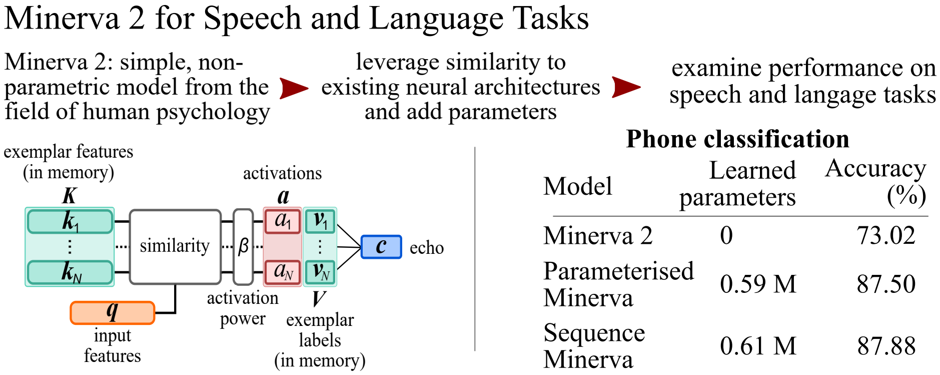
Graphical Abstract

Minerva 2 for speech and language tasks

Rhiannon Mogridge, Anton Ragni



Minerva 2 for Speech and Language Tasks

Minerva 2: simple, non-parametric model from the field of human psychology ➤ leverage similarity to existing neural architectures and add parameters ➤ examine performance on speech and langage tasks

Phone classification

Model	Learned parameters	Accuracy (%)
Minerva 2	0	73.02
Parameterised Minerva	0.59 M	87.50
Sequence Minerva	0.61 M	87.88

Highlights

Minerva 2 for speech and language tasks

Rhiannon Mogridge, Anton Ragni

- Minerva 2's similarity to the attention mechanism found in the transformer architecture is leveraged to create a sequence-based version of Minerva which shows promising performance on the TIMIT phone recognition task.
- Minerva 2's previously proposed echo-of-echoes process, an iterative inference technique, is shown to be in general ineffective, but by relaxing a single assumption, it becomes class-equivalent with deep equilibrium models.
- The effect of memory size on Minerva 2's performance is tested using three different experimental speech and language tasks. In general, performance improves with increasing exemplar set size, although with diminishing returns and higher computational overhead.

Minerva 2 for speech and language tasks

Rhiannon Mogridge[a], Anton Ragni[a]

[a]*The University of Sheffield, 210 Portobello St., Sheffield, S1 4DP, UK*

Abstract

Most artificial neural networks do not directly incorporate a memory of previous experiences, instead using training data to parameterise a model, and then discarding the training data prior to inference. While some recent models have included a memory, this has typically been added to an already highly parameterised model. An alternative option is to use a purely memory-based model, and then add parameters. This has been shown to work for Minerva 2, a simple, non-parametric, memory-based model which has been widely used in the field of human psychology. We revisit the use of Minerva 2 for speech and language tasks, drawing comparisons between Minerva 2 and other architectures, and showing that an iterative process that Minerva 2 uses for inference is a close relative of deep equilibrium models. We assess parameterised models based on Minerva 2, including a sequence model inspired by Minerva 2's similarity to the transformer architecture, which shows promising results.

Keywords: exemplars, Minerva 2, phone recognition, emotion classification, speech intelligibility

1. Introduction

Computational models based on theories of human cognition are widely used in machine learning, where neural networks currently provide state-of-the-art performance in a wide variety of tasks. Humans still outperform machines at many speech and language tasks, so there is clearly more that can be learned from human cognition. One aspect of cognition that has not been widely explored in machine learning is that of memory, and in particular, memories of specific experiences. Modern deep learning models frequently use previous examples for training, but these examples are rarely retained for inference. Such examples are referred to as *exemplars*, and models that make use of them are *exemplar models*. Exemplar models can be contrasted with *prototype models*, which use training data to parameterise a model. The training data is then discarded, and only the parameterised model is used for inference.

There are benefits to using exemplar models. They are usually interpretable, since the contribution of specific exemplars can be easily traced. They can also be used when data is scarce; theoretically, only a single exemplar for each class is required in order to perform classification, for example. Exemplar approaches are not typically scalable, however, so while they may be effective for small data sets, they are rarely used for large-scale data. This inability to scale leads to poor performance compared with modern, data-driven approaches.

The terms *prototype* and *exemplar* come from the field of human psychology. Experiments suggest that humans may not exclusively use either prototype or exemplar approaches. [1] suggest that human categorisation is largely rule-based, but with specific exceptions (exemplars) also being referred to. [2] propose a more balanced approach, with transitions between prototype and exemplar-based approaches. [3] concluded that humans use an exemplar-based approach when the objects to be classified are clearly distinct, but a prototype-based approach when they are easily confused, for example, classifying lines as “long” or “short” is more likely to be prototype-based, whereas classifying them as “red” or “yellow” is more likely to be exemplar-based. [4] found that the choice of exemplar or prototype-based approach might vary depending on the person. More recently, researchers have used techniques such as functional Magnetic Resonance Imaging (fMRI) to determine which approach is used. [5] found evidence for a largely exemplar-based approach; [6] found results consistent with a prototype approach. The literature is therefore mixed with regard to *when* exemplar and prototype approaches are used, but there is evidence that humans use both prototype and exemplar-based approaches at least some of the time. This is not reflected in automated speech and language tasks, in which data-driven, parametric, prototype-based deep learning approaches are currently overwhelmingly more popular.

While Artificial Neural Networks (ANNs) are usually entirely prototype-based, there are some exceptions. The use of memory, particularly for language modelling, has been explored with some success [7, 8, 9]. In such cases, the memory is typically added to an existing, highly-parameterised architecture. In previous work, we explored an alternative: create a hybrid model by taking an existing exemplar model, and parameterising it [10]. The memory model in question is Minerva 2 [11, 12], which originated in the field of human psychology, and has been used to test theories of human cognition [13, 14, 15], as well as being used previously for vowel classification [16] and limited vocabulary automatic speech recognition [17, 18]. More recently, new parameterised versions of Minerva 2, making use of modern machine learning techniques, were proposed and tested on a range of speech and language tasks, demonstrating that good feature representation is crucial to the performance of this type of model [10].

While exemplar and prototype models use different approaches, parallels can be drawn between Minerva 2 and existing architectures used in machine learning. Firstly, the core process within Minerva 2 bears a strong resemblance to the attention mechanism found in modern transformers [19, 10, 20], despite predating the term ‘attention’ by some years. Secondly, it has been demonstrated that, when using a fixed memory, Minerva 2 is a special case of Feed-Forward Neural Network (FFNN). Minerva 2’s use of ‘hidden nodes’ was first noted in 1990 [21], and an argument for a biologically plausible neural implementation of Minerva 2 is given in [15]. An ANN interpretation of Minerva 2 from a machine learning perspective is given in [10].

In this paper we explore the use of Minerva 2 for speech and language tasks, both theoretically and empirically, and provide recommendations for using it. Firstly, an iterative process, proposed to allow Minerva 2 to perform inference on undefined classes [12], and referred to here as ‘echo-of-echoes’, has been found to be ineffective for some practical applications [22, 16]. We examine it mathematically and with examples in §3.1. Secondly, we consider exemplar set size and activation power. Large datasets offer the opportunity to use large exemplar sets, but this comes with a corresponding computational overhead. Minerva’s performance with randomly sampled exemplar sets of increasing size has been previously explored on a single frame-based phone recognition task [10]. Here, the work is expanded to include two additional tasks, including a regression task, in §5.1. Thirdly, we leverage the similarities between Minerva 2 and transformers to propose a new sequence-based Minerva model in §3.4, which is compared with previously proposed Minerva-based models, other exemplar-based models, and a FFNN baseline in §5.2.

2. Minerva 2

Minerva 2 is a global memory model proposed by [11], created to test theories of human cognition. It has been widely used for comparison with human experiments [13, 23, 24, 25, 26, 27]. Minerva 2 is an exemplar-based model which uses previous experiences, or exemplars, to label new experiences. The mechanism by which the new label is produced, shown in Figure 1, is a form of attention, although Minerva 2 predates the term "attention" by several decades.

2.1. Generating an echo

Let $\boldsymbol{q}$ be an input query, representing a new experience to be labelled, shown in orange to the bottom left of Figure 1. Each exemplar is represented by a feature vector and a label vector. Let $\boldsymbol{K} = \begin{bmatrix}\boldsymbol{k}_1 & \dots & \boldsymbol{k}_N\end{bmatrix}$ be a matrix of column vectors, each of which represents an exemplar feature vector; in Figure 1, this is shown in green, to the left. Let $\boldsymbol{V} = \begin{bmatrix}\boldsymbol{v}_1 & \dots & \boldsymbol{v}_N\end{bmatrix}$ be matrix of column vectors representing the exemplar labels; in Figure 1, this is shown in green to the mid-right. The vector $\boldsymbol{q}$ and the matrices $\boldsymbol{K}$ and $\boldsymbol{V}$ are similar to the query, keys and values used in attention.

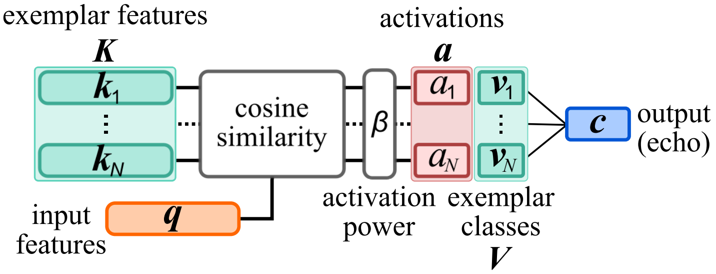


Figure 1: Minerva 2.

The elements of the query and exemplar vectors are restricted to ±1. Minerva 2 also permits values of 0, where information is either irrelevant or not available, but for this work we will assume that all the information is available. In its original form, the exemplar feature and label elements are in a single vector; for convenience they have been split into separate vectors here. This is a notation change only, and does not affect the underlying model. To label a new query, $\boldsymbol{q}$, it is compared against each of the stored exemplar feature vectors, using dot product similarity,

$$\boldsymbol{s} = \frac{1}{F}\boldsymbol{K}^\top\boldsymbol{q}, \tag{1}$$

where F is the dimension of the query and exemplar feature vectors, and provides scaling so that each element of $\boldsymbol{s}$ falls in the range $[-1, 1]$. The activation, $\boldsymbol{a}$, of the exemplars is a positively-accelerated function of the similarities, with [11] suggesting raising each element in the vector to an odd power, $a_i = s_i^{\beta}$. This differs from conventional attention, in which a softmax function is used. The activation function helps prevent exemplars that are very similar to the probe from being drowned out by large numbers of exemplars with only limited similarity. In principle, any positively accelerated function that preserves the sign could be used. As with attention, the new label, $\boldsymbol{c}$, referred to as the echo, is generated as a weighted sum of the exemplar labels, with the activations as weights,

$$\boldsymbol{c} = \boldsymbol{V}\boldsymbol{a}. \tag{2}$$

Finally, the echo is normalised by dividing by its largest absolute element value, which is equivalent to the L∞ norm,

$$\langle \boldsymbol{c} \rangle_{\infty} = \frac{\boldsymbol{c}}{||\boldsymbol{c}||_{\infty}}. \tag{3}$$

Minerva 2 has mechanisms by which exemplars can be learned and forgotten, but for this work we will be using a fixed exemplar set which is neither added to nor degraded. Under these circumstances, it has been shown that Minerva 2 is a form of FFNN with pre-determined parameters [10], and as such, its performance is not expected to exceed that of an equivalently sized FFNN trained on sufficient data. Instead, Minerva 2's strengths lie in the information contained in the exemplars, allowing it to function *without* training. It can also serve as a template on which to base more flexible models, discussed in §3.3 and §3.4.

2.2. Echo-of-echoes

Classes in machine learning are typically assumed to be discrete and are represented by one-hot vectors. This is not assumed for Minerva 2; instead, class representation can be any vector of length J with elements ± 1. This has the benefit of allowing classes to be correlated with each other, but introduces the problem of *ambiguous recall*. Ideally, the normalised echo (Equation 3) will closely resemble a known class, but this is not guaranteed. A class could be identified using a similarity or distance-based measure, comparing the normalised echo with some 'true' class representation, but [28] also offers

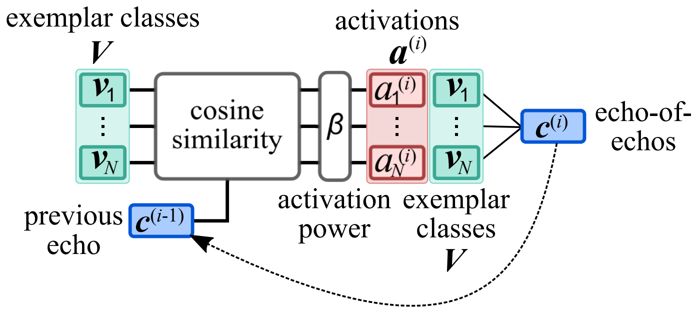


Figure 2: Minerva 2's echo-of-echoes process.

the option of iterating the process of echo generation to produce an 'echo-of-echoes'. This process is shown in Figure 2.

Let $\langle \boldsymbol{c} \rangle_\infty^{(0)}$ be the initial echo, as described in Equation 3, and let the notation $\boldsymbol{X}^{\circ\beta}$ denote raising each element in the vector or matrix $\boldsymbol{X}$ to the power β. The ith echo-of-echoes is obtained from the previous echo,

$$\boldsymbol{s}^{(i)} = \frac{1}{J}\boldsymbol{V}^\top \langle \boldsymbol{c}^{(i-1)} \rangle_\infty, \tag{4}$$

$$\boldsymbol{c}^{(i)} = \boldsymbol{V}\boldsymbol{a}^{(i)} \qquad \text{where } \boldsymbol{a}^{(i)} = \boldsymbol{s}^{(i)^{\circ\beta}} \tag{5}$$

$$\langle \boldsymbol{c}^{(i)} \rangle_\infty = \frac{\boldsymbol{c}^{(i)}}{||\boldsymbol{c}^{(i)}||_\infty}. \tag{6}$$

There is a history of using iterative 'deblurring' techniques in psychologically-motivated memory models to reduce or eliminate within-class variability in class labels. The 'brain-state-in-a box' model [29] incorporates an iterative feedback system, a version of which was also incorporated into the Theory of Distributed Associative Memory (TODAM) [30] memory model [31]. The model's results were found to be inconsistent with previous data, however [32], and the same can be said for Minerva 2's echo-of-echoes process. While the originator of Minerva 2 found that the results of iterating this process for any initial echo rapidly converged to one of the exemplar classes [28], others have had problems replicating the results. The echo-of-echoes process gave no benefit on a phone classification task [16], and was found to cause active deterioration in the output in an artificial grammar learning task [22].

The idea of an iterative approach to inference remains compelling, especially given the recent popularity of diffusion models in machine learning, which iteratively refine on initial predictions [33]. There is therefore value in exploring Minerva 2's echo-of-echoes process in more detail, which we do in §3.1.

2.3. Similarity with other approaches

As well as its similarity with the attention mechanism found in transformers, Minerva 2 is related to exemplar-based Nearest Neighbour (NN) approaches. In particular, in the limit as $\beta \to \infty$, Minerva 2 becomes equivalent to 1-nearest neighbour. The activation power β serves a broadly similar purpose to the k in k-nearest neighbour: a high value of k, or a low value of β, means that a large number of exemplars have an influence on the output. A low value of k, or a high value of β, means that relatively few exemplars will have a meaningful impact on the output. NN approaches have been widely used for speech and language tasks, including phone classification [34, 35, 36], voice recognition [37], speech emotion recognition [38] and voice conversion [39].

Sparse Representation (SR) is another exemplar method that has been used for phone recognition [40]. A key assumption of Minerva 2 is that any input's *label* can be represented as a linear combination of the exemplar labels (see Equation 2). SR methods make a similar assumption that the input *features* can be represented as a linear combination of the exemplar features. In contrast to Minerva 2, however, SR techniques enforce sparsity in the linear combination; that is, many of the exemplars' weights are forced to zero. This would be similar to forcing many of the weight in the vector $\boldsymbol{s}$ in Equation 1 to be zero. This enforced sparsity reduces overfitting. Although they are not equivalent, the degree of sparsity has a similar purpose to the activation power β, ensuring that irrelevant exemplars are excluded.

Unlike many exemplar approaches, such as k-NN, Minerva 2 is inherently differentiable. This opens up the possibility of incorporating learned parameters, and training using backpropagation. This has been shown to substantially improve the performance of Minerva on relatively small exemplar sets, while reducing the computation required for inference [41, 10, 42], and is explored further here. This is a hybrid exemplar/prototype approach, and has some similarity to previous hyrbid approaches. In particular, memory networks used for text-based question answering tasks [43, 44] incorporate a memory of exemplars as well as ANNs to transform the input, update the memory, and process and output the result. While the memory can be updated during both training and inference, the parameters of the ANNs are fixed once training is complete. Differentiable Neural Computers (DNCs)

[45] similarly have an updatable memory with a ‘controller’ ANN, although in this case the memory is not specifically composed of previous examples.

3. Theoretical framework

3.1. Echo-of-echoes

To solve the problem of ambiguous recall using echo-of-echoes, the $\boldsymbol{c}^{(i)}$ must converge to a recognisable class representation with little or no ambiguity. The ith echo-of-echoes, $\boldsymbol{c}^{(i)}$, can be written in terms of $\boldsymbol{c}^{(i-1)}$ by combining Equations 4 to 6,

$$\boldsymbol{c}^{(i)} = \frac{1}{J^{\beta}}\boldsymbol{V}\left(\boldsymbol{V}^{\top}\langle\boldsymbol{c}^{(i-1)}\rangle_{\infty}\right)^{\circ\beta}. \tag{7}$$

In order to be useful, the sequence must converge,

$$\lim_{i\to\infty}\boldsymbol{c}^{(i)} = \boldsymbol{c}^{*}. \tag{8}$$

If the sequence converges, then $\boldsymbol{c}^{*}$ must satisfy,

$$\begin{aligned}\boldsymbol{c}^{*} &= \frac{1}{J^{\beta}}\boldsymbol{V}\left(\boldsymbol{V}^{\top}\langle\boldsymbol{c}^{*}\rangle_{\infty}\right)^{\circ\beta} && (9)\\ &= \alpha\boldsymbol{V}\left(\boldsymbol{V}^{\top}\boldsymbol{c}^{*}\right)^{\circ\beta}, \qquad \text{where } \alpha = \frac{1}{J^{\beta}\left(||\boldsymbol{c}^{*}||_{\infty}\right)^{\beta}}. && (10)\end{aligned}$$

We first consider the case where there are W unique, linearly independent class representations, each of which is represented in the exemplar set once, and denoting this restricted set of exemplars $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_W]$. According to the echo-of-echoes process, each of these class representations must be a fixed point,

$$\left\{\boldsymbol{u}_w = \alpha_w\boldsymbol{U}\left(\boldsymbol{U}^{\top}\boldsymbol{u}_w\right)^{\circ\beta}\right\}_{w=1}^{W} \tag{11}$$

These can be written in matrix form as,

$$\boldsymbol{U} = \boldsymbol{U}\left(\boldsymbol{U}^{\top}\boldsymbol{U}\right)^{\circ\beta}\boldsymbol{\Lambda} \qquad \text{where } \boldsymbol{\Lambda} = \begin{bmatrix}\alpha_1 & \dots & 0\\ \vdots & \ddots & \vdots\\ 0 & \dots & \alpha_W\end{bmatrix}. \tag{12}$$

Since $\boldsymbol{U}$ has been defined to have full rank, it must have an inverse, so,

$$\boldsymbol{U}^{-1}\boldsymbol{U} = \boldsymbol{U}^{-1}\boldsymbol{U}\left(\boldsymbol{U}^{\top}\boldsymbol{U}\right)^{\circ\beta}\boldsymbol{\Lambda} \tag{13}$$

$$\boldsymbol{I}_W = \left(\boldsymbol{U}^{\top}\boldsymbol{U}\right)^{\circ\beta}\boldsymbol{\Lambda} \tag{14}$$

$$\boldsymbol{\Lambda}^{-1} = \left(\boldsymbol{U}^{\top}\boldsymbol{U}\right)^{\circ\beta}. \tag{15}$$

where $\boldsymbol{I}_W$ is the W-dimensional identity matrix. Since $\boldsymbol{\Lambda}$ is diagonal, its inverse is diagonal, so the matrix $\left(\boldsymbol{U}^{\top}\boldsymbol{U}\right)^{\circ\beta}$ must also be a diagonal matrix. Since β acts element-wise, so too is $\boldsymbol{U}^{\top}\boldsymbol{U}$. This can only be the case if the column vectors $\boldsymbol{u}_1, ..., \boldsymbol{u}_W$ are orthogonal. Thus, class representations that are linearly independent must also be orthogonal. The simplest way to represent classes orthogonally is to use one-hot vectors. In this case, the computationally intensive echo-of-echos process is unnecessary, since the correct class can be identified by taking the *argmax* of the output. This option is discussed in more detail in §3.3.

The results above were derived by assuming linear independence between the exemplars, but what if this is not the case? Equation 12 is then replaced with,

$$\boldsymbol{V} = \boldsymbol{V}(\boldsymbol{V}^{\top}\boldsymbol{V})^{\circ\beta}\boldsymbol{\Lambda}_V. \tag{16}$$

In this case, $\boldsymbol{V}$ is a $W \times N$ matrix, where N is the number of exemplars. Since the exemplars have some linear dependence, it has rank $r < N$, and as such has no left inverse. This means that there may be multiple solutions. Equation 16 is not true for every $\boldsymbol{V}$, however; the exemplar labels would need to be chosen to be suitable.

3.2. Equivalence to deep equilibrium models

Minerva 2 has been shown to be a form of FFNN [10], and the same is true for the echo-of-echoes process. Rewriting Equations 4 to 6, it can be seen that,

$$\boldsymbol{a}^{(i)} = \boldsymbol{\sigma}^{(a)}\left(\boldsymbol{W}^{(a)}\langle\boldsymbol{c}^{(i-1)}\rangle_\infty + \boldsymbol{b}^{(a)}\right) \tag{17}$$

$$\langle\boldsymbol{c}^{(i)}\rangle_\infty = \boldsymbol{\sigma}^{(c)}\left(\boldsymbol{W}^{(c)}\boldsymbol{a}^{(i)} + \boldsymbol{b}^{(c)}\right) \tag{18}$$

where

$$\boldsymbol{W}^{(a)} = \frac{1}{J}\boldsymbol{V}^{\top} \qquad \boldsymbol{b}^{(a)} = \boldsymbol{0} \qquad \boldsymbol{\sigma}^{(a)}(\boldsymbol{x}) = \boldsymbol{x}^{\circ\beta} \tag{19}$$

$$\boldsymbol{W}^{(c)} = \boldsymbol{V} \qquad \boldsymbol{b}^{(c)} = \boldsymbol{0} \qquad \boldsymbol{\sigma}^{(c)}(\boldsymbol{x}) = \langle\boldsymbol{x}\rangle_\infty. \tag{20}$$

Each iteration of the echo-of-echoes process is a 2-layer FFNN, with parameters shared with previous iterations. As an infinite-depth FFNN in which the layers share parameters, this is a form of Deep Equilibrium Model (DEM) [46]. The echo-of-echoes process differs from a typical DEM in two ways, however. Firstly, DEMs usually have a single layer repeated, rather than two. More importantly, for the echo-of-echoes process, fixed points are expected to be the exemplar class representations: $\boldsymbol{c}^* \in \{\boldsymbol{v}_1, ..., \boldsymbol{v}_N\}$. This is not assumed for DEMs. Relaxing this constraint by allowing the exemplar labels and fixed points to be separate allows us to rewrite Equation 16,

$$\boldsymbol{C}^* = \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{C}^*)^{\circ\beta} \boldsymbol{\Lambda}, \tag{21}$$

where $\boldsymbol{C}^* = \begin{bmatrix} \boldsymbol{c}_1^* & \dots & \boldsymbol{c}_M^* \end{bmatrix}$ is a matrix of column vectors making up the fixed points. The exemplar labels $\boldsymbol{V}$ and the prediction labels $\boldsymbol{C}^*$ have different label spaces, as with conventional DEMs, meaning that the echo-of-echoes process could in principal be trained using known fixed-point algorithms, and making use of the efficient backpropagation of DEMs.

3.3. Previous adaptations of Minerva 2 to speech and language tasks

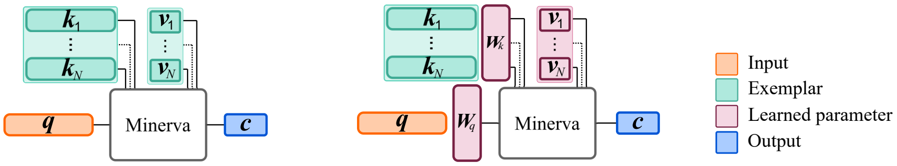
Figure 3 shows two variants of Minerva 2 with differing levels of parameterisation, first described in [10]. Both of these models use the cosine similarity in place of Equation 1, so that the input and exemplar features can take any real value, rather than being restricted to ±1.

$$\boldsymbol{s} = \tilde{\boldsymbol{K}}^\top \tilde{\boldsymbol{q}}, \quad \text{where } \tilde{\boldsymbol{q}} = \frac{\boldsymbol{q}}{||\boldsymbol{q}||_2}, \ \tilde{\boldsymbol{k}}_n = \frac{\boldsymbol{k}_n}{||\boldsymbol{k}_n||_2} \text{ and } \tilde{\boldsymbol{K}} = \begin{bmatrix} \tilde{\boldsymbol{k}}_1 & \dots & \tilde{\boldsymbol{k}}_N \end{bmatrix}. \tag{22}$$

For classification, the models make use of one-hot representation for the classes, which is a common choice in machine learning. Under these conditions, the predicted class label is given by,

$$\hat{w} = \underset{w=1,...,W}{argmax} \left(c_w \right). \tag{23}$$

For the simpler version of the model, referred to as Minerva-R, this is the only difference from Minerva 2. Minerva-R has no learned parameters, and the L∞ norm in Equation 3 is not required for classification tasks using one-hot representation and Equation 23, but for regression tasks, some form of scaling or calibration will be required for the output echo.



a. Minerva-R.

b. Minerva-RPE.

Figure 3: Minerva variations.

Minerva-RPE, a parameterised version of Minerva 2 shown in Figure 3b., incorporates learned parameters in two ways: a linear transformation of the input features; and learned class labels for the exemplars. Learning a linear feature transformation allows the model to emphasise relevant information in the features and discard irrelevant information, making the similarity measure between the input and the exemplars more meaningful. Equation 1 is replaced with,

$$\boldsymbol{s} = \tilde{\boldsymbol{K}}_w^\top \tilde{\boldsymbol{q}}_w \tag{24}$$

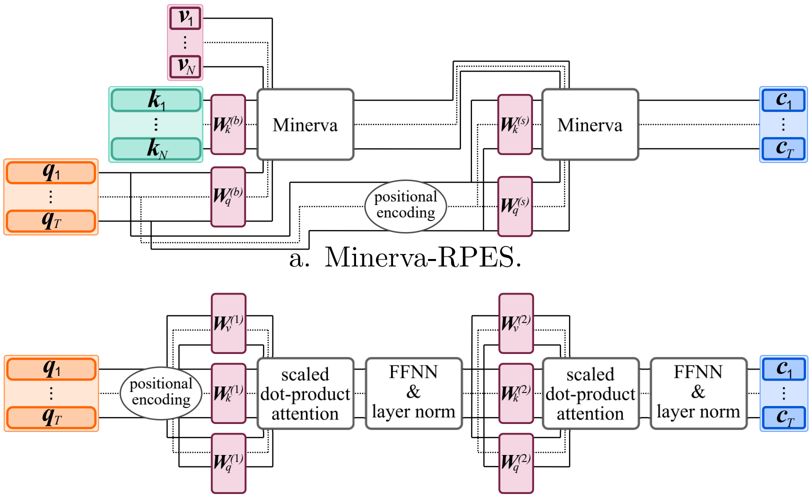
where $\boldsymbol{q}_w = \boldsymbol{W}\boldsymbol{q}$ and $\boldsymbol{K}_w = \boldsymbol{W}\boldsymbol{K}$, and $\boldsymbol{W}$ is a learned transformation matrix. Learning exemplar labels has multiple benefits. In the case of regression, it can reduce the 'noise' associated with the exemplars. In the case of classification, it allows an exemplar to fall on a spectrum between classes. In both cases, it also allows for correction of mislabelled data in the exemplars, and increases modelling power.

Minerva-RPE has two different sets of parameters: learned and unlearned. The learned parameters are the linear transformation and the exemplar class labels, $\{\boldsymbol{W}, \boldsymbol{V}\}$, and the unlearned parameters are the exemplar features, $\{\boldsymbol{K}\}$. The model can be used either for classification or regression. For classification, a softmax is applied to the output echo, and cross-entropy loss is used for training. For multi-class classification, where an input can be a member of more than one class at once, the magnitude of the outputs is adjusted by including learned scaling for each class, $\boldsymbol{z}_i = \boldsymbol{a} \cdot \boldsymbol{c}_i + \boldsymbol{b}$, followed by a sigmoid activation and binary cross-entropy loss across all classes. For regression, as with Minerva-R, calibration is likely to be necessary, so a final learned affine transformation is applied to the output, $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{b}$, and mean squared error (MSE) loss is used for training.

The computational complexity of Minerva-R and Minerva-RPE scale linearly with exemplar set size N, and may become prohibitively expensive for very large exemplar sets. One option for reducing the computation load is to reduce the size of the exemplar set, and the effect of this is explored in §5.1. A further option is available for the Minerva-RPE model: reduce the size of the feature transformation dimension, which is discussed in Section 4.5 of [42].

3.4. Minerva-RPES

Given the parallels between Minerva and attention, the transformer architecture can be used as inspiration for a sequence version of Minerva, which we shall refer to as Minerva-RPES, and which is shown in Figure 4a. It is composed of two Minerva modules: the first is Minerva-RPE, which assigns initial labels to the input sequence. The second Minerva module can be thought of as 'self-Minerva', in which the input sequence also forms the exemplars. This allows the prediction for each frame to take into account the rest of the utterance, making use of the initial labels produced by the first Minerva module.



a. Minerva-RPES.

b. Stacked self-attention.

Figure 4: Comparison of Minerva-RPES with 2-layer stacked self-attention.

Let $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1 & \dots & \boldsymbol{q}_T \end{bmatrix}$ be a sequence input of length T. The first Minerva module, shown to the left in Figure 4a., is Minerva-RPE, which can be represented in matrix form for the entire utterance by,

$$\boldsymbol{S}^{(b)} = \tilde{\boldsymbol{K}}^{(b)\top} \tilde{\boldsymbol{Q}}^{(b)} \tag{25}$$

where $\boldsymbol{Q}^{(b)} = \boldsymbol{W}_q^{(b)}\boldsymbol{Q}$ and $\boldsymbol{K}^{(b)} = \boldsymbol{W}_k^{(b)}\boldsymbol{K}$ and where the superscript (b) denotes the *base* module. The output of the first module is,

$$\boldsymbol{C}^{(b)} = \boldsymbol{V}\boldsymbol{A}^{(b)} \qquad\qquad \text{where } \boldsymbol{A}^{(b)} = \boldsymbol{S}^{(b)^{\circ\beta}}, \tag{26}$$

where $\boldsymbol{C}^{(b)} = \begin{bmatrix}\boldsymbol{c}_1^{(b)} & \dots & \boldsymbol{c}_T^{(b)}\end{bmatrix}$ gives the base (non-sequence) predicted labels of the input sequence. The Minerva process takes no account of the order of an input, so we make use of a positional embedding such as Rotary Positional Embeddings (RoPE) [47]. Let $pos(\boldsymbol{Q})$ be the input sequence with positional embeddings. Combined with the estimated labels, $\boldsymbol{C}^{(b)}$, we can use Minerva with the now-labelled sequence as both input and exemplars:

$$\boldsymbol{S}^{(s)} = \tilde{\boldsymbol{K}}^{(s)\top}\tilde{\boldsymbol{Q}}^{(s)} \tag{27}$$

where $\boldsymbol{Q}^{(s)} = pos\left(\boldsymbol{W}_q^{(s)}\boldsymbol{Q}\right)$ and $\boldsymbol{K}^{(s)} = pos\left(\boldsymbol{W}_k^{(s)}\boldsymbol{Q}\right)$, and where the superscript (s) denotes the *sequence* module. The model output is,

$$\boldsymbol{C} = \boldsymbol{C}^{(b)}\boldsymbol{A}^{(s)} \qquad\qquad \text{where } \boldsymbol{A}^{(s)} = \boldsymbol{S}^{(s)^{\circ\beta}}. \tag{28}$$

The model can be trained for classification or regression using cross-entropy loss or Mean Squared Error (MSE) loss respectively, with unlearned parameters $\{\boldsymbol{K}\}$ and learned parameters $\{\boldsymbol{W}_q^{(b)}, \boldsymbol{W}_k^{(b)}, \boldsymbol{W}_q^{(s)}, \boldsymbol{W}_k^{(s)}, \boldsymbol{V}\}$.

Figure 4 shows Minerva-RPES in direct comparison with a 2-layer stacked self-attention transformer-encoder. The most crucial difference between Minerva and scaled self-attention is the nature of the **input** to the first module: in self-attention, the queries, keys and values are all derived from the same sequence input; in Minerva, the query is derived from the sequence input, but the keys and values are derived from the exemplars. Further, the **similarity measure** employed is different: scaled dot-product attention produces positive weights that sum to one; Minerva produces weights that, individually, fall in the range $[-1, 1]$. And finally, although not a key feature, stacked self-attention is typically supplemented with feed-forward layers and layer normalisation [48]. These three traits, **input**, **similarity measure** and **FF + layer norm**, can be mixed-and-matched to produce class-equivalent models that fall on a spectrum between Minerva-RPES and stacked self-attention.

The computational complexity of the Minerva-RPES model scales with NT, where N is the number of exemplars and T is the input sequence length. The effect of reducing the length of the context used within the input sequence is explored in §5.2.

4. Experiments

Experiments were conducted to test: the effect of exemplar set size and activation power on a three different tasks; and the performance of the Minerva-RPES model compared with previously published Minerva-based models and a baseline FFNN. Three speech and language tasks were chosen:

1. Frame-based phone classification using the TIMIT dataset [49]. This is a classification task, and allows testing of a sequence model. The detailed phonetic labelling of this dataset allows exploration of a speech-based classification model.
2. Emotion classification of text using the GoEmotions dataset [50]. This task is a multiple-classification task, in which an input can be classified into more than one class. It allows us to explore both the use of Minerva for multi-label classification, and for a text-based task.
3. Speech intelligibility prediction using the Clarity Prediction Challenge 2 (CPC2) dataset [51]. Since Minerva is, by default, a regression model, it is useful to test its regression capabilities.

Feature representation has been shown to be important to Minerva 2's performance [10], so three different feature representations of varying quality were chosen for each task. The objective of this work is not to find the best possible feature representation for the tasks, but rather to explore the effect that feature representation has on the model outputs.

4.1. TIMIT frame-based phone recognition

TIMIT is a dataset composed of short, single-sentence sentence utterances labelled with phonetic information as well as a text transcription. The *training set* has 3696 utterances from 462 speakers (326 male, 136 female), the *development set* has 311 utterances from 50 speakers (32 male, 18 female), and the *test set* has 192 utterances from 24 speakers (16 male, 8 female). There is no speaker overlap between training, development and test sets. This work makes use of TIMIT's phonetic labelling, using a reduced set of 39 labels [52], rather than the 61 labels used in the original data, as is common with this dataset. The classes are imbalanced, with the 'silence' label representing almost a quarter of the training data, while the least represented label is /g/, at 0.3%. The development and test sets are similarly imbalanced. Three feature representations were used:

Log mel spectrogram: 96-dimensional features were obtained from the 32 channel log mel spectrogram. The stride was 20 ms, chosen to match the Wav2vec and HuBERT features (see next), and the window was 32 ms. Delta (differential) and delta-delta (acceleration) features were also included.

Wav2vec: 768-dimensional features were obtained from the final layer of a Wav2vec Self-Supervised Speech Representation (SSSR) model, pre-trained on Librispeech, using 960 hours of unlabelled data [53].

HuBERT: 768-dimensional features were obtained from the final layer of a HuBERT SSSR model, pre-trained on Librispeech, using 960 hours of unlabelled data [54].

Better performance could likely be achieved by choosing a suitable layer for feature extraction from the the Wav2vec and HuBERT models [55], but using the final layer allows direct comparison with the Minerva models reported in [10].

4.2. GoEmotions

GoEmotions [50] is dataset of reddit posts paired with annotated human emotion labels: *positive*, *negative*, *neutral* and *ambiguous*. This is a multi-classification task, with each utterance potentially belonging to more than one class; for example, one post might be considered both *neutral* and *positive* by different annotators. There are 58,009 annotated posts in total, divided into defined training/development/test splits of size 43,410 / 5,426 / 5427 (80% / 10% / 10%). More detailed annotations are available, but initial work on the models made use of the simple labels described here. Three feature representations were used:

LSA: The LSA features are described in [56], and are pretrained on the Touchstone Applied Science Associates (TASA) corpus. They are word-based 300-dimensional vectors, which were averaged over the words to produce a single vector representation of each sentence. These features have previously been used by [13] in conjunction with Minerva for comparison with human studies.

Word2vec: The Word2vec features [57] were obtained from a model pretrained on a part of the Google News dataset (around 100 billion words), resulting in 1024-dimensional word vectors, which were averaged over words / tokens to produce a single vector representation of each sentence.

BERT: The BERT features were obtained using the sentence-transformers python package [58], producing 768-dimensional vectors. The model is based on MPNet [59], then further trained on a variety of datasets. The model output is a single vector to represent the entire sentence.

4.3. Clarity prediction challenge 2

The CPC2 dataset [51] consists of utterances that have artificial noise added, before being enhanced by an enhancement system (a simulated hearing aid). The enhancement system is matched to a specific hearing-impaired listener, who listens to the enhanced noisy utterance, and repeats it back. The utterance is labelled with the 'correctness': the percentage of words the listener was able to repeat back correctly. The correctness is used as a measure of intelligibility. The objective of this task is to predict the correctness from the speech waveform. Additional information is available, such as the clean audio and basic information abut the listener's degree of hearing loss, but this work made use of only the enhanced noisy speech waveform and the correctness label.

The CPC2 data is divided into three training/evaluation pairs. The listeners and enhancement systems present in each evaluation set are not present in the corresponding training set, requiring models to generalise to unseen listeners and enhancement systems. There is overlap between the three training sets, however; the setup is effectively enforced 3-fold cross validation. There are no defined development sets, so for each training/evaluation pair, two listeners and two enhancement systems were selected at random to form a disjoint development set. All data using these listeners and enhancement systems were removed from the training data. Of the remaining training data, 10% was separated into a non-disjoint development set. The original Split 1 has 8599/305 data pairs for training/evaluation; Split 2 has 8135/294; and Split 3 has 7896/298. Following the creation of disjoint and non-disjoint development sets, the training/non-disjoint/disjoint sizes were: 5190/577/170 for Split 1; 5087/566/169 for spit 2; and 5213/580/166 for Split 3. The disjoint development sets were used for hyperparameter tuning and model selection. The non-disjoint validation set was used to assess the difference performance of models on previously-seen listeners and enhancement systems, which in turn gives information on how well the model generalises to unseen listeners and enhancement systems. For further details on the disjoint/non-disjoint development sets, see Appendix A.

The 32 kHz CPC2 waveforms were downsampled to 16 kHz. Three different feature representations were used:

Log Spectrogram: Fast Fourier transforms were used to compute 257-dimensional log magnitude spectrogram features, with a window of 32 ms and a stride of 16 ms.

XLSR: 1024-dimensional Cross-Lingual Representation Learning for Speech Recognition (XLSR) features were obtained from a model pretrained on 436k hours of multilingual data [60], which has been found to be effective for speech intelligibility prediction [61].

Whisper: 768-dimensional features were taken from the 8th decoder layer of a pretrained Whisper ASR model [62], which has been found to be effective for speech intelligibility prediction [41].

Feature representations were averaged over the time-domain to provide a single vector representation for each utterance. This simple method has been shown to perform competitively [10].

4.4. Exemplar set size and activation power

For each of the nine task/feature combinations, Minerva-R models were evaluated with a range of different exemplar set sizes and activation powers. For the TIMIT models, exemplars were selected randomly from the training set and stratified by phonetic class. This guarantees that all exemplar sets include examples from all classes, despite the class imbalance in the training data. The exemplar set size started at 39 exemplars (1 per class), and was doubled for progressive models until the size reached 79872 (2048 per class). The TIMIT results are reported in terms of the classification accuracy. Randomly selecting a class for each new input would result in an accuracy of around 2.6%, whereas reporting 'silence' for all frames would achieve around 24%, due to TIMIT's class imbalance. Higher values show better performance. Phone accuracy differs from Phone Error Rate (PER), a metric that is commonly used for segment-level phonetic modelling, and cannot be directly compared with it.

For the GoEmotions models, exemplars were selected randomly from the training set. The exemplar set size started at 8 exemplars, and was doubled for progressive models until the size reached 65536. This is a multi-classification task, where an input can belong to multiple classes. Since the model applies no scaling to the output, Area Under Receiver Operating Characteristic Curve (AUC) was used as the performance metric, which does not require class thresholds to be set. A value of 50% is equivalent to chance. Higher values show better performance.

For the CPC2 task, exemplars were selected randomly from the training set. The exemplar set size started at 4 exemplars, and was doubled for progressive models until the size reached 8192. Since the Minerva-R model has no innate scaling, a calibration was performed on the model output: the exemplar features were used as input to the model, and the resulting model output was matched to the true exemplar labels using least-squares logistic regression. The parameters for this regression were then used to scale model output at inference. The results are reported in terms of Root Mean Squared Error (RMSE) of the predicted correctness, with 0 RMSE indicating perfect prediction. Predicting the mean correctness for every utterance in the CPC2 test set results in an RMSE of 40.0%. Values at or above this level show performance no better than chance.

For the smallest model for each task/feature combination, the activation power was set to 1, and then incremented by 2 (to ensure odd powers) to find the optimal value. For subsequent models, the initial value of the activation power was set to the optimal value of the previous model, and the activation power was incremented and decremented by 2 until the optimal value was found.

4.5. Sequence Minerva

Minerva-RPES models were trained on the TIMIT task using log mel spectrogram and HuBERT features. Both models had an exemplar set of size 14976 (384 exemplars per class) and a feature transformation dimension of 64 on the non-sequence variants of Minerva. This is smaller than either the log mel spectrogram or HuBERT feature representations (96 and 768 respectively), but increasing the feature transformation dimension above 64 was found to give no meaningful improvement on non-sequence Minerva models (Section 4.5 of [42]), and the same dimension has been used here for comparability with previous models. The models were trained with categorical cross-entropy loss. These models were designed to be comparable to previous Minerva-based models reported by [10]. Training was conducted using a single NVIDIA RTX 3080 GPU with 10 GB RAM, and Minerva-RPES models took around 20 minutes to train.

Further Minerva-RPES / stacked self-attention models were trained to explore the class equivalence between these models. The **input**, **similarity measure** and **FF + layer norm** traits described in §3.4 were mixed-and-matched to produce 8 new models, which were each trained with log mel spectrogram and HuBERT features. Since the self-attention models use the current input sequence in place of an exemplar set, the exemplar set size of the Minerva-based models was set to be approximately equivalent in size to an input utterance. TIMIT utterances are on average around 152 frames (with stride 20 ms), so the exemplar sets used 4 exemplars per class, for 156 exemplars overall. The feature transformation dimension for all models was 64. Training was conducted using a single NVIDIA RTX 3080 GPU with 10 GB RAM, and Minerva-RPES models took around 14 minutes to train. For all models described in this section, hyperparameters were tuned on the development set, and hyperparameter values are given in Appendix A.

5. Results and discussion

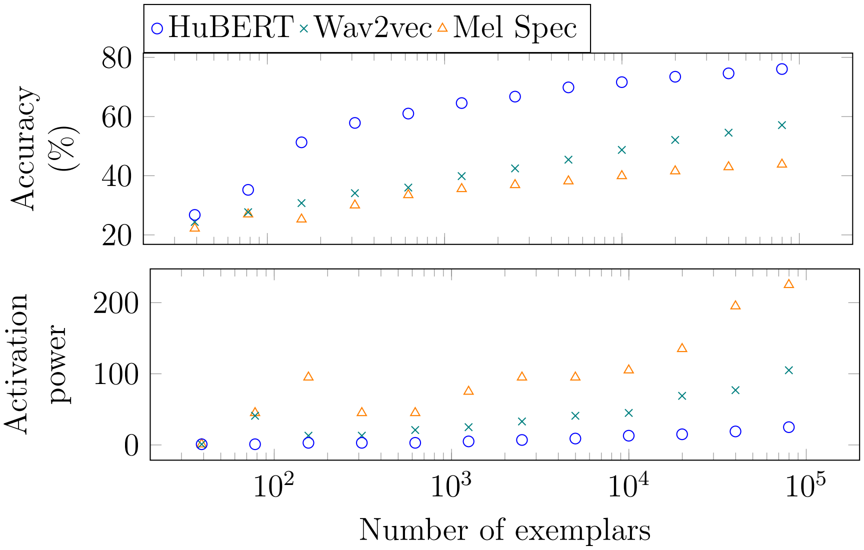


Figure 5: Performance and optimal β for different exemplar set sizes on TIMIT.

5.1. Activation power and exemplar set size

Figures 5 to 7 show the performance and optimal activation power β of the untrained Minerva-R model on the TIMIT, GoEmotions and CPC2 tasks respectively, for each feature representation, for increasing exemplar set size. Note that the results for the TIMIT task using HuBERT features were previously reported in [10]. The spectrogram features for the CPC2 tasks are

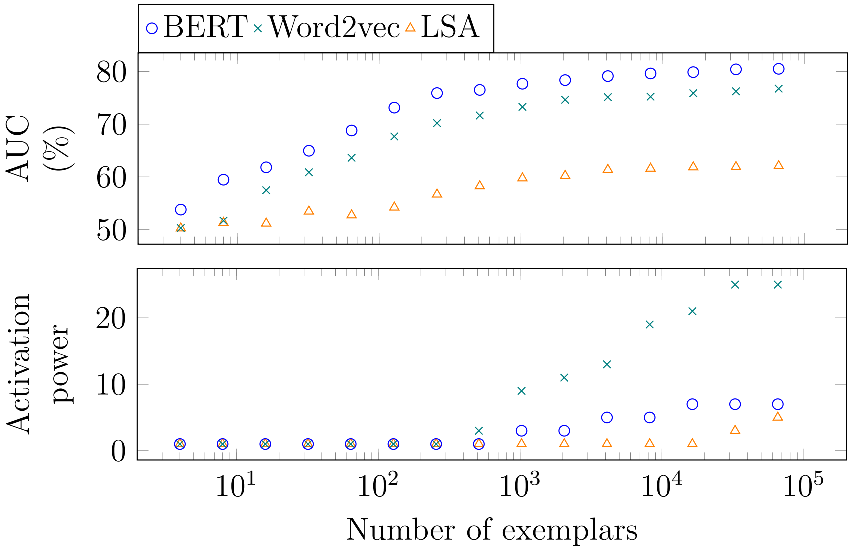


Figure 6: Performance and optimal β for different exemplar set sizes on GoEmotions.

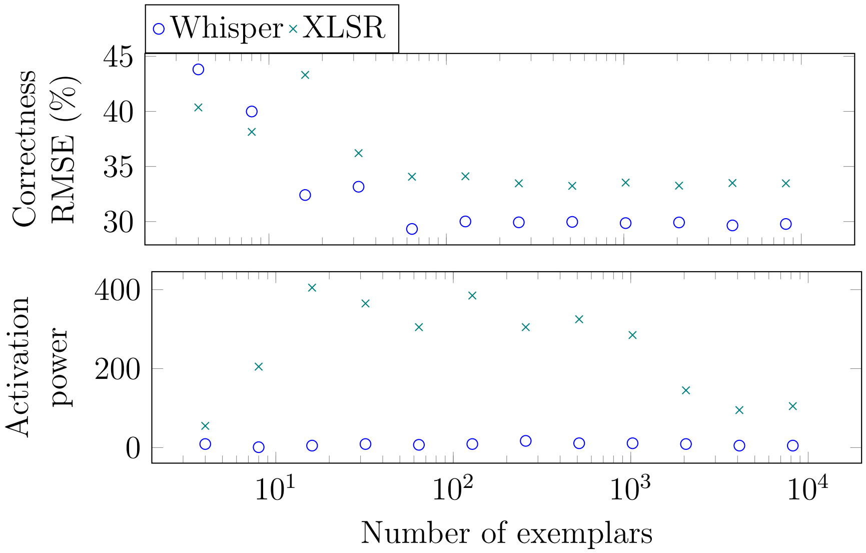


Figure 7: Performance and optimal β for different exemplar set sizes on CPC2.

excluded, since they did not generalise effectively to the disjoint evaluation set.

For the TIMIT and GoEmotions tasks, as the exemplar set size increases, performance improves, although with diminishing returns. Furthermore, as the exemplar set size increases, the optimal value of the activation power β increases, for both tasks and all feature representations This matches previous results [10].

For the CPC2 task, performance improved up to 64 exemplars for the Whisper features, and up to 512 exemplars for the XLSR features. No performance improvement was seen for exemplar sets larger than these. Unlike the TIMIT and GoEmotions tasks, the optimal value for β appears to peak and then fall with increasing exemplar set size. This may be due to CPC2 being a regression task, rather than classification. In regression, the scaling of the output is crucial, and the value of β affects this scaling, since activations close to ± 1 are relatively unaffected, while activations with lower magnitude are reduced (see Equation 2). It is possible that the non-linearity added by the activation power becomes detrimental to the calibration process with large numbers of exemplars. It should be noted that, while the activation power for the XLSR features falls for larger exemplar sets, it is still very high at 105 for the maximum size of exemplar set tested, with 8192 exemplars. This is still high enough to highly emphasise the most similar exemplars.

Alternative activation functions could be considered in future work. The activation power used here preserves the sign of the similarities, meaning that an exemplar may be considered an 'opposite' of the input. This may be a useful trait if classes can be considered opposite to each other, for example positive and negative classes in emotion classification. A softmax, by comparison, assumes orthogonality between classes, and ensures consistent scaling of the activation power. This might be more appropriate for the CPC2 task, in which scaling is an important consideration.

5.2. Minerva-RPES

All statistical comparisons reported here are *t*-tests based on 5 repeats with different random initialisations and (in the case of Minerva models) different randomly selected exemplar sets. Table 1 shows results for the sequence Minerva-RPES model, compared with Minerva-R and Minerva-RPE models reported in [10], which use the same size exemplar set. A FFNN baseline model of approximately equivalent computational complexity, also previously reported in [10], is shown for comparison, as well as two exemplar-based models: a k-NN model from [35]; and a semi-supervised Measure Propagation (MP) model from [36]. The latter two are not directly comparable, since they use different feature representation and substantially different exemplar selection techniques, but they are useful for putting the Minerva models' results into context.

Table 1: Frame-based phone classification on TIMIT.

Features	Model	Learned params	Feature trans.	Learned labels	β	Accuracy (%) Dev	Test
MFCC	k-NN*	0	-	-	-	-	60.07
	MP**	0	-	-	-	-	58
Mel Spec	R†	0	No	No	135	41.51	40.65
	RPE†	0.59 M	Yes	Yes	7	68.48	67.02
	FFNN†	1.19 M	-	-	-	**69.64**	**68.19**
	RPES	0.61 M	Yes	Yes	5	68.29	66.95
HuBERT	R†	0	No	No	15	73.13	73.02
	RPE†	0.68 M	Yes	Yes	5	88.32	87.50
	FFNN†	1.88 M	-	-	-	88.36	87.60
	RPES	0.78 M	Yes	Yes	5	**88.73**	**87.88**

*From [35]; **from [36]; †from [10]

The k-NN model reported in [35] and MP model reported in [36], both shown in Table 1, outperform the Minerva-R model, despite also being non-parameterised exemplar-based approaches. The reasons for this are likely related to exemplar selection and feature representation. Both the k-NN model and the MP model use Mel Frequency Cepstral Coefficients (MFCC) features, rather than the log mel spectrogram used by Minerva-R, which is likely to affect the results somewhat. Log mel spectrogram features are highly correlated, which may have an impact given that Equation 24 determines the similarity between the input and each exemplar based on their correlation. Perhaps more importantly, the k-NN model uses the entire TIMIT training set as exemplars, as does the MP model, although the MP model is semi-supervised, with only 5% of the data being labelled. The Minerva-R model, in contrast, uses less than 3% of the training data, and as previously noted, increasing the exemplar set size leads to improved performance. Further, the Minerva-R exemplar set is stratified by phonetic class. This ensures that small exemplar sets contain representatives of all classes, but it has been shown that a randomly selected exemplar set (in which class imbalances are preserved) results in better performance using Minerva-R. This is because it increases the chance of predicting common classes such as silence, and reduces the chance of predicting rare classes such as /g/. For a more detailed discussion of exemplar set stratification, see section 4.4.2.2 of [42]. The stratified exemplar set is used here for comparability with the parameterised Minerva models.

The best performing model overall is the Minerva-RPES model with HuBERT features. Although the gain over the baseline HuBERT FFNN is modest, it is statistically significant ($p < 0.01$). The log mel spectrogram Minerva-RPES model, in contrast, performs substantially worse that the FFNN model, and worse than the simpler Minerva-RPE model. This is counter-intuitive: the HuBERT features include a lot of context information, which means that gains are likely to be modest. The log mel spectrogram features include little context information, so gains should be larger. Some insight can be gained from Table 2, which shows the performance of models that mix-and-match characteristics from Minerva-RPES and transformer architectures. It can be seen that the log mel spectrogram models all benefit from extra feedforward layers and layer normalisation, whereas the HuBERT models in general do not. The best performing log mel spectrogram model in Table 2 substantially outperforms all log mel spectrogram models in Table 1, with a test accuracy of 69.58 % (compared to 68.19 % for the FFNN), despite having a much smaller exemplar set. Using Minerva input (i.e. exemplars) and the Minerva-style activation function yields the best performance on this task for both feature representations, outperforming similar stacked self-attention models. This form of model therefore shows promise.

Minerva-RPES uses the entire utterance by default, but further experiments were performed with HuBERT features to determine how much context is useful. Context is measured in frames, with a stride of 10 ms. The measured contexts ranged from zero (effectively the same as Minerva-RPE) to 1024, which is sufficient to give full context on the short utterances found in TIMIT. Results for mid, backward and forward context are shown in Figure 8. Mid, backward and forward context are all useful, but mid-context is the most useful. Performance plateaus at around 32 frames.

The Minerva-RPES results reported here show promise, but further work comparing this model with alternative architectures on additional, ideally larger, datasets would be a logical next step.

Table 2: Performance of models on the Minerva-Transformer spectrum, using log mel spectrogram and Hubert features.

Base model	Activations	FF + LN	Learned params	Test accuracy (%)	
			MelSpec/Hubert	MelSpec	Hubert
Minerva	Minerva	None	31 k / 203 k	65.66	**87.77**
		Both	34 k / 206 k	**69.58**	87.52
	scaled dot-product	None	31 k / 203 k	64.65	87.72
		Both	34 k / 206 k	68.10	87.38
Transformer	Minerva	None	29 k / 158 k	53.50	86.03
		Both	35 k / 164 k	69.24	87.10
	scaled dot-product	None	29 k / 158 k	48.78	86.45
		Both	35 k / 164 k	67.29	86.35

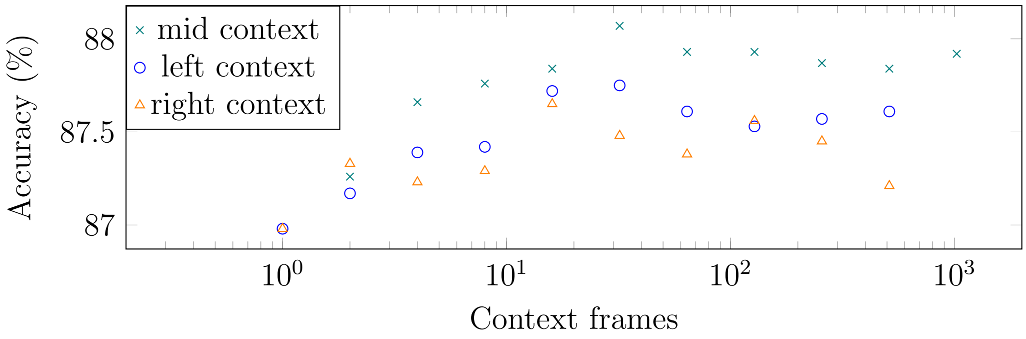


Figure 8: Accuracy of Minerva-RPES on the TIMIT task with increasing mid, left and right context.

6. Conclusions

We have shown that the iterative echo-of-echoes process proposed by its creator is a close relative of DEMs, and that relaxing a single assumption makes it a DEM. Since DEMs have shown excellent performance in a variety of tasks, further work exploring this option is warranted.

Based on experimental work, increasing Minerva's memory increases performance, although with diminishing returns. The activation power has a major effect on the quality of the model output, and should therefore ideally be tuned as a hyperparameter.

Minerva 2 is closely related to the attention mechanism found in transformers, and this similarity can be leveraged to convert Minerva to a sequence model, which shows promising experimental results.

Appendix A. Hyperparameter tuning

Appendix A.1. CPC2 development set

Both TIMIT and GoEmotions have established development sets in their references [49, 50]. CPC2 does not.

Table A.3: Listeners and enhancement systems used for the disjoint validation sets.

Training split	**Listeners**	**Enhancement systems**
Split 1	L0243	E032
	L0254	E038
Split 2	L0250	E031
	L0252	E036
Split 3	L0252	E031
	L0254	E032

CPC2 is divided into three paired training and evaluation splits, but has no established development sets for model selection / hyperparameter tuning. The training sets have overlap with each other, but the evaluation sets do not. For each of the three splits, two listeners and two systems were randomly selected to form a disjoint development set. All data with these listeners and systems were removed from the training set. A randomly selected non-disjoint development set consisting of 10% of the remaining training data was also formed. This enabled assessment of model performance both on previously seen and unseen listeners and enhancement systems. Hyperparameter tuning was performed using the disjoint development sets, to ensure generalisation to unseen listeners and systems. The development set results in Table 1 of the report are from the non-disjoint development set. Table A.3 gives the listeners and enhancement systems selected to form the disjoint development sets.

Appendix A.2. Hyperparameters for best-performing models

Tables A.4 to A.5 give the hyperparameters for each of the models reported.

Table A.4: Tuned hyperparameter values for the TIMIT Minerva-RPES models (Table 1).

Features	**Learning rate**	**Weight decay**	**Dropout**
Mel Spec	10^{-3}	10^{-7}	0.0
HuBERT	10^{-3}	10^{-7}	0.4

Table A.5: Tuned hyperparameter values for the TIMIT Minerva-RPES and transformer models with log mel spectrogram features (Table 2).

Base model	**Activations**	**FF + layer norm**	**Learning rate**	**Weight decay**	**Dropout**
Yes	Minerva	None	10^{-3}	10^{-3}	0.0
		Both	10^{-3}	10^{-7}	0.0
	scaled dot-product	None	10^{-3}	10^{-7}	0.0
		Both	10^{-3}	10^{-6}	0.0
No	Minerva	None	10^{-2}	10^{-7}	0.0
		Both	10^{-2}	10^{-7}	0.0
	scaled dot-product	None	10^{-4}	10^{-7}	0.1
		Both	10^{-3}	10^{-5}	0.0
FFNN	-	No	10^{-3}	10^{-5}	0.0

References

[1] R. M. Nosofsky, T. J. Palmeri, M. Stephen C, Rule-plus-exception model of classification learning, Psychological Review 101 (1) (1994) 53–79.

[2] M. A. Erickson, J. K. Kruschke, Rules and exemplars in category learning., Journal of Experimental Psychology: General 127 (1998).

[3] J. N. Rouder, R. Ratcliff, Comparing exemplar- and rule-based theories of categorization, Current Directions in Psychological Science 15 (2006).

[4] S. Natal, I. McLaren, E. Livesey, Generalization of feature- and rule-based learning in the categorization of dimensional stimuli: evidence for dual processes under cognitive control, J Exp Psychol Anim Behav Process 39 (2) (2013) 140–51.

[5] M. L. Mack, A. R. Preston, B. C. Love, Decoding the brain's algorithm for categorization from its neural implementation, Current Biology 23 (20) (2013) 2023–2027.

Table A.6: Tuned hyperparameter values for the TIMIT Minerva-RPES and transformer models with HuBERT features (Table 2).

Base model	**Activations**	**FF + layer norm**	**Learning rate**	**Weight decay**	**Dropout**
Yes	Minerva	None	10^{-4}	10^{-5}	0.2
		Both	10^{-3}	10^{-6}	0.3
	scaled dot-product	None	10^{-3}	10^{-6}	0.2
		Both	10^{-3}	10^{-7}	0.4
No	Minerva	None	10^{-3}	10^{-7}	0.2
		Both	10^{-3}	10^{-7}	0.2
	scaled dot-product	None	10^{-4}	10^{-6}	0.2
		Both	10^{-3}	10^{-3}	0.1
FFNN	-	No	10^{-4}	10^{-7}	0.1

[6] C. R. Bowman, D. Zeithamova, Abstract memory representations in the ventromedial prefrontal cortex and hippocampus support concept generalization, Journal of Neural Science 38 (10) (2018) 2605–2614.

[7] Y. Wu, M. N. Rabe, D. Hutchins, C. Szegedy, Memorizing transformers, in: 2022 International Conference on Learning Respresentations (ICLR), 2022.

[8] Z. Zhong, T. Lei, D. Chen, Training language models with memory augmentation, arXiv preprint arXiv:2205.12674 (2022).

[9] W. Wang, L. Dong, H. Cheng, X. Liu, X. Yan, J. Gao, F. Wei, Augmenting language models with long-term memory, Advances in Neural Information Processing Systems 36 (2024).

[10] R. Mogridge, A. Ragni, Learning from memory-based models, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2024, 2024, pp. 2360–2364.

[11] D. Hintzman, Minerva 2: a simulation model of human memory, Behav. Res. Methods Instrum. Comput. 16 (1984) 96–101. doi:10.3758/BF03202365.

[12] D. L. Hintzman, "schema abstraction" in a multiple-trace memory model., Psychological review 93 (4) (1986) 411.

[13] J. Nick Reid, R. K. Jamieson, True and false recognition in minerva 2: Extension to sentences and metaphors, Journal of Memory and Language 129 (2023) 104397. doi:https://doi.org/10.1016/j.jml.2022.104397.
URL https://www.sciencedirect.com/science/article/pii/S0749596X22000845

[14] K. McNeely-White, D. McNeely-White, A. Huebert, B. Carlaw, A. Cleary, Specifying a relationship between semantic and episodic memory in the computation of a feature-based familiarity signal using minerva 2, Memory & Cognition 50 (09 2021). doi:10.3758/s13421-021-01234-6.

[15] E. D. Reichle, A. Veldre, L. Yu, S. Andrews, A neural implementation of minerva 2, in: Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 44, 2022, pp. 2278–2284.

[16] V. Maier, R. K. Moore, An investigation into a simulation of episodic memory for automatic speech recognition, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2005, ISCA, 2005, pp. 1245–1248.

[17] V. Maier, R. K. Moore, Temporal episodic memory model: an evolution of Minerva2, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2007, 2007, pp. 866–869.

[18] R. Moore, V. Maier, Preserving fine phonetic detail using episodic memory: Automatic Speech Recognition using MINERVA2, in: International Congress of Phonetic Sciences, 2007, 2007, pp. 197–203.

[19] R. Ozen, R. L. West, M. A. Kelly, Minerva-q: A multiple-trace memory system for reinforcement learning, Learning 1 (1) (2022) 3.

[20] S. Dennis, K. Shabahang, H. Yim, The antecedents of transformer models, Current Directions in Psychological Science 34 (1) (2025) 3–11.

[21] D. L. Hintzman, Human learning and memory: connections and dissociations, Annual review of psychology (1990) 109–139.

[22] Z. Dienes, Connectionist and memory-array models of artificial grammar learning, Cognitive Science 16 (1) (1992) 41–79.

[23] M. R. Dougherty, C. F. Gettys, E. E. Ogden, Minerva-dm: A memory processes model for judgments of likelihood., Psychological Review 106 (1) (1999) 180.

[24] P. Sedlmeier, R. Hertwig, G. Gigerenzer, Are judgments of the positional frequencies of letters systematically biased due to availability?, Journal of Experimental Psychology: Learning, Memory, and Cognition 24 (3) (1998) 754.

[25] E. R. Smith, Illusory correlation in a simulated exemplar-based memory, Journal of Experimental Social Psychology 27 (2) (1991) 107–123.

[26] K. C. Klauer, T. Meiser, A source-monitoring analysis of illusory correlations, Personality and Social Psychology Bulletin 26 (9) (2000) 1074–1093.

[27] P. J. Kwantes, D. J. Mewhort, Evidence for sequential processing in visual word recognition., Journal of Experimental Psychology: Human Perception and Performance 25 (2) (1999) 376.

[28] D. L. Hintzman, Judgments of frequency and recognition memory in a multiple-trace memory model., Psychological review 95 (4) (1988) 528.

[29] J. A. Anderson, J. W. Silverstein, S. A. Ritz, R. S. Jones, Distinctive features, categorical perception, and probability learning: Some applications of a neural model, Psychological review 84 (5) (1977) 413.

[30] B. B. Murdock, Learning in a distributed memory model, in: Current issues in cognitive processes, Psychology Press, 2014, pp. 69–106.

[31] S. Lewandowsky, B. B. Murdock Jr, Memory for serial order, Psychological Review 96 (1) (1989) 25.

[32] J. S. Nairne, I. Neath, Critique of the retrieval/deblurring assumptions of the theory of distributed associative memory, Psychological Review (1994).

[33] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[34] L. Golipour, D. O'Shaughnessy, Context-independent phoneme recognition using a k-nearest neighbour classification approach, in: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 1341–1344.

[35] J. Labiak, K. Livescu, Nearest neighbors with learned distances for phonetic frame classification, in: Interspeech, 2011, pp. 2337–2340.

[36] A. Subramanya, J. Bilmes, Semi-supervised learning with measure propagation, Journal of Machine Learning Research 12 (11) (2011).

[37] Ranny, Voice recognition using k nearest neighbor and double distance method, in: 2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA), 2016, pp. 1–5.

[38] M. Venkata Subbarao, S. K. Terlapu, N. Geethika, K. D. Harika, Speech emotion recognition using k-nearest neighbor classifiers, in: P. Shetty D., S. Shetty (Eds.), Recent Advances in Artificial Intelligence and Data Engineering, Springer Singapore, Singapore, 2022, pp. 123–131.

[39] M. Baas, B. van Niekerk, H. Kamper, Voice conversion with just nearest neighbors, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2023, 2023, pp. 2053–2057.

[40] T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, D. Kanevsky, Exemplar-based sparse representation features: From timit to lvcsr, IEEE Transactions on Audio, Speech, and Language Processing 19 (8) (2011) 2598–2613.

[41] R. Mogridge, G. Close, R. Sutherland, T. Hain, J. Barker, S. Goetze, A. Ragni, Non-intrusive speech intelligibility prediction for hearing-impaired users using intermediate asr features and human memory models, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 306–310.

[42] R. Mogridge, An exemplar-informed approach to speech and language tasks, Ph.D. thesis, University of Sheffield (2024).

[43] J. Weston, S. Chopra, A. Bordes, Memory networks, arXiv preprint arXiv:1410.3916 (2014).

[44] S. Sukhbaatar, J. Weston, R. Fergus, et al., End-to-end memory networks, Advances in neural information processing systems 28 (2015).

[45] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al., Hybrid computing using a neural network with dynamic external memory, Nature 538 (7626) (2016) 471–476.

[46] S. Bai, J. Z. Kolter, V. Koltun, Deep equilibrium models, Advances in neural information processing systems 32 (2019).

[47] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, Neurocomputing 568 (2024) 127063.

[48] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[49] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, V. Zue, TIMIT acoustic-phonetic continuous speech corpus, Linguistic Data Consortium (1992).

[50] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, S. Ravi, Goemotions: A dataset of fine-grained emotions, arXiv preprint arXiv:2005.00547 (2020).

[51] J. Barker, M. Akeroyd, W. Bailey, T. J. Cox, J. F. Culling, J. Firth, S. Graetzer, , G. Naylor, The 2nd clarity prediction challenge: A machine learning challenge for hearing aid intelligibility prediction, in: 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024, pp. 11551–11555.

[52] C. Lopes, F. Perdigão, Phone recognition on timit database, Speech Technologies 1 (2011) 285–382. doi:10.5772/17600.

[53] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 12449–12460.

[54] 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 (2021).

[55] A. Pasad, J.-C. Chou, K. Livescu, Layer-wise analysis of a self-supervised speech representation model, in: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2021, pp. 914–921.

[56] F. Günther, C. Dudschig, B. Kaup, Lsafun - an r package for computations based on latent semantic analysis, Behavior Research Methods 47 (2015) 930–944.
URL https://link.springer.com/article/10.3758/s13428-014-0529-0

[57] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[58] N. Reimers, I. Gurevych, ”sentence-bert: Sentence embeddings using siamese bert-networks”, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019, pp. 3982–3992.

[59] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Advances in neural information processing systems 33 (2020) 16857–16867.

[60] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, M. Auli, Unsupervised cross-lingual representation learning for speech recognition, arXiv preprint arXiv:2006.13979 (2020).

[61] G. Close, T. Hain, S. Goetze, Non Intrusive Intelligibility Predictor for Hearing Impaired Individuals using Self Supervised Speech Representations, in: Proc. Workshop on Speech Foundation Models and their Performance Benchmarks (SPARKS), ASRU sattelite workshop, Taipei, Taiwan, 2023. arXiv:2307.13423.

[62] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust Speech Recognition via Large-Scale Weak Supervision (2022). arXiv:2212.04356.