

# Bootstrapping meaning through listening: Unsupervised learning of spoken sentence embeddings

Jian Zhu<sup>β,α</sup>, Zuoyu Tian<sup>ζ</sup>, Yadong Liu<sup>α</sup>, Cong Zhang<sup>ξ</sup>, Chia-wen Lo<sup>μ</sup>

<sup>β</sup>University of Michigan, Ann Arbor    <sup>α</sup>University of British Columbia

<sup>ζ</sup>Indiana University Bloomington    <sup>ξ</sup>Newcastle University

<sup>μ</sup>Max Planck Institute for Human Cognitive and Brain Sciences

<sup>β</sup>lingjzhu@umich.edu, <sup>ζ</sup>zuoytian@iu.edu

## Abstract

Inducing semantic representations directly from speech signals is a highly challenging task but has many useful applications in speech mining and spoken language understanding. This study tackles the unsupervised learning of semantic representations for spoken utterances. Through converting speech signals into hidden units generated from acoustic unit discovery, we propose WavEmbed, a multimodal sequential autoencoder that predicts hidden units from a dense representation of speech. Secondly, we also propose S-HuBERT to induce meaning through knowledge distillation, in which a sentence embedding model is first trained on hidden units and passes its knowledge to a speech encoder through contrastive learning. The best performing model achieves a moderate correlation (0.5~0.6) with human judgments, without relying on any labels or transcriptions. Furthermore, these models can also be easily extended to leverage textual transcriptions of speech to learn much better speech embeddings that are strongly correlated with human annotations. Our proposed methods are applicable to the development of purely data-driven systems for speech mining, indexing and search.

## 1 Introduction

In Spoken Language Understanding (SLU), a goal is to understand the semantic content of spoken utterances. Traditionally, research in speech processing focus on tasks that process the low-level sensory information in speech, such as automatic speech processing (ASR), under the assumption that language understanding can be handled by NLP modules after speech is transcribed (Wang et al., 2005; De Mori et al., 2008; Serdyuk et al., 2018). Yet speech-based semantic representations allow us to bypass texts in some scenarios, not only simplifying the pipeline and but also beneficial for certain domains without much transcribed data or some languages without writing systems.

For speech processing, the spoken term detection tasks such as keyword detection (e.g., Mamou et al., 2007; Miller et al., 2007; Can and Saraclar, 2011; Wang et al., 2018) and query-by-example search (e.g., Hazen et al., 2009; Parada et al., 2009; Chen et al., 2015) focus on the exact matching of audio terms in speech databases. Yet the speech-to-speech search enabled by spoken sentence embeddings can further expand our capacity to search speech in meaning rather than only in form. This capacity marks a significant advancement in the machine’s capacity to perform speech mining, voice search and indexing and spoken information retrieval (Duquenne et al., 2021).

While learning textual sentence similarity is a classic task in NLP (e.g., Agirre et al., 2012, 2015, 2016; Cer et al., 2017), the task is still relatively unexplored in speech research. The main challenge in learning spoken sentence embeddings lies in the lack of labeled data for supervised learning. Given the costs associated by creating semantic ratings, it is important to explore unsupervised methods to induce semantic representations directly from speech signals.

In this study, we present two approaches to tackle the challenge of inducing semantic representations directly from speech signals without any semantic labeling. The first model, Waveform Embedding Transformer (WavEmbed) (Figure 1), is a multimodal sequential autoencoder that encodes a speech signal into a bottleneck vector and reconstructs a sequence of ‘hidden units’, which are generated using unsupervised acoustic unit discovery. The second model, Sentence HuBERT (S-HuBERT), learns the semantic representation through aligning with a frozen unsupervised text embedding model, which is trained with the hidden units (Figure 2). We make the following contributions.

- We propose simple yet effective unsupervised methods to learn spoken sentence rep-

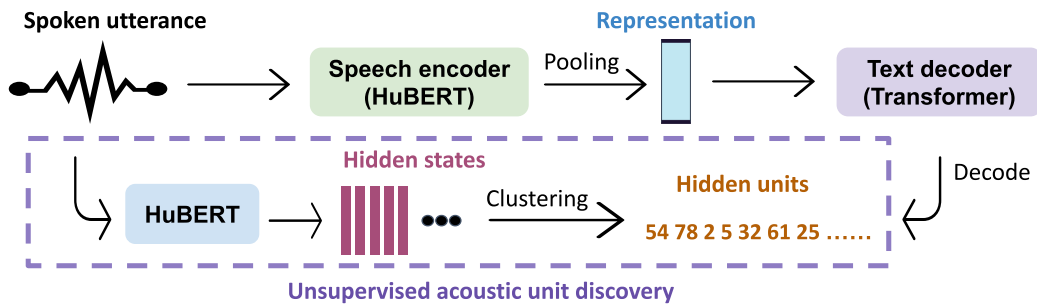


Figure 1: The architecture of WavEmbed. WavEmbed first projects a speech signal into a fixed-dimensional vector representation, and then decodes it back to discrete acoustic units, which are generated through clustering on the hidden states from the sixth layer of the (frozen) pretrained HuBERT model. The learned fixed-dimensional vector encodes semantic information in the latent space. No texts are required in this training loop. However, if text transcripts are available, the decoder targets can also be textual sequences.

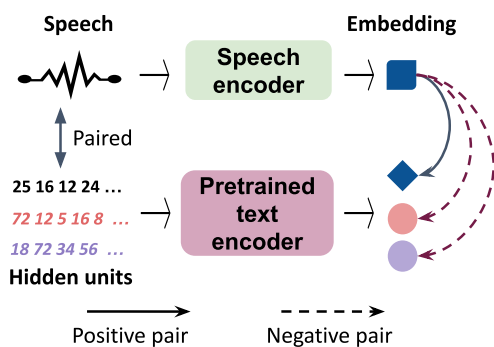


Figure 2: Illustration of S-HuBERT. An unsupervised text embedding model is first trained on either hidden units or textual transcripts. Then it is used as a teacher model to transfer semantic knowledge to a speech encoder through contrastive model distillation.

representations. Our best performing unsupervised model achieves moderate Spearman’s rank correlations (0.5~0.6) with human judgments without relying on any labels or text transcripts.

- Our proposed methods can be easily extended to speech-text pairs to enhance performance. With text transcripts, the performance can further be increased to 0.7~0.8 in terms of Spearman’s correlation. We made extensive comparisons and analyses of model performance under different conditions.
- We have also created a speech dataset for evaluating spoken sentence similarities, which were rated by multiple human raters and encompassed various speech accents to measure the robustness of models.

Our code, data and pretrained checkpoints necessary for replicating the experiments are avail-

able at [https://github.com/lingjzhu/spoken\\_sent\\_embedding](https://github.com/lingjzhu/spoken_sent_embedding).

## 2 Background

**Self-supervised speech modeling** Most speech technologies including ASR and text-to-speech synthesis (TTS) nowadays heavily rely on the availability of text transcripts. Yet such textual resources can sometimes be hard to collect for many languages, some of which might not have writing systems. Many efforts have since been made to explore effective methods to learn speech representations directly from speech signals, such as the ZeroSpeech Workshop (Versteegh et al., 2015; Dunbar et al., 2017, 2019, 2020, 2021).

Recently, large-scale self-supervised models including CPC (Oord et al., 2018), Wav2Vec (Schneider et al., 2019), Wav2Vec2 (Baevski et al., 2020), HuBERT (Hsu et al., 2021) and WavLM (Chen et al., 2021) have learned effective speech representations that can benefit a wide range of downstream speech tasks (Yang et al., 2021). In particular, Hsu et al. (2021) proposed using clustering algorithm to cluster hidden states of HuBERT into hidden units, which were then used to create training masks. These clusters were shown to encode rich phonemic information (Hsu et al., 2021; Baevski et al., 2021). Later it is found that discretizing speech into ‘hidden units’ allows the application of NLP algorithms to process speech via the proxy of these discrete hidden units, without the need of actual textual transcripts (‘textless NLP’) (Lakhotia et al., 2021; Nguyen et al., 2022). This discovery has greatly benefited a variety of tasks, some of which were traditionally not performed with speech, including unsupervised ASR (Baevski et al., 2021), spoken language modeling

(Ao et al., 2022; Hsu et al., 2021; Wu et al., 2022; Nguyen et al., 2022), speech resynthesis (Polyak et al., 2021), spoken language generation (Lakhotia et al., 2021; Kharitonov et al., 2022b), speech-to-speech translation (Lee et al., 2021; Popuri et al., 2022; Lee et al., 2022; Wu et al., 2022), and spoken named entity recognition (Wu et al., 2022). Following these prior studies, our work also falls in the domain of ‘textless NLP’.

**Unsupervised sentence embeddings** Learning semantic sentence embedding has been extensively studied in NLP community, such as Skip-Thought vectors (Kiros et al., 2015), InferSent (Conneau et al., 2017), Universal Sentence Encoder (Cer et al., 2018), and SBERT (Reimers and Gurevych, 2019). Recently, unsupervised sentence embeddings have considerably narrowed the performance gap between unsupervised and supervised methods. Contrastive learning has been utilized to learn a vector space in which semantically similar sentences are close to each other, such as DeCLUTR (Giorgi et al., 2021), SimCSE (Gao et al., 2021), TransEncoder (Liu et al., 2021) and DiffCSE (Chuang et al., 2022). Another approach relies on autoencoders to compress a sentence into a latent vector representation and then reconstruct the original sentence, such as VGVAE (Chen et al., 2019) and TSDAE (Wang et al., 2021).

Most unsupervised methods are based on textual sentences. In speech, current studies tend to center on acoustic word embeddings (e.g., Kamper et al., 2016; Settle and Livescu, 2016; Settle et al., 2017; Holzenberger et al., 2018; Kamper, 2019). Despite the progress, learning sentence-level embeddings for speech still remains under-explored. In SUPERB benchmark for evaluating speech representations (Yang et al., 2021), spoken sentence similarity ranking is not yet listed as a downstream task. In recent works, it has been shown that spoken sentence semantic similarities can be learned via the visually grounded speech models (Merx et al., 2021). Multilingual spoken sentence embeddings can also be learned by using supervised multilingual text models as teacher models (Duquenne et al., 2021; Khurana et al., 2022). These methods more or less relied on labeled data such as speech-image pairs or multilingual sentence pairs. However, we propose unsupervised methods to induce semantic embeddings from speech signals only, and our methods can also utilize textual transcriptions to improve performance if they are available.

### 3 Method

**Task formulation** The current task is to encode spoken utterances into low dimensional dense vectors such that semantically similar utterances are close to each other in the learned latent space. Given a speech signal  $\mathbf{x} \in \mathbb{R}^{1 \times N} = [x_1, x_2, \dots, x_N]$ , our goal is to learn a neural network function  $f_{enc}$  that converts  $\mathbf{x}$  to a fixed-dimensional vector  $\mathbf{z} \in \mathbb{R}^d = f_{enc}(\mathbf{x})$ , such that  $\mathbf{z}$  encodes the semantic content of the original signal  $\mathbf{x}$ . For a certain semantically similar pair  $\{\mathbf{z}, \mathbf{z}^+\}$  and a semantically dissimilar pair  $\{\mathbf{z}, \mathbf{z}^-\}$  (as determined by human raters), it is expected that  $sim(\mathbf{z}, \mathbf{z}^+) > sim(\mathbf{z}, \mathbf{z}^-)$ , where  $sim()$  is a similarity scoring function.

It is further assumed that some forms of transcriptions of the original speech signal exist. Usually, a transcription of  $\mathbf{x}$  take the form of a textual sequence  $\mathbf{y} \in \mathbb{R}^{1 \times M} = [y_1, y_2, \dots, y_M]$ ,  $N > M$ . Such data are sometimes available as most speech datasets for ASR and TTS are organized as pairs of speech and texts. However, in most scenarios the textual transcriptions are not available or too costly to create. In these cases, the transcriptions can be in the form of pseudo-units  $\hat{\mathbf{y}} \in \mathbb{R}^{1 \times L} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_L]$ ,  $N > L$ , which could be generated by an unsupervised system for acoustic unit discovery. During training, these transcripts are used as the targets for the proxy tasks. However, in inference, the pretrained model can directly project speech into semantic embeddings.

#### 3.1 Discretizing speech signals

Acoustic unit discovery refers to the task of segmenting speech signals into discrete word-like or phone-like units (e.g., Lee and Glass, 2012; Lee et al., 2015; Ondel et al., 2016; Kamper, 2019; van Niekerk et al., 2020). Annotating speech signals can sometimes be prohibitively costly for many languages and application domains. Unsupervised discovery of acoustic units can be used as a proxy of transcriptions to train speech systems, if the discovered acoustic units are consistent representations of speech. In our approach, acoustic units are treated as ‘pseudo-texts’ to bootstrap the learning of semantic representations.

We used pretrained speech transformer, HuBERT, to discretize speech signals into ‘hidden units’, which were proposed in Baevski et al. (2021) and Lakhotia et al. (2021). After passing speech signals into HuBERT, the hidden states of the sixth

layer were extracted and a k-means clustering algorithm was applied on the hidden states to quantize them into discrete clusters. The sequence of cluster indexes, after deduplication by merging consecutive same indexes, are the hidden units representing the original speech (see Figure 1). The discrete hidden units remove certain paralinguistic and non-linguistic variations such as speaker voice traits and background noises, so they can be considered a normalized representations of the speech content (though many phonetic variations are still present) (Lee et al., 2021).

We used the `textless-lib` (Kharitonov et al., 2022a) to convert speech signals into discrete hidden units. We selected `hubert-base-ls960` as the base speech encoder and set the number of clusters to 50, 100 and 200. After speech were discretized into sequences of hidden units, sentence-piece tokenizers (Kudo and Richardson, 2018) were trained on them to shorten the sequence length (see Appendix A). There is evidence showing that re-tokenizing hidden units are generally beneficial for language modeling and downstream tasks (Ren et al., 2022; Wu et al., 2022).

### 3.2 S-HuBERT

The first approach, S-HuBERT, is to transfer the knowledge of a well-learned text embedding model to a speech embedding model (Duquenne et al., 2021; Khurana et al., 2022), in which pretrained supervised textual embeddings are adopted as the teacher models and speech models are trained to align with the text embeddings in the same latent space.

Here we also extend this approach to the unsupervised learning domain. The proposed utilizes an unsupervised sentence embedding model with transcriptions, and then transfers the knowledge of a textual sentence embedding model to an acoustic sentence embedding model (S-HuBERT) by leveraging the correspondence between speech and its transcriptions. In the absence of textual transcriptions, the hidden units can be processed as pseudo-texts to induce unsupervised meaning embeddings.

We mainly investigate two approaches to train unsupervised (pseudo-)text embedding models, namely, SimCSE (Gao et al., 2021) and TSDAE (Wang et al., 2021). If these two types of models are trained with hidden units, they are referred to as Hu-SimCSE and Hu-TSDAE respectively, in order

to distinguish them from the text-based models.

**SimCSE** The unsupervised SimCSE (Gao et al., 2021) is a contrastive learning framework for textual sentence embeddings. It takes a sentence as input and uses the same sentence as the target with only the dropout noises. As pretrained transformers such as BERT and RoBERTa apply a dropout mask of 10%, the same sentence will result in slightly different hidden states in multiple passes and can be treated as positive pairs in contrastive learning. We trained SimCSE models to induce sentence meaning from text transcripts before transferring the knowledge to a speech model. For modeling hidden units, we first pretrained a BERT model on hidden units, which were converted from the whole speech corpus. Then Hu-SimCSE was initiated with the pretrained hidden-unit BERT for training.

**TSDAE** Transformer-based Sequential Denoising AutoEncoder (TSDAE) is a denoising encoder-decoder model that encodes a corrupted text sequence into a dense vector and decodes the original text sequence. We trained text-based TSDAE models following as closely as possible the settings specified by Wang et al. (2021). However, slightly different hyperparameters were adopted for Hu-TSDAE. In the original TSDAE, tokens in the input sequence are randomly deleted with a ratio of 0.6. We found that deleting tokens in the input hidden units significantly hurt performance. Instead, using the same uncorrupted sequence of hidden units as both inputs and targets achieved much better performance in our hyperparameter tuning experiments (see Appendix D.1).

**Language modeling on discrete units** Both SimCSE and TSDAE models were initialized with pretrained transformer checkpoints. In addition to publicly available text-based pretrained models, we also pretrained hidden-unit based pretrained transformers. Given a corpus of hidden units converted from raw speech, transformer-based language models were pretrained to learn the statistical regularities in sequences of hidden units. We adopted the same model architecture as BERT (Devlin et al., 2019) (`bert-base-uncased`) and used the masked language modeling task with a masking rate of 15%. However, the next sentence prediction task was discarded, because it was not found to significantly affect the model performance (Liu et al., 2019; Lan et al., 2019).

**Knowledge distillation** We transferred the knowledge from a pretrained textual sentence embedding model (SimCSE or TSDAE) into a speech embedding model through teacher-student training (Duquenne et al., 2021). Here the teacher model was pretrained text embedding model whereas the student model was the pretrained speech model, HuBERT (Hsu et al., 2021).

We used contrastive learning for training S-HuBERT (Sun et al., 2020; Wu et al., 2021; Ye et al., 2022). Given a speech embedding  $z_i$  and its corresponding (pseudo-)text embedding  $\tilde{z}_i^+$  with in-batch negative samples, the InfoNCE loss (Oord et al., 2018) is computed as

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{e^{\text{sim}(z_i, \tilde{z}_i^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(z_i, \tilde{z}_j^+)/\tau}} \quad (1)$$

where  $\tau$  is the temperature parameter and  $\text{sim}()$  is the cosine similarity function  $\text{sim}(z_1, z_2) = z_1^T z_2 / (\|z_1\| \cdot \|z_2\|)$ .  $\tau$  was set to 0.05 in all experiments. In order to keep a large number of negative samples, we maintained a dynamic memory bank of negative samples (He et al., 2020). In each iteration, textual representations in the last mini-batch are enqueued into the memory bank, whereas the oldest textual representations in the bank are dequeued. The text model is frozen throughout training. A comparison of InfoNCE and MSE loss is available at Table 14 in Appendix D.3.

### 3.3 WavEmbed

The WavEmbed is a sequential autoencoder (Vincent et al., 2010; Hill et al., 2016; Wang et al., 2021), which encodes a speech signal  $\mathbf{x}$  into a fixed-dimensional vector  $\mathbf{z}$  and decodes the vector representation  $\mathbf{z}$  using only the encoded vector. The vector  $\mathbf{z}$  is used as the semantic representation. The decoded discrete representations can be actual texts  $\mathbf{y}$  or sequences of hidden acoustic units  $\hat{\mathbf{y}}$ . The proposed method is inspired by the TSDAE (Wang et al., 2021), which learns effective unsupervised sentence embeddings through a denoising encoder-decoder model that encodes a corrupted text sequence into a dense vector and decodes the uncorrupted one. WavEmbed generalizes the original TSDAE to acoustic signals and can learn semantic representations of speech through reconstructing not only the texts but also the hidden acoustic units discovered unsupervisedly.

Yet WavEmbed differs from TSDAE in some aspects. TSDAE’s encoder and decoder components

are all text-based, whereas WavEmbed utilizes a speech encoder. TSDAE relies on the denoising reconstruction as a proxy task, in which the model is trained to recover the original sentence from the embedding of the corrupted sentence (word deletion with a ratio of 0.6). However, WavEmbed reconstructs a discrete sequence from the embedding of a corresponding spoken sentence but no corruptions except the standard dropout is applied to the speech signal. In addition, WavEmbed uses self-attention pooling to pool the encoder hidden states rather than the average pooling in TSDAE, as self-attention pooling is more effective than mean or max pooling for sentence-level speech embeddings (see Khurana et al., 2022, and Table 14 in Appendix D.3).

The encoder  $f_{\text{enc}}$  consists of two parts, a pretrained speech transformer  $f_S$  for speech feature extraction and a self-attention pooling layer for pooling. Let  $\mathbf{H} \in \mathbb{R}^{T \times d} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_T]$  be the hidden states of a speech transformer model  $f_S$  given a speech signal  $\mathbf{x}$ . The self-attention pooling operation (Safari et al., 2020) can be computed as:

$$\mathbf{H} = f_S(\mathbf{x}) \quad (2)$$

$$\mathbf{z} = \text{Softmax}(\mathbf{W}\mathbf{H}^T)\mathbf{H} \quad (3)$$

where  $\mathbf{W} \in \mathbb{R}^d$  is a learnable parameter during training. Given a semantic representation  $\mathbf{z}$  of speech signal  $\mathbf{x}$ , the autoregressive decoder  $f_{\text{dec}}$  predicts the hidden units  $\hat{\mathbf{y}}$  that correspond to the content of the speech signal  $\mathbf{x}$ .

$$\hat{\mathbf{y}} = f_{\text{dec}}(\mathbf{z}) \quad (4)$$

The encoder-decoder model is trained with the standard negative log likelihood loss.

$$\begin{aligned} \mathcal{L} &= -\sum_{l=1}^L \log P(\hat{y}_l | \mathbf{z}, \hat{y}_{l-1}, \dots, \hat{y}_1) \\ &= -\sum_{l=1}^L \log P(\hat{y}_l | f_{\text{enc}}(\mathbf{x}), \hat{y}_{l-1}, \dots, \hat{y}_1) \end{aligned} \quad (5)$$

The WavEmbed is trained to predict discrete acoustic units  $\hat{\mathbf{y}}$  based on the speech signals  $\mathbf{x}$ . However, when textual transcripts for speech signals are available, the prediction targets can also be replaced with textual sequences  $\mathbf{y}$  to enhance the learning of semantic content. Once the model is trained, the decoder is discarded, leaving only the speech encoder for extracting the semantic embeddings.

## 4 Data

**Training data** We used the training set of the 1700-hour English subset of Common Voice (Ardila et al., 2020) for training all speech encoders, which can be assessed through the HuggingFace Datasets library (Lhoest et al., 2021). Speech signals were all downsampled from 44.1kHz to 16kHz to be compatible with HuBERT. To avoid memory issues, we limited the maximal length of speech to be 10 seconds, which only excluded less than 1% of data.

### Common Voice spoken sentence similarity

Given that there is few evaluation data for spoken sentence similarities, we created the Common Voice Spoken Sentence Similarity (CVS) dataset based on the test set of the English subset of Common Voice, which contains a wide range of accents. Following the criteria of annotating STS test data (Agirre et al., 2016), four proficient English users independently rated the similarity of 1149 sentence pairs on a scale of 0 to 5 (not similar to most similar). The average Spearman’s rank correlation between the four raters reached 0.937. The annotated data was randomly partitioned into the development set (40% , 459 sentence pairs) and the test set (60%, 690 sentence pairs). Score distributions and additional details are provided in Appendix B.

**Spoken STS** Additionally, we used the Spoken STS data collected by Merx et al. (2021) for evaluation, which is publicly available online<sup>1</sup>. The Spoken STS include synthetic and natural recordings of sentences in the Semantic Textual Similarity (STS) dataset, covering benchmarks from STS12 to STS16 (Agirre et al., 2012, 2013, 2014, 2015, 2016). The STS datasets include paired sentences with human labelled similarity scores. The synthetic speech data contain all sentence pairs in STS datasets, which were synthesized via Google’s Wavenet using six voices (three males and three females) with a US accent. The natural speech data contains 638 pairs of randomly selected sentences evenly distributed across the STS datasets. These sentences were produced by four speaker (2 females and 2 males) with a North American accent. The synthetic and natural speech was only used as the test set. None of our models had seen any STS sentences during training. During evaluation, for each sentence pair, we averaged the similarity

<sup>1</sup><https://easy.dans.knaw.nl/ui/datasets/id/easy-dataset:237533>

scores for all possible combinations of speakers to derive the final score.

## 5 Experiments

Three different types of transcriptions were compared in this study, namely, (ground truth) text transcripts, text transcripts recognized by an ASR model, and hidden units. ASR transcripts were additionally considered because for some languages ASR systems can be used to assist the development of speech retrieval. We used the `hubert-large-ls960-ft` (Hsu et al., 2021) via HuggingFace Hub to transcribe the Common Voice English subset to texts.

For WavEmbed models, we trained multiple variants based on texts, ASR transcriptions and hidden units. The audio encoder was initiated with the `hubert-base-ls960` checkpoint. The decoder was also initiated with pretrained weights according to their outputs, including `bert-base-uncased`, `gpt2-base`, `roberta-base`, as well as pretrained transformers for hidden units on different vocabulary sizes.

For S-HuBERT, we first trained SimCSE and TS-DAE on Common Voice texts, ASR transcriptions and hidden units, all of which were also initiated with pretrained weights according to respective inputs. Specifically for SimCSE, additionally four existing model checkpoints from Gao et al. (2021) were adopted without any training nor fine-tuning. Then the HuBERT base model with a self-attention pooling layer and a projection head on top was fine-tuned to aligned with the semantic embeddings of the text models in the same latent space. For Hu-SimCSE and Hu-TSDAE, we only selected the best performing models to perform model distillation.

Given that there are several different models implemented here, detailed hyperparameter settings are given at Appendix C. The implementation of transformers was modified based on the `transformers` library (Wolf et al., 2020) and the `sentence_transformer` library (Reimers and Gurevych, 2019). All our models were trained on a single A40 GPU with 48GB of memory. Unless otherwise stated, all models were initialized with pretrained models, as pretrained weights were found to improve downstream performance (see Rothe et al., 2020, and Table 15 in Appendix D.4).

Models	Synthetic speech					Natural Speech		
	STS12	STS13	STS14	STS15	STS16	STS	CVS-dev	CVS-test
<i>A. Unsupervised textual sentence embeddings (evaluated on texts)</i>								
<i>TSDAE</i>	<i>55.2</i>	<i>67.4</i>	<i>62.4</i>	<i>74.3</i>	<i>73.0</i>	-	-	-
TSDAE-CV-text	51.3	67.3	60.6	71.5	74.9	<b>65.5</b>	<b>90.7</b>	<b>89.5</b>
TSDAE-ASR-text	50.8	67.9	58.5	70.2	74.6	64.6	90.0	88.5
<i>B. Unsupervised spoken sentence embeddings (trained with texts)</i>								
WavEmbed (BERT decoder)	48.8	53.8	52.7	66.1	63.6	55.4	<b>78.4</b>	78.7
WavEmbed (RoBERTa decoder)	<b>52.1</b>	<b>56.1</b>	<b>54.7</b>	<b>68.4</b>	<b>66.1</b>	<b>58.6</b>	78.1	<b>79.3</b>
<i>C. Unsupervised spoken sentence embeddings (trained with ASR transcripts)</i>								
WavEmbed (BERT decoder)	<b>52.1</b>	<b>55.9</b>	<b>55.6</b>	<b>68.0</b>	<b>66.2</b>	<b>57.9</b>	77.5	<b>79.5</b>
WavEmbed (RoBERTa decoder)	49.3	53.0	52.2	64.9	64.0	54.6	<b>77.9</b>	78.1
<i>D. Unsupervised spoken sentence embeddings (trained with hidden units)</i>								
HuBERT - Avg. Last hidden states	<b>47.3</b>	49.5	44.1	58.3	49.2	33.2	1.5	4.4
WavEmbed - 50 C	44.0	51.5	47.6	62.7	56.3	49.8	62.6	59.8
WavEmbed - 100 C	45.3	<b>53.4</b>	<b>50.3</b>	63.3	<b>59.7</b>	<b>50.0</b>	65.6	<b>61.2</b>
WavEmbed - 200 C	43.8	49.9	48.7	<b>63.5</b>	58.4	47.4	64.8	59.2
WavEmbed - 100 C - 1000 V	45.2	49.7	47.1	61.8	58.1	45.7	67.8	60.9
WavEmbed - 100 C - 3000 V	44.3	50.1	47.1	61.4	57.3	45.5	68.4	61.0
WavEmbed - 100 C - 5000 V	43.8	48.9	45.9	60.9	57.5	44.8	67.9	61.0
WavEmbed - 100 C - 8000 V	43.1	48.8	45.7	60.3	58.8	43.5	<b>68.9</b>	61.1
WavEmbed - 100 C - 12000 V	43.3	47.6	44.4	60.3	57.5	42.8	66.0	60.2

Table 1: Evaluations based on the Spearman’s rank correlation ( $\times 100$ ). The italicized blue fonts suggest that results were directly taken from Wang et al. (2021). The bold numbers indicate the best performance within each subsection. Note. C  $\rightarrow$  number of clusters; V  $\rightarrow$  size of vocabulary; CV-text  $\rightarrow$  Common Voice texts.

## 6 Results and discussions

In general, we show that semantic representations can be learned directly from acoustic signals through the proxy of hidden units. Here we summarize the main results and the ablation analyses (see Appendix D for additional results).

**The self-reconstruction approach is more effective in modeling hidden units than contrastive learning.** In general, the WavEmbed method is far more effective than S-HuBERT when transcriptions are not available. The WavEmbed models with hidden-unit targets all achieve moderate correlation with human judgments (Table 1 D), as compared to the weak correlations with humans for the S-HuBERT from Hu-SimCSE (Table 2 E). Even for directly modeling on hidden units without audio, the self-reconstruction based Hu-TSDAE far outperform the contrastive learning based Hu-SimCSE in terms of Spearman correlations. Gao et al. (2021) noted that SimCSE depends heavily on a good pretrained LM. Yet our hidden unit-based BERT is trained on a relatively small corpus without a sentence-level task, which presumably is the

reason why the performance of Hu-SimCSE is sub-optimal.

**Re-tokenizing hidden units does not always help.** Ren et al. (2022) and Wu et al. (2022) found that re-tokenizing hidden units can improve the performance of downstream tasks, yet our findings here are inconclusive. For WavEmbed (Table 1 D), re-tokenizing hidden units does not improve model performance. However, for Hu-SimCSE and Hu-TSDAE, the results are almost contrary. Using sentence piece units learned from the raw hidden units tends to improve Spearman correlation in these hidden unit-based models (see Table 12 and 13 in Appendix D.2).

**Supervised training with texts is the most effective method.** This holds even if supervised data are in another modality. In Table 2 C, it is apparent that transferring weights of supervised embedding models to S-HuBERT achieves the best evaluation performance across all evaluation benchmarks. This result may suggest that, given well trained supervised text models, the need for created labeled speech data might not be very critical.

Models	Synthetic speech					Natural Speech		
	STS12	STS13	STS14	STS15	STS16	STS	CVS-dev	CVS-test
<i>A. Supervised text embeddings</i>								
<i>SimCSE-sup-BERT</i>	<i>75.3</i>	<i>84.7</i>	<i>80.2</i>	<i>85.4</i>	<i>80.8</i>	<b>81.4</b>	<b>92.9</b>	<b>92.3</b>
<i>SimCSE-sup-RoBERTa</i>	<i>76.5</i>	<i>85.2</i>	<i>81.0</i>	<i>86.0</i>	<i>82.6</i>	80.2	91.2	90.9
<i>B. Unsupervised text embeddings</i>								
<i>SimCSE-unsup-BERT</i>	<i>68.4</i>	<i>82.4</i>	<i>74.4</i>	<i>80.9</i>	<i>78.6</i>	77.6	89.6	<b>89.0</b>
<i>SimCSE-unsup-RoBERTa</i>	<b>70.2</b>	<b>81.8</b>	<b>73.2</b>	<b>81.4</b>	<b>80.7</b>	<b>77.7</b>	88.0	87.3
SimCSE-unsup-BERT-CV-text	61.2	77.9	69.9	77.6	76.8	73.1	<b>89.7</b>	87.7
SimCSE-unsup-BERT-ASR-text	57.7	69.0	64.2	74.5	68.7	66.5	85.5	86.1
<i>C. Supervised text embeddings → spoken sentence embeddings</i>								
S-HuBERT ( <i>SimCSE-sup-BERT</i> )	57.3	63.9	62.6	72.6	67.6	64.5	81.1	79.7
S-HuBERT ( <i>SimCSE-sup-RoBERTa</i> )	<b>61.5</b>	<b>68.0</b>	<b>65.1</b>	<b>75.2</b>	<b>70.8</b>	<b>69.0</b>	<b>81.3</b>	<b>81.9</b>
<i>D. Unsupervised text embeddings → spoken sentence embeddings</i>								
S-HuBERT ( <i>SimCSE-unsup-BERT</i> )	53.0	59.6	58.5	67.8	65.4	60.5	<b>80.0</b>	80.8
S-HuBERT ( <i>SimCSE-unsup-RoBERTa</i> )	<b>60.8</b>	<b>70.8</b>	<b>65.9</b>	<b>76.5</b>	<b>72.8</b>	<b>70.1</b>	79.8	<b>81.6</b>
S-HuBERT (TSDAE-CV-text)	38.6	43.6	45.6	55.4	54.6	45.4	77.8	76.2
S-HuBERT (TSDAE-ASR-text)	43.1	47.0	48.0	58.7	55.9	48.0	79.5	75.6
<i>E. Unsupervised hidden-unit embeddings → spoken sentence embeddings</i>								
S-HuBERT (Hu-SimCSE-100 C-8000 V)	37.4	36.8	32.3	44.5	39.6	26.6	35.2	34.6
S-HuBERT (Hu-TSDAE-100 C-1000 V)	<b>47.8</b>	<b>53.8</b>	<b>51.6</b>	<b>63.8</b>	<b>60.4</b>	<b>50.3</b>	<b>61.5</b>	<b>55.3</b>

Table 2: Evaluations based on the Spearman’s rank correlation ( $\times 100$ ). The italicized blue font suggests that results or checkpoints were directly taken from Gao et al. (2021). Models in parentheses are teacher models whereas models outside parentheses are student models. The bold numbers indicate the best performance within each subsection. Note. C  $\rightarrow$  number of clusters; V  $\rightarrow$  vocabulary size; CV-text  $\rightarrow$  Common Voice texts.

### Training with text transcripts are better than with pseudo-units.

In both Table 1 and 2, a comparison between text-based and hidden unit-based models clearly shows that text-based WavEmbed or S-HuBERT outperformed their hidden unit-based counterparts by a very large margin (ranging from 5~40 in absolute points). Compared with texts, hidden units are noisy representations of the speech content, such that speaker identities were still detectable from hidden units converted from speech (Kharitonov et al., 2022a). Such content-irrelevant information introduces many noises into model training, therefore leading to sub-optimal performance. WavEmbed and Hu-TSDAE are more robust to local noises, as they force the speech embedding to capture the global property of the sentence through self-reconstruction (Wang et al., 2021).

In order to measure the quality of the learned embeddings, we compute the metrics for alignment and uniformity proposed by Wang and Isola (2020), which are as follows.

$$\ell_{align} \triangleq \mathbb{E}_{x, x^+ \sim P_{pos}} \|f(x) - f(x^+)\|^2 \quad (6)$$

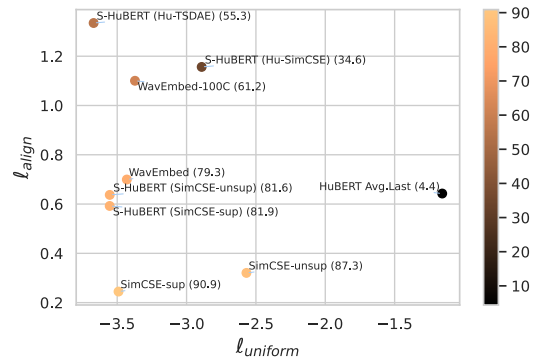


Figure 3:  $\ell_{align} - \ell_{uniform}$  plot. All SimCSE models and the WavEmbed decoder were based on RoBERTa-base. The bracketed number indicates Spearman correlation ( $\times 100$ ). Low  $\ell_{align}$  and low  $\ell_{uniform}$  are desirable qualities of embeddings.

$$\ell_{uniform} \triangleq \log \mathbb{E}_{x, y \stackrel{i.i.d.}{\sim} P_{data}} e^{-2\|f(x) - f(y)\|^2} \quad (7)$$

where  $x^+$  refers to a positive sample for  $x$  and  $P_{pos}$  is the set of positive samples. The analysis was based on the CVS test set. Sentence pairs with a score  $\geq 4$  were treated as positive pairs for



Equation 6. All possible combinations of sentence pairs were used to compute Equation 7. Figure 3 shows the  $\ell_{align}$ - $\ell_{uniform}$  plot for different models (Wang and Isola, 2020; Gao et al., 2021). Lower  $\ell_{align}$  suggests that positive instances are closer to each other in the embedding space, whereas lower  $\ell_{uniform}$  indicates that random representations are more uniformly scattered, both of which imply a good embedding space. Text-based SimCSE models have both low  $\ell_{align}$  and low  $\ell_{uniform}$ . For speech based WavEmbed and S-HuBERT, they tend to have low  $\ell_{uniform}$  but high  $\ell_{align}$ , so distances between positive samples are relatively far from each other. The average last hidden states of HuBERT has a very low  $\ell_{uniform}$ , suggesting that different embeddings are not well separated. It seems to perform well on STS (Table 1 D), because STS datasets were recorded from limited speakers and it might be using similarities of voice traits (see the first row of Table 6), which also explains why this method fails on the CVS dataset that contains a diversity of speakers. Generally speaking, for speech models not trained with texts, spoken utterances from the same speaker are always less separated than utterances from different speakers (see Figure 6 and 7 in Appendix E).

**ASR transcripts can be a good proxy for ground truth transcripts.** Modern ASR systems can achieve very low word error rate (WER) on certain high resource languages. Our experiments with recognized texts of Common Voice audio indicates that they are slightly inferior to the ground truth transcriptions in performance but the gap is pretty narrow (see Table 1. B and C; Table 2. B). In the absence of texts, it is possible to utilize ASR systems to quickly create training data for speech retrieval models. This option currently might only work well for a few high resource languages such as English, Chinese and Spanish. The recent advancement in unsupervised ASR (Baeovski et al., 2021; Liu et al., 2022) or multilingual ASR (Li et al., 2022), however, might make it possible to extend the proposed methods to more low resource languages.

**WavEmbed and S-HuBERT are useful in different scenarios.** Our results have shown that WavEmbed works better when only speech signals are available, whereas S-HuBERT can leverage more powerful text models if transcripts are provided. Furthermore, since S-HuBERT aligns

speech embeddings with text embeddings in a common semantic space, it is also capable of performing cross-modal semantic retrieval between speech and texts.

## 7 Conclusions and future directions

In summary, we investigated two unsupervised approaches, WavEmbed and S-HuBERT, to induce semantic embeddings from speech signals, which are significantly correlated with human judgements of semantic similarities. We experimented these approaches with different settings and found that performance can be improved through leveraging transcribed texts, even when the transcriptions were from slightly inaccurate ASR systems. We believe that our study is an essential step towards direct speech-to-speech search and simplifying the pipeline of SLU.

Currently, the proposed methods do not depend on any specific features of certain languages, and therefore should also work for other languages. However, it remains unclear how language specific properties can affect the actual performance. In the future, we plan to extend our methods to other languages than English.

Generally speaking, spoken sentence similarity task is more challenging than its text counterparts, because speech has far more variability than texts, which can be caused by various physiological, psychological and social factors (e.g., vocal tract morphology, gender, accents and emotional states). It remains to be explored how these variabilities (some of which are related to expressing meaning) can be taken into account in computational models. Speech conveys information beyond its textual content. For example, prosody can reverse the meaning of a sentence from being non-ironic to ironic. While we only focus on the semantic similarity of textual content in speech, we also envision that spoken sentence similarity tasks in the future should include non-textual features such as prosody and emotion.

## Acknowledgements

We thank Professor David Jurgens at University of Michigan for his comments on the early versions of this manuscript. We are grateful for Danny Merckx for sharing the Spoken STS dataset. We would also like to thank three anonymous reviewers and the area chair for their suggestions, which helped improve this article greatly.

## Limitations

Our study is still limited in several ways. First, there is still a large gap between unsupervised learning from speech and unsupervised methods with texts. Part of the reason is that the performance is bounded by the relatively inaccurate acoustic unit discovery method that still retains many paralinguistic information such as voice traits (Lee et al., 2021). One potential future direction to overcome this limitation is to develop better unsupervised acoustic unit discovery systems.

Secondly, our proposed methods still require a large amount of data to pre-train and train, making it still difficult to extend this method to low resources. Large-scale pretrained models such as HuBERT, BERT and RoBERTa are only available for a limited number of languages that can offer huge data for pretraining. It is paramount to investigate data efficient methods that can work on low resource languages. Multilingual pretrained models such as XLS-R (Babu et al., 2021) might be a potential base model for low resource languages, but more work should be done.

Given that transformers have a complexity of  $\log(n^2)$ , it might not be the optimal model for processing speech, which often have very long sequence length. In our experiments, we also notice that training and finetuning HuBERT on speech is very inefficient. An important future direction is to search for better model architectures that are optimized for processing long sequences of speech.

## References

- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. [SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability](#). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.
- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. [SemEval-2014 task 10: Multilingual semantic textual similarity](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.
- Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. [SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.
- Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. [SemEval-2012 task 6: A pilot on semantic textual similarity](#). In *\*SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.
- Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. [\\*SEM 2013 shared task: Semantic textual similarity](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Junyi Ao, Ziqiang Zhang, Long Zhou, Shujie Liu, Haizhou Li, Tom Ko, Lirong Dai, Jinyu Li, Yao Qian, and Furu Wei. 2022. Pre-training transformer decoder for end-to-end asr model with unpaired speech data. *arXiv preprint arXiv:2203.17113*.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. [Common voice: A massively-multilingual speech corpus](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.
- Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhota, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.
- Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, 34:27826–27839.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. Wav2Vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33.
- Doğan Can and Murat Saraclar. 2011. [Lattice indexing for spoken term detection](#). *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2338–2347.

- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Guoguo Chen, Carolina Parada, and Tara N Sainath. 2015. Query-by-example keyword spotting using long short-term memory networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5236–5240. IEEE.
- Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. [A multi-task approach for disentangling syntax and semantics in sentence representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2021. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*.
- Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. [Supervised learning of universal sentence representations from natural language inference data](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
- Renato De Mori, Frédéric Bechet, Dilek Hakkani-Tur, Michael McTear, Giuseppe Riccardi, and Gokhan Tur. 2008. Spoken language understanding. *IEEE Signal Processing Magazine*, 25(3):50–58.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W. Black, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2019. [The Zero Resource Speech Challenge 2019: TTS Without T](#). In *Proc. Interspeech 2019*, pages 1088–1092.
- Ewan Dunbar, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux. 2021. [The Zero Resource Speech Challenge 2021: Spoken Language Modelling](#). In *Proc. Interspeech 2021*, pages 1574–1578.
- Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux. 2017. The zero resource speech challenge 2017. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 323–330. IEEE.
- Ewan Dunbar, Julien Karadayi, Mathieu Bernard, Xuan-Nga Cao, Robin Algayres, Lucas Ondel, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2020. [The Zero Resource Speech Challenge 2020: Discovering Discrete Subword and Word Units](#). In *Proc. Interspeech 2020*, pages 4831–4835.
- Paul-Ambroise Duquenne, Hongyu Gong, and Holger Schwenk. 2021. Multimodal and multilingual embeddings for large-scale speech mining. *Advances in Neural Information Processing Systems*, 34.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. [DeCLUTR: Deep contrastive learning for unsupervised textual representations](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.
- Timothy J Hazen, Wade Shen, and Christopher White. 2009. Query-by-example spoken term detection using phonetic posteriorgram templates. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 421–426. IEEE.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. [Learning distributed representations of sentences from unlabelled data](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.
- Nils Holzenberger, Mingxing Du, Julien Karadayi, Rachid Riad, and Emmanuel Dupoux. 2018. Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments. In *Proc. Interspeech*, pages 2683–2687.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhota, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*.
- Herman Kamper. 2019. Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6535–6539. IEEE.
- Herman Kamper, Weiran Wang, and Karen Livescu. 2016. Deep convolutional acoustic word embeddings using word-pair side information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4950–4954. IEEE.
- Eugene Kharitonov, Jade Copet, Kushal Lakhota, Tu Anh Nguyen, Paden Tomasello, Ann Lee, Ali Elkahky, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, et al. 2022a. textless-lib: a library for textless spoken language processing. *arXiv preprint arXiv:2202.07359*.
- Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhota, Tu Anh Nguyen, Morgane Riviere, Abdelrahman Mohamed, Emmanuel Dupoux, and Wei-Ning Hsu. 2022b. [Text-free prosody-aware generative spoken language modeling](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8666–8681, Dublin, Ireland. Association for Computational Linguistics.
- Sameer Khurana, Antoine Laurent, and James Glass. 2022. Samu-xlsr: Semantically-aligned multimodal utterance-level cross-lingual speech representation. *arXiv preprint arXiv:2205.08180*.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Kushal Lakhota, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. [On generative spoken language modeling from raw audio](#). *Transactions of the Association for Computational Linguistics*, 9:1336–1354.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Sravya Popuri, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, and Wei-Ning Hsu. 2022. [Direct speech-to-speech translation with discrete units](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3327–3339, Dublin, Ireland. Association for Computational Linguistics.
- Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. 2021. Textless speech-to-speech translation on real data. *arXiv preprint arXiv:2112.08352*.
- Chia-ying Lee and James Glass. 2012. [A nonparametric Bayesian approach to acoustic model discovery](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea. Association for Computational Linguistics.
- Chia-ying Lee, Timothy J. O’Donnell, and James Glass. 2015. [Unsupervised lexicon discovery from acoustic input](#). *Transactions of the Association for Computational Linguistics*, 3:389–403.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Guntjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. [Datasets: A community library for natural language processing](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xinjian Li, Florian Metze, David R. Mortensen, Alan W Black, and Shinji Watanabe. 2022. [ASR2K: Speech](#)

- [Recognition for Around 2000 Languages without Audio](#). In *Proc. Interspeech 2022*, pages 4885–4889.
- Alexander H Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2022. Towards end-to-end unsupervised speech recognition. *arXiv preprint arXiv:2204.02492*.
- Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2021. Trans-encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. In *International Conference on Learning Representations*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. 2007. Vocabulary independent spoken term detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–622.
- Danny Merckx, Stefan L. Frank, and Mirjam Ernestus. 2021. [Semantic Sentence Similarity: Size does not Always Matter](#). In *Proc. Interspeech 2021*, pages 4393–4397.
- David R. H. Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, and Herbert Gish. 2007. [Rapid and accurate spoken term detection](#). In *Proc. Interspeech 2007*, pages 314–317.
- Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux. 2022. Are discrete units necessary for spoken language modeling? *arXiv preprint arXiv:2203.05936*.
- Lucas Ondel, Lukáš Burget, and Jan Černocký. 2016. Variational inference for acoustic unit discovery. *Proceedia Computer Science*, 81:80–86.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran. 2009. Query-by-example spoken term detection for oov terms. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 404–409. IEEE.
- Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharonov, Kushal Lakhota, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. In *INTERSPEECH 2021-Annual Conference of the International Speech Communication Association*.
- Sravya Popuri, Peng-Jen Chen, Changhan Wang, Juan Pino, Yossi Adi, Jiatao Gu, Wei-Ning Hsu, and Ann Lee. 2022. Enhanced direct speech-to-speech translation using self-supervised pre-training and data augmentation. *arXiv preprint arXiv:2204.02967*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Shuo Ren, Shujie Liu, Yu Wu, Long Zhou, and Furu Wei. 2022. Speech pre-training with acoustic piece. *arXiv preprint arXiv:2204.03240*.
- Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. [Leveraging pre-trained checkpoints for sequence generation tasks](#). *Transactions of the Association for Computational Linguistics*, 8:264–280.
- Pooyan Safari, Miquel India, and Javier Hernando. 2020. [Self-Attention Encoding and Pooling for Speaker Recognition](#). In *Proc. Interspeech 2020*, pages 941–945.
- Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. [wav2vec: Unsupervised Pre-Training for Speech Recognition](#). In *Proc. Interspeech 2019*, pages 3465–3469.
- Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.
- Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu. 2017. [Query-by-example search with discriminative neural acoustic word embeddings](#). In *Proc. Interspeech 2017*, pages 2874–2878.
- Shane Settle and Karen Livescu. 2016. Discriminative acoustic word embeddings: Teccurrent neural network-based approaches. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 503–510.
- Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020. [Contrastive distillation on intermediate representations for language model compression](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 498–508, Online. Association for Computational Linguistics.
- Benjamin van Niekerk, Leanne Nortje, and Herman Kamper. 2020. [Vector-Quantized Neural Networks for Acoustic Unit Discovery in the ZeroSpeech 2020 Challenge](#). In *Proc. Interspeech 2020*, pages 4836–4840.
- Maarten Versteegh, Roland Thiollière, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. 2015. [The zero resource speech](#)

challenge 2015. In *Proc. Interspeech 2015*, pages 3169–3173.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. **TSDAE: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.** In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding. *IEEE Signal Processing Magazine*, 22(5):16–31.

Yu-Hsuan Wang, Hung-yi Lee, and Lin-shan Lee. 2018. Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6269–6273. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. **Transformers: State-of-the-art natural language processing.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Felix Wu, Kwangyoung Kim, Shinji Watanabe, Kyu Han, Ryan McDonald, Kilian Q Weinberger, and Yoav Artzi. 2022. Wav2seq: Pre-training speech-to-text encoder-decoder models using pseudo languages. *arXiv preprint arXiv:2205.01086*.

Xing Wu, Chaochen Gao, Jue Wang, Liangjun Zang, Zhongyuan Wang, and Songlin Hu. 2021. Disco: Effective knowledge distillation for contrastive learning of sentence embeddings. *arXiv preprint arXiv:2112.05638*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, and Cheng-I Jeff Lai et al. 2021. **SUPERB: Speech Processing Universal PERFORMANCE Benchmark.** In *Proc. Interspeech 2021*, pages 1194–1198.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. *arXiv preprint arXiv:2205.02444*.

## A Sentence-piece Tokenizers

All tokenizers for hidden units were trained with the open-source package `SentencePiece`<sup>2</sup>. They were trained on the whole Common Voice English training set, which was transcribed into hidden units with cluster numbers of 50, 100, and 200. The vocabulary sizes were {1000, 3000, 5000, 8000, 12000, 20000} for each cluster number. We used the arguments in Table 3, otherwise default arguments were kept.

Arguments	Value
<code>model_type</code>	BPE
<code>hard_vocab_limit</code>	true
<code>split_by_whitespace</code>	false
<code>pad_id</code>	0
<code>bos_id</code>	1
<code>eos_id</code>	2
<code>unk_id</code>	3
<code>bos_piece</code>	[CLS]
<code>eos_piece</code>	[SEP]
<code>unk_piece</code>	[UNK]
<code>pad_piece</code>	[PAD]
<code>user_defined_symbols</code>	[MASK]

Table 3: Arguments for training sentence-piece tokenizers on hidden units.

## B Common Voice Sentence Similarity

The Common Voice Spoken Sentence Similarity dataset was created based on the test set of the English subset of Common Voice. To get the similarity of every pair of sentences in the original test set, we used two pretrained SimCSE models, that is, `sup-simcse-roberta-large` and `sup-simcse-bert-large-uncased`<sup>3</sup>.

The similarity scores were averaged across two models, and divided into ten equal intervals between 0 and 1. Then 120 sentence pairs were randomly sampled from each interval, resulted in 1200 sentence pairs. Some pairs were removed due to low quality, leaving 1149 pairs in the final data. Four proficient English users independently rated the sentence pairs based on the textual content on a scale from 0-5, where 0 implies completely irrelevant and 5 means completely the same. We

<sup>2</sup><https://github.com/google/sentencepiece>

<sup>3</sup><https://github.com/princeton-nlp/SimCSE>

	rater_A	rater_B	rater_C	rater_D
rater_A	1	0.94	0.85	0.92
rater_B	0.94	1	0.87	0.96
rater_C	0.85	0.87	1	0.92
rater_D	0.92	0.96	0.92	1

Figure 4: Pairwise Spearman’s rank correlations between all four human raters. Sentence similarity ratings are highly consistent across human raters.

tried to make this dataset similar to those in STS SemEval tasks and followed their annotation criteria. The instructions were the same as the Figure 2 in Agirre et al. (2012) and sample rated sentences were from the Table 1 of Agirre et al. (2016). The final similarity scores were averaged across four human raters. The rated data was partitioned into the development set and the test set with a 40/60 split. The distributions of similarity scores in these two sets are provided in Figure 5.

## C Hyperparameters

### C.1 BERT on hidden units

We trained 15 BERT models on hidden units from scratch. The detailed hyperparameters are given in Table 4, which were the same for all models. The number of clusters ranged from {50, 100, 200}. A BERT model was trained for each number of clusters. For hidden units of 100 and 200 clusters, they were further tokenized into vocabulary sizes of {1000, 3000, 5000, 8000, 12000, 20000} respectively. Then a BERT model was trained for each vocabulary size, resulting in 15 in total. Each took about 3.5~6 hours to train.

### C.2 WavEmbed

For WavEmbed models, we apply the same hyperparameters in Table 5.

### C.3 Hu-TSDAE

Table 6 presents the hyperparameters for all Hu-TSDAE models, which were trained for the following vocabulary sizes {50, 100, 200, 1000, 3000, 5000, 8000, 12000}. For vocabulary sizes larger than 200, they were learned from the hidden units with 100 clusters.

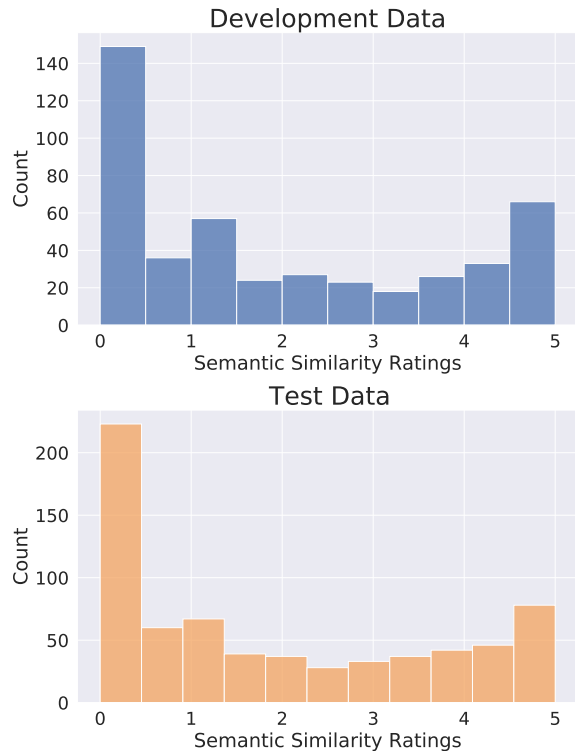


Figure 5: Distribution of ratings in Common Voice Spoken Sentence Similarity dataset.

Hyperparameters	Value
Epoch	5
Total update steps	~8k
Optimizer	AdamW
Initial learning rate	1e-4
Effective batch size	512
FP 16	True
Masking probability	15%
Weight Initialization	random init.
Training task	MLM
Loss	Cross-entropy loss
Checkpointing	Keep last
GPU	A40
Time	3.5~6 hours

Table 4: Hyperparameters for training BERT on hidden units.

Hyperparameters	Value
Epoch	1
Total update steps	~13.5k
Optimizer	AdamW
Initial learning rate	5e-4
Effective batch size	64
FP 16	False
Encoder initialization	HuBERT
Decoder initialization	Pretrained weights
Training task	Autoregressive LM
Loss	Cross-entropy loss
Checkpointing	Keep best
GPU	A40
Time	~22 hours

Table 5: Hyperparameters for training WavEmbed models.

Hyperparameters	Value
Epoch	3
Total update steps	~10.4k
Optimizer	AdamW
Initial learning rate	3e-5
Effective batch size	256
FP 16	False
Encoder initialization	Pretrained weights
Decoder initialization	Same as encoder
Training task	Autoregressive LM
Loss	Cross-entropy loss
Checkpointing	Keep best
GPU	A40
Time	6~10.5 hours

Table 6: Hyperparameters for training Hu-TSDAE models.

Hyperparameters	Value
Epoch	1
Optimizer	AdamW
FP 16	False
Encoder initialization	Pretrained weights
Training task	Contrastive learning
Loss	InfoNCE loss
Checkpointing	Keep best
Early stopping	true
GPU	A40
Time	5~28 minutes

Table 7: Hyperparameters for training Hu-SimCSE models.

#### C.4 Hu-SimCSE

The general configurations were specified in Table 7 and 8. For Hu-SimCSE models, the training was highly unstable. The loss dropped to a very low value in about 20 steps. Most models usually reached the best validation performance in less than 200 iterations, and then the performance began to drop severely. So early stopping was applied when the validation performance dropped for more than 80 steps. In our experiments, Hu-SimCSE models was highly sensitive to hyperparameters, so we tuned hyperparameters by carrying out grid search if effective batch size  $\in \{64, 128, 256, 512\}$ , number of in-batch negative samples  $\in \{63, 127\}$  and learning rate  $\in \{5e-5, 1e-5\}$ .  $\tau$  was set to 0.05 for all experiments.

#### C.5 S-HuBERT

The hyperparameters for S-HuBERT were shown in Table 9. The text encoder was pretrained text embeddings including Hu-SimCSE, Hu-TSDAE, SimCSE, and TSDAE. The weights of text encoder were frozen throughout training.

#### C.6 Unsupervised text embeddings: SimCSE and TSDAE

For both SimCSE and TSDAE, we trained both text-based models using `sentence_transformer` library (Reimers and Gurevych, 2019), as it provides a easy-to-use interface. They were trained on the full texts of Common Voice English training subset. Models were initialized with either `bert-base-uncased` or `roberta-base-uncased`. The following hyperparameters were used during model pretraining.



Models	Clusters	Vocab.	Effective batch size	Learning rate	Num. Negatives
Hu-SimCSE	50	50	128	5e-5	63
Hu-SimCSE	100	100	128	5e-5	63
Hu-SimCSE	200	200	128	5e-5	63
Hu-SimCSE	100	1000	256	5e-5	127
Hu-SimCSE	100	3000	64	5e-5	63
Hu-SimCSE	100	5000	256	5e-5	127
Hu-SimCSE	100	8000	256	5e-5	63
Hu-SimCSE	100	12000	256	5e-5	127
Hu-SimCSE	100	20000	256	5e-5	127
Hu-SimCSE	200	1000	128	5e-5	63
Hu-SimCSE	200	3000	512	5e-5	127
Hu-SimCSE	200	5000	128	5e-5	127
Hu-SimCSE	200	8000	256	5e-5	127
Hu-SimCSE	200	12000	256	5e-5	127
Hu-SimCSE	200	20000	128	5e-5	63

Table 8: Final hyperparameters for Hu-SimCSE models.

- Epoch: 2 (for SimCSE) or 3 (for TSDAE)
- Initial learning rate: 1e-4
- Effective batch size: 64

Otherwise we used the default settings in `sentence_transformer` for hyperparameters not mentioned above.

In our experiments, it shows that using `bert-base-uncased` as initial model perform better (see Table 1, Table 2, and Table 10).

## D Ablation analysis

### D.1 Deletion ratio of Hu-TSDAE

In the original implementation of TSDAE, the input data were subject to random deletion with a ratio of 0.6, as this augmentation led to the best evaluation performance (Wang et al., 2021). However, we found that training TSDAE on hidden units did not benefit from random deletion. In Table 11, the Spearman’s rank correlation was the highest when no deletion was applied.

### D.2 Impact of hidden units

The impact of vocabulary sizes on the validation performance was illustrated in Table 12 and 13. The vocabularies were induced from the whole Common Voice English corpus of hidden units. All models were tested on the CVS development set, which was also discretized into hidden units.

Both tables suggest that re-tokenizing hidden units into a higher level of sub-word units generally improve performance for directly modeling hidden units without speech involved. Generally speaking, smaller vocabulary size usually leads to

Hyperparameters	Value
Epoch	1
Total update steps	~4k
Optimizer	AdamW
Initial learning rate	1e-4
Effective batch size	192
Temperature	0.05
Memory bank	256
FP 16	False
Speech Encoder	HuBERT
Text Encoder	Pretrained weights
Training task	Contrastive learning
Loss	InfoNCE
Checkpointing	Keep best
GPU	A40
Time	~21 hours

Table 9: Hyperparameters for S-HuBERT.

Models	Synthetic speech					Natural Speech		
	STS12	STS13	STS14	STS15	STS16	STS	CVS-dev	CVS-test
<i>Unsupervised text embeddings</i>								
SimCSE-unsup-RoBERTa (CV text)	52.7	70.0	63.2	72.2	72.8	66.6	86.5	86.9
SimCSE-unsup-RoBERTa (ASR text)	53.4	67.0	62.2	72.7	71.8	67.5	86.9	87.1
TSDAE-RoBERTa (CV text)	45.0	59.2	54.7	65.3	70.9	57.1	88.0	86.7
TSDAE-RoBERTa (ASR text)	44.9	61.1	55.7	68.2	72.8	60.5	88.2	86.8

Table 10: Evaluation results based on the Spearman’s rank correlation. All models (e.g., HuBERT, RoBERTa, BERT) refer to the 12-layer base models.

Deletion ratio	CVS-dev
0	<b>52.7</b>
0.1	49.2
0.2	42.9
0.3	40.5
0.4	29.4
0.5	23.7
0.6	18.2

Table 11: The impact of deletion rate on the Spearman’s rank correlations ( $\times 100$ ) of CVS-dev evaluation. Here the base model was the Hu-TSDAE model, which was trained on pseudo-texts with 100 clusters, which were further tokenized to 8000 vocabulary tokens.

Models	Clusters	Vocab. Size	CVS-dev
Hu-TSDAE	50	50	49.9
Hu-TSDAE	100	100	55.8
Hu-TSDAE	200	200	56.9
Hu-TSDAE	100	1000	<b>58.4</b>
Hu-TSDAE	100	3000	57.5
Hu-TSDAE	100	5000	56.9
Hu-TSDAE	100	8000	54.3
Hu-TSDAE	100	12000	52.8

Table 12: For Hu-TSDAE models, the impact of the number of hidden-unit clusters and the vocabulary size on the Spearman’s correlation ( $\times 100$ ). Results were based on the hidden-unit transcriptions of CVS development set.

higher performance for Hu-TSDAE, as 1000 is the best vocabulary size. However, for Hu-SimCSE, having a vocabulary size that is not too large nor too small is more beneficial for training (8000 was the optimal size in our experiments).

### D.3 Pooling methods and loss functions

Here we compared how different pooling methods and loss functions can impact model performance. For loss functions, we compared InfoNCE and mean square errors (MSE), both of which are

Models	Clusters	Vocab. Size	CVS-dev
Hu-SimCSE	50	50	30.7
Hu-SimCSE	100	100	31.9
Hu-SimCSE	200	200	31.9
Hu-SimCSE	100	1000	35.2
Hu-SimCSE	100	3000	37.1
Hu-SimCSE	100	5000	37.3
Hu-SimCSE	100	8000	<b>42.5</b>
Hu-SimCSE	100	12000	35.6
Hu-SimCSE	100	20000	36.8
Hu-SimCSE	200	1000	36.8
Hu-SimCSE	200	3000	34.5
Hu-SimCSE	200	5000	36.1
Hu-SimCSE	200	8000	<b>39.4</b>
Hu-SimCSE	200	12000	38.5
Hu-SimCSE	200	20000	33.7

Table 13: For Hu-SimCSE models, the impact of the number of hidden units and the vocabulary size on the evaluation Spearman’s rank correlation ( $\times 100$ ). Results were based on the hidden-unit transcriptions of CVS development set.

commonly used for knowledge distillation.

For pooling methods, [Khurana et al. \(2022\)](#) has shown that self-attention pooling is better than mean or max pooling, so we did not repeat the same experiments here.

Two pooling methods for HuBERT models were compared: self-attention pooling and CLS pooling. The CLS pooling was proposed by [Duquenne et al. \(2021\)](#), in which a CLS token (a vector of ones) is inserted before speech feature extractors before feeding them into the transformer part of HuBERT. Then the last hidden state of the CLS token, or the first embedding of the last hidden states, was used as the representation of the whole speech signal. This is similar to pretrained text transformers like BERT ([Devlin et al., 2019](#)) and RoBERTa ([Liu et al., 2019](#)). Here we inserted a learnable embedding for the CLS token at the beginning of the sequence. Results in Table 14 suggest that self-attention pooling and the InfoNCE loss work best.

Models	Pooling	Loss	STS	CVS-dev	CVS-test
S-HuBERT ( <i>SimCSE-unsup-RoBERTa</i> )	CLS	MSE	68.6	79.5	77.9
S-HuBERT ( <i>SimCSE-unsup-RoBERTa</i> )	Self-attention	MSE	68.6	<b>80.7</b>	80.3
S-HuBERT ( <i>SimCSE-unsup-RoBERTa</i> )	CLS	InfoNCE	67.8	80.0	79.1
S-HuBERT ( <i>SimCSE-unsup-RoBERTa</i> )	Self-attention	InfoNCE	<b>70.2</b>	79.8	<b>81.6</b>

Table 14: Impact of different combination of pooling methods and loss functions on evaluation performance. The italicized blue font suggests that results or checkpoints were directly taken from [Gao et al. \(2021\)](#). Models in parentheses are teacher models whereas models outside parentheses are student models.

Models	Encoder Init.	Decoder Init.	STS	CVS-dev	CVS-test
WavEmbed	HuBERT-base-ls960	GPT-2-base	52.4	76.5	77.4
WavEmbed	HuBERT-base-ls960	BERT-base-uncased	55.4	<b>78.4</b>	78.7
WavEmbed	HuBERT-base-ls960	RoBERTa-base	<b>58.6</b>	78.1	<b>79.4</b>
WavEmbed	HuBERT-base-ls960	Random	37.5	74.6	76
WavEmbed	Random	Random	28.9	60.4	53.4
WavEmbed - 100 C	HuBERT-base-ls960	BERT pretrained on hidden units	50.0	65.6	61.2
WavEmbed - 100 C	HuBERT-base-ls960	Random	46.8	67.5	61.9

Table 15: Impact of weight initialization on the performance of WavEmbed. These results show that initializing with pretrained weights is important for improving performance.

#### D.4 Weight initialization

Here we compared how initializing with pretrained weights affect model performance. For WavEmbed, we initialized the WavEmbed decoder with different model weights, including pretrained weights or random weights. When initializing a decoder with encoders such as BERT and RoBERTa, all cross-attention layers were reinitialized. Despite this, initializing the decoder with pretrained weights can help improve the performance, as shown in Table 15.

#### E Visualizing predictions

Details predictions of selected models were shown at Figure 6 and 7.

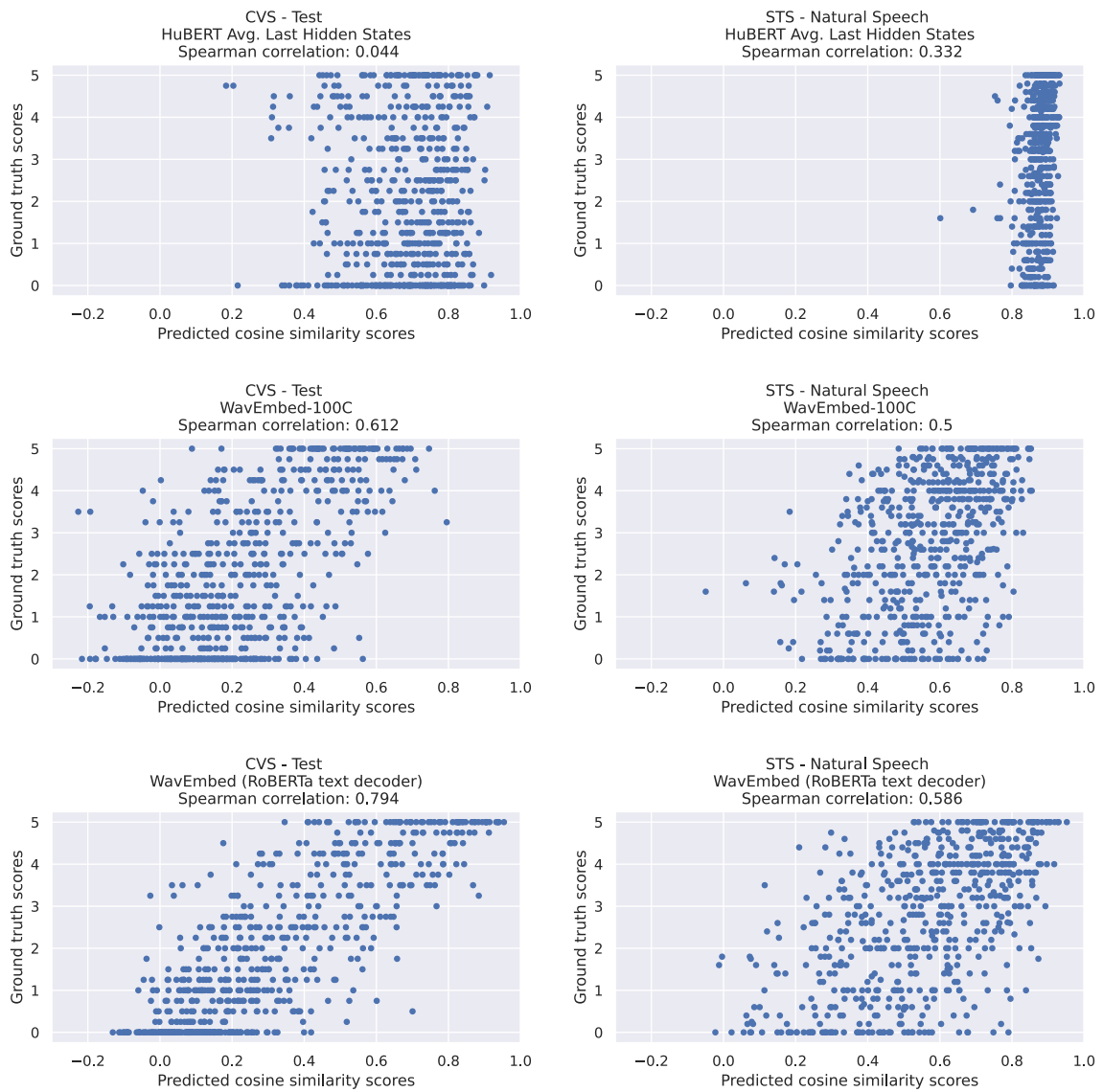


Figure 6: Plots of predicted semantic similarity scores by different WavEmbed models against ground truth scores.



Figure 7: Plots of predicted semantic similarity scores by different S-HuBERT models against ground truth scores.