values of the selected keys, and the output is concatenated with the output of the previous attention mechanism:

$$x_{out} = x + w \cdot U^V, \tag{4}$$

where w are the weights of the selected keys obtained with (3), and U^V are the memory values associated with the keys.

4. SYSTEM DESCRIPTION

In the following section, we describe the architecture of the system which combines RN, multi-head attention and mem-

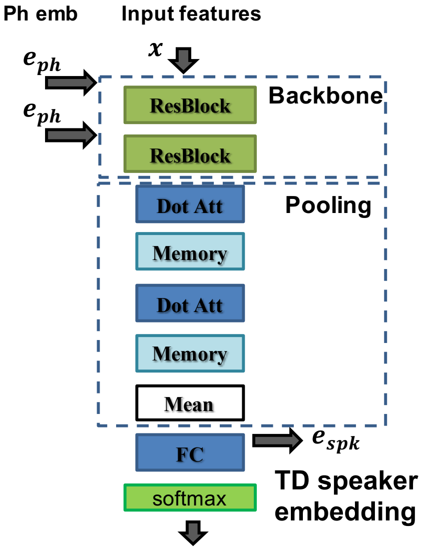

Fig. 2. Architecture for RN, Attention and Memory layers network, composed of a backbone, a pooling and a embedding extraction.

ory layers. Table 1 and Fig.2 depicts this architecture which is composed of two main parts: the backbone and the pooling. The backbone uses two RN blocks with three layers each block and Rectified Linear Units (ReLU) as non-linearities.

Moreover, this architecture needs positional information [7] for the self attention layers to provide a good performance. Instead of using temporal positional information as many language modelling applications, we use the output of a phonetic classifier bottleneck [10, 14]. The architecture of the phonetic classifier is an evolution of [10]. In this system, we use a modification of efficient net [17] to operate with 1D group convolutions as backbone. Efficient net can produce several temporal scales. We combine them using a modification of [18], where we substitute the linear combinations by the operation concatenation of channels and 1D group convolution. We concatenate this information before each RN block.

Table 1. Topology for RN, Attention and Memory layers architecture.

Layer	Layer type	Channels	Output
1	Conv1D	128	128×T
2	ResBlock-ReLU(x3)	160	160×T
3	ResBlock-ReLU(x3)	256	256×T
4	BatchNorm1D	256	256×T
5	DotAtt (heads=16)	256	256×T
6	Memory	256	256×T
7	DotAtt (heads=16)	256	256×T
8	Memory	256	256×T
9	Mean	–	256
10	FC+softmax	–	N
	Cross-Entropy Loss		

Following these RN blocks, the pooling part alternates two multi-head attention layers with two memory layers. The multi-head attention layers can be seen analogous to an alignment method which allows assigning embeddings to several categories. This approach has been found useful for text-dependent tasks [5, 19]. In addition, we introduce in our architecture memory layers that can store a significant amount of information for a relatively small inference computing cost. Furthermore, with the integration of the phonetic embeddings in the backbone part, the performance of the attention mechanism improves since the phonetic embeddings help to guide to the attention mask similarly to the positional embedding in language modeling transformers [7].

5. EXPERIMENTAL SETUP

In this paper, two different text-dependent speaker verification datasets have been used to carry out the experiments. First, we reported the results on the RSR2015 database [3]. It consists of speech samples from 157 males and 143 females. For each speaker, there are 9 sessions pronouncing 30 different phrases. The corpus is divided into three speaker subset: background (bkg), development (dev), and evaluation (eval). In this work, we develop our experiments with Part II, which is based on short control commands which have a strong overlap of lexical content, and we employ only the bkg data for training. The second database employed is DeepMine database [4]. This database is composed of three different parts with English and Persian phrases. For the experiments using this database, we employ the selected files from Part 1, which are used in Task 1 of the Short-duration Speaker Verification (SdSV) Challenge 2020 [20]. Part 1 corresponds to text-dependent part and consists of 5 Persian phrases and 5 English phrases, and 963 females and males speakers. Finally, we have also employed LibriSpeech [21] to train a phonetic classification network which is used to extract phonetic embeddings.

5.1. Experimental description

To develop the experiments using DeepMine data, we have employed as input for the system a feature vector based on mel-scale filter banks. With this feature extractor, we obtain two log filter banks of sizes 24 and 32 which are concatenated with the log energy. While for the experiments with RSR2015 database, 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) stacked with their first and second derivates is employed as input to train the architecture. Furthermore, in both cases, we have used phonetic embeddings of 256 dimensions as positional information which are extracted from a phonetic classifier network.

In this paper, two sets of experiments were carried out to evaluate the architecture proposed. In the first set of experiments with RSR2015-part II, we use, as baseline to compare, the result obtained in [22] where one model for each phrase with Gaussian Mixture Model (GMM) as alignment method instead of attention mechanisms is trained using exactly the same data as in this work. Additionally, we compare the architecture using memory layers (MEM) with different sizes of the layer to the architecture using feed-forward layers (FF) as in original transformer network [7].

In the second set of experiments using DeepMine-part I, the baseline to compare is based on x-vectors presented by organizers of SdSV Challenge [20] which apart from training with the mention DeepMine database, this baseline was trained using the popular VoxCeleb 1 and 2 databases [23, 24]. Moreover, as in the first set, we compare the architecture using memory layers (MEM) to the architecture with feed-forward layers (FF).

6. RESULTS

Table 2 shows Equal Error Rate (EER), NIST 2008 (DCF08) and NIST 2010 minimum detection cost (DCF10) results for the experiments focused on RSR2015-part II database. We can observe that the proposed architecture using memory layers with multi-head attention mechanism achieve the best result. Note that independently of the size of MEM layer, the result is better than using the original FF layer. Additionally to the previous metrics, in the last row of the table, we show the relative improvement achieved comparing the architecture with the best result using the MEM layer and the architecture with the FF layer.

Table 2. Experimental results on RSR2015 part II [3] evaluation set, showing EER% and NIST 2008 and 2010 min costs (DCF08, DCF10). These results were obtained with train set and varying the use of feed-forward layer or memory layer in the architecture, and the sizes of memory layer.

FF	**MEM**	**Size**	**EER%**	**DCF08**	**DCF10**
Baseline*			5.10	0.276	0.850
yes	no	-	5.28	0.255	0.743
no	yes	2^{11}	4.88	0.238	**0.700**
		2^{12}	**4.80**	**0.237**	0.706
		2^{13}	4.99	0.245	0.721
MEM vs FF Improv. %			**9.09**	**7.05**	**5.78**

Furthermore, in [5], we demonstrated that approaches similar to x-vectors do not work correctly with RSR2015 database due to small training data size, and the lack of special treatment of phonetic information. For this reason, we used an alignment mechanism and trained one model for each phrase in the baseline system [22]. However, in this work, we can check that a competitive result can be obtained with a single network for all the phrases thanks to the architecture based on multi-head attention with phonetic embeddings allowing a precise handling of phonetic information, which is one of the key points in the text-dependent SV. As we can see in the results, the memory layer enhances the performance of the attention mechanism, which confirms that the information stored during training is useful at inference time.

Table 3. Experimental results on DeepMine part I [4] evaluation set, showing EER% and NIST 2008 min cost (DCF08). These results were obtained with train set and varying the use of feed-forward layer or memory layer in the architecture, and the sizes of memory layer.

FF	**MEM**	**Size**	**EER%**	**DCF08**
Baseline x-vectors*			9.05	0.529
yes	no	-	3.94	0.151
no	yes	2^{11}	3.70	0.143
		2^{12}	**3.58**	**0.136**
		2^{13}	3.62	0.137
MEM vs FF Improv. %			**9.14**	**9.93**

The results obtained in DeepMine-part I database are shown in Table 3. Unlike previous text-dependent SV works, in these experiments where the training data available in this database is larger, we observe that the use of one model trained with all the phrases works better. Moreover, we can see that the architecture with different sizes of MEM layers achieves better result than the architecture with FF layers, following the same trend than the other database. In Table 3, we can also see that both architectures outperform the baseline with x-vectors. Therefore, the importance of the phonetic information combined with the temporal attention using MEM layers and multi-head attention to train DNN architectures in text-dependent SV is shown again.

7. CONCLUSIONS

In this paper, we have introduced a new architecture for text-dependent SV task. This kind of architecture allows taking advantage of the knowledge acquired by the temporal attention mechanisms to keep the phonetic information. Moreover, the use of memory layers improves the model capacity keeping the efficiency of the architecture based on the original transformer network. The evaluation was carried out in two text-dependent SV databases to confirm the improvement achieved. The first one is RSR2015-part II, which due to the lack of data had suffered problems with deep architectures when an alignment mechanism was not incorporated into the network, but in this work, we have demonstrated that with this architecture is possible to achieve competitive results in this database. Using the other database, DeepMine-part I, the architecture proposed outperforms the baseline with x-vectors, so we have demonstrated the relevance of keeping the temporal information during the training even with larger databases.

8. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 ICASSP*, pp. 5329–5333.

[2] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 ICASSP*, pp. 5115–5119.

[3] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[4] Hossein Zeinali, Lukas Burget, and Jan Cernocky, "A Multi Purpose and Large Scale Speech Corpus in Persian and English for Speaker and Speech Recognition: the DeepMine Database," in *Proc. ASRU 2019*.

[5] Victoria Mingote, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Supervector Extraction for Encoding Speaker and Phrase Information with Neural Networks for Text-Dependent Speaker Verification," *Applied Sciences*, vol. 9, no. 16, pp. 3295, 2019.

[6] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[8] Gautam Bhattacharya, Md Jahangir Alam, and Patrick Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification.," in *Interspeech*, 2017, pp. 1517–1521.

[9] FA Rezaur rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in *2018 ICASSP*, pp. 5359–5363.

[10] Ignacio Viñals, Dayana Ribas, Victoria Mingote, Jorge Llombart, Pablo Gimeno, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Phonetically-Aware Embeddings, Wide Residual Networks with Time-Delay Neural Networks and Self Attention Models for the 2018 NIST Speaker Recognition Evaluation," *Proc. Interspeech 2019*, pp. 4310–4314, 2019.

[11] Miquel India, Pooyan Safari, and Javier Hernando, "Self multi-head attention for speaker recognition," *Proc. Interspeech 2019*, pp. 4305–4309, 2019.

[12] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, "Large memory layers with product keys," in *Advances in Neural Information Processing Systems*, 2019, pp. 8546–8557.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] Tianyan Zhou, Yong Zhao, Jinyu Li, Yifan Gong, and Jian Wu, "CNN with phonetic attention for text-independent speaker verification," in *2019 ASRU*. IEEE, 2019, pp. 718–725.

[15] Alex Graves, Greg Wayne, and Ivo Danihelka, "Neural turing machines," *preprint arXiv:1410.5401*, 2014.

[16] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al., "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[17] Mingxing Tan and Quoc V Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[18] Mingxing Tan, Ruoming Pang, and Quoc V Le, "Efficientdet: Scalable and efficient object detection," *arXiv preprint arXiv:1911.09070*, 2019.

[19] Weicheng Cai, Zexin Cai, Xiang Zhang, Xiaoqi Wang, and Ming Li, "A novel learnable dictionary encoding layer for end-to-end language identification," in *2018 ICASSP*, pp. 5189–5193.

[20] Hossein Zeinali, Kong Aik Lee, Jahangir Alam, and Lukaš Burget, "Short-duration Speaker Verification (SdSV) Challenge 2020: the Challenge Evaluation Plan.," Tech. Rep., arXiv preprint arXiv:1912.06311.

[21] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 ICASSP*, pp. 5206–5210.

[22] Victoria Mingote, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida, "Training Speaker Enrollment Models by Network Optimization," *Proc. Interspeech 2020*.

[23] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, pp. 2616–2620.

[24] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.