# Deep neural networks for automatic speech processing: a survey from large corpora to limited data

Vincent Roger, Jérôme Farinas, Julien Pinquier

▶ **To cite this version:**

Vincent Roger, Jérôme Farinas, Julien Pinquier. Deep neural networks for automatic speech processing: a survey from large corpora to limited data. EURASIP Journal on Audio, Speech, and Music Processing, 2022, article 19, pp.1-15. 10.1186/s13636-022-00251-w . hal-03755976

**HAL Id: hal-03755976**
**https://hal.science/hal-03755976**

Submitted on 22 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Roger *et al.*
*EURASIP Journal on Audio, Speech, and Music Processing* (2022) 2022:19
https://doi.org/10.1186/s13636-022-00251-w

EURASIP Journal on Audio, Speech, and Music Processing

REVIEW

Open Access

# Deep neural networks for automatic speech processing: a survey from large corpora to limited data

Vincent Roger*, Jérôme Farinas and Julien Pinquier

**Abstract**

Most state-of-the-art speech systems use deep neural networks (DNNs). These systems require a large amount of data to be learned. Hence, training state-of-the-art frameworks on under-resourced speech challenges are difficult tasks. As an example, a challenge could be the limited amount of data to model impaired speech. Furthermore, acquiring more data and/or expertise is time-consuming and expensive. In this paper, we focus on the following speech processing tasks: automatic speech recognition, speaker identification, and emotion recognition. To assess the problem of limited data, we firstly investigate state-of-the-art automatic speech recognition systems, as this is the hardest task (due to the wide variability in each language). Next, we provide an overview of techniques and tasks requiring fewer data. In the last section, we investigate few-shot techniques by interpreting under-resourced speech as a few-shot problem. In that sense, we propose an overview of few-shot techniques and the possibility of using such techniques for the speech problems addressed in this survey. It is true that the reviewed techniques are not well adapted for large datasets. Nevertheless, some promising results from the literature encourage the usage of such techniques for speech processing.

**Keywords:** Audio processing, Deep learning techniques, Deep neural networks, Few-shot learning, Speech analysis, Under-resourced languages

## 1 Introduction

Automatic speech processing systems have improved dramatically over the past few years, especially automatic speech recognition (ASR) systems. This is also the case for other speech processing tasks such as speaker identification or emotion classification. This success was made possible by the large amount of annotated data available combined with the extensive use of deep learning techniques and the capacity of modern graphics processing units. Some modelings are already deployed for everyday use, such as personal assistants in smartphones and connected speakers. Nevertheless, challenges remain for automatic speech processing systems. They lack the robustness to deal with extensive vocabularies in a real-world environment: this includes noises, distance from the speaker, paucity of robustness to speech variations, reverberations, and other alterations [1]. Some challenges, such as CHiME [2], provide data to let the community try to handle some of these problems. Ways are sought to improve the generalization of modern models by avoiding the inclusion of additional annotated data for each possible environment.

State-of-the-art (SOTA) techniques for most speech tasks require large datasets. Indeed, with modern DNN speech processing systems, having more data usually implies better performance. The TED-LIUM 3 (from [3], with 452 h) provides more than twice the data of the TED-LIUM 2 dataset. The authors thus obtain better results by training their model on TED-LIUM 3 than training their model using TED-LIUM 2 data.

*Correspondence: Vincent.Roger@irit.fr; roger.vins.11@gmail.com

IRIT, Université de Toulouse, CNRS, Toulouse, France

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**Table 1** SOTA results over test-clean set from LibriSpeech and quantity of data used. Some self-supervised results are provided from [7]

| Model type | Quantity of data used | | WER |
|---|---|---|---|
| | Pre-training | Training | |
| Pase+ [8] | 50h | 960h | 16.62 |
| Wav2Vec2.0 [9] | 960h | | 4.79 |
| | 60k h | | 3.10 |
| HuBERT [10] | 960h | | 4.79 |
| | 60k h | | 2.94 |
| Hybrid model [11] | - | | 2.7 |
| End to end supervised [12] | - | | 2.44 |
| Wav2Vec2.0 using conformers and spec augment [13] | 60k h | | 1.4 |
| Wav2Vec using BERT XXL [14] | 60k h | | 1.4 |

**Table 2** SOTA results over test-other set from LibriSpeech and quantity of data used. Some self-supervised results come from [7] experiments

| Model type | Quantity of data used | | WER |
|---|---|---|---|
| | Pre-training | Training | |
| End to end supervised [12] | - | 960h | 8.29 |
| Hybrid model [11] | - | | 5.7 |
| Wav2Vec2.0 using conformers and spec augment [13] | 60k h | | 2.6 |
| Wav2Vec using BERT XXL [14] | 60k h | | 2.5 |

**Table 3** SOTA results over IEMOCAP using 4 emotions (happiness, neutral, anger, and sadness) and quantity of data used. Self-supervised results come from [7] experiments

| Model type | Quantity of data used | | Accuracy |
|---|---|---|---|
| | Pre-training | Training | |
| Pase+ [8] | 50h | 12h | 57.86 |
| Wav2Vec2.0 [9] | 960h | | 63.43 |
| | 60k h | | 65.64 |
| HuBERT [10] | 960h | | 64.92 |
| | 60k h | | 67.62 |
| Multitask approach [15] | - | + labels for the other task | 81.6 |
| DAAN [16] | 1 billion words for lexical | | 82.7 |

**Table 4** SOTA results over VoxCeleb1 and quantity of data used. Self-supervised results come from [7] experiments

| Model type | Quantity of data used | | Accuracy |
|---|---|---|---|
| | Pre-training | Training | |
| Pase+ [8] | 50h | 350h | 37.99 |
| Wav2Vec2.0 [9] | 960h | | 75.18 |
| | 60k h | | 86.14 |
| HuBERT [10] | 960h | | 81.42 |
| | 60k h | | 90.33 |
| AutoSpeech [17] | - | | 87.66 |

This improvement in performance for ASR systems is also observed with the LibriSpeech dataset, from [4]. V. Panayotov et al. obtained better results on the Wall Street Journal (WSJ) test set by training a model using the LibriSpeech dataset (1000 h) than with the WSJ training set (82 h) [4].

This phenomenon, that having more data leads to better performance, is also observable in speaker recognition with the VoxCeleb 2 dataset compared to the VoxCeleb dataset [5]: the authors increased the number of sentences from 100,000 to one million and increased the number of individuals from 1251 to 6112 compared to the previous version of VoxCeleb. They thus obtained better performance than by training their model with the previous VoxCeleb dataset.

We summarized state-of-the-art approaches for automatic speech recognition in Tables 1 and 2, for emotion recognition in Table 3, and for speaker recognition in Table 4. From these tables, we can see that having more data does not always induce better results. Nevertheless, using more speech as pretraining of an unsupervised model (such as the libri-light [6] 60k hours) and bigger models (model having more parameters) helps to obtain some state-of-the-art results on unimpaired speech. Hence, scaling either the number of parameters or the amount of data can be useful when possible.

With under-resourced languages (such as [18]) and/or some tasks (pathology detection with speech signals), we lack large datasets [19]. By under-resourced, we mean limited digital resources (limited acoustic and text corpora) and/or a lack of linguistic expertise. For a more precise definition and details of the problem, see [20]. Some non-conventional speech tasks such as disease detection (such as Parkinsons, gravity of head and neck cancer and others) using audio are examples of under-resourced tasks [19]. Training deep neural network models in such contexts is a challenge for these under-resourced speech datasets. This is especially the case for tasks involving a large vocabulary. M. Moore et al. showed that recent ASR systems are not well adapted for impaired speech [21], and M. B. Mustafa et al. showed the difficulties in adapting such models with limited amounts of data [22].

Few-shot learning consists of training a model using *k*-shot (where shot means an example per class), where

$k \geq 1$ and $k$ is a low number. Training an ASR system on a new language, adapting an ASR system on pathological speech, or performing speaker identification (with impaired voice) with few examples are still complicated tasks [19, 21]. We think that few-shot techniques may be useful for tackling these problems.

This survey will be focused on how to train deep neural network (DNN) models with low resources for speech data with non-overlapping mono signals. Therefore, we will first review SOTA ASR techniques that use a large amount of data (Section 2). Then, we will review techniques and speech tasks (speaker identification, emotion recognition) requiring fewer data than SOTA techniques (Section 3). We will also look into pathological speech processing for ASR using adaptation techniques (Section 3.2). Finally, we will review few-shot techniques for audio (Section 4) which is the focus of this survey.

## 2 Automatic speech processing

In this section, we will review SOTA ASR, speaker identification, and emotion recognition systems using multi-models and end-to-end models. Here, we are focused on mono speech sequences $\mathbf{x}=[x_1,x_2,\ldots,x_n]$ where $x_i$ can be speech features or audio samples. ASR systems consist in matching $\mathbf{x}$ into a sequence of words $\mathbf{y}=[y_1,y_2,\ldots,y_u]$ (where $u \leq n$), while speaker and emotion recognition systems map the sequence $\mathbf{x}$ into a single outcome $y$ representing a class. The reviewed systems were evaluated using word error rate (WER) as a measure for ASR systems and the accuracy of recognition for speaker identification and emotion recognition.

### 2.1 Multi-models

A multi-model approach consists in solving a problem using multiple models. These models are designed to solve sub-tasks (related to the problem) and the targeted task. The minimum configuration is with two models (let us say $f$ and $g$) to solve a given task. Classically for the ASR task, we can first train an acoustic model (a phoneme classifier or equivalent sound unit) then train on top of it a language model that output the desired sequence of words. Hence, we have:

$$\hat{\mathbf{y}} = f(g(\mathbf{x})), \tag{1}$$

with $f$ being the language model predicting the output sequence $\hat{\mathbf{y}}$ (which can be different from the real sequence $\mathbf{y}$) and $g$ being the acoustic model. Note that for emotion and speaker recognition, the output of $f$ is $\hat{y}$ instead of $\hat{\mathbf{y}}$ and is not necessarily a language model. Both can be trained separately or conjointly. Usually, hybrid models are used as acoustic models.

Hybrid models consist in using probabilistic models with deterministic ones. Probabilistic models involve randomness using random variables combined with trained parameters. Hence, every prediction is slightly different on a given example $\mathbf{x}$. Gaussian mixture models (GMMs) are an example of such models. Deterministic models do not involve randomness and every prediction is the same, given an input $\mathbf{x}$. DNNs are an example of such models. A popular and efficient hybrid model is the DNN-hidden Markov model (DNN-HMM). DNN-HMM consists in replacing the GMMs that estimate the probability density functions by DNNs. The DNNs can be trained as phoneme classifiers. They form the acoustic model. This acoustic model is combined with a language model (LM) that maps the phonemes into a sequence of words. Lüscher et al. used DNN-HMMs combined with a LM to obtain SOTA on the LibriSpeech test-other set (official augmented test set) [11]. This model processes mel frequency cepstral coefficients (MFCCs), which are computed on the audio signals. Their best LM approach consisted in the use of transformer from [23]. Transformers are autoregressive models (depending on the previous outputs of the models) using soft attention mechanisms. Soft attention consists in determining a glimpse $g$, which is a selection of characteristics from the input $x$ that help to filter non-useful information. Their best hybrid model got a WER of 5.7% for the test-other set and 2.7% for test-clean set.

Hybrid model is also used in emotion recognition and obtains SOTA on IEMOCAP using context-dependent domain adversarial neural network (DAAN) [16]. The IEMOCAP database [24] here was modified to obtain in a four-class emotion problem. Those emotions are angry, happy, neutral, and sad.

DAAN consists in using a lexical model (pretrained over one billion words) and audio features (such as MFCC and energy) that represent 6373 features for each input frame. They fuse these inputs using attention. Then, multiple GRU layers are used. Note that their approach use two tasks to be learned (emotion recognition and domain recognition). Such approach requires multiple labels, which IEMOCAP database provides. Doing so, they achieved an accuracy of 82.7% over all the emotions.

### 2.2 End-to-end systems

In end-to-end approaches, the goal is to determine a model $f$ that can do the mapping:

$$\hat{\mathbf{y}} = f(\mathbf{x}). \tag{2}$$

It will be trained to directly map $\mathbf{x}$ to $\hat{\mathbf{y}}$ (which can be different from the real sequence $\mathbf{y}$) with a single learnable function. Only supervised methods can function end-to-end to solve the speech tasks we are focused on.

In ASR systems, Kim et al. got SOTA on LibriSpeech test-clean official set [12]. Compared to [11], they used

vocal tract length perturbation as the input of their end-to-end model. The model is based on the encoder-decoder architecture using stacked long short-term memory (LSTM) for the encoder and LSTM combined with soft attention for the decoder [12]. They obtained a WER of 2.44% on test-clean set and 8.29% on test-other set. Those results are close to [11] (best hybrid model results) and show that end-to-end approaches are competitive compared to multi-model approaches.

In emotion recognition, Lian et al. got SOTA results with a the same modified version of the IEMOCAP database [24] as other methods mentioned for [16]. They used an end-to-end multi-task system with only supervised tasks: gender identification and emotion identification [15]. The resulting model achieved an overall accuracy for the emotion task (which is the main target) of 81.6% and an average accuracy of each emotion category of 82.8%. Using such an approach allowed them to achieve balanced results from unbalanced data. Furthermore, their results are similar to hybrid SOTA approach from [16]. Nevertheless, using only supervised tasks requires multiple ground truths for the targeted dataset.

In speaker identification, autoSpeech [17] architecture is SOTA over the VoxCeleb1 dataset [25]. Their approach consists in an algorithm that automatically search the best convolutional neural network architecture to solve the task. With their approach, they obtained an accuracy of 87.66%.

## 3 Techniques requiring fewer data

Some techniques require fewer data than the techniques of the previous section. In this section, we will enumerate the principal ways to leverage (to our best knowledge) the lack of large datasets such as impaired speech. We will not discuss semi-supervised techniques that use a large amount of unsupervised data.

### 3.1 Data augmentation

The first way to leverage the lack of data is to artificially augment the number of samples. To do so, the classic approach consists in adding noise or deformation, as in [26]. The authors obtain near SOTA results on LibriSpeech (1000 hours from [4]) with an end-to-end model. Nevertheless, they obtain SOTA results on SwitchBoard (300 h from [27]) with a WER of 6.8% on Switchboard and 14.1% on the CallHome portion using shallow fusion and their data augmentation. But these are handcrafted augmentations, and some of them require additional audio (such as adding noise).

Some other approaches use generative models to have new samples, as in [28, 29]. Chatziagapi et al. used conditional generative adversarial networks (GANs) to generate new samples [28]. With conditional GANs, we can control the mode of the generated samples [30]. By doing so, they balanced their initial dataset and obtained better results. Jiao et al. used deep convolutional GANs to generate dysarthric speech and improved their results [29].

### 3.2 Domain transposition

Another way to leverage the lack of data is to use data domain transposition. The idea consists of mapping speech features from one domain (such as spectrogram containing speech and noises) to another domain to reduce the data complexity (such as spectrogram with only speech). In machine learning, data complexity is defined by the size of the model needed to replicate the data, the size of the shortest encoding possible (that let the reconstruction of the initial data), and the error rate of the best model possible given a task [31]. Here are some recent examples on speech:

- Wang et al. used GAN to dereverberate speech signals [32]. In their work, the generator is used as a mapping function for converting reverberated signals into dereverberated speech signals.
- Chen et al. performed vocal conversion using GAN with a controller mapping impaired speech to a representation space $z$ [33]. $z$ is the input of the generator that is used as a mapping function to have unimpaired speech signals.
- Zhao et al. used Cycle GAN (framework designed for domain transfer) as an audio enhancer [34]. Their resulting model is SOTA on the CHiME-4 dataset.

### 3.3 Models requiring fewer parameters

Having fewer data means that overfitting can occur if neural network models require too many parameters. This is why some experimental techniques tried models requiring fewer parameters. Here, we highlight some recent techniques that we find interesting:

- The use of SincNet layers, from [35], to replace classic 1D convolutions over raw audio. Here, instead of requiring *window_size* parameters (with *window_size* being the window size of the 1D convolution) per filter, we only need two parameters per filter for every window size. These two parameters indirectly represent the values of the bandwidth at high and low energy.
- The use of LightGRU (LiGRU), from [36], based on the gated recurrent unit (GRU) framework. LiGRU is a simplification of the GRU framework given some assumptions concerning the speech signal. They removed the reset gate of the GRU and used the ReLU activation function (combined with Batch Nor-

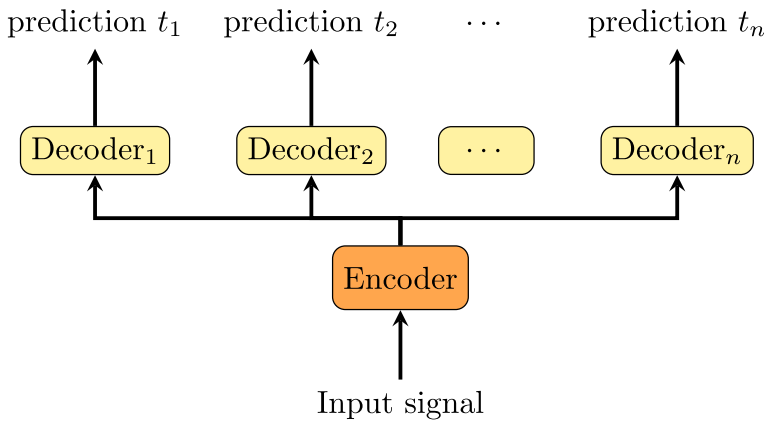


**Fig. 1** Illustration of multi-task architecture. The output of the encoder is given to each decoder to produce the prediction for each $t_i$ task

malization [37]) instead of the tanh activation function.

- The use of quaternion neural networks, from [38], for speech processing. The quaternion formulation allows 4 dimensions to be fused into one inducing a drastic reduction of required parameters in their experiments (almost 4 times fewer).

### 3.4 Multi-task approach

Multi-task models can be viewed as an extension of the Encoder-Decoder architecture where you have a decoder per task with a shared encoder (as in Fig. 1). These tasks are then trained conjointly with classic feed-forward algorithms. The goal of multi-task learning is to obtain an encoder that outputs sufficient information for each task. It can thus potentially improve the performance of each task compared to mono-task architectures. It is a way to have a more representative encoder given the same amount of data.

Pascual et al. used a combination of self-supervised tasks to tackle the lack of ground truth and used the resulting encoder for transfer learning [39]. They recently improved this work in [8] where they use more tasks, a recurrent unit on top of the encoder and denoising mechanisms using multiple data augmentation techniques on their system.

### 3.5 Transfer learning

Transfer learning techniques consist in using a pretrained model and using it as a feature extractor or using its parameters as initializers (that can be fine-tuned) to solve a related problem/task.

Contrastive predictive coding (CPC from [40]) is a framework for training self-supervised audio representation using a 2-level architecture combined with a self-supervised loss. The authors achieved better results by transferring the obtained parameters onto a speaker identification task and a phoneme classification task (on the LibriSpeech dataset) than with the use of MFCC features.

Some binary tasks of the multi-task model from Pascual et al. use predictive coding like in CPC. The latter developed an unsupervised multi-task model to obtain better encoders for transfer learning [39]. They applied it on multiple tasks and obtained acceptable results on speaker identification (using VTCK [41]), emotion recognition (using INTERFACE [42]), and ASR (using TIMIT [43]).

Nowadays, CPC approaches such as Wav2Vec [44] (last version Wav2Vec2.0 [9]) and HuBERT [10] represent SOTA self-supervised techniques. Such techniques are able to pretrain models over 60k hours of unlabeled speech such as libri-light [6]. Wav2Vec and Wav2Vec2.0 consist in learning representations from the waveform using the CPC framework. Wav2Vec2.0 combines convolutional layers to process the waveform and obtain a low latent representation, which is fed to transformer layers to obtain a contextual representation. HuBERT is using a BERT encoder (which is a transformer) inside the CPC framework and penalizes the loss function using the BERT loss [45]. These approaches have been compared in [7] benchmarks (among others). For speaker identification, the best model obtained an accuracy of 90.33% over the VoxCeleb test set [46]. Then, for emotion recognition, the best model obtained an accuracy of 67.62% over the IEMOCAP [47] test set. And for ASR, the best model obtained a word error rate of 2.94% over librispeech [4] test-clean. A more detailed overview (with a complete benchmark across several speech tasks) of self-supervised approaches is available

in [7]. Note that this benchmark does not include pathological speech data or related tasks.

The benefits of pre-trained networks for transfer learning decrease as the target task diverges from the original task of the pre-trained network [48]. To tackle this, Van den Oord et al. attempted to have generic tasks with their unsupervised approach, and they obtained promising results [40]. The benefits of transfer learning also decrease when the dissimilarity between the datasets increases [48]. This problem can discourage the use of transfer learning for some pathological speech. However, dysarthric and accented speech seem similar to speech in the LibriSpeech dataset, according to [49]. Shor et al. successfully used transfer learning to improve their results with a 36.7-h dataset.

Nevertheless, Mustapha et al. showed that the acoustic characteristics of unimpaired and impaired speech are very different [22]. Where few data are available, fine-tuning generic representations from unimpaired speech to impaired speech can be critical. Furthermore, learning from scratch could be a hard task. This is why looking into few-shot techniques could be helpful as a replacement or combined with generic pre-training.

## 4 Few-shot learning and speech

In the previous sections, we reviewed models that require a large amount of data to learn directly a task or to train more generic representations (to allow tackling some tasks with fewer data). But there is not always enough data to train a model from generic representations (by using them or adapting them); this is the case for pathological speech [19]. Google is trying to acquire more data of that nature[1]. But acquiring such data can be expensive and time consuming. Mustafa et al. recommend the use of adaptive techniques to tackle the limited amount of data problem in such case [22]. We think few-shot techniques can be another solution to this problem. Nevertheless, some non-common tasks such as pathological speech or dialect identification with few examples are still hard to train with SOTA techniques based on large speech datasets. This is why we investigate the following few-shot techniques and see the adaptations required for using them on speech datasets.

### 4.1 Few-shot notations

Let us consider a distribution $P$ from which we draw independent identically distributed (*iid*) episodes ($\mathcal{E}$ or datasets). $\mathcal{E}$ is composed of a support set $\mathcal{S}$, unlabeled data $\bar{\mathbf{x}}$ and a query set $\mathcal{Q}$. The support set corresponds to the supervised samples to which the model has access:

$$\mathcal{S} = \{(x_1, y_1), \dots (x_s, y_s)\}, \tag{3}$$

with $x_i$ being samples and $y_i$ being the corresponding labels, such as $y_i \in \{1,2,\dots,K\}$ and $K$ being the number of classes appearing in $P$. The query set is composed of samples to classify $\hat{\mathbf{x}}$ with $\hat{\mathbf{y}}$ being the corresponding ground truth.

To summarize, episodes drawn from $P$ have the following form:

$$\begin{aligned} \mathcal{E} = \{&\mathcal{S} = \{(x_1, y_1), \dots, (x_s, y_s)\}, \\ &\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_r), \\ &\mathcal{Q} = \{(\hat{x}_1, \hat{y}_1), \dots, (\hat{x}_t, \hat{y}_t)\} \end{aligned} \tag{4}$$

with $s$, $r$, and $t$ fixed values that respectively represent the number of supervised samples for the support set, the number of unsupervised samples, and the number of supervised samples for the query set.

In this survey, we will focus on few-shot learning techniques where $r=0$, $t \geq 1$ and $s=kn$, with $n$ being the number of times each label appears for the support set and $k$ the number of classes selected from $P$, such as $k \leq K$. Hence, we have an $n$-shot with $k$-ways (or classes) for each episode. One-shot learning is just a special case of few-shot learning where $n=1$. In some few-shot frameworks, we only sample one episode from $P$ and it represents our task.

### 4.2 Few-shot learning techniques

In this section, we will review frameworks that impacted the few-shot learning field in image processing, frameworks with a formulation that seems suitable for speech processing, and frameworks already successfully used by the speech community.

#### *4.2.1 Siamese technique*

Siamese neural networks are designed to be used per episode [50]. They involve measuring the distance between two samples and judging whether or not they are similar. Hence, a siamese network uses the samples from the support set $\mathcal{S}$ as references for each class. It is then trained using all the combinations of samples from $\mathcal{S} \bigcup \mathcal{Q}$ which provides much more training than having only $s+t$ samples in classical feedforward frameworks. Siamese Networks take two samples ($x_1$ and $x_2$) as input and compute a distance between them, as follows:

$$\phi(x_1, x_2) = \sigma\left(\sum \boldsymbol{\alpha} |Enc(x_1) - Enc(x_2)|\right), \tag{5}$$

with *Enc* being a DNN encoder that represents the signal input, $\sigma$ being the sigmoid function, $\boldsymbol{\alpha}$ learnable

[1] https://blog.google/outreach-initiatives/accessibility/impaired-speech-recognition/

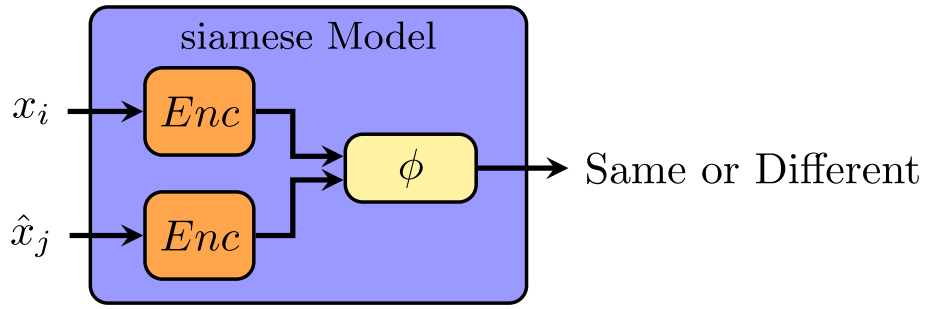


**Fig. 2** Example of comparison between a reference ($x_i$) and a new example ($\hat{x}_j$) from the query set, where *Enc* is the same network applied to both $x_i$ and $\hat{x}_j$. The model outputs the distance between the classes $x_i$ and $\hat{x}_j$

parameters that weight the importance of each component of the encoder, and $x_1$ and $x_2$ sampled from either the support set or the queries set.

To define the class of a new sample from $\mathcal{Q}$ or any new data, we must compute the distance between each reference from $\mathcal{S}$ and the new sample. An example of comparison between a reference and a new example is shown in Fig. 2. The class of the reference with the lowest distance then becomes the prediction of the model. To train such a model, [50] used this loss function:

$$\mathcal{L} = \mathbb{E}_{y(x_i)=y(\tilde{x}_j)} \log(\phi(x_i, \tilde{x}_j)) + \mathbb{E}_{y(x_i)\neq y(\tilde{x}_j)} \log(1 - \phi(x_i, \tilde{x}_j))$$

with $\tilde{\mathbf{x}} = [x_1, \ldots, x_s, \hat{x}_1, \ldots, \hat{x}_t]$ from $\mathcal{S}$ and $\mathcal{Q}$. $y(x)$ is a function that returns the label corresponding to the example $x$. The last layer of $\phi$ should be a softmax function.

Eloff et al. used a modified version of this framework for multimodal learning with the modalities being speech and image signals [51], but to our knowledge, there is no study yet concerning just speech processing. The speech signals used consist of 11-digit numbers (zero to nine and the "oh" used in phone numbers) with the corresponding 10 images ("oh" and zero give the same images). The problem is to associate speech signals with the corresponding images. In their experiment, the model shows some invariance to speakers (accuracy of 70.12% $\pm$0.68) using only a one-shot configuration, which is a promising result.

Siamese neural networks are not very suitable when the number of classes $K$ or the number of shots $q$ become too high. It increases the number of references to be compared and the computation time to forward the model. The primary problem concerns training the model. Once the model has been trained, we can reduce this effect by pre-calculating all encodings of the examples of the support set. This also dramatically increases the number of combinations for the training phase, which can be viewed as a positive point. This framework does not seem appropriate for end-to-end ASR with large vocabularies, such as in English (around 470,000 words), though it may be sufficient for languages such as Esperanto (around 16,780 words). The other way to use such a framework in ASR systems is to use it in hybrid models as an acoustic model, where we can train it on every phoneme (for example 44 phonemes/sounds in English) or more refined sound units.

The siamese framework seems interesting for tasks such as speaker identification, as a new speaker can be added without retraining the model (supposing the model had generalized) or changing the architecture of the model. We only have to add at least one sample of the new speaker to the references. Furthermore, the siamese formulation seems well adapted for speaker verification. We only need to replace the pair (**x**,*speaker_id*) by the pair ($\mathbf{x}, \mathcal{S}_{top5}$), where $\mathcal{S}_{top5}$ is a support set composed of signals from the top 5 predictions of the identification sub-task.

Nevertheless, this framework will be of limited use if the number of speakers to identify become too high. Even so, it is possible to use such techniques in an end-to-end ASR system when the vocabulary is limited, such as in the experiment described in [51]. Also, this framework was used in emotion recognition [52]. In their experiments, they used their approach over the IEMOCAP [47] using a 3-way task (which is different from all other papers reviewed in this work that use 4 classes). Nevertheless, they managed to obtain an unweighted average recall of 67.4% using a 10-shot configuration, which is an encouraging result.

#### *4.2.2 Matching network*

The matching networks system described in [53] is a few-shot framework designed to be trained with a set of multiple episodes (with typically 5-ways to 25-ways), which consists of a single model $\phi$. This model evaluates

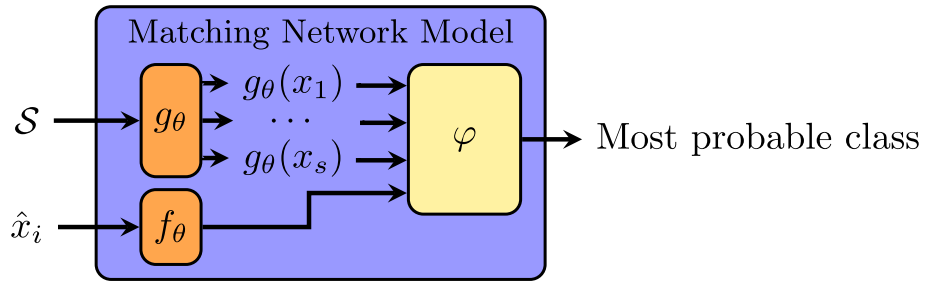


**Fig. 3** Illustration of the matching network model to predict the class of a new example $\hat{x}_i$

new examples given the support set $\mathcal{S}$ as in the siamese framework:

$$\varphi(\hat{x}, \mathcal{S}) :\rightarrow \hat{y}. \tag{6}$$

In matching learning, $\phi$ is as follows:

$$\varphi(\hat{x}, \mathcal{S}) = \sum_{(x_i, y_i) \in \mathcal{S}} a(\hat{x}, x_i) y_i, \tag{7}$$

with, $a$ being the attention kernel.
In [53], this attention kernel is as follows:

$$a(\hat{x}, x_i) = \text{softmax}(c(f(\hat{x}), g(x_i))), \tag{8}$$

where $c$ is the cosine distance, and $f$ and $g$ are embedding functions.

Vinyals et al. used a recurrent architecture to modulate the representation of $f$ using the support set $S$ [53]. The goal is to have $f$ following the same type of representation as $g$. To do this, the $g$ function is as follows:

$$g(x_i) = \overrightarrow{h_i} + \overleftarrow{h_i} + g'(x_i), \tag{9}$$

where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ represent a bi-LSTM output over $g'(x_i)$, which is a DNN.

The $f$ function is as follows:

$$f(\hat{x}) = attLSTM(f'(\hat{x}), g(S), m), \tag{10}$$

with *attLSTM* being an LSTM requiring a fixed number of recurrences (here $m$) and $g(S)$ representing the application of $g$ to each $x_i$ from the $S$ set. $f$ is a DNN with the same architecture as $g'$, but not necessarily sharing the parameter values.

Training this framework therefore consists in the maximization of the log likelihood of $\phi$ given the parameters of $g$ and $f$.

Figure 3 illustrates forward time of the matching network model. For forward time on new samples, $g(\mathcal{S})$ can be pre-calculated to gain computation time. Nevertheless, matching networks have the same disadvantages as siamese networks when $q$ and/or $K$ become too high. Furthermore, adding new classes to a trained matching network model is not as easy as for siamese network models. As this requires retraining the matching network model to add an element to the support set. Despite these disadvantages, matching learning showed better results than the siamese framework on image datasets [53]. This is why it should be investigated in speech processing to see if it is still the case.

#### 4.2.3 Prototypical networks

Prototypical networks [54] are designed to work with multiple episodes. In the prototypical framework, the model $\phi$ makes its predictions given the support set $\mathcal{S}$ of an episode such as the previously seen frameworks. This framework uses training episodes as mini-batches to obtain the final model. This model is formulated as follows:

$$\varphi(\hat{x}, S) = softmax_k(-d(f(\hat{x}), \mathbf{c}_k)), \tag{11}$$

where $\mathbf{c}_k$ is the prototype of the class $k$, $d$ being a Bregman divergence (for their useful properties in optimization, see [54] for more details), which also has the following property: $\mathbf{R}^n \times \mathbf{R}^n \rightarrow [0, +\inf[$.

Snell et al. used the Euclidean distance for $d$ instead of the cosine distance used in meta learning and matching learning papers [54]. As a result, they obtain better results in their experiments. Next, they go further by reducing the Euclidean distance to a linear function.

In the prototypical framework, there is only one prototype for each class $k$ as illustrated in Fig. 4. It is computed as follows:

$$\mathbf{c}_k = \frac{1}{|\mathcal{S}_k|} \sum_{(x_i, y_i) \in \mathcal{S}_k} f(x_i), \tag{12}$$

with $f$ being a mapping function such as $\mathbb{R}^D \rightarrow \mathbb{R}^M$ and $\mathcal{S}_k$ being the samples with $k$ of the support set.

Prototypical networks require only one comparison per class and not $q$ per class for $q$-shot learning as in siamese

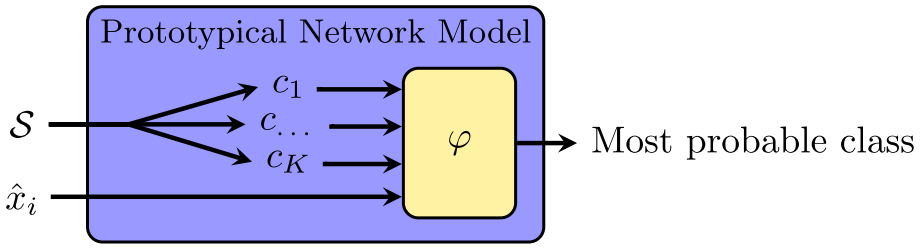


**Fig. 4** Illustration of the prototypical network model to predict class of a new example $\hat{x}_i$

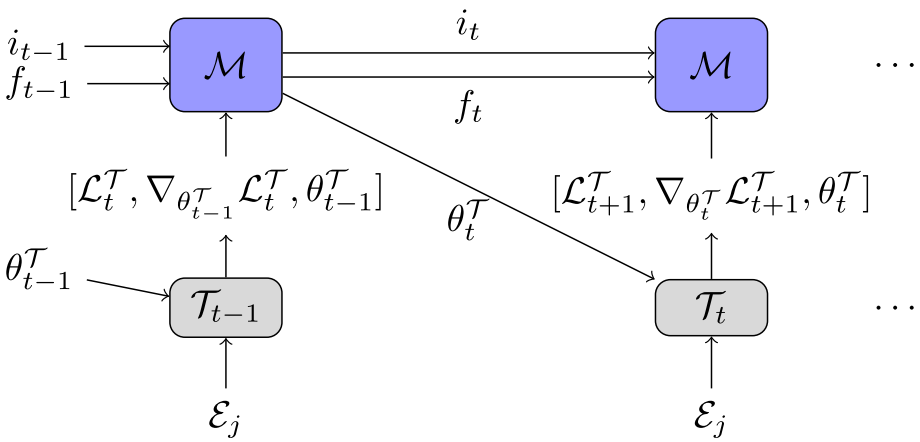


**Fig. 5** Illustration of Meta-Learning for training with episode $\mathcal{E}_j$ at step $t$. Here the Meta model $\mathcal{M}$ processes the different training steps of the trainee $\mathcal{T}$ as a sequence

and matching learning networks. That is why this framework is less subject to the high computation problem for prediction of new samples, as it is only influenced by high $K$. It will certainly be insufficient for end-to-end ASR systems on the English language due to the large vocabulary issues described in Section 4.2.1, but it is a step towards it.

In speaker recognition, prototypical networks were used over a portion of Voxceleb1 [25] by [55]. They obtained under 20-ways and using 5-shot an accuracy of 72.77 which are promising results.

#### *4.2.4 Meta-learning*

Meta-learning systems [56] are designed to be trained over multiple episodes (also called datasets). In this framework, a trainee model ($\mathcal{T}$) with parameters $\theta^{\mathcal{T}}$ trained from the start of every episode, usually has a classic DNN architecture. The support set and the query set in the episodes are considered as the training set and the test set for the trainee model.

Along with this trainee model, a second model is trained: the meta model ($\mathcal{M}$) with $\theta^{\mathcal{M}}$ parameters. This meta model is the key of meta learning, it consists in monitoring the trainee model by updating $\theta^{\mathcal{T}}$ parameters. To train this meta model, Ravi et al. suggests sampling *iid* episodes from $P$ to form the meta-dataset ($\mathcal{D}$) [56]. This meta-dataset is composed of a training set ($\mathcal{D}_{train}$), a validation set ($\mathcal{D}_{valid}$), and a testing set ($\mathcal{D}_{test}$).

While the trainee model is training on an episode $\mathcal{E}_j$, the meta model is used to update its parameters:

$$\theta_t^{\mathcal{T}_j} = \mathcal{M}(\theta_{t-1}^{\mathcal{T}_j}, \mathcal{L}^{\mathcal{T}_j}, \nabla_{\theta_{t-1}^{\mathcal{T}_j}} \mathcal{L}^{\mathcal{T}_j}), \tag{13}$$

with $\mathcal{L}^{\mathcal{T}_j}$ being the loss function of the trainee model learned with the episode $\mathcal{E}_j$ and $\theta_{t-1}^{\mathcal{T}_j}$ being the parameters of the trainee model at step $t-1$. Also, $\mathcal{M}$ has to guess initial weights of the trainee models at step $t=0$ ($\theta_0^{\mathcal{T}_j}$).

The learning curve (loss) of the trainee model with $\mathcal{E}_j$ is viewed in [56] as a sequence that can be the input of the meta model $\mathcal{M}$. For simplicity, we will use the notation of $\mathcal{T}$ instead of $\mathcal{T}_j$ for the next few paragraphs. Figure 5 illustrates the learning steps of the trainee using the meta model.

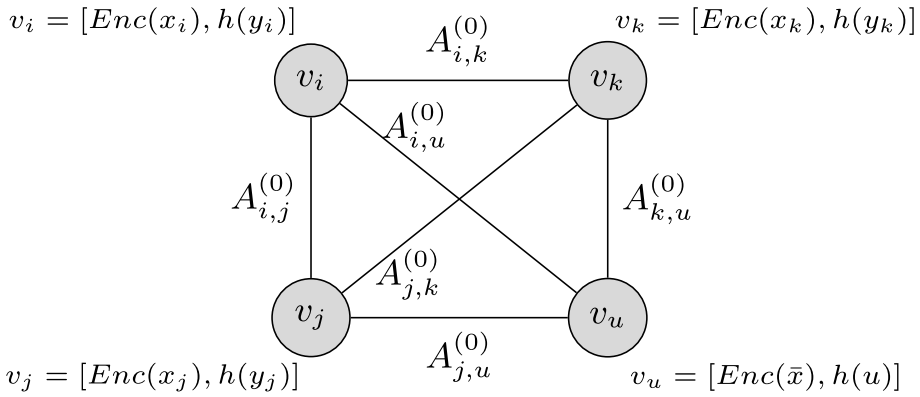


**Fig. 6** Illustration of the input of the first layer (here a graph convolution) of a GNN. Here, we have three samples (represented by vertices $v_i$, $v_j$ and $v_k$) in the support set and one query (represented by the vertex $v_u$)

*4.2.4.1 Trainee parameters update* Ravi and Larochelle identify the learning process of $\mathcal{T}$ using classic feedforward update on the episode $E_j$ to be similar to the $c_t$ update gate of the LSTM framework [56]. In the meta learning framework, $c_t$ is used as the $\theta_t^{\mathcal{T}}$ estimator, as follows:

$$\theta_t^{\mathcal{T}} = f_t \odot \theta_{t-1}^{\mathcal{T}} + i_t \odot \tilde{\theta_t^{\mathcal{T}}}, \tag{14}$$

with $\tilde{\theta_t^{\mathcal{T}}} = -\alpha_t \nabla_{\theta_{t-1}^{\mathcal{T}}} \mathcal{L}_t^{\mathcal{T}}$ being the update term of the parameters $\theta_{t-1}^{\mathcal{T}}$, $f_t$ being the forget gate, and $i_t$ the update gate.

*4.2.4.2 Parameters of the meta model* Both $i_t$ and $f_t$ are part of the meta learner. In the meta-learning framework, the update gate is formulated as follows:

$$i_t = \sigma(\mathbf{W}_I.[\nabla_{\theta_{t-1}^{\mathcal{T}}} \mathcal{L}_t^{\mathcal{T}}, \mathcal{L}_t^{\mathcal{T}}, \theta_{t-1}^{\mathcal{T}}, i_{t-1}] + \mathbf{b}_I), \tag{15}$$

with $\mathbf{W}_I$ and $\mathbf{b}_I$ being parameters of $\mathcal{M}$. The update gate is used to control the update term in equation 14, like the learning rate in the classic feedforward approach.

Next, the forget gate in the meta-learning framework is formulated as follows:

$$f_t = \sigma(\mathbf{W}_F.[\nabla_{\theta_{t-1}^{\mathcal{T}}} \mathcal{L}_t^{\mathcal{T}}, \mathcal{L}_t^{\mathcal{T}}, \theta_{t-1}^{\mathcal{T}}, f_{t-1}] + \mathbf{b}_F), \tag{16}$$

with $\mathbf{W}_F$ and $\mathbf{b}_F$ parameters of $\mathcal{M}$.

This gate is here to decide whether the training of the trainee should restart or not. This can be useful to avoid the problem of a sub-optimal local minimum. Note that this gate is not present in classic feedforward approaches (where this gate is equal to one).

The trainee model ($\mathcal{T}$) of this framework can be any kind of model, such as a siamese neural network. It can therefore have the advantages of this framework. It can also avoid the disadvantages of the siamese neural network, as it can use any other framework (usually classic DNN). This framework is interesting to training efficient models for speech processing (in terms of learning speed) when we have multiple ASR tasks with different vocabularies. For example, suppose we have the following kinds of speech episodes: dialing numbers, commands to a robot A, and commands to a robot B. The model can initialize good filters for the first layers (as this still involves speech processing). Another example could be training acoustic models for multiple languages (with each episode corresponding to a language).

#### *4.2.5 Graph neural network*

Graph neural networks (GNNs) are used by Garcia and Bruna to introduce their few-shot framework [57]. This framework is designed to be used with multiple episodes, they called tasks. In this framework, one model is used over a complete graph $G$: $G$=($V$,$E$) and every node corresponds to an example. For few-shot learning, a GNN consists in applying graph convolution layers over the graph $G$.

Initial vertices to guess the ground truth of a query $\tilde{x_i}$ from the query set $\mathcal{Q}$ are constructed as follows:

$$V^{(0)} = \begin{array}{l}((Enc(x_1), h(y_1)), \ldots, (Enc(x_s), h(y_s)),\\ (Enc(\bar{x_1}), u), \ldots, (Enc(\bar{x_r}), u)\\ (Enc(\tilde{x_i}), u))\end{array}, \tag{17}$$

where *Enc* is an embedding extraction function (a neural network or any classic feature extraction technique), $h$ the one-hot encoding function, and $u$=$K^{-1}\mathbf{1}_K$ a uniform distribution for examples with unknown labels (the unsupervised ones from $\bar{\mathbf{x}}$ and/or from the query set $\mathcal{Q}$).

The vertices at each layer $l$ (with 0 being the initial vertices) will henceforth be denoted:

$$V^{(l)} = (v_1, \ldots, v_n), \tag{18}$$

where $n{=}s{+}r{+}1$ and $V^{(l)} \in \mathbb{R}^{n*d_l}$.

Every layer (with an illustration of a layer in Fig. 6) in a GNN is computed as follows:

$$V^{(l+1)} = Gc(V^{(l)}, A^{(l)}), \tag{19}$$

with $A^{(l)}$ being the adjacency operators constructed from $V^{(l)}$ and $Gc$ being the graph convolution.

*4.2.5.1 Construction of the adjacency operators* The adjacency operator uses a set:

$$A^{(l)} = \{\tilde{A}^{(l)}, \mathbf{1}\}, \tag{20}$$

with $\tilde{A}^{(l)}$ being the adjacency matrix of $V^{(l)}$.

For every $(i,j){\in}E$ (remember that we have complete graphs), we compute the values of the adjacency matrix as follows:

$$\tilde{A}^{(l)}_{i,j} = \phi(v_i^{(l)}, v_j^{(l)}), \tag{21}$$

where:

$$\phi(v_i^{(l)}, v_j^{(l)}) = f(|v_i^{(l)} - v_j^{(l)}|), \tag{22}$$

with $f$ being a multi-layer perceptron with its parameters denoted $\theta_f$. $\tilde{A}^{(l)}$ is then normalized using the softmax function over each line.

*4.2.5.2 Graph convolution* The graph convolution requires the construction of the adjacency operators set and is computed as follows:

$$Gc(V^{(l)}, A^{(l)}) = \rho(\sum_{B \in A} BV^{(l)}\theta^{(k)}_{B,l}), \tag{23}$$

with $B$ being an adjacency operator from $A$, $\theta^{(k)}_{B,l} \in \mathbb{R}^{d_{l-1},d_l}$ learnable parameters and $\rho$ being a pointwise linearity (usually leaky ReLU).

*4.2.5.3 Training the model* The output of the resulting GNN model is a mapping of the vertices to a $K$-simplex that gives the probability of $\tilde{x_i}$ being in class $k$. V. Garcia and J. Bruna used the cross-entropy to train the model using all other samples in the query set $\mathcal{Q}$ [57]. Hence, the GNN few-shot framework consists in learning $\theta_f$ and $\theta_{1,l}...\theta_{card(A),l}$ parameters with all episodes.

*4.2.5.4 Few-shot GNN on audio* This framework was used by [58] on 5-way audio classification problems. The 5-way episodes are randomly selected from the initial dataset: AudioSet [59] for creating the 5-ways training episodes and TV program (from [60]) data to create the 5-ways test episodes.

Zhang et al. compare the use of per class (or intra-class) attention and global attention, which gave the best results [58]. They applied it for each layer. Their experiments were performed for 1-shot, 5-shots, and 10-shots with the respective accuracy of 69.4% ± 0.66, 78.3% ± 0.46, and 83.6% ± 0.98. Such results are an encouragement for the use of few-shot learning for speech signals. Nevertheless, this framework does not allow the use of multiple classes and shots per episode, which increase the number of nodes and thus the computations in forward time. Hence, it is not suitable for large vocabulary problems.

## 5 Preliminary results on phoneme recognition

Following this review of few-shot techniques, we implemented our first approach of a few-shot solution for phoneme recognition on TIMIT. We did not try word recognition due to the large vocabulary issues described in Section 4.2.1. To select our architecture, we reused the five first layers of the PyTorch-Kaldi model using MFCC [61]. We use this architecture as an encoder for the siamese framework and the prototypical framework, as they require the same type of architecture. To evaluate our results, we used the phone accuracy (where PER=1−phone accuracy) to match the usual metric used in few-shot learning. Note that this is only a first attempt; more complete experiments (including all frameworks, speaker recognition, and emotion recognition) will be done in our future work. Nevertheless, our initial results may help others to avoid some difficulties.

In our experiments, we encountered difficulties in getting these models to learn (the problems being similar for the siamese and prototypical networks). In our initial experiments, both architectures converged to a decrease in the cost function, but this resulted in a decrease in the phone accuracy score on the test set (which is the unwanted behavior). To solve this problem, we had to make the following changes:

1. When learning the model, we presented as many positive pairs as negative ones for each batch (specific to siamese networks)
2. We made sure to balance the classes present for each batch (siamese and prototypical networks).
3. We balanced the numbers of male and female speakers for the reference examples (siamese and prototypical networks).
4. We observed that reducing the number of parameters of the chosen architecture improved the results. The new architecture used is in Table 5. Thus, we go from an architecture of about 23 million parameters to about 5 million parameters. Therefore, we assume

**Table 5** Architecture used for siamese and prototypical networks

| Layer number | Layer type | Parameters |
|---|---|---|
| 0 | Input data | MFCC with a windowing of 25 ms and a 10 ms stride |
| 1 | Stacked bidirectional GRUs | 5 GRUs of 256 cells each |
| 2 | Dropout | Of 0.2 |
| 3 | Batch normalization | For each direction |
| 4 | Linear layer | 128 filters |

**Table 6** Results on the 39 phonemes of TIMIT with the siamese network

| #shots | Accuracy on test |
|---|---|
| 10 | 22.76 |
| 25 | 32.58 |
| 40 | 27.90 |

**Table 7** Results on the 39 phonemes of TIMIT with the prototypical network

| #shots | #queries | Accuracy on test |
|---|---|---|
| 10 | 15 | 39.76 |
| 15 | 15 | 37.82 |
| 10 | 20 | 41.16 |
| 20 | 15 | 41.12 |
| 15 | 20 | 41.33 |
| 20 | 20 | 41.38 |

that these methods do not allow learning from scratch models with a large number of parameters.

5 The choice of examples from different dialect regions[2] allows an increase in the scores but is not always possible depending on the available data. It represents an improvement of 5% in accuracy.

Thus, for the recognition of the 39 English phonemes of TIMIT, we obtained at best an accuracy of 32.58% for the siamese network and an accuracy of 41.38% for the prototypical network. These initial results are encouraging, especially considering that we used 1.1% of the data from the TIMIT training set (when we used 40 samples for training the model). To compare these two methods, we have made two compilation tables (Tables 6 and 7) where we used the accuracy as a metric. It should be noted that for the siamese network architecture, the computation time of our implementation was higher (about one day for an experiment) than our implementation of the prototypical networks (about 6h for an experiment). Even if our implementation can be improved, this difference is due to the nature of the frameworks. In siamese networks, each reference is compared to a new sample, while in prototypical only, the prototypes are compared to new samples. Note that our implementation can be improved to diminish the computational time for both techniques, but the difference between the two frameworks will remain. This difference in computational time also explains why the number of trials we attempted is less important for siamese networks. Considering these initial results, we consider that the prototypical architecture is more interesting since the computation times are lower and the results are better. Moreover, we notice that the prototypical network seems more stable if the number of supervised examples increases (compared to the siamese network).

Nevertheless, these results are just preliminary ones. We will include data augmentations in our next experiments, hoping it will increase the accuracy, as all reviewed few-shot techniques use data-augmentation.

## 6 Summary and future directions

In this survey, we investigated few-shot techniques for speech usage. In order to do so, we started with state-of-the-art speech processing systems. These systems require a large amount of data and are not suited for under-resourced speech problems. We also looked into techniques requiring fewer data using data augmentation, domain transposition, models requiring fewer parameters, the multi-task approach, and transfer learning. Nevertheless, these techniques are not always sufficient in a data-limited context, especially for pathological speech [19]. Next, we studied few-shot techniques and how well the different frameworks are adapted for classical speech tasks.

The main drawback of the reviewed techniques is the amount of computation required for large datasets (such as LibriSpeech from [4]) compared to SOTA models we reviewed in Section 2. Nevertheless, we considered some recent works already using few-shot techniques on speech with promising results. Such techniques seem useful for classical speech tasks on impaired speakers. Moreover, we think it can be useful for unconventional speech tasks such as measuring the intelligibility of a person (with impaired or unimpaired speakers) to help the

[2] In the TIMIT dataset, data come from eight dialect regions from the United States. The dialects are as follows: New England, Northern, North Midland, South Midland, Southern, New York City, Western and Army Brat (moved around). Note, the authors cannot ensure dialect boundaries for the two last ones.

re-education process (by identifying the problems faster). Acquiring a large amount of data is time consuming and laborious for some patients (with severe pathologies). We believe that few-shot techniques may help the community to tackle this problem.

Our initial results over the TIMIT dataset indicate that such an approach is adaptable to phoneme recognition. Indeed, we managed to obtain an accuracy of 41% using only 40 supervised samples per phoneme with the prototypical network over MFCC features. Furthermore, this result was obtained without requiring data augmentation, which should improve this first result (as all reviewed few-shot techniques use data augmentation). To better see the potential of such techniques, we will continue our work by establishing a benchmark for phoneme recognition, speaker recognition tasks and emotion recognition. Afterwards, we plan to use this technique with the best results on this benchmark as a base for teaching the concept of intelligibility.

#### Abbreviations

ASR: Automatic speech recognition; DNN: Deep neural network; GMM: Gaussian mixture model; GRU: Gated recurrent units; GNN: Graphical neural network; GAN: Generative adversarial network; HMM: Hidden Markov model; MFCC: Mel frequency cepstral coefficients; PER: Phone error rate; SOTA: State of the art; WER: Word error rate.

#### Acknowledgements

This work is part of the ANR-18-CE45-0008 RUGBI project funded by the French National Research Agency.

#### Authors' contributions

This work was mainly carried out by Vincent Roger, under the direction of Jérôme Farinas and Julien Pinquier. All authors read and approved the final manuscript.

#### Authors' information

Vincent Roger obtains his master's degree in computer science in 2015 at university Paul Sabatier Toulouse. He is a PhD student in computer science working on corpora with a limited amount of impaired speech.
Jérôme Farinas received a PhD (computer science specialty) in 2002 and proposed and rhythm model for automatic language identification. Since 2003, he is an assistant professor at the Toulouse III Paul Sabatier University where he works in the IRIT laboratory. He works more specifically in automatic speech processing, in the field of pathological speech, and in the framework of the ANR RUGBI project (looking for relevant linguistic units to improve the intelligibility measurement of speech production disorders).
Julien Pinquier received a PhD (computer science specialty) in 2004, related to audio indexing and structuring by search of primary components: speech, music, and keysounds. He received the HDR diploma of the University of Toulouse in 2014: this work was based on audio segmentation (speech, music, and environmental sounds) and audiovisual segmentation. Since 2005, he is an assistant professor at the Toulouse III Paul Sabatier University where he works in the IRIT laboratory. His objectives relate to the combination of the audio and the video and speech intelligibility/comprehensibility measurements.

#### Funding

Vincent Roger's doctorate is funded by Federal University of Toulouse and Occitanie Region, N°2018-1290 (ALDOCT N°500).

#### Availability of data and materials

TIMIT dataset is available here: https://www.kaggle.com/datasets/nltkdata/timitcorpus. The code used can be found there: https://github.com/vroger11/audio_loader.

### Declarations

#### Ethics approval and consent to participate

The authors approve and consent to participate.

#### Consent for publication

The authors consent for publication.

#### Competing interests

The authors declare that they have no competing interests.

Received: 19 November 2021 Accepted: 15 July 2022
Published online: 17 August 2022

#### References

1 P. Sahu, M. Dua, A. Kumar, in *Speech and Language Processing for Human-machine Communications*. Challenges and issues in adopting speech recognition, (2018), pp. 209–215. https://doi.org/10.1007/978-981-10-6626-9_23.
2 J. Barker, S. Watanabe, E. Vincent, J. Trmal, in *Interspeech 2018*. The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines, (2018), pp. 1561–1565. https://doi.org/10.21437/Interspeech.2018-1768.
3 F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, Y. Estève, in *Speech and Computer - 20th International Conference, vol. 11096*. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation, (2018), pp. 198–208. https://doi.org/10.1007/978-3-319-99579-3_21.
4 V. Panayotov, G. Chen, D. Povey, S. Khudanpur, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Librispeech: An ASR corpus based on public domain audio books (IEEESouth Brisbane, 2015), pp. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964.
5 J. S. Chung, A. Nagrani, A. Zisserman, in *Interspeech 2018*. VoxCeleb2: Deep Speaker Recognition, (2018), pp. 1086–1090. https://doi.org/10.21437/Interspeech.2018-1929.
6 J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P. -E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, et al, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Libri-light: A benchmark for asr with limited or no supervision (IEEE, Barcelona 2020), pp. 7669–7673.
7 S. -w. Yang, P. -H. Chi, Y. -S. Chuang, C. -I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. -T. Lin, et al., Superb: Speech processing universal performance benchmark. (Proc. Interspeech 2021), pp. 1194–1198. https://doi.org/10.21437/Interspeech.2021-1775.
8 M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, Y. Bengio, Multi-task self-supervised learning for Robust Speech Recognition. arXiv:2001.09239 [cs, eess] (2020). http://arxiv.org/abs/2001.09239.
9 A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst.**33:**, 12449–12460 (2020).
10 W. -N. Hsu, B. Bolte, Y. -H. H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process.**29:**, 3451–3460 (2021).
11 C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, H. Ney, in *Interspeech 2019*. RWTH ASR Systems for LibriSpeech: Hybrid vs Attention, (2019), pp. 231–235. https://doi.org/10.21437/Interspeech.2019-1780.
12 C. Kim, M. Shin, A. Garg, D. Gowda, in *Interspeech 2019*. Improved vocal tract length perturbationfor a state-of-the-art end-to-end speech recognition system, (2019), pp. 739–743. https://doi.org/10.21437/Interspeech.2019-3227.

13 Y. Zhang, J. Qin, D. S. Park, W. Han, C. -C. Chiu, R. Pang, Q. V. Le, Y. Wu, Pushing the limits of semi-supervised learning for automatic speech recognition. arXiv preprint arXiv:2010.10504 (2020). http://dblp.uni-trier.de/db/journals/corr/corr2010.html#abs-2010-10504.

14 Y. -A. Chung, Y. Zhang, W. Han, C. -C. Chiu, J. Qin, R. Pang, Y. Wu, W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244-250.

15 Y. Li, T. Zhao, T. Kawahara, in *Interspeech 2019*. Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning, (2019), pp. 2803–2807. https://doi.org/10.21437/Interspeech.2019-2594.

16 Z. Lian, J. Tao, B. Liu, J. Huang, Z. Yang, R. Li, in *INTERSPEECH*. Context-dependent domain adversarial neural network for multimodal emotion recognition, (2020), pp. 394–398.

17 S. Ding, T. Chen, X. Gong, W. Zha, Z. Wang, Autospeech: Neural architecture search for speaker recognition. (2020).

18 B. Deka, J. Chakraborty, A. Dey, S. Nath, P. Sarmah, S. R. Nirmala, S. Vijaya, in *2018 Oriental COCOSDA - International Conference on Speech Database and Assessments, Miyazaki, Japan, May 7-8, 2018*. Speech corpora of under resourced languages of north-east india, (2018), pp. 72–77. https://doi.org/10.1109/ICSDA.2018.8693038.

19 S.. Latif, J.. Qadir, A.. Qayyum, M.. Usama, S.. Younis, Speech technology for healthcare: opportunities, challenges, and state of the art. IEEE Rev. Biomed. Eng. **14**, 342–356 (2020)

20 L. Besacier, E. Barnard, A. Karpov, T. Schultz, Automatic speech recognition for under-resourced languages: a survey. Speech Comm.**56:**, 85–100 (2014). https://doi.org/10.1016/j.specom.2013.07.008.

21 M. Moore, H. Venkateswara, S. Panchanathan, in *Interspeech 2018*. Whistle-blowing ASRs: evaluating the need for more inclusive speech recognition systems, (2018), pp. 466–470. https://doi.org/10.21437/Interspeech.2018-2391.

22 M. B. Mustafa, S. S. Salim, N. Mohamed, B. Al-Qatab, C. E. Siong, Severity-based adaptation with limited data for ASR to aid dysarthric speakers. PLoS ONE. **9**(1), 86285 (2014). https://doi.org/10.1371/journal.pone.0086285.

23 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, in *Advances in Neural Information Processing Systems, vol. 30*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Attention is All you Need, (2017), pp. 5998–6008.

24 C. Busso, M. Bulut, C. -C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, S. S. Narayanan, Iemocap: Interactive emotional dyadic motion capture database. Lang. Resour. Eval.**42**(4), 335–359 (2008).

25 A. Nagrani, J. S. Chung, A. Zisserman, in *Interspeech 2017*. VoxCeleb: A large-scale speaker identification dataset, (2017), pp. 2616–2620. https://doi.org/10.21437/Interspeech.2017-950.

26 D. S. Park, W. Chan, Y. Zhang, C. -C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, SpecAugment: a simple data augmentation method for automatic speech recognition, (2019). https://doi.org/10.21437/Interspeech.2019-2680.

27 J. J. Godfrey, E. C. Holliman, J. McDaniel, in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1*. SWITCHBOARD: Telephone speech corpus for research and development, (1992), pp. 517–520.

28 A. Chatziagapi, G. Paraskevopoulos, D. Sgouropoulos, G. Pantazopoulos, M. Nikandrou, T. Giannakopoulos, A. Katsamanis, A. Potamianos, S. Narayanan, in *Interspeech 2019*. Data augmentation using GANs for speech emotion recognition, (2019), pp. 171–175. https://doi.org/10.21437/Interspeech.2019-2561.

29 Y. Jiao, M. Tu, V. Berisha, J. Liss, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Simulating dysarthric speech for training data augmentation in clinical speech applications, (2018). http://arxiv.org/abs/1804.10325.

30 M. Mirza, S. Osindero, Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014). https://doi.org/10.48550/arxiv.1411.1784

31 L. Li, Y. S. Abu-Mostafa, *Data complexity in machine learning* (California Institute of Technology, Pasadena, USA, 2006). https://resolver.caltech.edu/CaltechCSTR:2006.004.

32 K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, L. Xie, Investigating generative adversarial networks based speech dereverberation for robust speech recognition, (2018). https://doi.org/10.21437/Interspeech.2018-1780.

33 L. -W. Chen, H. -Y. Lee, Y. Tsao, in *Interspeech 2019*. Generative adversarial networks for unpaired voice transformation on impaired speech, (2019), pp. 719–723. https://doi.org/10.21437/Interspeech.2019-1265.

34 S. Zhao, C. Ni, R. Tong, B. Ma, in *Interspeech 2019*. Multi-Task multi-network joint-learning of deep residual networks and cycle-consistency generative adversarial networks for robust speech recognition, (2019), pp. 1238–1242. https://doi.org/10.21437/Interspeech.2019-2078.

35 M. Ravanelli, Y. Bengio, in *NIPS 2018 Workshop IRASL*. Interpretable convolutional filters with SincNet, (2018). http://arxiv.org/abs/1811.09725. Accessed 19 Nov 2021.

36 M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, Light gated recurrent units for speech recognition. IEEE Trans. Emerg. Top. Comput. Intell.**2**(2), 92–102 (2018). https://doi.org/10.1109/TETCI.2017.2762739.

37 S. Ioffe, C. Szegedy, in *International Conference on Machine Learning*. Batch normalization: accelerating deep network training by reducing internal covariate shift (PMLR, Montréal, 2015), pp. 448–456.

38 T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, R. De Mori, in *NeurIPS 2018 - IRASL*. Speech recognition with quaternion neural networks, (2018). http://arxiv.org/abs/1811.09678. Accessed 19 Nov 2021.

39 S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio, in *Interspeech 2019*. Learning problem-agnostic speech representations from multiple self-supervised tasks, (2019), pp. 161–165. https://doi.org/10.21437/Interspeech.2019-2605.

40 A Van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding. CoRR (2018). http://arxiv.org/abs/1807.03748. Accessed 19 Nov 2021.

41 J. Yamagishi, C. Veaux, K. MacDonald, et al., Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). University of Edinburgh. The Centre for Speech Technology Research (CSTR). (2019). https://doi.org/10.7488/ds/2645

42 V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte, A. Nogueiras, in *LREC*. Interface databases: design and collection of a multilingual emotional speech database, (2002).

43 J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon Tech. Rep. N. **93:**, 27403 (1993).

44 S. Schneider, A. Baevski, R. Collobert, M. Auli, wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862, (2019), pp. 3465-3469. https://doi.org/10.21437/Interspeech.2019-1873

45 J. Devlin, M. -W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018), pp 4171-4186. https://doi.org/10.18653/v1/N19-1423

46 A. Nagrani, J. S. Chung, W. Xie, A. Zisserman, Voxceleb: Large-scale speaker verification in the wild. Comput. Speech Lang.**60:**, 101027 (2020). https://doi.org/10.1016/j.csl.2019.101027.

47 C. Busso, M. Bulut, C. -C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, S. S. Narayanan, IEMOCAP: Interactive emotional dyadic motion capture database. Lang. Resour. Eval.**42**(4), 335–359 (2008). https://doi.org/10.1007/s10579-008-9076-6.

48 J. Yosinski, J. Clune, Y. Bengio, H. Lipson, in *Advances in Neural Information Processing Systems, vol. 27*, ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. How transferable are features in deep neural networks? (2014), pp. 3320–3328.

49 J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt, A. Hassidim, Y. Matias, in *Interspeech 2019*. Personalizing ASR for dysarthric and accented speech with limited data, (2019), pp. 784–788. https://doi.org/10.21437/Interspeech.2019-1427.

50 G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition. ICML Deep Learning Workshop, vol. 2, (2015). https://ieeexplore.ieee.org/document/9529076.

51 R. Eloff, H. A. Engelbrecht, H. Kamper, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Multimodal one-shot learning of speech and images, (2019), pp. 8623–8627. https://doi.org/10.1109/ICASSP.2019.8683587.

52 K. Feng, T. Chaspari, Few-shot learning in emotion recognition of spontaneous speech using a siamese neural network with adaptive sample pair formation. IEEE Trans. Affective Computing (2021).

53 O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, in *Advances in Neural Information Processing Systems, vol. 29*, ed. by D. D.

Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Matching networks for one shot learning, (2016), pp. 3630–3638.
54 J. Snell, K. Swersky, R. Zemel, in *Advances in Neural Information Processing Systems, vol. 30*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Prototypical networks for few-shot learning, (2017), pp. 4077–4087.
55 P. Anand, A. K. Singh, S. Srivastava, B. Lall, Few shot speaker recognition using deep neural networks. arXiv preprint arXiv:1904.08775 (2019).
56 S. Ravi, H. Larochelle, in *ICLR 2017*. Optimization as a model for few-shot learning, (2017), p. 11.
57 V. Garcia, J. Bruna, in *ICLR 2018*. Few-shot learning with graph neural networks, (2018), p. 13.
58 S. Zhang, Y. Qin, K. Sun, Y. Lin, in *Interspeech 2019*. Few-shot audio classification with attentional graph neural networks, (2019), pp. 3649–3653. https://doi.org/10.21437/Interspeech.2019-1532.
59 J.. F.. Gemmeke, D.. P.. W.. Ellis, D.. Freedman, A.. Jansen, W.. Lawrence, R.. C.. Moore, M.. Plakal, M.. Ritter, Audio Set: an ontology and human-labeled dataset for audio events, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. (IEEE, New Orleans, 2017), pp. 776–780 https://doi.org/10.1109/ICASSP.2017.7952261
60 S. Zhang, H. Jiang, S. Zhang, B. Xu, in *INTERSPEECH 2006 - ICSLP*. Fast SVM training based on the choice of effective samples for audio classification, (2006), p. 4.
61 M. Ravanelli, T. Parcollet, Y. Bengio, in *Proc. of ICASSP*. The pytorch-kaldi speech recognition toolkit, (2019).

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen journal and benefit from:**

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

**Submit your next manuscript at ► springeropen.com**