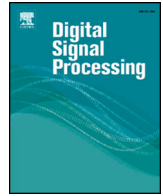




ELSEVIER

Contents lists available at ScienceDirect

## Digital Signal Processing

journal homepage: [www.elsevier.com/locate/dsp](http://www.elsevier.com/locate/dsp)

# Class token and knowledge distillation for multi-head self-attention speaker verification systems

Victoria Mingote<sup>\*</sup>, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

*ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain*

## ARTICLE INFO

### Article history:

Available online 1 December 2022

### Keywords:

Class token  
Teacher-student learning  
Distillation token  
Speaker verification  
Multi-head self-attention  
Memory layers

## ABSTRACT

This paper explores three novel approaches to improve the performance of speaker verification (SV) systems based on deep neural networks (DNN) using Multi-head Self-Attention (MSA) mechanisms and memory layers. Firstly, we propose the use of a learnable vector called Class token to replace the average global pooling mechanism to extract the embeddings. Unlike global average pooling, our proposal takes into account the temporal structure of the input what is relevant for the text-dependent SV task. The class token is concatenated to the input before the first MSA layer, and its state at the output is used to predict the classes. To gain additional robustness, we introduce two approaches. First, we have developed a new sampling estimation of the class token. In this approach, the class token is obtained by sampling from a list of several trainable vectors. This strategy introduces uncertainty that helps to generalize better compared to a single initialization as it is shown in the experiments. Second, we have added a distilled representation token for training a teacher-student pair of networks using the Knowledge Distillation (KD) philosophy, which is combined with the class token. This distillation token is trained to mimic the predictions from the teacher network, while the class token replicates the true label. All the strategies have been tested on the RSR2015-Part II and DeepMine-Part 1 databases for text-dependent SV, providing competitive results compared to the same architecture using the average pooling mechanism to extract average embeddings.

© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The performance in speaker verification (SV) tasks has improved greatly in recent years thanks to the deep learning (DL) advances in signal representations and optimization metrics [1–5] that have been adapted from the state-of-the-art face verification, image recognition, or text-modelling systems. In these systems, Convolutional Neural Network (CNN) or Time Delay Neural Network (TDNN) [2] are still the most employed approaches to obtain the signal representations or embeddings. Nevertheless, self-attention mechanisms are becoming a dominant approach in many fields beyond text-related tasks. For example, Transformers [6] are spreading to many tasks [7–10] where large scale databases are available. In SV tasks, this kind of architecture has started to be successfully applied in text-independent SV [11–14] where there are no constraints in the uttered phrase and big databases are available. However, in text-dependent SV, there is still room for

improvement since the amount of public data is not very large. Besides, text-dependent SV consists of deciding whether a speech sample has been uttered by the correct speaker pronouncing the fixed passphrase selected. So, the phonetic information of the signal is relevant to determine the identity. Therefore, keeping the temporal structure is needed to obtain representations that encode correctly both phrase and speaker information.

In the context of text-dependent SV tasks, our previous works [15–17] showed the advantages of replacing the traditional pooling mechanism based on averaging the temporal information with an external alignment mechanism to obtain a supervector embedding. This supervector allowed to keep the temporal structure and represent both phrase and speaker information, but the temporal alignment had to be performed by using an external method as a phone decoder, a Gaussian Mixture Model (GMM) [18,19] or a Hidden Markov Model (HMM) [20]. As an alternative approach, in [21], we introduced Multi-head Self-Attention (MSA) mechanisms [6] combined with memory layers [22] to substitute the alignment mechanisms. The use of MSA allowed the model to focus on the most relevant frames of the sequence to discriminate better among utterances and speakers. However, the proposed architecture based

<sup>\*</sup> Corresponding author.

E-mail addresses: [vmingote@unizar.es](mailto:vmingote@unizar.es) (V. Mingote), [amiguel@unizar.es](mailto:amiguel@unizar.es) (A. Miguel), [ortega@unizar.es](mailto:ortega@unizar.es) (A. Ortega), [lleida@unizar.es](mailto:lleida@unizar.es) (E. Lleida).

on MSA employed an average pooling mechanism to obtain the final representation embedding.

In this work, to substitute the global average pooling, we have introduced a learnable vector known as Class token, which is inherited from Natural Language Processing (NLP) [7], and recently, many image recognition systems [8]. However, this approach has not yet been applied to SV tasks. To introduce this vector into the system based on DNN with MSA and memory layers, the class token is concatenated to the input before the first MSA layer, and the state at the output is employed to perform the class prediction. During training, the temporal information is encoded in the token, and this token interacts with the whole input sequence through self-attention and learns a global description similar to a supervector approach [16,23] since the multiple heads act as slots of the supervector. A similar mechanism has also been used recently in [10]. Therefore, the average pooling mechanism is not needed to obtain a representation. The multiple heads can encode more details about the sequence order than the average, playing the role of the states and improving the results as shown in [16], [17] with the use of external alignment mechanisms based on HMM and GMM. In addition, the information encoded in these multiple heads can be represented and analyzed, which improves the interpretability of the results of this kind of approach. To improve the performance obtained with the class token approach, we also introduce a novel multiple initialization sampling mechanism to reduce possible initialization problems and give more robustness against the lack of data to model predictions. Since it is a case of use in the industry to develop custom specific systems with small in-domain datasets and this kind of approach could be a possible solution.

Moreover, this work contributes with another approach based on Transformer architecture and Knowledge Distillation (KD) [24, 9]. We propose a teacher-student approach combined with Random Erasing data augmentation [25,26] which allows modelling the uncertainty in the parameters of a teacher model with a compact student model and get more reliable predictions. Following the idea proposed in [9], we have also introduced the Distillation token in the student network to replicate the predictions of the teacher network, while the class token is trained to reproduce the true label as Fig. 3 depicts. Unlike the objective in [9], in our work, the distillation process is not intended to compress the teacher model, but rather both models are trained together and the student model learns to better capture the intrinsic variability of the teacher predictions.

To summarize, the main contributions are:

- We replace the global average pooling mechanism by a learnable class token to obtain a global utterance descriptor associated to the concept of supervector in speaker verification.
- We propose a new approach based on a sampling approximation to estimate the class token.
- We introduce a teacher-student architecture with an additional token known as distillation token which is combined with the class token to provide robustness to the learned student model.

This paper is organized as follows. In Section 2, we show an overview of the MSA and memory layers. Section 3 explains the strategy of introducing a learnable class token using sampling. In Section 4 we introduce the approach based on KD combined with the tokens employed to develop our system. Section 5 describes the system used. In Section 6, we present the experimental data, and Section 7 explains the results achieved. Conclusions are presented in Section 8.

## 2. Overview of transformer encoder

The original transformer architecture [6] is composed of two main parts: the encoder and decoder parts. However, in many tasks, the transformer encoder is the only part used to create the DL systems. The core mechanism of each encoder block is Multi-head Self-Attention (MSA) layer which is composed of multiple dot-product attention. As we only employ the encoder part, the input to this attention mechanism is the same for the query, key and value signals ( $Q, K, V$ ):

$$Q_h = x \cdot W_h^Q, K_h = x \cdot W_h^K, V_h = x \cdot W_h^V, \quad (1)$$

where  $x$  is the input to this layer, and  $W_h^Q, W_h^K, W_h^V$  are learnable weight matrices to make the linear projections. After these projections, a softmax operation is performed over the temporal axis, which allows each head to focus on certain frames of the input sequence. The result of this softmax operation is known as the self-attention matrix for each head and can be defined as:

$$A_h = \text{softmax}_t \left( \frac{Q_h \cdot K_h^T}{\sqrt{d_k}} \right), \quad (2)$$

where  $d_k$  is the number of dimensions of the query/key vector, and  $\top$  denotes transpose. This self-attention matrix learns the most relevant information among the different data. Using this information, the value  $V$  feature vectors are aggregated to obtain the output of each head. The final output of each head can be calculated as,

$$H_h = A_h \cdot V_h. \quad (3)$$

Thus, MSA is defined as the concatenation of the outputs from each head  $H_h$ :

$$MSA(X) = [H_1, H_2 \dots H_{d^{head}}] \cdot W^{head}, \quad (4)$$

where  $X$  is the input to the 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 the  $h$ -th layer.

The transformer encoder alternates the MSA layer with a second layer which is the feed-forward (FF) layer. However, in [21], we proposed the replacement of FF layers by memory layers as in [22]. With this layer, the input data is compared with all the keys using a product key-attention, and the scores obtained are used to select the closest keys, which have the highest scores. After that, the associated weight vectors are computed with the following expression:

$$w = \text{softmax}_n(x \cdot U^K), \quad (5)$$

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

$$x_{out} = x + w \cdot U^V, \quad (6)$$

where  $w$  are the weights of the selected keys obtained with (5), and  $U^V$  are the memory values associated with the keys. After the encoder blocks are applied, an average pooling mechanism is usually employed to reduce the temporal information and represent variable-length utterances with fixed-length vectors. However, this averaging may neglect the order of the phonetic information, which is relevant for text-dependent SV tasks.

### 3. Representation using class token

In many tasks of NLP and computer vision, the transformer architecture uses a learnable vector called Class Token ( $x_{CLS}$ ), as in the original BERT model [7] or Vision Transformer (ViT) [8], instead of a global average pooling. To employ this token in the transformer encoder, the vector is concatenated to the input of the first MSA layer to perform the classification task. With this token, the self-attention is forced to capture the most relevant information with the class token to obtain a representation as a global utterance descriptor similar to the supervector approach. Instead of mixing all the information with an average pooling mechanism, the temporal structure can be kept since the attention mechanism acts as a weighted sum of the temporal tokens for each layer. The output vector is the concatenation of different head subvectors and each of them is the result of a different attention outcome. Thus, the mechanism can be seen similar to those used in our previous work [16], where the heads play the role of the states and the supervector in [23]. The supervector mechanism is also similar to [27] but in that case, the task was text-independent SV and MSA layers were not used. Besides, this type of mechanism allows to enhance the interpretability of what the neural network learns through the self-attention layers.

In [23], this mechanism to obtain the supervector is defined similar to a conventional GMM supervector with the following expression:

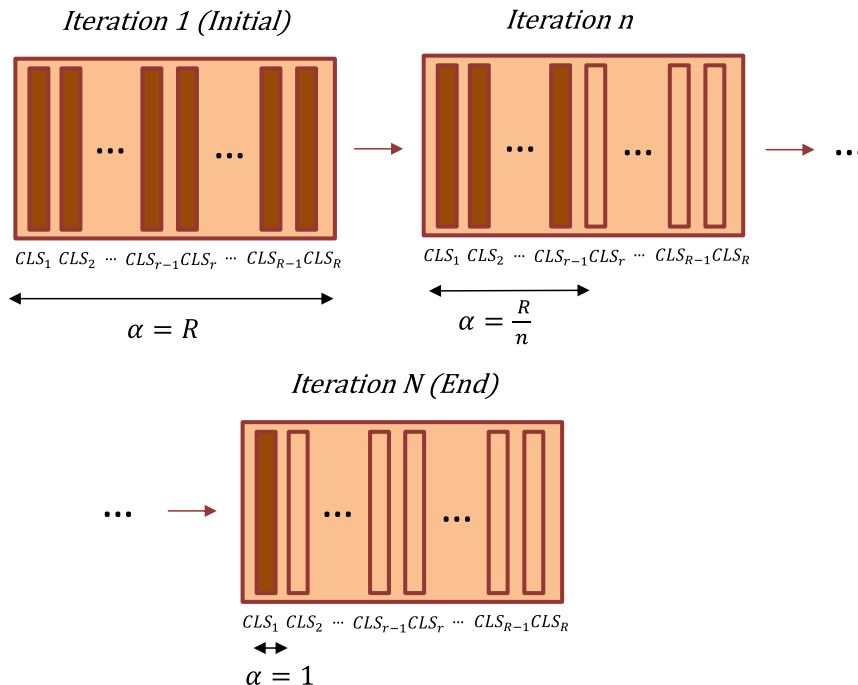
$$s_c = \frac{\sum_t x_t \cdot w_{tc}}{\sum_t w_{tc}} = \sum_t x_t \cdot \bar{w}_{tc}, \quad (7)$$

where  $w_{tc}$  are the weights obtained by a softmax function on the output of a learnable layer,  $s_c$  are vectors per state/component  $C$  of dimension  $D$  that summarize the information associated along the sequence of feature vectors  $x_t$  of dimension  $D$ , and  $\bar{w}_{tc}$  are the normalized weights defined as  $w_{tc}/\sum_t w_{tc}$ . The final supervector is built by the concatenation of these vectors  $S = \{s_1, \dots, s_C\}$  and is used to represent the whole sequence. In this work, the output feature vectors for each head  $H$  of the MSA layer are ob-

tained with (3) as a weighted sum equivalent to (7), where  $\bar{w}_{tc}$  corresponds to the rows of the matrix of self-attention weights  $A_h$  obtained with (2). In particular, for the class token, the normalized weights would be obtained from the last row of  $A_h$ . Therefore, the final class token obtained with this mechanism is the concatenation of the different head subvectors corresponding to the class token position, which can be expressed as the supervector presented previously  $S_{CLS} = \{s_{1-CLS}, \dots, s_{H-CLS}\}$ .

To introduce the class token in the system, one trainable vector parameter with the dimension of the feature vectors is defined when the network is initialized. For each batch, it is replicated and concatenated at the end of each input feature sequence in the training batch as an additional token. Hence, a single shared vector is trained to learn the final embedding representation.

In this work, we propose the use of a new sampling approach [28], and instead of having a single class token shared for the whole batch, we assume this sensitive parameter is the result of sampling a list of several vectors to be selected during the training by sampling them. In order to do that, we define a matrix of  $R$  vectors (*Token Matrix*) and sample it to take one of them for each example in the batch introducing uncertainty in the class token (*CLS Token*). However, the use of this approach leads to a complex and slower evaluation process since a sampling inference would have to be carried out to obtain the representations. For this reason, to avoid making the sampling inference, we have scheduled a forced reduction of the available vectors in the *Token Matrix* throughout the training process. Thus, at the end of this process, only one weight is different to zero, and the class token vector parameter is fixed. This strategy allows us to start the training (*Iteration 1*) with a matrix of several vectors to sample from and, gradually, we reduce the number of vectors as the training progresses to finish (*Iteration N*) with only one as the original class token as Fig. 1 depicts. Therefore, the training leads the system progressively to focus the relevant information on the first vector of the matrix. In addition, using this sampling approach, the system is trained to capture the uncertainty introduced by initially having a *Token Matrix* with  $R$  vectors to combine with the training batch data. Thus, each example from the batch is combined



**Fig. 1.** Evolution of the number of vectors in the token matrix that are available for sampling from the beginning of the training process (iteration 1) to the final iteration (iteration N). In each iteration, the dark vectors represent the enabled class tokens, while the light vectors are the disabled tokens.

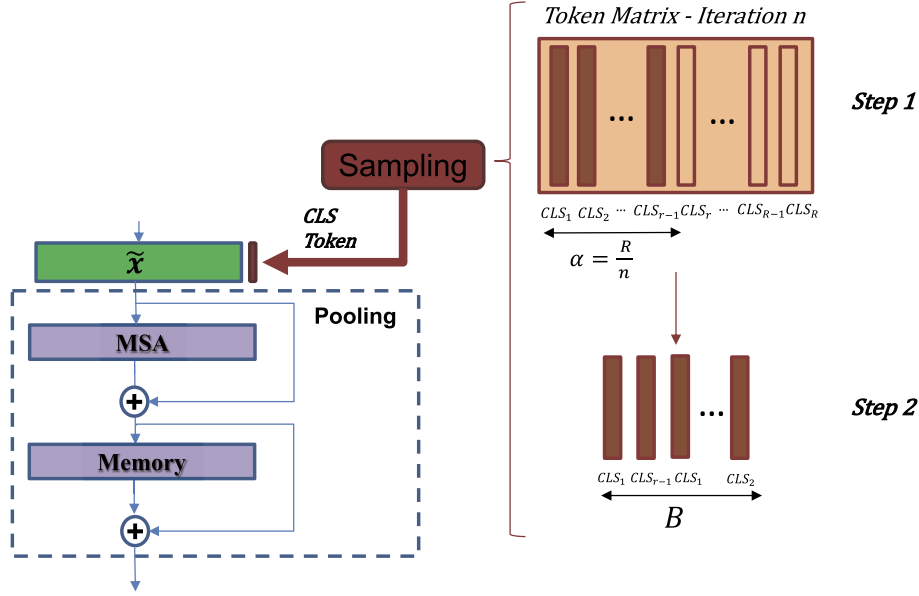


Fig. 2. Example of the sampling steps in iteration  $n$  of the training process. In Step 1, the available vectors of the token matrix in that iteration are defined and the random indexes of batch size ( $B$ ) are calculated. In Step 2, the class tokens are selected and added to the input of MSA layer.

**Algorithm 1:** Algorithm for sampling class token and introducing it before the pooling part.

**Input:** Input examples  $\mathcal{X}$ , batch size  $B$ , examples of batch  $x$ , the number of layers  $L$ , the total number of tokens to sample  $R$ , and the number of epochs  $N$

1. Define the vector with the number of sample vectors  $\alpha$  available to select each epoch:

$$\alpha = (\alpha_1, \dots, \alpha_n, \dots, \alpha_N), \alpha_n = R, \dots, 1, \text{ with } R \in \mathbb{R}$$

2. Define the random matrix of class token vectors:

$$TokenMatrix = random\_matrix(R)$$

for  $n = 1$  to  $N$  do

for  $x \in \mathcal{X}$  do

3. Sampling Process

3.1 Step 1, every update  $B$  integer indexes are randomly generated from the available  $\alpha_n$  vectors:

$$inds = random\_integer(\alpha_n, B)$$

3.2 Step 2, the correspondent tokens are selected:

$$CLS_{Token} = TokenMatrix[inds]$$

4. Network Training

4.1 Step 1, the class token is concatenated with the input to the MSA layer:

$$x_l = [x || CLS_{Token}]$$

4.2 Step 2, the new input is introduced to the first MSA layer in the pooling part and the  $L$  layers are applied:

for  $l = 1$  to  $L$  do

$$x'_l = x_l + MSA(x_l)$$

$$x_l = x'_l + Memory(x'_l)$$

end

4.3 Step 3, the state at the output of the last layer in the pooling block of the class token is used as final representation:

$$x_{CLS} = x'_l^{nd}$$

end

end

$$\alpha = (\alpha_1, \dots, \alpha_n, \dots, \alpha_N), \alpha_n = R, \dots, 1, \text{ with } R \in \mathbb{R} \quad (8)$$

where  $R$  is the number of tokens defined in the matrix, and  $N$  is the total number of iterations for training process. Among the number of tokens available at each iteration, a random selection of the batch size is made to select the index of the vectors. These vectors are selected from the distribution (*Token Matrix*) and used as class tokens (*CLS Token*) in the batch to concatenate to the input of the first MSA layer. The overall process is described in Algorithm 1. Besides, Fig. 2 shows a graphical example of how this sampling process is made in an intermediate iteration (*Iteration n*). In this graphical explanation, it can be observed how in an intermediate iteration the number of vectors has been reduced forcing the network to put the relevant information to represent the utterances in the vectors still available.

**4. Knowledge distillation with tokens**

Motivated by the benefits obtained when the training databases are not very large with Teacher-Student architecture based on CNNs [26], we have implemented this architecture using two transformer networks as Fig. 3 depicts. Using a Bayesian approach similar to [29], the teacher-student architecture allows providing robustness to the system. In this approach, the teacher and student networks are trained at the same time, unlike previous works [30,31] in which the teacher network is usually a pre-trained model to reduce complexity. Whether the teacher network had been a frozen model, negative training examples that obtain high posterior values in the teacher network would be learned as positive examples by the student network. Besides, different sources of distortion are applied to each of the input signals of both networks, so we have employed a data augmentation method called Random Erasing (RE) [25] to provide more variability to the input training data. With this kind of architecture, the teacher network has to predict augmented unseen data and the student network tries to mimic the label predictions produced by the teacher network using the class token output. This training strategy allows the student network to capture the variability in the predictions produced by the first network and model this uncertainty in the parameters during the training process. However, inspired by [9], we have also

with a random vector from the matrix which is reduced in size after each epoch until only one vector remains at the end, so more variability has to be modelled which helps to improve the robustness of the system.

To carry out this process, we define the following vector, which indicates to the neural network the number of tokens available at each iteration of the training process:

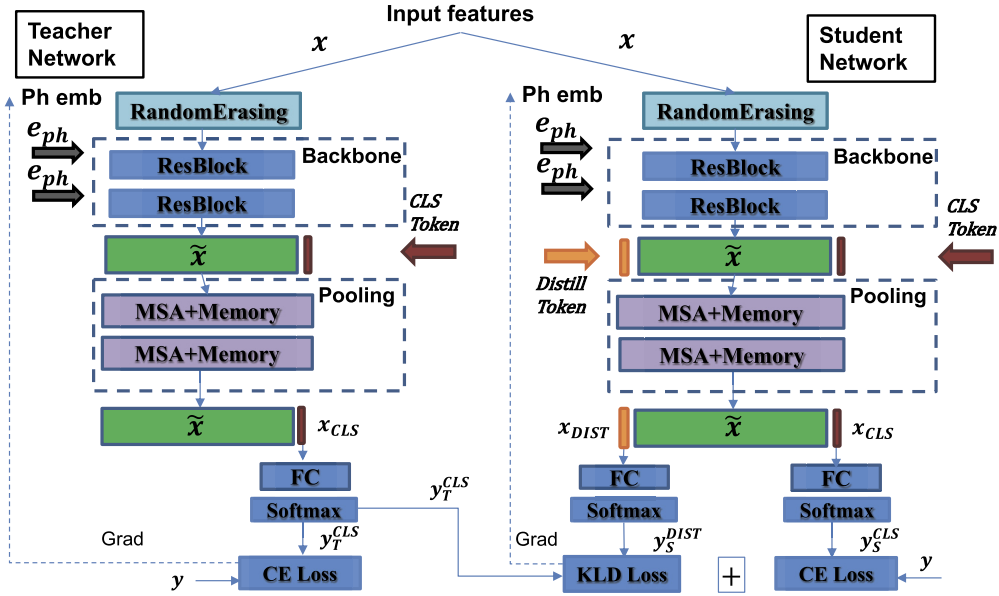


Fig. 3. Teacher-student architecture used to create the system, where the dashed line indicates the process of backpropagation of the gradients. Both networks are employed to train while for testing, the student network is the only used.

included an extra learnable token in the student network which is known as Distillation Token (*Distill Token*). The introduction of this extra token allows to implement of a multi-objective optimization by using the class token to reproduce the true label while the distillation token is trained to mimic the predictions of the teacher network. To achieve this, the Kullback-Leibler Divergence (KLD) loss between the student and teacher distributions is minimized. The KLD loss can be formulated as,

$$KLD = -\sum_{i=1}^I \sum_{j=1}^J p_T(y_i^{cls}|x_j) \cdot \log(p_S(y_i^{dist}|x_j)) + const, \quad (9)$$

where  $i$  and  $j$  are the speaker and utterance indices,  $x_j$  is the input signal,  $p_T(y_i^{cls}|x_j)$  is the output posterior probability of the label  $y_i^{cls}$  from the class token of the teacher model,  $p_S(y_i^{dist}|x_j)$  is the output posterior probability of the label  $y_i^{dist}$  from the distillation token of the student network for the same example, and  $const$  is defined in [29]. Hence, to train the teacher-student architecture showed in Fig. 3, we employ the following two loss expressions for teacher and student networks:

$$Loss_T = CE(y_T^{cls}, y), \quad (10)$$

$$Loss_S = KLD(y_S^{dist}, y_T^{cls}) + CE(y_S^{cls}, y), \quad (11)$$

where  $CE$  is the cross-entropy loss,  $y_S^{cls}$  is the class token output from the student network, and  $y$  are the ground truth labels.

## 5. System description

In this section, we describe the system architecture used in this work for text-dependent SV. Fig. 3 depicts this architecture where a teacher-student approach is employed. Both architectures follow the structure described in [21] with the same backbone and pooling parts. The backbone is based on two Residual Network (RN) [32] blocks with three layers each block. Additionally, these architectures need embeddings with positional information to help guiding the attention mask in the MSA layers. In this work, these embeddings ( $e_{ph}$ ) are extracted by a phonetic classifier network instead of using temporal position information [27]. For the pooling part, two MSA layers of 16 heads combined with two memory layers are employed. Moreover, before the first MSA layer, the

class token is concatenated to the input. In the case of the student network, the distillation token is also included. Thanks to the self-attention mechanism, these tokens learn to obtain a global representation for each utterance without applying the global average pooling. These representations, similar to supervector, are more convenient for text-dependent SV task since these global representations do not neglect the sequence order and are obtained automatically thanks to the self-attention mechanism. So external alignment mechanisms are not necessary to obtain them as in [15–17], where GMM or HMM posterior probabilities are needed to align speech frames to supervectors. Besides, the use of memory layers increases the amount of knowledge obtained by the network that can be stored. After training the system, a cosine similarity over the token representations is applied to perform the verification process. Note that this kind of system based on teacher-student consists of training of two architectures at the same time. Therefore, this process may involve a higher computational cost. However, during inference, only the student network is employed to extract the embeddings, so there is no extra inference time.

## 6. Experimental setup

### 6.1. Datasets

For the experiments, two text-dependent speaker verification datasets have been employed. The first set of experiments has been reported on the RSR2015 text-dependent speaker verification dataset [33]. This dataset comprises recordings from 157 males and 143 females. For each speaker, there are 9 sessions with 30 different phrases. This data is divided into three speaker subsets: background (bkg), development (dev) and evaluation (eval). In this paper, we develop our experiments with Part II, which is composed of short control command with a strong overlap of lexical content, and we employ only the bkg data for training. The eval part is used for enrollment and trial evaluation. This dataset has three evaluation conditions, but in this work, the most challenging, which is the Impostor-Correct case, is the only one that has been evaluated and employed in the text-dependent SV. Note that there are other systems that obtain relevant results for this dataset, similar to those presented below. Nevertheless, such systems are based on

traditional models such as Hidden Markov Models (HMMs) [33,34] or neural network architectures focused on two different streams for speaker and utterance information [35,36].

The second dataset used is the DeepMine database [37]. This corpus consists of three different parts of which we employ the files selected for the Short-duration Speaker Verification (SdSV) Challenge 2020 [38] from Part 1. Part 1 is the text-dependent part which is composed of 5 Persian and 5 English phrases and contains 963 females and males speakers. This data is divided into two subsets: train with 101.063 audio files and evaluation with 69.542 audio files. Finally, the phonetic classification network [27] has been trained using LibriSpeech [39] to extract phonetic embeddings. Unlike other works presented in the challenge [40,41], we have not used VoxCeleb 1 and 2 datasets [42,43] in the neural network training process. Motivated by the fact that in some situations and applications is required the implementation of custom systems with the few available in domain-data. For this reason, we have developed systems only with the in-domain data.

## 6.2. Experimental description

To carry out the experiments with the RSR2015 dataset, a set of features composed of 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) with their derivatives are employed as input. While for the experiments using the DeepMine dataset, we have employed 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 to obtain a final input dimension of 57. Moreover, phonetic embeddings of 256 dimensions have been used as positional information. As the optimizer for the experiments in this work, the Adam optimizer is employed with a learning rate that increases from  $10^{-3}$  to  $5 \times 10^{-3}$  during 60 epochs and then decays from  $5 \times 10^{-3}$  to  $10^{-4}$ . In addition, training data is fed into the systems with a minibatch size of 32.

## 7. Results

In this paper, two sets of experiments have been carried out to evaluate the proposals with both databases. We compare the different approaches to obtain the representations with a single neural network using the same architecture as the teacher network: the use of the traditional global average pooling (AVG), the attentive pooling (ATT) and the introduction of the learnable class token (CLS). For the class token approach, we evaluate our proposal of sampling a matrix of  $R$  vectors and reducing it until having a single vector (Sampling). This parameter is also swept for different values of  $R$ , including  $R = 1$  that corresponds to the original idea of having a single token and repeating it. Moreover, we analyze the effect produced by the fact of using a teacher-student architecture with an extra distillation token (CLS - DIST).

In order to evaluate these experiments, we have measured the performance using three metrics. Equal Error Rate (EER) which measures the discrimination ability of the system. NIST 2008 and 2010 minimum Detection Cost Functions (DCF08, DCF10) [44,45] which measure the cost of detection errors in terms of a weighted sum of false alarm and miss probabilities for a decision threshold, and a priori probability.

### 7.1. Class token study

A first set of experiments was performed to compare the use of a class token to obtain global utterance descriptors with the use of a global average pooling method or the attentive pooling proposed in [46]. Thus, we study the two approaches to introduce this vector

explained during this work and the effect of the number of vectors chosen for the sampling approach.

Table 1 presents EER, DCF08 and DCF10 results for the experiments with RSR2015-part II dataset. Regardless of the number of vectors in the sampling for class tokens, if we apply our proposed strategy to introduce the tokens with a sampling alternative, the obtained performance is better. In addition, the results show how employing a learnable token outperforms the use of an average embedding or an attentive pooling embedding. Note that the token is trained through self-attention and keeping the temporal structure to obtain a global utterance representation, while the average embedding neglects this information that is relevant to the SV task. As we can also observe with the sweep of  $R$  value, the use of several vectors to create the token matrix is better than using a single vector and repeating it for the whole batch, which corresponds to the original way of applying this approach. The case of having a single vector and repeat it corresponds with the experiments with  $R = 1$ . However, when the number of available tokens is too large, the performance begins to degrade. This degradation could be caused by the introduction of too much variability that the system is not able to model as the architectures employed are not so large, which means that there are a limited number of different tokens to carry out the training process.

In Table 2, the results obtained in DeepMine-part I database are shown. Unlike the other dataset, the training data in DeepMine is larger, which indicates that the lack of data is not so critical to train a powerful and robust system. Therefore, the replacement of the average embedding or attentive pooling embedding by a class token improves the performance only slightly. Besides, the sweep of  $R$  value shows that the evolution of the female and male results separately do not follow the same trend as occurs in the RSR-Part II results.

### 7.2. Effect of knowledge distillation using tokens

In this section, we analyze the effect of introducing an approach based on Knowledge Distillation philosophy which consists of a teacher-student architecture. Furthermore, in this approach, an extra distillation token (CLS - DIST) is incorporated [9]. This approach has been employed to compare the performance obtained in the case of the average global pooling as well as in the proposed sampling approach to use the class token. In this second case, we have developed the teacher-student architecture using the  $R$  value of the best configuration obtained in the previous section, and also, the case of  $R = 1$  as it is the usual way to apply this class token approach in the literature.

Results of these experiments in RSR-Part II are shown in Table 3. Regardless of the kind of approach to obtain the representations used, we can observe that the use of an architecture based on a teacher-student approach improves the robustness and achieves better performance in all the alternatives to extract the representations. Moreover, the best performance is obtained applying our proposed strategy to introduce the tokens with a sampling alternative with more than a single vector.

On the other hand, Table 4 presents the performance of systems with DeepMine-part I. In this case, the results show that the application of only the teacher-student architecture does not improve the systems. However, the use of the teacher-student architecture and the extra distillation token (CLS - DIST), combined with the sampling strategy with several token vectors also allows achieving a more robust system and a significant improvement in the results.

### 7.3. Analysis of class token self-attention representations

In view of the relevant results obtained, we have also conducted an analysis to interpret what the self-attention matrix  $A$  is learn-

**Table 1**

Experimental results on RSR2015 Part II [33] eval subset, showing EER%, DCF08 and DCF10. These results were obtained to compare the different approaches to obtain the representations: average, attentive or sampling strategies.

Architecture			Female			Male			Female + Male		
Type	T/S	Sampling	EER%	DCF08	DCF10	EER%	DCF08	DCF10	EER%	DCF08	DCF10
AVG	no	—	4.64	0.228	0.669	4.92	0.244	0.716	4.79	0.237	0.706
ATT	no	—	4.07	0.188	0.618	4.44	0.211	0.615	4.53	0.213	0.646
CLS	no	R=1	3.71	0.174	0.580	4.27	0.215	0.679	4.12	0.201	0.634
		R=50	3.37	0.169	0.580	4.04	0.199	0.601	3.75	0.187	0.606
		R=100	<b>3.33</b>	<b>0.158</b>	<b>0.552</b>	<b>3.68</b>	<b>0.182</b>	<b>0.552</b>	<b>3.57</b>	<b>0.173</b>	<b>0.565</b>
		R=200	3.55	0.171	0.562	4.09	0.199	0.607	3.86	0.189	0.587

**Table 2**

Experimental results on DeepMine [37] eval subset, showing EER%, DCF08 and DCF10. These results were obtained to compare the different approaches to obtain the representations: average, attentive or sampling strategies.

Architecture			Female			Male			Female + Male		
Type	T/S	Sampling	EER%	DCF08	DCF10	EER%	DCF08	DCF10	EER%	DCF08	DCF10
AVG	no	—	3.92	0.135	0.411	3.02	0.137	0.676	3.58	0.136	0.521
ATT	no	—	5.73	0.193	0.468	5.39	0.210	0.664	5.60	0.200	0.551
CLS	no	R=1	3.81	0.128	0.389	3.32	0.143	0.697	3.60	0.134	0.520
		R=50	3.92	0.131	0.393	3.19	0.140	0.668	3.62	0.134	0.519
		R=100	<b>3.69</b>	<b>0.124</b>	<b>0.379</b>	3.09	0.137	0.658	<b>3.43</b>	<b>0.129</b>	<b>0.505</b>
		R=200	3.89	0.133	0.417	<b>2.92</b>	<b>0.133</b>	<b>0.655</b>	3.50	0.133	0.521

**Table 3**

Experimental results on RSR2015 Part II [33] eval subset, showing EER%, DCF08 and DCF10. These results were obtained to compare the use of a teacher-student architecture for the different approaches to obtain the representations: average or sampling strategies.

Architecture			Female			Male			Female + Male		
Type	T/S	Sampling	EER%	DCF08	DCF10	EER%	DCF08	DCF10	EER%	DCF08	DCF10
AVG	no	—	4.64	0.228	0.669	4.92	0.244	0.716	4.79	0.237	0.706
	yes	—	3.52	0.170	0.587	3.78	0.186	0.579	3.74	0.185	0.602
CLS	no	R=1	3.71	0.174	0.580	4.27	0.215	0.679	4.12	0.201	0.634
CLS-DIST	yes	R=1	3.01	0.148	0.548	3.40	0.173	0.557	3.31	0.167	0.558
CLS	no	R=100	3.33	0.158	0.552	3.68	0.182	0.552	3.57	0.173	0.565
CLS-DIST	yes	R=100	<b>2.47</b>	<b>0.122</b>	<b>0.414</b>	<b>2.83</b>	<b>0.138</b>	<b>0.463</b>	<b>2.68</b>	<b>0.133</b>	<b>0.443</b>

**Table 4**

Experimental results on DeepMine [37] eval subset, showing EER%, DCF08 and DCF10. These results were obtained to compare the use of a teacher-student architecture for the different approaches to obtain the representations: average or sampling strategies.

Architecture			Female			Male			Female + Male		
Type	T/S	Sampling	EER%	DCF08	DCF10	EER%	DCF08	DCF10	EER%	DCF08	DCF10
AVG	no	—	3.92	0.135	0.411	3.02	0.137	0.676	3.58	0.136	0.521
	yes	—	4.07	0.135	0.401	3.04	0.141	0.646	3.65	0.138	0.501
CLS	no	R=1	3.81	0.128	0.389	3.32	0.143	0.697	3.60	0.134	0.520
CLS-DIST	yes	R=1	3.80	0.131	0.395	3.25	0.144	<b>0.621</b>	3.57	0.135	0.494
CLS	no	R=100	3.69	0.124	<b>0.379</b>	3.09	0.137	0.658	3.43	0.129	0.505
CLS-DIST	yes	R=100	<b>3.51</b>	<b>0.122</b>	0.385	<b>2.68</b>	<b>0.122</b>	0.652	<b>3.19</b>	<b>0.123</b>	<b>0.492</b>

ing in each system. To perform this analysis, we have employed the system with the best performance from each database, and within these systems, the last MSA layer of the student model has been selected to make the representations. In addition, we have chosen different utterances to analyze in Fig. 4 and Fig. 5. For each utterance, three figures are plotted: the spectrogram of the utterance, the matrix of attention weights corresponding to the class token for each of the 16 heads of the MSA layer, and the sum of the weights of these class token attentions.

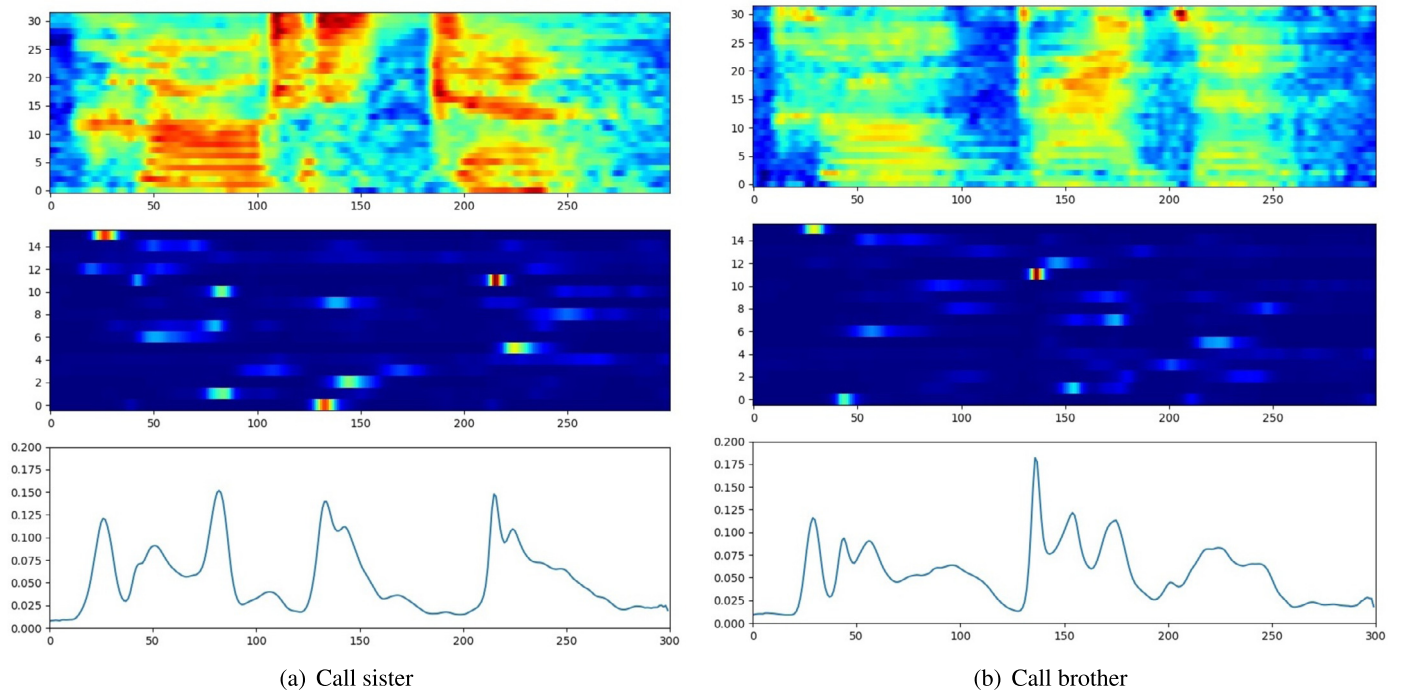
In Fig. 4, two examples of utterances of different phrases (“Call sister”, “Call brother”) pronounced by the same speaker are shown. These examples are obtained from the evaluation set of the RSR-Part II database. Whether we look in the middle and bottom figures, we can observe the relevant information learned by the self-attention weights to correctly determine the phrase and speaker of each utterance using the class token. Note that these two phrases of example begin exactly the same with the word *Call*, so focusing on the beginning of the figures, we observe how the self-

attention gives similar relevance in both cases to the areas of same phonemes. Moreover, we can also see that the weights do not pay attention to the area at the beginning and end of the utterances that correspond to moments of silence.

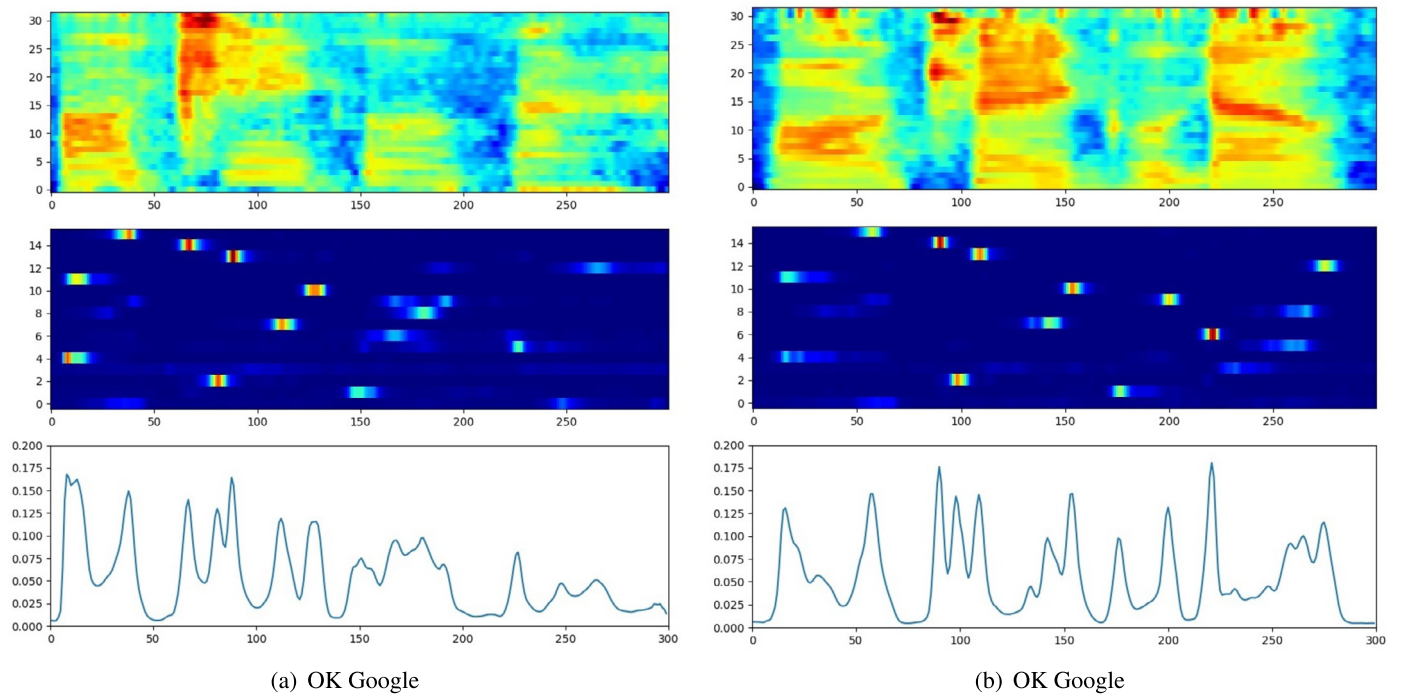
Fig. 5 represents two examples of utterances of the same phrase (“OK Google”) pronounced by different speakers. In this case, the examples are obtained from the evaluation set of the DeepMine database. Note that since these figures are of the same phrase, self-attention is focused on the same areas, but different relevance is given to some of them. Besides, the effect of not focusing on the beginning and end of the utterance also occurs in these examples.

## 8. Conclusion

In this paper, we have presented a novel approach for the SV task. This approach is based on the use of a learnable class token to obtain a global utterance descriptor instead of employing the average pooling. Moreover, we have developed an alternative to create the class token with a sampling strategy that introduces



**Fig. 4.** Visualizing two examples of different phrases of RSR-Part II which are pronounced by the same speaker. In both cases, three representations are presented. The figure on top shows the spectrogram of each phrase. In the middle, the attention weights learnt by the class token for each of the 16 heads in the last MSA layer are depicted. Finally, the plot on bottom is the sum of the rows of the previous weight attention matrix.



**Fig. 5.** Visualizing two examples of the same phrase of DeepMine which are pronounced by different speakers. In both cases, three representations are presented. The figure on top shows the spectrogram of each phrase. In the middle, the attention weights learnt by the class token for each of the 16 heads in the last MSA layer are depicted. Finally, the plot on bottom is the sum of the rows of the previous weight attention matrix.

uncertainty that helps to generalize better. Apart from the previous approach, we have also employed a teacher-student architecture combined with an extra distillation token to develop a more robust system. Using this architecture, the distillation token in the student network learns to replicate the predictions from the teacher network. Both proposals were evaluated in two text-dependent SV databases. Results achieved show in RSR2015-part II that each of

the approaches introduced to obtain a robust system and reduce potential underperformance due to the lack of data improves the overall performance. However, in DeepMine-part 1, the results obtained replacing only the average embedding by the class token present a small improvement, while the use of a teacher-student architecture achieves a great improvement and confirms the power of this kind of approach to train the systems.



## CRediT authorship contribution statement

**Victoria Mingote:** Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Antonio Miguel:** Conceptualization, Investigation, Methodology, Software, Supervision, Writing – review & editing. **Alfonso Ortega:** Supervision, Writing – review & editing. **Eduardo Lleida:** Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie Grant 101007666; in part by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR under Grant PDC2021-120846-C41, by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the grant PRE2018-083312, by the Government of Aragón (Grant Group T36\_20R), and by Nuance Communications, Inc.

## References

- [1] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: closing the gap to human-level performance in face verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708, <https://doi.org/10.1109/CVPR.2014.220>.
- [2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, X-vectors: robust dnn embeddings for speaker recognition, in: 2018 ICASSP, 2018, pp. 5329–5333.
- [3] E. Hoffer, N. Ailon, Deep metric learning using triplet network, Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9370 (2010) (2015) 84–92, [https://doi.org/10.1007/978-3-319-24261-3\\_{\\_}7](https://doi.org/10.1007/978-3-319-24261-3_{_}7).
- [4] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
- [5] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, S. Khudanpur, Deep neural network-based speaker embeddings for end-to-end speaker verification, in: Spoken Language Technology Workshop (SLT), 2016, IEEE, 2016 IEEE, pp. 165–170.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova Bert, Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAAACL-HLT, 2019, pp. 4171–4186.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, arXiv preprint, arXiv:2010.11929.
- [9] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, arXiv preprint, arXiv:2012.12877.
- [10] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, T. Kipf, Object-centric learning with slot attention, in: NeurIPS 2020, 2020.
- [11] M. India, P. Safari, J. Hernando, Self multi-head attention for speaker recognition, in: Proc. Interspeech 2019, 2019, pp. 4305–4309.
- [12] H.-j. Shim, J. Heo, J.-h. Park, G.-h. Lee, H.-j. Yu, Graph attentive feature aggregation for text-independent speaker verification, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 7972–7976.
- [13] R. Wang, J. Ao, L. Zhou, S. Liu, Z. Wei, T. Ko, Q. Li, Y. Zhang, Multi-view self-attention based transformer for speaker recognition, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 6732–6736.
- [14] B. Han, Z. Chen, Y. Qian, Local information modeling with self-attention for speaker verification, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 6727–6731.
- [15] V. Mingote, A. Miguel, A. Ortega, E. Lleida, Differentiable supervector extraction for encoding speaker and phrase information in text dependent speaker verification, in: Proceedings of IberSPEECH 2018, 2018, pp. 1–5, <https://doi.org/10.21437/IberSPEECH.2018-1>.
- [16] V. Mingote, A. Miguel, A. Ortega, E. Lleida, Supervector extraction for encoding speaker and phrase information with neural networks for text-dependent speaker verification, Appl. Sci. 9 (16) (2019) 3295.
- [17] V. Mingote, A. Miguel, A. Ortega, E. Lleida, Optimization of the area under the ROC curve using neural network supervectors for text-dependent speaker verification, Comput. Speech Lang. 63 (2020) 101078.
- [18] D.A. Reynolds, R.C. Rose, et al., Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Trans. Speech Audio Process. 3 (1) (1995) 72–83.
- [19] D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models, Digit. Signal Process. 10 (1–3) (2000) 19–41.
- [20] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
- [21] V. Mingote, A. Miguel, A. Ortega, E. Lleida, Memory layers with multi-head attention mechanisms for text-dependent speaker verification, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6154–6158, <https://doi.org/10.1109/ICASSP39728.2021.9414859>.
- [22] G. Lample, A. Sablayrolles, M. Ranzato, L. Denoyer, H. Jégou, Large memory layers with product keys, in: Advances in Neural Information Processing Systems, 2019, pp. 8546–8557.
- [23] W. Cai, Z. Cai, X. Zhang, X. Wang, M. Li, A novel learnable dictionary encoding layer for end-to-end language identification, in: 2018 ICASSP, 2018, pp. 5189–5193.
- [24] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: NIPS 2014 Deep Learning Workshop, 2015, pp. 1–9, <https://doi.org/10.1063/1.4931082>, <http://arxiv.org/abs/1503.02531>.
- [25] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 13001–13008.
- [26] V. Mingote, A. Miguel, D. Ribas, A. Ortega, E. Lleida, Knowledge distillation and random erasing data augmentation for text-dependent speaker verification, in: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 6824–6828.
- [27] I. Viñals, D. Ribas, V. Mingote, J. Llombart, P. Gimeno, A. Miguel, A. Ortega, E. Lleida, Phonetically-aware embeddings, wide residual networks with time-delay neural networks and self attention models for the 2018 NIST speaker recognition evaluation, in: Proc. Interspeech 2019, 2019, pp. 4310–4314.
- [28] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International Conference on Machine Learning, PMLR, 2015, pp. 1613–1622.
- [29] A. Korattikara, V. Rathod, K. Murphy, M. Welling, Bayesian dark knowledge, Adv. Neural Inf. Process. Syst. (2015) 1–9, <https://doi.org/10.1017/CBO9781107415324.004>.
- [30] P. Shen, X. Lu, S. Li, H. Kawai, Feature representation of short utterances based on knowledge distillation for spoken language identification, in: Interspeech, 2018, pp. 1813–1817.
- [31] P. Shen, X. Lu, S. Li, H. Kawai, Interactive learning of teacher-student model for short utterance spoken language identification, in: Proc. ICASSP, IEEE, 2019, pp. 5981–5985.
- [32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [33] A. Larcher, K.A. Lee, B. Ma, H. Li, Text-dependent speaker verification: classifiers, databases and RSR2015, Speech Commun. 60 (2014) 56–77.
- [34] R.K. Das, M. Madhavi, H. Li, Compensating utterance information in fixed phrase speaker verification, in: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 2018, pp. 1708–1712.
- [35] T. Liu, M.C. Madhavi, R.K. Das, H. Li, A unified framework for speaker and utterance verification, in: Proc. Interspeech 2019, 2019, pp. 4320–4324.
- [36] T. Liu, R.K. Das, M. Madhavi, S. Shen, H. Li, Speaker-utterance dual attention for speaker and utterance verification, Proc. Interspeech 2020 (2020) 4293–4297.
- [37] H. Zeinali, L. Burget, J. 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.
- [38] H. Zeinali, K.A. Lee, J. Alam, L. Burget, Short-duration Speaker Verification (SdSV) Challenge 2020: the Challenge Evaluation Plan., Tech. Rep., arXiv preprint arXiv:1912.06311.

- [39] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, Librispeech: an ASR corpus based on public domain audio books, in: 2015 ICASSP, 2015, pp. 5206–5210.
- [40] A. Lozano-Diez, A. Silnova, B. Pulugundla, J. Rohdin, K. Vesely, L. Burget, O. Plhot, O. Glembek, O. Novotny, P. Matejka, BUT text-dependent speaker verification system for SdSV challenge 2020, in: INTERSPEECH 2020, 2020, pp. 761–765.
- [41] Z. Chen, Y. Lin, Improving X-vector and PLDA for text-dependent speaker verification, in: INTERSPEECH, 2020, pp. 726–730.
- [42] A. Nagrani, J.S. Chung, A. Zisserman, VoxCeleb: a large-scale speaker identification dataset, in: Proc. Interspeech 2017, 2017, pp. 2616–2620.
- [43] J.S. Chung, A. Nagrani, A. Zisserman, VoxCeleb2: deep speaker recognition, in: Proc. Interspeech 2018, 2018, pp. 1086–1090.
- [44] The NIST Year 2008 Speaker Recognition Evaluation Plan, [https://www.nist.gov/sites/default/files/documents/2017/09/26/sre08\\_evalplan\\_release4.pdf](https://www.nist.gov/sites/default/files/documents/2017/09/26/sre08_evalplan_release4.pdf), 2008.
- [45] The NIST Year 2010 Speaker Recognition Evaluation Plan, [https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST\\_SRE10\\_evalplan-r6.pdf](https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_evalplan-r6.pdf), 2010.
- [46] B. Desplanques, J. Thienpondt, K. Demuynck ECAPA-TDNN, Emphasized channel attention, propagation and aggregation in TDNN based speaker verification, in: Proc. Interspeech 2020, 2020, pp. 3830–3834.



**Victoria Mingote** received the Bachelor's and Master's degree in Telecommunication Engineering from the University of Zaragoza, Spain, in 2014 and 2016, respectively. After that, she joined ViVoLab research group as a Ph.D student and received her Ph.D. degree in 2022 from the University of Zaragoza. She is currently a post-doctoral researcher in the ViVoLab research group. She has achieved several publications of her work in different international journals and conference proceedings.

Her research interests expands through the areas of signal processing, machine learning, multimodal verification and identification (voice and face), and language identification.



**Antonio Miguel** was born in Zaragoza, Spain. He received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the University of Zaragoza (UZ), Zaragoza, Spain, in 2001 and 2008, respectively. From 2000 to 2006, he was with the Communication Technologies Group, Department of Electronic Engineering and Communications, UZ, under a research grant. Since 2006, he has been an Associate Professor in the Department of Electronic Engineering

and Communications, UZ. His current research interests include acoustic modeling for speech and speaker recognition.



**Alfonso Ortega** received the Telecommunication Engineering and the Ph.D. degrees from the University of Zaragoza, Zaragoza, Spain, in 2000 and 2005, respectively. He is Associate Director of the Aragon Institute for Engineering Research (I3A), University of Zaragoza. In 2006, he was visiting scholar in the Center for Robust Speech Systems, University of Texas at Dallas, USA. He has participated in more than 50 research projects funded by national or international public institutions and more than 30 research projects for several companies. He is the author of more than 100 papers published in international journals or conference proceedings and several international patents. He is currently Associate Professor in the Department of Electronic Engineering and Communications, University of Zaragoza. His research interests include speech processing, analysis and modeling, automatic speaker verification, and automatic speech recognition. His Ph.D. thesis, advised by Dr. E. Lleida, received the Ph.D. Extraordinary Award and the Telefonica Chair Award to the best technological Ph.D.



**Eduardo Lleida** received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in signal processing from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1985 and 1990, respectively. From 1986 to 1988, he was involved in his doctoral work in the Department of Signal Theory and Communications, UPC. From 1989 to 1990, he was an Assistant Professor, and from 1991 to 1993, he was an Associated Professor in the Department of Signal Theory and Communications, UPC. From February 1995 to January 1996, he was with AT&T Bell Laboratories, Murray Hill, NJ, USA, as a Consultant in speech recognition. He is currently a full Professor of signal theory and communications in the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain, and a member of the Aragon Institute for Engineering Research, where he is heading the ViVoLab research group in speech technologies. He has been the doctoral advisor of 12 doctoral students. He has managed more than 50 speech-related projects, being inventor in seven worldwide patents, and coauthored more than 200 technical papers in the field of speech, speaker and language recognition, speech enhancement and recognition in adverse acoustic environments, acoustic modeling, confidence measures, and spoken dialogue systems.