SpeechLM: Enhanced Speech Pre-Training With Unpaired Textual Data

Ziqiang Zhang, Sanyuan Chen, Long Zhou, Yu Wu, Shuo Ren, Shujie Liu, Zhuoyuan Yao, Xun Gong, Lirong Dai, Member, IEEE, Jinyu Li, Senior Member, IEEE, and Furu Wei

Abstract—How to boost speech pre-training with textual data is an unsolved problem due to the fact that speech and text are very different modalities with distinct characteristics. In this paper, we propose a cross-modal Speech and Language Model (SpeechLM) to explicitly align speech and text pre-training with a pre-defined unified discrete representation. Specifically, we introduce two alternative discrete tokenizers to bridge the speech and text modalities, including phoneme-unit and hidden-unit tokenizers, which can be trained using unpaired speech or a small amount of paired speech-text data. Based on the trained tokenizers, we convert the unlabeled speech and text data into tokens of phoneme units or hidden units. The pre-training objective is designed to unify the speech and the text into the same discrete semantic space with a unified Transformer network. We evaluate SpeechLM on various spoken language processing tasks including speech recognition, speech translation, and universal representation evaluation framework SUPERB, demonstrating significant improvements on content-related tasks.

Index Terms—Speech-text joint pre-training, discrete tokenization, speech recognition, speech translation.

I. INTRODUCTION

SPEECH and text are two important carriers of human communication, and they can be converted into each other through speech recognition and synthesis systems. In past years, the unimodal self-supervised representation learning has been well explored in natural language [1], [2] and speech [3], [4]. According to neuroscience, humans first pre-process speech and text with different cortices, and then extract the meaning with the same area, called the Wernicke-Geschwind area [5]. Motivated by this, it is a very promising direction to design two pre-nets and a unified representation space (similar to the Wernicke area) so that the speech model would benefit greatly from text modality.

In terms of joint speech-text modeling, most approaches employ a speech encoder and a text encoder to map the speech and text inputs to hidden states, based on which, a shared encoder is used to learn cross-modality content information [6], [7], [8]. To align the speech and text modalities, two alignment losses (TLM and STM) in SLAM [6] are introduced with supervised ASR data. Extending SLAM to the multilingual scenario, mSLAM [7] introduces CTC losses and uses SpanBERT [9] to replace the BERT objective for pre-training on character-level text. Based on the RNN-T framework, Maestro [8] learns shared representations with modality matching, duration prediction, and sequence alignment. Almost all previous work follows the same structure with a speech/text encoder and a shared encoder, however, the interface between the speech encoder and the text encoder is not well studied, which probably leads to the outputs of the two encoders in different spaces, and suffers from transfer interference and capacity dilution for the shared encoder [6].

In this paper, we aim at unifying speech and text modalities via a well-defined interface, with which the model can benefit from additional textual data. We argue that such an interface should provide a shared semantic space for both speech and text, and preferably have strong interpretability and learnability. To this end, we explore two alternative representation spaces satisfying the above characteristics of the interface, which are based on phoneme units and hidden units. With them, we can convert the speech and text to a shared intermediate modality (phoneme/hide units) and decouple the joint speech-text modeling into two sub-modules, transformation components between speech/text modalities and the discrete units, and a unified unit representation learning component. More specifically, we employ a Speech Transformer encoder and an embedding layer to convert unlabeled speech and text-derived units into representations with the same dimension, respectively. After that, we leverage a functionally defined Shared Transformer encoder to model the representations for both speech and text jointly. As for the learning objective, we propose two pre-training tasks. One is Unit-based Masked Language Modeling (UMLM) task trying to predict the unit tokens from the masked text. The other one is Unit-based Connectionist Temporal Classification (UCTC) task, aiming at reconstructing the complete text sequences from the masked content.
unit sequences. To better align the representations from speech and text, we also adopt a Random Swapping Mechanism for the UMLM task, which swaps the intermediate representations from the speech and the corresponding discrete units before feeding them into the Shared Transformer.

To prepare the discrete unit tokens for model pre-training, we introduce two discrete tokenizers named **phoneme-unit tokenizer** and **hidden-unit tokenizer**. For the phoneme-unit tokenizer, we use a hybrid ASR model to transcribe unlabeled speech sequences to frame-level phoneme units, and convert unpaired text by the lexicon. For the hidden-unit tokenizer, we use the HuBERT-based k-means model to cluster speech into hidden units, and utilize a non-autoregressive model to transform the unlabeled text into hidden units. All tokenizer models are obtained with unsupervised data or a small amount of ASR data, and are used offline before pre-training. By converting to the shared phoneme/hidden-unit tokens, both unpaired speech and unpaired text data can be leveraged as the training data for pre-training.

The contributions of this paper are summarized as follows.

- We propose two alternative tokenizers which can convert unlabeled speech and text into the discrete shared space and relieve the influence of modality difference.
- The proposed SpeechLM (Speech and Language Model), equipped with two simple and clear learning objectives and the random swapping mechanism, can unify and simplify the cross-modal speech-text pre-training.
- Experiments demonstrate that SpeechLM enhanced by textual data significantly outperforms its speech-only counterparts on various spoken language tasks, e.g., ASR, speech translation (ST), and universal representation evaluation framework SUPERB [10].

The rest of the paper is organized as follows. Section II introduces the related works. Section III describes the proposed speech-text joint pre-training framework, SpeechLM. Experiments and results are present in Section IV with analysis discussed in Section V. Finally, we summarize the conclusions in Section VI.

II. RELATED WORK

A. Discrete Predictive Representation Learning for Speech

Unlike natural language processing (NLP), speech signals are continuous, making it not straightforward to find the predictive labels for pre-training. To tackle this issue, a tokenizer, also referred to as a quantizer, is required to map continuous speech features into discrete tokens [4], [11], [12]. HuBERT [4] is the pioneer in the exploration of predictive speech representation learning (SSL), which utilizes a k-means model on the middle layer of the Transformer as the tokenizer to convert speech into discrete tokens. In [12] tries to combine a contrastive loss and a masked prediction loss in a self-supervised speech representation learning framework. And [13] extends the prediction task for intermediate Transformer layers. In addition to the unsupervised tokenizers, [14] proposes a supervision-guided tokenizer, which is an acoustic model trained on limited labeled data, and can generate frame-level aligned phonemes as the predictive targets for SSL. In this work, we inherit the concept of discrete predictive learning, while our goal is to take advantage of textural data to improve speech representations.

B. Joint Speech-Text Pre-Training

With the rapid development of unimodal pre-training in speech and natural language processing [1], [4], joint speech-text pre-training obtains more and more attention from research and industrial communities [6], [15], [16], [17], [18], [19]. Most previous studies [15], [16], [17] design modality-specific modules for speech and text for lower representations, and then use a shared module to jointly model them. And individual self-supervised losses are calculated for unpaired speech and text data for pre-training. However, in this way, the speech and the text are not guaranteed to lie in the same hidden space in the shared module, since they are unpaired data that are unaligned. This is known as the transfer interference issue [6] and will cause capacity dilution resulting in performance degradation. To alleviate this issue, some work [6], [8], [20], [21] try to achieve speech-text alignment by leveraging extra speech-text paired data. SLAM [6] and mSLAM [7] leverage extra supervised ASR tasks to enhance the speech-text alignment. MAESTRO [8] also uses paired speech-text data in RNN-T framework to encourage the alignment of the intermediate representations at a more fine-grained level. However, these approaches still leave unpaired speech and text data modeled separately by using different pre-training targets, which might still lead the shared module to use individual capacities to handle each modality. Unlike SLAM and MAESTRO which directly use paired data at the pre-training stage, we utilize trained tokenizers to convert all unpaired speech and text data into the same discrete space and eliminate the influence of modal difference, so that the two modalities can interact naturally via the shared interface during the pre-training. While SpeechUT [19] also leverages hidden units as the tokenization of speech and text modalities, while it only works in an encoder-decoder architecture where the encoder mainly models the speech and the decoder mainly models the text. Instead, this work focuses on the encoder architecture which is more suitable for general speech representations, and explores different types of tokenizers and pre-training tasks.

Our proposed method introduces several notable advancements compared to prior research: (1) it establishes a formal alignment between speech and text within an explicit semantic space, thereby improving speech representation; (2) we introduce two types of tokenizers, including hidden-unit and phone-unit tokenizers, to take advantage of large-scale unpaired speech and text data; (3) the SpeechLM design is straightforward and efficient, employing unit-based masked language modeling and connectionist temporal classification tasks for pre-training.

III. METHODS

Given unpaired speech and text data, SpeechLM is pre-trained to learn a unified representation of speech and text modalities with the help of offline discrete tokenizers. In this section, we will present the overall framework of SpeechLM, as well as the pre-training procedures and the tokenizers.
A. Phoneme/Hidden Unit as the Bridge

Speech and language are two different modalities with different characteristics. We explore bridging speech and text pre-training with an explicitly defined discrete representation, where speech and text could be tokenized into a shared discrete space easily. Leveraging phoneme/hidden units as the bridge between speech and text has the following advantages: First, it is easier to separately align speech and text into a shared intermediate representation than to align them directly. Second, we can make full use of additional unpaired data to improve the alignment; Thirdly, we can leverage more fine-grained alignment information, i.e., at the frame level, to facilitate joint modeling.

To achieve this goal, we implement two tokenizers for both speech and text, a phoneme-unit tokenizer and a hidden-unit tokenizer, which will be described in detail in Section III-D. The former aims to convert speech and text into the phoneme space, while the latter converts them into an acoustic clustering space. Given a speech sample \( S \) or a text sample \( Y \), a tokenizer (\( T_S \) for speech, \( T_T \) for text) yields a sequence of discrete units \( Z \),

\[
Z_S \triangleq (z_{S_1}, \ldots, z_{S_M}) = T_S(S),
\]

\[
Z_T \triangleq (z_{T_1}, \ldots, z_{T_N}) = T_T(Y)
\]

where \( M \) and \( N \) are the lengths of the unit sequences from speech and text, respectively.

B. Model Architecture

SpeechLM consists of a Speech Transformer and a Shared Transformer, which are enhanced with the random swapping mechanism, as illustrated in Fig. 1. Next, we will introduce the main modules with the input of unpaired speech \( S \) and text \( Y \).

1) Speech Transformer: Following HuBERT [4], we use a standard Transformer [22] as the backbone of the Speech Transformer, equipped with relative position embedding [23]. A speech waveform \( S \) is first processed into a sequence of speech features \( X \triangleq (x_1, x_2, \ldots, x_M) \) by a stack of 1-D convolutional layers. We follow HuBERT to mask the speech features \( X \) with the mask probability of 8% and the mask length of 10, resulting in nearly half of the features being masked. Then the masked features, \( \hat{X} \), are fed into the Speech Transformer for intermediate representations,

\[
H_S^l = \text{Transformer}(H_S^{l-1})
\]

where \( l \) means the layer and \( H_S^l \triangleq \hat{X} \) indicating the input. Let \( L \) be the total number of layers of all Transformer modules in SpeechLM architecture, and the Speech Transformer accounts for half. Consequently, the output of the Speech Transformer can be formulated as \( H_S^{L/2} \triangleq (h_{S_1}^{L/2}, \ldots, h_{S_M}^{L/2}) \).

2) Shared Transformer: The Shared Transformer has the same architecture with the Speech Transformer and handles two types of input with respect to speech and text. The first input is the previous output of the Speech Transformer, \( H_S^{L/2} \), and it is processed by the Shared Transformer into higher-level representations, \( H_T^{L/2} \). The second input is the unit embedding sequence \( U_T \triangleq (u_{T_1}, \ldots, u_{T_N}) \) that is embedded from the text-derived units, \( Z_T \), by the unit embedding layer,

\[
U_T = \text{Emb}(Z_T)
\]

It is then processed by the Shared Transformer into \( H_T^L \), where \( H_T^{L/2} \triangleq U_T \) indicates the input. Consequently, \( H_T^L \) and \( H_T^L \) are used as the encoded representations for speech and text, respectively. For textual representations, we further employ a CTC layer [24] that converts \( H_T^L \) to character-level representations.

3) Random Swapping Mechanism: To better align the speech and textual representations into a shared latent space at the early layer of the Shared Transformer, we introduce a random swapping mechanism which was proposed in [25] and also used in [19]. As each speech sequence can be tokenized into a sequence of discrete units, we can randomly select some time positions (denoted as \( i \in R \)) from a speech sequence and replace each \( h_{S_i}^{L/2} \) with the corresponding unit embedding \( u_{S_i} \), where \( u_{S_i} = \text{Emb}(z_{S_i}) \) is derived from speech units \( z_{S_i} \) by the unit embedding layer. To avoid information leakage, the swapping positions \( R \) are only selected within unmasked regions of speech.
sequence. In this way, we can shuffle two modalities into one sequence and enforce the model to treat them equally.

C. Pre-Training Tasks

SpeechLM is jointly optimized by a unit-based masked language modeling task with unlabeled speech data and a unit-based connectionist temporal classification task with unlabeled text data.

1) Unit-Based Masked Language Modeling (UMLM): The unit-based masked language modeling task is designed for speech pre-training, like HuBERT [4] and ILS-SSL [13]. Given $l$th-layer speech representations $H_S^l \triangleq (h_{S_1}^l, \ldots, h_{S_M}^l)$, UMLM tries to predict the corresponding tokenized units $Z_S \triangleq (z_{S_1}, \ldots, z_{S_M})$ at the masked positions. The probability of the predicted unit at position $i$ is calculated with

$$p(z_i|h_{S_i}^l) = \frac{\exp(\cos(W h_{S_i}^l, e(z_i))/\tau)}{\sum_{z' \in Z} \exp(\cos(W h_{S_i}^l, e(z'))/\tau)}$$

(4)

where $W$ is a projection matrix, $e(\cdot)$ is an embedding matrix, $\tau$ is the temperature coefficient which is usually set to 0.1, and $Z$ is the set of phoneme/hidden-unit categories, i.e. the vocabulary. Similar to ILS-SSL, here the UMLM loss is computed on both the outputs of Speech Transformer ($H_S^{L/2}$) and Shared Transformer ($H_S^1$), with the loss formulated as,

$$L_{UMLM} = - \sum_{i \in M} \left( \log p(z_{S_i}|h_{S_i}^{L/2}) + \log p(z_{S_i}|h_{S_i}^1) \right)$$

(5)

where $z_{S_i}$ is the corresponding speech unit at position $i$ and $M$ is the set of masked positions.

2) Unit-Based Connectionist Temporal Classification (UCTC): Connectionist temporal classification (CTC) [24] is first proposed to address the sequence label problem where the output is shorter than the unsegmented input sequence. Here, we partially mask the phoneme-unit or hidden-unit sequences $Z_T$ tokenized from the unlabeled text as the input, and aim at recognizing the complete original text through the Shared Transformer and CTC layer. The input sequence is masked in the same way as the speech features described above. Regarding the text sequence $Y$, the unit-based CTC loss is calculated as,

$$L_{UCTC} = -\log p_{CTC}(Y|H_T^L)$$

(6)

where $p_{CTC}(\cdot)$ is modeled by the CTC layer, whose goal is to transform the encoded unit representation $H_T^L$ into the target characters $Y$.

By taking advantage of unlabeled speech and text data, SpeechLM performs multi-task pre-training with UMLM and UCTC tasks,

$$L = L_{UMLM} + \lambda L_{UCTC}$$

(7)

where $\lambda$ is used to control the weight of two losses. Through joint optimization and the random swapping mechanism, SpeechLM is expected to align speech and text into a unified unit representation.

D. Unified Tokenizers

Figs. 2 and 3 show the overview of the proposed phoneme-unit tokenizer and hidden-unit tokenizer. Besides, the tokenizers are offline models, which are used to pre-process the unlabeled speech and text data before the pre-training.

1) Phoneme-Unit Tokenizer: Inspired by PBERT [14], which leverages phoneme labels as the pre-training targets, we introduce the phoneme-unit tokenizer ($T^P_S$) to discretize speech signals ($T^P_S$) as well as text sequences ($T^P_T$). For speech data, the tokenizer is composed of a phoneme recognition model, whose goal is to convert acoustic features into phoneme units. It contains an acoustic encoder that estimates the prior phoneme probability of speech, a language model with a lexicon that provides the phoneme distribution, and a weight finite-sate transducer (WFST) [26] decoder that computes the final posterior phoneme probability. We implement it using the open-source Kaldi toolkit [27] with a small amount of paired ASR data and language model data. More details are described in Section IV-B.

For text data, we can directly convert words into phonemes by looking up the provided lexicon. To alleviate the mismatch in the length of speech and text, we further upsample the phoneme sequence of text by randomly repeating each phoneme many times to make sure they have similar lengths to that of the speech.

2) Hidden-Unit Tokenizer: We follow HuBERT to tokenize speech into hidden units with a k-means cluster model, called $T^H_S$, where the clustering feature is the intermediate hidden states of the 2nd round HuBERT model.
To tokenize the text data into the same hidden-unit space, we propose a non-autoregressive text to hidden-unit model \((\mathcal{T}_S^H)\), which is based on FastSpeech [28]. The model consists of a text encoder that models the textual representations, a duration model that estimates the output length, and a unit decoder that predicts the final HuBERT-style hidden units. \(\mathcal{T}_F^H\) is trained with a small number of text-unit pairs derived from ASR data, where the text side is the phoneme transcriptions with phoneme’s durations, and the units are tokenized from the corresponding speech by \(\mathcal{T}_S^H\).

IV. EXPERIMENT

SpeechLM is evaluated on various spoken language tasks, including automatic speech recognition (ASR), speech translation (ST), and the universal representation evaluation benchmark SUPERB [10]. According to the tokenizers, the model can be divided into SpeechLM-H and SpeechLM-P using hidden-unit and phone-unit as discrete tokens, respectively.

A. Data

We use unlabeled speech data from LibriSpeech [29] and LibriLight [30] to pre-train Base and Large models respectively. LibriSpeech contains 960 hours of labeled speech where the labels are not used in pre-training. The speech data are collected from English audiobooks in a reading style. LibriLight has about 60,000 hours of unlabeled speech in the same acoustic domain as LibriSpeech. The unpaired text data are from LibriSpeech LM corpus, containing about 40 million English sentences. The paired data for optimizing the tokenizers (\(\mathcal{T}_S^H\) and \(\mathcal{T}_F^H\)) are the full LibriSpeech data in the Large setting and the 100-hour subset (train-clean-100) in the Base setting. For downstream tasks, we use LibriSpeech for ASR evaluation, and four translation directions of CoVoST-2 [31] for ST evaluation. CoVoST-2 [31] is a large-scale multilingual Speech Translation corpus based on CommonVoice [32], containing about 430 hours of reading-style speech from the Internet. For all tasks of SUPERB evaluation, the data details can be found in [10].

B. Tokenizer Setup

a) Phone-unit tokenizer for speech: In the Base setting, we follow the standard Kaldi [27] recipe\(^1\) to train a hybrid GMM-HMM ASR model on 100 hours of labeled LibriSpeech data, denoted as \texttt{tri4b} model. To boost the performance, we use the \texttt{tri4b} model to decode the remaining 860 hours of speech and combine them into the final 960 hours of pseudo-labeled data. We then train a bigger GMM-HMM model on the pseudo-labeled data (denoted as \texttt{tri6b}) as the final phone-unit tokenizer. The effectiveness of this strategy is demonstrated in Section V-E. In the Large setting, we train a neural network (DNN) instead of GMM on 960-hour labeled LibriSpeech data, which can further boost the phoneme accuracy. Once the GMM/DNN-HMM hybrid model is obtained, the tokenization process involves decoding and transducing the best phoneme-level alignment path from the unlabeled speech. The original two Kaldi models in the Base and Large setting have a frame-shift of 10 ms and 30 ms, respectively. Thus we re-sample the phonemes to a frame-shift of 20 ms to match SpeechLM model architecture. Detaiully, in the Base setting we downsample the phoneme-units and in the Large setting we perform linear interpolation to upsample the phoneme-units.

b) Phone-unit tokenizer for text: We use the 200 K word-to-phone lexicon provided by LibriSpeech dataset to convert words to phonemes, the OOV words are replaced by \(<\text{unk}>\) symbol. Following [33], we randomly insert the silence phoneme \(<\text{SIL}>\) between words with a probability of 25%. Then we upsample the phoneme sequence by repeating the phonemes. The length of phonemes follows Gaussian distribution estimated from the train-clean-100 set of LibriSpeech. Though setting individual mean and variance for each phoneme category seems better for simulating proper sequence length, our preliminary experiments show it performs similarly compared with setting unified mean and variance for all non-silence phoneme categories. Thus we simply set the global mean which is 5, and variance which is 25 for non-silence phonemes. The silence phoneme \(<\text{SIL}>\) has a mean of 14 and a variance of 25.

c) Hidden-unit tokenizer for speech: We use the released HuBERT [4] model following a K-Means model as the tokenizer for speech. Recent work by [34] has highlighted the importance of the quality of hidden units for the performance of self-supervised pre-training models. While in this paper we leave the exploration with other more advanced tokenizers such as WavLM as future work. HuBERT model is pre-trained on the same 960-hour unpaired LibriSpeech data. The K-Means model has 500 clustering centers, regarded as hidden units. It produces the hidden units at a frame rate of 50.

d) Hidden-unit tokenizer for text: To build the text-to-hidden-unit tokenizer, we modify FastSpeech [28] by replacing the task from spectrogram regression to hidden-unit classification. The model consists of a text encoder, a duration module, and a unit decoder. We maintain the same model architecture as [28] except for the last classification layer. The input to the encoder is a phoneme sequence without duration information. Then upsampling is performed by the duration module, which predicts the duration of each phoneme embedding and repeats the embeddings before feeding them into the decoder. At last, the decoder models the repetitive phoneme embeddings and output units. The training data are processed from ASR data, where the input phonemes are from the text and the targets are derived from the speech by the Hidden-unit tokenizer for speech. We train the model on LibriSpeech train-clean-100 subset for 10 K steps, with a learning rate of 5e-4 and a batch size of 10 K input phonemes. The final model achieves 41.3 and 34.6 BLEU scores on dev-clean and dev-other sets.

C. Model Setup

a) Network Architecture: The network architecture of SpeechLM follows that of HuBERT [4] for a fair comparison, which is also used for many following work [13], [14], [35].
Specifically, the Base model consists of $L=12$ Transformer layers where both the Speech Transformer and the Shared Transformer have 6 layers. Each Transformer layer is with an attention dimension of 768, a feed-forward dimension of 3072, and attention heads of 12. The Large model doubles the number of Transformer layers, and increases the attention dimension to 1024, the feed-forward dimension to 4096, and attention heads to 16. The convolutional layers in front of the Speech Transformer process and downsample the 16 kHz input waveform into lower-level representations. They have 512 channels for each convolutional layer and the kernel sizes are [10, 3, 3, 3, 3, 2, 2], resulting in a downsampling rate of 320. The CTC layer consists of a single 1-D convolutional layer with a kernel size of 2, it is then followed by a linear layer that outputs the probabilities of text characters.

b) Pre-training configuration: All models are pre-trained on 32 GPUs for 400 K steps. To align with HuBERT, we additionally set the update frequency to 4 for Large models to simulate 128 GPUs. We use Adam [36] optimizer with $\beta_1=0.9, \beta_2=0.98$ for optimization. The maximum learning rate is set to $5e-4$ and decays linearly to zero after the warming-up steps. The batch size for the Base model is 4375 tokens after down(sampling for both speech and text input, and for the Large model it is set to 2800. The hyper-parameter $\lambda$ used for weighting $L_{UCTC}$ is determined after searching from $\{0.01, 0.1, 1.0, 10.0\}$. We found that a lower weight achieves the best performance of ASR on the valid set, and the best choice is 0.1.

c) Fine-tuning configuration: For ASR fine-tuning, when fine-tuned on 100-hour LibriSpeech, the total steps are 60 K with a batch size of 800 seconds. When fine-tuned on the full 960-hour LibriSpeech, the total steps are 200 K with a batch size of 1800 seconds. All ASR models on LibriSpeech are tuned with a maximum learning rate of 1e-5 and a tri-stage learning rate schedule with the warming-up, holding, and decay periods of $[0.1, 0.4, 0.5]$.

And for CoVoST-2, both the Base and the Large models are fine-tuned for 50 K steps with a batch size of 1600 seconds. The learning rate warms up to 1e-4 in 5 K steps and then decays linearly to zero. After fine-tuning, we select the model with the best accuracy on the valid set in the Base setting and average the top 5 models with the best accuracy on the valid set in the Large setting. The decoding beam size is 5 without external language model fusion.

For SUPERB evaluation, we follow the setup in [10].

D. Evaluation on Speech Recognition

We further evaluate our SpeechLM models on SUPERB [10], which is designed to provide a standard and comprehensive testbed for pre-trained models on various speech tasks, including Speaker Identification (SID), Automatic Speaker Verification (ASV), Speaker Diarization (SD), Phoneme Recognition (PR), and attention. Table I shows that in the Base setting, by taking advantage of textual data, SpeechLM significantly outperforms previous models, such as wav2vec 2.0 [37], HuBERT [4], WavLM [35], PBERT [14], ILS-SSL [13], and data2vec [38]. Particularly, the proposed SpeechLM-P obtains about 39% and 16% relative WER reductions over HuBERT and data2vec on test-other set without LM. Even with LM the superiority of SpeechLM is reduced, it still remains 10% relative WER reduction over data2vec. That demonstrates the effectiveness of our proposed method for taking advantage of textual data. Unlike external language models, SpeechLM encodes the text data directly into the model with shared model parameters, and it shows complementarity with external LM. More ASR results and analyses can be found in Section V. Furthermore, in the Large setting, SpeechLM also achieves competitive or better performance than previous work. Results with other previous work with larger models3 like SLAM and MAESTRO are listed in Table VII, note that they use 2× model size and larger amount of paired data, or different inference framework (e.g., RNN-T in MAESTRO), making their results not comparable with the setting in Table I.

E. Evaluation on Speech Translation

We then evaluate SpeechLM on speech-to-text translation tasks. Following [40], we use four language directions from English to German (de), Catalan (ca), Arabic (ar), and Turkish (tr) in CoVoST-2 [31] as the downstream tasks. When fine-tuning, the pre-trained model serves as the encoder, then it is followed by a randomly initialized decoder. The decoder consists of 6 Transformer layers with an attention dimension of 768. We use character vocabulary for the output of target languages in all translation tasks. The final evaluation is the word-level case-sensitive detokenized BLEU [41] on the test set. The results are shown in Table II, including the baselines fine-tuned from other pre-trained models. The numbers in brackets represent the standard deviation of three fine-tuning results. Table II shows that by boosting the quality of speech representation learning with textual data, SpeechLM achieves significant improvement over the speech-only pre-trained baseline, HuBERT, as well as the ASR pre-trained model [31]. In the Base setting, SpeechLM-H and SpeechLM-P achieve comparable results, with 2.4 BLEU improvement over HuBERT Base. Moreover, SpeechLM-P Large model surprisingly outperforms previous work such as SLAM X-Large [6], even though the latter has a much bigger model size.

F. Universal Representation Evaluation

We further evaluate our SpeechLM models on SUPERB [10], which is designed to provide a standard and comprehensive testbed for pre-trained models on various speech tasks, including Speaker Identification (SID), Automatic Speaker Verification (ASV), Speaker Diarization (SD), Phoneme Recognition (PR),

3We do not compare our work with other text injection methods like [39], because [39] is designed to adapt CTC-based ASR models to a target domain using unpaired text data, while our objective is to align speech and text in the pre-training stage.
Automatic Speech Recognition (ASR), Out-Of-Domain Automatic Speech Recognition (OOD-ASR), Keyword Spotting (KS), Query by Example Spoken Term Detection (QbE), Speech Translation (ST), Intent Classification (IC), Slot Filling (SF), Emotion Recognition (ER). These tasks can be grouped into five aspects of speech: content, speaker, semantics, and paralinguistics (ParaL). Table III shows the universal speech representation evaluation results. Compared to the previous self-supervised learning methods, SpeechLM achieves good performance on several content-related and semantic-related tasks, such as PR, ASR, ST, and SF. Particularly, the proposed SpeechLM-P model obtains 36% and 20% relative PER/WER reductions on PR and ASR tasks. Meanwhile, we can observe performance degradation for the speaker and paralinguistics-related tasks, especially for SpeechLM-P. It indicates that with our joint speech and text pre-training method, the model learns more about extracting the
content-related information while discarding the other aspects of speech signals.

V. ANALYSIS

To better understand the effectiveness of the proposed method, we conduct several experiments to investigate its main components, such as the random swapping mechanism, the speech/text pre-training ratio, the comparison of two tokenizers, the amounts of unpaired text data and paired data, and further visualization analysis.

A. Effect of Random Swapping Mechanism

We argue the introduced random swapping mechanism is the key component of SpeechLM to align the speech and text modalities in the same space. Here, we explore its effectiveness by removing it from the SpeechLM-P model. As shown in lines 1-2 of Table IV, without the random swapping mechanism, the performance degrades dramatically from 8.1 WER to 9.1 WER on test-other set without LM, and from 6.2 WER to 6.7 WER with LM. The WER on test-clean set also increases. That confirms that the random swapping mechanism is beneficial for the downstream ASR tasks, demonstrating a better pre-trained model.

B. Comparison of Two Tokenizers

We further compare the influence of the choice of two tokenizers. Tables I and II show that two tokenizers result in comparable performance on ASR and ST tasks. To further remove the influence of text data, we pre-train the models with these two tokenizers using only speech data, with results shown in lines 3-4 of Table IV. Lines 3-4 show that two tokenizers still perform comparably for downstream ASR tasks. Such results demonstrate that both phoneme-units and hidden-units could be good pseudo targets for pre-training speech models. Moreover, we explore whether we can obtain improvement by not relying on paired speech-text data for training tokenizers, i.e., using phoneme-units for text data and hidden-units for speech data. We conduct an experiment (line 6) in which the speech side predicts the hidden units and the text side is trained with masked phoneme-to-character CTC loss. In this case, the intermediate modality is not unified for speech and text so the random swapping mechanism can not be performed. Compared to the results using pair data (line 1 and line 5), the performance is degraded drastically, which is over 10% relative WER degradation. That demonstrates paired data, or unified tokenizers, are necessary for aligning the modalities for the SpeechLM framework.

C. Noisy Data Evaluation

To assess the noise robustness of our method, we also evaluate the pre-trained models on noisy test data for ASR. The test set is made by [50], which is a mixture of LibriSpeech test-clean set and a noise subset of FreeSound [51] at different levels of signal-to-noise ratio (SNR). We use three noise types including ‘Traffic’, ‘Metro’, and ‘Car’ for evaluation. For comparison, both the Wav2vec 2.0 baseline and our models were pre-trained and fine-tuned using the unaltered LibriSpeech dataset, without the incorporation of any noise from FreeSound. Additionally, no external language models are utilized during the ASR decoding process. The outcomes of these evaluations are summarized in Table V, indicating that SpeechLM consistently outperforms Wav2vec 2.0 by a large margin in noisy conditions. Moreover, we notice that the SpeechLM-H model performs slightly better than the SpeechLM-P model in most noisy cases.

D. Effect of Text Data Size

Since the text corpus contains up to 40 M sentences which is much larger than the number of speech samples (960-hour Librispeech contains about 30 K sentences), we conduct experiments to explore the effect of text data size for pre-training by randomly sampling subsets from the original text corpus. Surprisingly, Fig. 4 shows that the performance does not degrade much until the text data are reduced to 40 K sentences. We speculate that the text data here are used at the lexical level, i.e., the transformation from phoneme/hidden units to characters, and 40 K data is sufficient to build such a lexicon. It is also noted that the WER of dev-other set is getting worse as the amount of text data increases for the SpeechLM-H models, while such degradation is not observed for SpeechLM-P. As a result, the best SpeechLM-H model is obtained using 400 K text data, which achieves WERs of 3.8, 8.3, 2.7, and 6.0 without or with LM on test-clean and test-other set. It is possible due to the hidden-unit tokenizer \( T_H \) trained on 100 h clean unit-to-text data thus the tokenizer tends to produce cleaner units, and this tokenization bias can accumulate as the amount of text data increases.
TABLE V
ASR PERFORMANCE (WER) ON NOISY TEST SETS AT DIFFERENT NOISE TYPE (TRAFFIC, METRO AND CAR) AND SNRs (0–20 dB)

| Model | Traffic (SNR) | Metro (SNR) | Car (SNR) |
|-------|---------------|-------------|-----------|
|       | 0  5  10  15 | 0  5  10  15 | 0  5  10  15 |
| Wav2vec 2.0 Base [37] | 44.8 34.1 23.2 18.4 12.8 | 45.7 31.5 23.9 15.3 10.6 | 17.6 12.2 9.6 7.8 7.2 |
| SpeechLM-H Base | 35.3 24.0 14.4 9.5 8.1 | 29.6 18.1 13.3 8.6 6.5 | 11.4 6.9 5.5 4.6 4.4 |
| SpeechLM-P Base | 38.0 26.0 16.5 11.6 8.9 | 34.2 20.6 14.8 8.7 7.2 | 12.7 7.3 4.9 4.5 4.1 |

All models are pre-trained/fine-tuned without noise injection.

Fig. 4. ASR performance fine-tuned on 100-hour LibriSpeech benchmark, models are pre-trained with different amounts of text data.

TABLE VI
ASR PERFORMANCE (WER) OF DIFFERENT PRE-TRAINED SPEECHLM-P BASE MODELS ON THE LIBRISPEECH 100-HOUR BENCHMARK WITH RESPECT TO DIFFERENT PHONEME-UNIT TOKENIZERS FOR SPEECH ($T_S^P$)

| $T_S^P$ model | Training data (h) of $T_S^P$ | WER of $T_S^P$ | WER of fine-tuned models |
|---------------|-------------------------------|----------------|--------------------------|
|               | train-clean-100 | train-other-500 | dev-clean | dev-other | test-clean | test-other |
| GMM-HMM | 30 | 6.79 | 20.39 | 3.35 | 8.42 | 3.43 | 8.40 |
| GMM-HMM | 100 | 6.40 | 19.58 | 3.38 | 8.48 | 3.40 | 8.43 |
| GMM-HMM | 100 + 860(unpaired) | 6.55 | 14.87 | 3.32 | 8.17 | 3.38 | 8.14 |
| FA | 960 | 0 | 0 | 3.24 | 7.33 | 3.30 | 7.30 |

All results are reported using language models.

TABLE VII
ASR PERFORMANCE (WER) ON 960 H LIBRISPEECH BENCHMARK, COMPARING SPEECHLM WITH SLAM AND MAESTRO

| Model       | Size       | Pre-training speech data | WER (†) w/o LM | WER (†) w/ LM |
|-------------|------------|--------------------------|----------------|--------------|
|             |            | Unpaired speech | Paired speech | test-clean | test-other | test-clean | test-other |
| SLAM [6]    | X-Large (0.6B) | 60kh | 960kh | 1.6 | 3.1 | - | - |
| Maestro [8] | X-Large (0.6B) | 60kh | ~5kh | 1.5 | 2.8 | Conformer | 1.5 | 2.7 |
| SpeechLM-P (ours) | Large (0.3B) | 60kh | 960kh | 1.9 | 3.6 | Transformer | 1.8 | 3.2 |

E. Effect of Paired Data Size

Besides the comparison of using or not using paired data for unified tokenizers at the pre-training stage in Section V-B, another question is how much data is sufficient for training these tokenizers, especially considering if it is potential to be used for low-resource languages. Since dozens of hours of paired data can be obtained for most of the languages [32], we start at using about 30-hour paired data randomly selected from train-clean-100 of LibriSpeech. Table VI lists the results of SpeechLM-P Base models pre-trained with different speech tokenizers ($T_S^P$). Other configurations of pre-training and fine-tuning including the text tokenizer keep the same for all models. Since the tokenizers are actually ASR models, we can evaluate their performance by computing the WER on two training sets of LibriSpeech, where the −other− set contains noisier speech than the −clean− set. Though the training sets are used as unpaired speech data at the pre-training stage, we could still use the labels to evaluate the tokenizers for just measuring their quality. As shown in Table VI, when reducing the training data of the tokenizers, the performance of the SpeechLM-P models degrades less than the tokenizers. For example, comparing the first two lines in Table VI, the fine-tuning WER barely changes (less than 0.1 points) when the reduction of training data causes obvious different tokenizers. Then, we find that improving the tokenizer for noisy data helps, as the semi-supervised trained tokenizer (line 3) using the rest of unpaired data in LibriSpeech improves the SpeechLM-P model, especially for dev/test-other sets. At last, we try to use the Oracle phoneme tokens obtained by force-alignment (denoted as FA) with true transcriptions to pre-train the model (the last line). The WER decreases marginally (about 0.1 points) on dev/test-clean sets while remarkably (over 0.8 points)
on dev/test-other sets. To summarise, Table VI shows that the tokenizer trained by 100-hour paired data seems to be sufficient in SpeechLM-P Base setting for clean data while still being improved for noisy data.

F. Visualization Analysis

Fig. 5 illustrates the distributions of representations extracted from different layers of the Shared Transformer in the SpeechLM-P Base model. The dimension is reduced to 2-D by T-SNE [32]. Data points are randomly sampled from unpaired speech and text samples from LibriSpeech dev-clean set. As text is represented by units for modeling in SpeechLM, we also convert the text to phoneme units, so that each data point represents 20 ms of speech or 1 phoneme unit. Layer = 6 denotes the representations before feeding into the Shared Transformer. It is shown that as the layer increases, SpeechLM tends to align speech and text representations into a shared space.

VI. CONCLUSION

In this work, we present SpeechLM, a text-augmented speech pre-trained model, which achieves competitive performance on various spoken language tasks, such as automatic speech recognition and speech translation. To make full use of unpaired data, we propose two alternative discrete tokenizers based on phoneme units and hidden units to tokenize speech and text into the same semantic space. With the shared interface, SpeechLM can learn better speech representations with the help of text modality. Quantitative and qualitative analyses demonstrate the superiority and effectiveness of the proposed method. For future work, we would like to advance the work by deeply integrating large language models and extending them to natural language tasks.

LIMITATIONS

While the proposed SpeechLM achieves competitive performance on various spoken language tasks, it still has some limitations: (1) the current method needs paired data, or phoneme lexicon to build the tokenizers. The lexicon might be language-specific, which restricts the cross/multi-lingual application; (2) the effectiveness of applying SpeechLM to other speech domains (e.g., noisy, conversation-style speech) and the minimum amount of paired data required to build well-performing tokenizers need to be further investigated; (3) due to our computation limits, the performance of SpeechLM X-Large models are not explored. We will consider these issues as future work.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., 2019, pp. 4171–4186.
[2] L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” in Proc. 33rd Conf. Neural Inf. Process. Syst., 2019, pp. 13063–13075. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/c02b0da5ddc1df7138b349f99ac5-Paper.pdf
[3] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” 2019, arXiv:1904.05862.
[4] W.-N. Hsu, B. Bolte, Y.-H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 29, pp. 3451–3460, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9585401.
[5] P. Tremblay and A. S. Dick, “Broca and Wernicke are dead, or moving past the classic model of language neurobiology,” Brain Lang., vol. 162, pp. 60–71, 2016.
[6] A. Baevski et al., “SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training,” 2021, arXiv:2110.10239.
[7] A. Baevski et al., “msSLAM: Massively multilingual joint pre-training for speech and text,” 2022, arXiv:2202.01374.
[8] Z. Chen et al., “MAESTRO: Matched speech text representations through modality matching,” 2022, arXiv:2204.03409.
[9] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “SpanBERT: Improving pre-training by representing and predicting spans,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 64–77, 2020.
[10] S.-W. Yang et al., “SUPERB: Speech processing universal performance benchmark,” 2021, arXiv:2105.01051.
[11] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in Proc. Int. Conf. Learn. Representations, 2020.
[12] Y.-A. Chung et al., “W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in Proc. IEEE Autom. Speech Recognit. Understanding Workshop, 2021, pp. 244–250.
[13] C. Wang et al., “Improving self-supervised learning for speech recognition with intermediate layer supervision,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Singapore, 2022, pp. 7092–7096. doi: 10.1109/ICASSP43922.2022.9747022
[14] C. Wang et al., “Supervision-guided codebooks for masked prediction in speech pre-training,” 2022, arXiv:2206.10125.
[15] M. Kim, G. Kim, S.-W. Lee, and J.-W. Ha, “St-BERT: Cross-modal language model pre-training for end-to-end spoken language understanding,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2021, pp. 7478–7482.
A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, “Unsupervised speech translation and recognition,” in Proc. 50th Annu. Meeting Assoc. Comput. Linguistics, Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 5732–5738. [Online]. Available: https://aclanthology.org/2022.acl-long.393

Y. Tang et al., “Unified speech-text pre-training for speech translation and recognition,” in Proc. Conf. Empirical Methods Natural Lang. Process., Abu Dhabi, UAE: Association for Computational Linguistics, 2022, pp. 1488–1499. [Online]. Available: https://aclanthology.org/2022.acl-long.105

Z. Zhang et al., “SpeechUT: Bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training,” in Proc. Conf. Empirical Methods Natural Lang. Process., Abu Dhabi, UAE: Association for Computational Linguistics, 2022, pp. 5504–5520. [Online]. Available: https://proceedings.neurips.cc/paper/2022/hash/6f76a523335925025a02f40a933031cc-Abstract.html

A. Vaswani et al., “Attention is all you need,” in Proc. 31st Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010. doi: 10.5555/3295222.3295349.

P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., 2018, pp. 464–468.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in Proc. 23rd Int. Conf. Mach. Learn., 2006, pp. 369–376. doi: 10.1145/1143844.1143891.

W. Wang et al., “Optimizing alignment of speech and language latent spaces for end-to-end speech recognition and understanding,” in Proc. IEEE Conf. Acoust. Speech Signal Process., 2022, pp. 7802–7806.

M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” Comput. Speech Lang., vol. 16, no. 1, pp. 69–88, 2002.

D. Povey et al., “The kaldi speech recognition toolkit,” in Proc. IEEE Workshop Autom. Speech Recognit. Understanding, 2011. [Online]. Available: http://infoscience.epfl.ch/record/192584

Y. Ren et al., “FastSpeech: Fast, robust and controllable text to speech,” in Proc. Adv. Neural Inf. Process. Syst., 2019, Art. no. 285.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2015, pp. 5206–5210.

J. Kahn et al., “Libri-light: A benchmark for ASR with limited or no supervision,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2020, pp. 7669–7673.

C. Wang, A. Wu, and J. Pino, “CoVoST 2 and massively multilingual speech-to-text translation,” 2020, arXiv:2007.10357.

R. Ardila et al., “Common voice: A massively-multilingual speech corpus,” in Proc. 12th Conf. Lang. Resour. Eval., 2020, pp. 4211–4215.

A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, “Unsupervised speech recognition,” in Proc. Adv. Neural Inf. Process. Syst., Curran Associates, Inc., 2021, pp. 27826–27839. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/ea1594dc9788f2cfac311592613b7f71ff-Paper.pdf

Y. Yang et al., “Towards universal speech discrete tokens: A case study for ASR and TTS,” 2023, arXiv:2309.07377.

S. Chen et al., “Wav2LM: Large-scale self-supervised pre-training for full stack speech processing,” IEEE J. Sel. Topics Signal Process., vol. 16, no. 6, pp. 1505–1518, Oct. 2022.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.