SPLAT: Speech-Language Joint Pre-Training for Spoken Language Understanding

Yu-An Chung[1*], **Chenguang Zhu**[2*], **Michael Zeng**[2]
[1]MIT Computer Science and Artificial Intelligence Laboratory
[2]Microsoft Cognitive Services Group
andyyuan@mit.edu,{chezhu,nzeng}@microsoft.com

Abstract

Spoken language understanding (SLU) requires a model to analyze input acoustic signal to understand its linguistic content and make predictions. To boost the models' performance, various pre-training methods have been proposed to learn rich representations from large-scale unannotated speech and text. However, the inherent disparities between the two modalities necessitate a mutual analysis. In this paper, we propose a novel semi-supervised learning framework, SPLAT, to jointly pre-train the speech and language modules. Besides conducting a self-supervised masked language modeling task on the two individual modules using unpaired speech and text, SPLAT aligns representations from the two modules in a shared latent space using a small amount of paired speech and text. Thus, during fine-tuning, the speech module alone can produce representations carrying both acoustic information and contextual semantic knowledge of an input acoustic signal. Experimental results verify the effectiveness of our approach on various SLU tasks. For example, SPLAT improves the previous state-of-the-art performance on the Spoken SQuAD dataset by more than 10%.

1 Introduction

Spoken language understanding (SLU) tackles the problem of comprehending audio signals and making predictions related to the content. SLU has been widely employed in various areas such as intent understanding (Tur and De Mori, 2011; Bhargava et al., 2013; Ravuri and Stolcke, 2015; Lugosch et al., 2019), question answering (Lee et al., 2018; Chuang et al., 2020), and sentiment analysis (Zadeh et al., 2018). Early approaches leverage a two-step pipeline: use automatic speech recognition (ASR) to transcribe input audio into text, and then employ language understanding models to produce results. However, such cascaded system has several drawbacks. First, the transcription produced by the ASR module often contains errors, which adversely affects the language understanding module's prediction accuracy. Second, even if the transcription is perfect, the rich prosodic information of speech (e.g., tempo, pitch, and intonation) is inevitably lost after ASR. In comparison, humans often leverage these information to better understand and disambiguate the content. Therefore, there has been a rising trend of end-to-end approaches to retain information from audio signals to carry out the understanding task (Serdyuk et al., 2018; Chen et al., 2018; Haghani et al., 2018).

While end-to-end SLU methods are effective, they often suffer from a shortage of labeled training data, especially when the target task is in a novel domain. One solution is to leverage self-supervised training as is done in pre-trained language models. Examples like BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and RoBERTa (Liu et al., 2019) are first pre-trained on large-scale unannotated text in a self-supervised fashion to learn rich textual representations before being fine-tuned on downstream tasks with a modest amount of labeled data. Borrowing this idea, several pre-training methods have been proposed for speech, e.g., wav2vec (Schneider et al., 2019; Baevski et al., 2020a), contrastive predictive coding (Oord et al., 2018; Rivière et al., 2020), autoregressive predictive coding (Chung et al., 2019a, 2020; Chung and Glass, 2020b), and DeCoAR (Ling et al., 2020; Ling and Liu, 2020), to capture contextual representations from unlabeled speech data. Nevertheless, these methods leverage only acoustic data and mainly focus on modeling the acoustic information during pre-training. As a result, the produced representations may not be optimal for language understanding tasks.

To solve these problems, we propose a novel SPeech-LAnguage joint pre-Training framework,

[*] Equal contribution. The work was done when Yu-An Chung was interning in Microsoft.

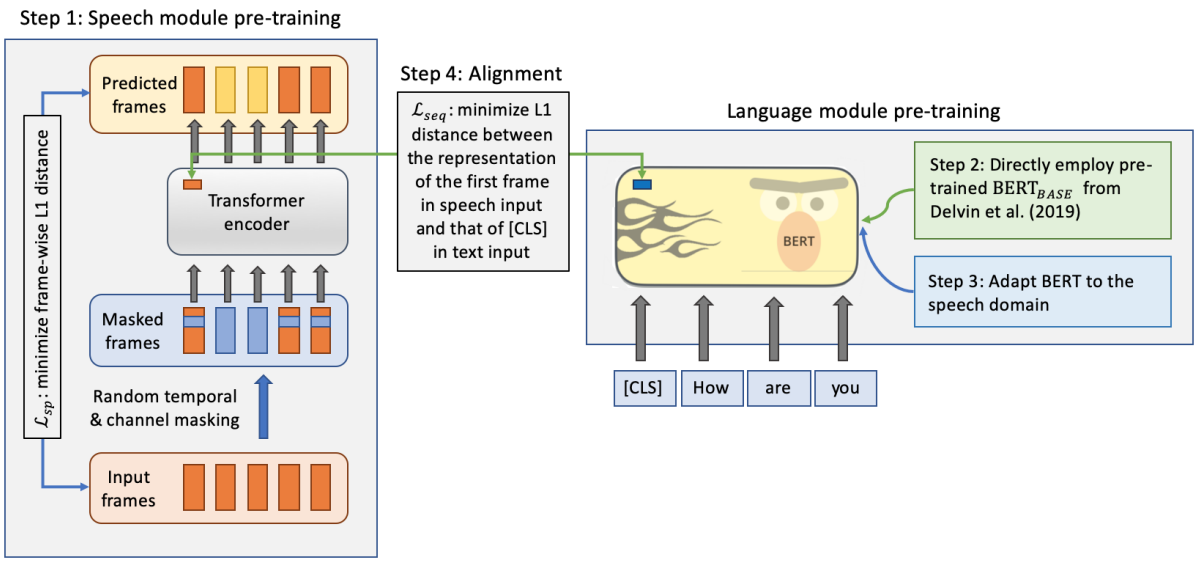

Figure 1: Overview of SPLAT. First, the speech and language modules are separately pre-trained using speech and text data via masked language modeling (MLM). In practice, we directly employ the $\text{BERT}_{\text{BASE}}$ model released by Devlin et al. (2019) to be the language module. Then, by leveraging a small amount of paired speech and text data, either a sequence-level alignment loss $\mathcal{L}_{seq}$ or a token-level alignment loss $\mathcal{L}_{tok}$ is applied to align the representations from both modules in a shared latent space (only $\mathcal{L}_{seq}$ is shown here). During alignment, the language module is kept frozen and only the speech module is updated. Before aligning the two modules, there is an optional step to update the $\text{BERT}_{\text{BASE}}$-initialized language module via MLM using the text portion from the paired data. This optional step aims to adapt the language module to the speech domain to facilitate later alignment. After pre-training, the language module is discarded and only the speech module is used in downstream tasks.

SPLAT. SPLAT contains a speech module and a language module for multi-modal understanding. The speech module is a Transformer encoder trained from scratch and the language module is initialized from BERT. Both modules leverage large-scale unannotated data for pre-training via masked language modeling. In the speech module, each frame is seen as a token and is replaced with zero vector with a certain probability. For each masked frame, we minimize the L1-distance between the predicted frame and the original frame.

Then, to make the speech module aware of the contextual information extracted from the language module, we design an alignment loss to align the representations from both modules in a shared latent semantic space. In detail, we propose two alignment methods, a sequence-level one and a token-level one, that leverage a small amount of paired speech and text to minimize the disparity between the acoustic representations from the speech module and the textual representations from the language module. In this way, the speech representations will carry not only the acoustic information but also the contextual knowledge from the text. After this alignment, when text input is absent during fine-tuning, the speech module alone can produce representations that bridge the speech input and the language understanding output.

We conduct extensive evaluations on several downstream SLU tasks, including Fluent Speech Commands for intent detection, Switchboard for dialog act classification, CMU-MOSEI for spoken sentiment analysis, and Spoken SQuAD for spoken question answering. SPLAT achieves superior results in all datasets. For example, SPLAT improves the previous state-of-the-art performance on the Spoken SQuAD dataset by more than 10%. Furthermore, we show that SPLAT can perform well even given just a tiny portion of the labeled training data in downstream tasks.

2 Related Work

Spoken language understanding In recent years, due to its flexibility and effectiveness, end-to-end spoken language understanding (SLU) has been proposed and applied to various tasks (Qian et al., 2017; Serdyuk et al., 2018; Lugosch et al., 2019). For instance, Qian et al. (2017) use an auto-encoder to initialize the SLU model. Lugosch et al. (2019) pre-train the model to recognize words and

phonemes, and then fine-tune it on downstream tasks. Chen et al. (2018) pre-train the model to categorize graphemes, and the logits are fed into the classifier. In most of these approaches, the model pre-training requires annotated speech, e.g., word or phonemes corresponding to audio signals. As a result, the massive unlabeled speech data cannot be utilized by these models.

Self-supervised pre-training for language Pre-trained models have achieved great success in both language and speech domains. In language, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), UniLM (Dong et al., 2019), and BART (Lewis et al., 2020) have been successfully applied to natural language inference (Zhang et al., 2020b), question answering (Zhu et al., 2018), and summarization (Zhu et al., 2019). These pre-trained models leverage self-supervised tasks such as masked language modeling (MLM), next sentence prediction, and de-noising autoencoder.

Self-supervised pre-training for speech In speech, wav2vec (Schneider et al., 2019) leverages contrastive learning to produce contextual representations for audio input; vq-wav2vec (Baevski et al., 2020a) and wav2vec 2.0 (Baevski et al., 2020b) further propose to discretize the original continuous audio signals in order to enable more efficient MLM training with Transformer (Vaswani et al., 2017). Pre-trained speech models have been applied to ASR (Ling et al., 2020; Chung and Glass, 2020a; Baevski et al., 2020b), phoneme recognition (Song et al., 2020; Liu et al., 2020a), speech translation (Nguyen et al., 2020; Chung et al., 2019c), and speech synthesis (Chung et al., 2019b), to name a few.

Nevertheless, an SLU model must incorporate both acoustic and language understanding capabilities to project speech signals to semantic outputs. Thus, a pre-trained model for SLU needs to address tasks beyond a single modality.

Speech and language joint pre-training Recently, SLU applications have prompted joint pre-training on both speech and text data. SpeechBERT (Chuang et al., 2020) applies MLM to pairs of audio and transcripts. However, there are several crucial differences to compared to our work. First, SpeechBERT contains a phonetic-semantic embedding module that requires forced alignment to first segment speech into word segments to obtain. Second, both the pre-training and fine-tuning phases of SpeechBERT require both speech and text input, since it is designed for a specific spoken question answering task. However, many SLU tasks only take speech as input, which does not align with the design of SpeechBERT. In contrast, our model can learn to align acoustic and textual representations using just (a small amount of) paired data during pre-training, and only needs speech input for downstream tasks.

Denisov and Vu (2020) propose to align speech and language embeddings in a method similar to ours. However, there are several key differences. First, Denisov and Vu (2020) employ the encoder of a pre-trained ASR model, which already requires plentiful of annotated speech to obtain. Our model, on the other hand, conducts self-supervised learning to pre-train the speech module using unannotated speech. Secondly, besides sequence-level alignment, we propose a token-level alignment method, which is suitable for token-level downstream tasks. Last but not least, our model uses a much smaller paired speech and text for alignment (10 hours) than Denisov and Vu (2020) (1,453 hours), yet still largely outperforms their method in intent detection and dialog act classification.

3 Method

In this section we present SPLAT, a framework for learning joint contextual representations of speech and language. The model consists of a speech module and a language module that share a similar architecture and learning algorithm. The pre-training of SPLAT is divided into two steps. First, we individually pre-train the speech and language modules using unannotated speech and text, respectively. Then, we leverage a simple yet effective alignment task that uses only a small amount of paired speech and text data to align the representations from both modules in a shared latent semantic space such that the information learned by the language module is transferred to the speech module. After pre-training, the language module is discarded and only the speech module is used in downstream tasks.

Below we formally describe the procedures for pre-training the speech (§3.1) and language modules (§3.2), and the alignment loss (§3.3) for aligning the representations from the two modules. Figure 1 provides an overview of the pre-training procedures of SPLAT.

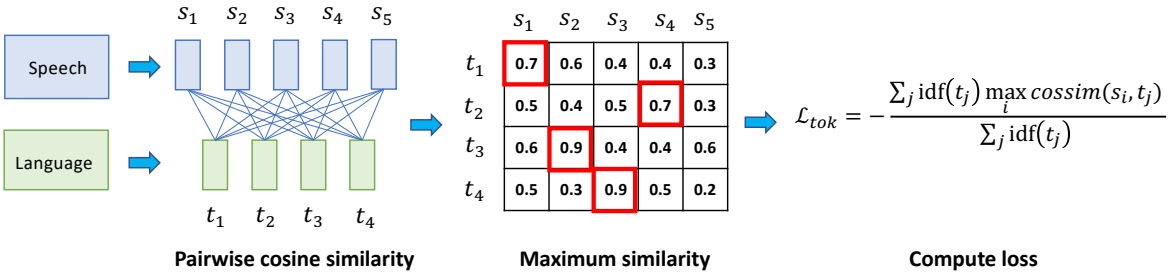

Figure 2: Token-level alignment between speech and language modules. $(\boldsymbol{s}_1, ..., \boldsymbol{s}_5)$ are the output embeddings of the speech module and $(\boldsymbol{t}_1, ..., \boldsymbol{t}_4)$ are those of the language module.

3.1 Speech module pre-training

The goal of this module is to leverage unlabeled speech data to learn representations that capture meaningful acoustic information about speech utterances such as their phonetic content and speaker characteristics. Formally, the input to the speech module is a 80-dimensional log Mel spectrogram, $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$, where $\boldsymbol{x}_i \in \mathbb{R}^{80}, 1 \leq i \leq n$. The speech module, which is implemented as a Transformer architecture, then produces hidden representations $(\boldsymbol{s}_1, ..., \boldsymbol{s}_n)$ and predictions $(\hat{\boldsymbol{x}}_1, ..., \hat{\boldsymbol{x}}_n)$, where $\boldsymbol{s}_i \in \mathbb{R}^{768}$ and $\hat{\boldsymbol{x}}_i \in \mathbb{R}^{80}$.

To boost its capacity for contextual understanding, we borrow the idea of masked language modeling (MLM) (Devlin et al., 2019; Liu et al., 2020c; Wang et al., 2020; Liu et al., 2020b). Specifically, each audio frame $\boldsymbol{x}_i$ is replaced with a zero vector with a probability of 15%. The corresponding output $\hat{\boldsymbol{x}}_i$ is trained to be close to the original frame $\boldsymbol{x}_i$ via minimizing their L1-distance. Additionally, since consecutive frames are highly correlated, it is possible that the model simply utilizes the local smoothness of speech signals for reconstructing a single frame and thus fails to capture useful information. To avoid such issue, when a frame $\boldsymbol{x}_i$ is selected to be masked, its following three frames $\boldsymbol{x}_{i+1}$, $\boldsymbol{x}_{i+2}$, and $\boldsymbol{x}_{i+3}$ are also masked, and the model is asked to reconstruct all these masked frames.

Furthermore, according to SpecAugment (Park et al., 2019), the input features $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ can be seen as comprising two dimensions: time, i.e., the subscript i, and channel, i.e., the elements in each $\boldsymbol{x}_i$. While conventional MLM masks along certain time steps, the input signals can also be masked along the channel dimension. In other words, each column vector $[\boldsymbol{x}_{1,j}, ..., \boldsymbol{x}_{n,j}]$ for $1 \leq j \leq 80$ has a 15% of chance to be masked, i.e., replaced with a zero vector. This channel masking is combined with temporal masking to reinforce the model's capability to utilize contextual information from both time and channel, and reduce the impact of co-adaptation between acoustic frames. The final pre-training objective for the speech module is to reconstruct the entire input sequence from the altered version of it:

$$\mathcal{L}_{sp} = \sum_{i=1,2,...,n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_1 \tag{1}$$

We use the speech portion of the train-clean-360 subset from the LibriSpeech corpus (Panayotov et al., 2015) to pre-train the speech module, i.e., to minimize $\mathcal{L}_{sp}$. This subset contains 360 hours of read speech produced by 921 speakers. We follow the standard Kaldi setting, using a frame size of 25ms and a time shift of 10ms for generating the 80-dimensional log Mel spectrograms. The spectrograms are normalized to zero mean and unit variance per speaker.

3.2 Language module pre-training

The language module aims to offer contextual understanding for text input. We directly employ the $\text{BERT}_{\text{BASE}}$ model released by Devlin et al. (2019), which is pre-trained on a large text corpus with the MLM task and contains rich textual representations, as the language module. We denote the cross-entropy loss for the language MLM task as $\mathcal{L}_{text}$.

Given input token embeddings $(\boldsymbol{y}_1, ..., \boldsymbol{y}_m)$, where $\boldsymbol{y}_1$ corresponds to the [CLS] token, the module produces contextual representations $(\boldsymbol{t}_1, ..., \boldsymbol{t}_m)$, where $\boldsymbol{t}_j \in \mathbb{R}^{768}, 1 \leq j \leq m$.

3.3 Aligning speech and language representations

The input to most SLU tasks consists of only audio signals, but the model is required to conduct semantic understanding, which can be best handled when

textual information is present. Therefore, we propose to align the pre-trained speech and language representations in a shared semantic latent space.

Suppose a pair of speech and text data consisting of an acoustic feature sequence $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ and its transcript $(\boldsymbol{y}_1, ..., \boldsymbol{y}_m)$. The speech and language modules separately produce the output representations $(\boldsymbol{s}_1, ..., \boldsymbol{s}_n)$ and $(\boldsymbol{t}_1, ..., \boldsymbol{t}_m)$. We then propose two methods to align the embeddings from the modules: sequence-level and token-level alignment.

Sequence-level alignment For sequence-level alignment, we treat the first embeddings from the two output representations, i.e., $\boldsymbol{s}_1$ and $\boldsymbol{t}_1$, as the sequence-level representations of their respective sequences, and minimize their L1-distance:

$$\mathcal{L}_{seq} = \|\boldsymbol{s}_1 - \boldsymbol{t}_1\|_1 \tag{2}$$

Since our goal is to transfer the textual knowledge contained by the language module to the speech module, we only update the speech module to minimize $\mathcal{L}_{seq}$ and keep the language module fixed.

After pre-training, when the transcript is absent in downstream tasks, the first output embedding of the speech module $\boldsymbol{s}_1$ will still be close to its corresponding text embedding $\boldsymbol{t}_1$ from the language module, as if the transcript were given. It follows that $\boldsymbol{s}_1$ can then be used to predict the property of the whole audio input, e.g., intent classification.

Token-level alignment To achieve a finer level of alignment, each audio feature should be compared with its each text token. Although forced alignment (Gorman et al., 2011) can establish this correspondence between audio signals and individual words, it requires a pre-trained ASR system to obtain. Here we propose a method that automatically aligns audio features with textual tokens.

Inspired by BERTScore (Zhang et al., 2020a), for each output text embedding $\boldsymbol{t}_j$, we first compute its cosine similarity with each output acoustic embedding $\boldsymbol{s}_i$, and select the acoustic feature with the highest similarity. Then, the alignment is performed by maximizing the sum of these maximum similarities over all tokens, weighted by each token's inverse document frequency (idf) to reduce the impact of common words:

$$\mathcal{L}_{tok} = -\frac{\sum_{j=1}^{m} \text{idf}(\boldsymbol{t}_j) \max_i \text{cossim}(\boldsymbol{s}_i, \boldsymbol{t}_j)}{\sum_{j=1}^{m} \text{idf}(\boldsymbol{t}_j)} \tag{3}$$

The token-level alignment loss is illustrated in Figure 2. Same as $\mathcal{L}_{seq}$, when minimizing $\mathcal{L}_{tok}$, the language module is kept fixed and only the speech module is updated.

Algorithm 1 Pre-training SPLAT

Input: An unlabeled speech corpus $\mathcal{X} = \{\boldsymbol{x}^{(p)}\}_{p=1}^{N}$, an unlabeled text corpus $\mathcal{Y} = \{\boldsymbol{y}^{(q)}\}_{q=1}^{M}$, and a paired speech-text corpus $\mathcal{Z} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^{K}$, where $K \ll N, M$.

1: Use $\mathcal{X}$ to train the speech module by minimizing $\mathcal{L}_{sp}$ (Equation 1).
2: Use $\mathcal{Y}$ to train the language module by minimizing $\mathcal{L}_{text}$ (we directly employ $\text{BERT}_{\text{BASE}}$ from Devlin et al. (2019) for this step).
3: Use $\{\boldsymbol{y}^{(k)}\}_{k=1}^{K}$ from $\mathcal{Z}$ to train the language module by minimizing $\mathcal{L}_{text}$.
4: Use $\mathcal{Z}$ to align the two modules by minimizing $\mathcal{L}_{seq}$ (Equation 2) or $\mathcal{L}_{tok}$ (Equation 3).
5: Discard the language module.

Output: The final speech module.

To minimize the alignment loss, we randomly sample 10 hours of audio paired with its transcripts from the train-clean-360 subset, of which the speech portion is used to pre-train the speech module (§ 3.1). In practice, before minimizing the alignment loss, we find it beneficial to train (i.e., minimize $\mathcal{L}_{text}$) the language module initialized with $\text{BERT}_{\text{BASE}}$ with the 10-hour LibriSpeech transcripts with the MLM task. This step allows the model to adapt to the speech domain and facilitates the following alignment task.

We summarize the complete procedure of pre-training SPLAT in Algorithm 1. After pre-training, the language module is discarded and only the speech module is used in downstream tasks.

4 Experiment Setup

4.1 Baselines

We include a number of strong baselines from recent literature for each downstream task (Lugosch et al., 2019; Duran and Battle, 2018; Ghosal et al., 2018; Chuang et al., 2020). We also compare with another speech-language joint pre-training framework (Denisov and Vu, 2020). For each baseline, the reported performance is achieved by system that either uses similar or more amounts of data than our model.

To verify the effectiveness of each component in SPLAT, we experiment with the following variants of it, including whether to pre-train the model,

Table 1: Variants of SPLAT. An ✗ indicates that the variant does not incorporate this step during pre-training. The step numbers correspond to those listed in Algorithm 1.

Model variant	Step 1. Pre-train speech module	Step 2. Pre-train language module	Step 3. Adapt language module before alignment	Step 4. Type of alignment loss
SPLAT-Scratch	✗	✗	✗	✗
SPLAT-Speech	✓	✗	✗	✗
SPLAT-Seq	✓	✓	✗	$\mathcal{L}_{seq}$
SPLAT-Seq-MLM	✓	✓	✓	$\mathcal{L}_{seq}$
SPLAT-Tok	✓	✓	✗	$\mathcal{L}_{tok}$
SPLAT-Tok-MLM	✓	✓	✓	$\mathcal{L}_{tok}$

Table 2: Summary of SLU datasets. For the rows of Train, Validation, and Test, the numbers indicate the number of utterances in the split.

Task	Intent detection	Dialog act classification	Spoken sentiment analysis	Spoken question answering
Dataset	FSC	SwBD	CMU-MOSEI	Spoken SQuAD
Num. of classes	31	42	7	-
Train/val/test	23.1k/3.1k/3.8k	97.8k/8.6k/2.5k	16.2k/1.8k/4.6k	35.1k/2.0k/5.4k

whether to use the language module and which alignment task to apply. Table 1 summarizes the considered model variants.

- **SPLAT-Scratch**: No pre-training is conducted at all. Speech module is trained from scratch on downstream tasks.
- **SPLAT-Speech**: Only the speech module is pre-trained. Language module and alignment loss are not incorporated.
- **SPLAT-Seq**: SPLAT with sequence-level alignment loss $\mathcal{L}_{seq}$, but language module is not trained on LibriSpeech transcripts with MLM before alignment.
- **SPLAT-Seq-MLM**: SPLAT with sequence-level alignment loss $\mathcal{L}_{seq}$, and language module is trained on LibriSpeech transcripts with MLM before alignment.
- **SPLAT-Tok**: SPLAT with token-level alignment loss $\mathcal{L}_{tok}$, but language module is not trained on LibriSpeech transcripts with MLM before alignment.
- **SPLAT-Tok-MLM**: SPLAT with token-level alignment loss $\mathcal{L}_{tok}$, and language module is trained on LibriSpeech transcripts with MLM before alignment.

The speech module of SPLAT is a 3-layer Transformer encoder where each layer has a hidden size of 768 and 12 self-attention heads. The language module is directly initialized from the pre-trained $\text{BERT}_{\text{BASE}}$ released by Devlin et al. (2019).

4.2 Downstream SLU Tasks

We evaluate our model on four different SLU applications: intent detection, dialog act classification, spoken sentiment analysis, and spoken question answering. The first three belong to multi-class classification tasks, and the last one is a span prediction problem, which will be described in more detail below. Table 2 summarizes the used dataset for each application. For all datasets, we use 80-dimensional log Mel spectrograms as input acoustic features as in the pre-training stage.

Intent detection We use the Fluent Speech Commands corpus (FSC) (Lugosch et al., 2019) for intent detection, where the goal is to correctly predict the intent of an input utterance. In this dataset, each utterance is annotated with three slots: action, object, and location, where each slot can take one of multiple values. The combination of slot values is defined as the intent of the utterance, and there are 31 unique intents in total. In this work we follow the original paper to formulate intent detection as a simple 31-class classification task.

Dialog act classification We use the NTX-format Switchboard corpus (SwDA) (Calhoun et al., 2010), a dialog corpus of 2-speaker conversations. The goal is to correctly classify an input

Table 3: Results on all downstream datasets. All numbers of our models are an average of three runs, of which variances are negligibly small and not included. The metric is classification accuracy for FSC, SwBD and CMU-MOSEI. The metric for Spoken SQuAD is Audio Overlapping Score (AOS).

Model	FSC	SwBD	CMU-MOSEI	Spoken SQuAD
Ours				
SPLAT-Scratch	97.6	65.8	68.8	30.4
SPLAT-Speech	99.5	67.5	69.0	57.7
SPLAT-Seq	99.5	74.6	72.5	62.7
SPLAT-Seq-MLM	99.5	**76.3**	74.7	**65.9**
SPLAT-Tok	99.2	71.2	70.4	58.0
SPLAT-Tok-MLM	99.2	72.7	71.2	63.8
SPLAT-Seq-MLM 1-hour	99.5	75.8	65.3	65.3
Baselines				
Lugosch et al. (2019)	98.8	-	-	-
Duran and Battle (2018)	-	75.5	-	-
Ghosal et al. (2018)	-	-	**75.9**	-
Chuang et al. (2020)	-	-	-	59.7
Denisov and Vu (2020)	**95.5**	60.2	-	-

utterance into one of the 42 dialog acts.

Spoken sentiment analysis We use the CMU-MOSEI dataset (Zadeh et al., 2018), where each utterance is annotated for a sentiment score on a $[-3, 3]$ Likert scale: [-3: highly negative, -2: negative, -1: weakly negative, 0: neutral, +1: weakly positive, +2: positive, +3: highly positive]. We treat the task as a 7-class classification problem. And we only use audio signals in the input data.

For the above three tasks, during fine-tuning, an MLP network with one hidden layer of 512 units is appended on top of the speech module. It converts the output representation of the first frame, i.e., $\boldsymbol{s}_1$, for class prediction. Both the pre-trained speech module and the randomly initialized MLP are fine-tuned on the training set for 10 epochs with a batch size of 64 and a fixed learning rate of 3e-4. We compute classification accuracy after each training epoch and pick the best-performing checkpoint on the validation set to report results on the test set.

Spoken question answering We use the Spoken SQuAD dataset (Li et al., 2018), which is augmented[1] from SQuAD (Rajpurkar et al., 2016) for spoken question answering. The model is given an article in the form of speech and a question in the form of text. The goal is to predict a time span in the spoken article that answers the question. In other words, the model outputs an audio segment extracted from spoken article as the answer. The model is evaluated by Audio Overlapping Score (AOS) (Li et al., 2018): the greater the overlap between the predicted span and the ground-truth answer span, the higher the score will be.

During fine-tuning, given a spoken article and a question in the text form, the pre-trained speech module extracts audio representations of the article and pass them to a randomly initialized 3-layer Transformer encoder along with the tokenized textual question as input. The Transformer then uses the self-attention mechanism to implicitly align elements of the input audio and textual features. For each time step of the audio input, the Transformer is trained to predict whether this is the start of the span with a simple logistic regression. A separate classifier is used for predicting the end of the span.

5 Results and Analysis

5.1 Main results

Table 3 shows the performance of models on all four downstream tasks. Each number from our model is an average over three runs. Based on the results, we make the following observations.

Firstly, compared with **SPLAT-Scratch**, all pre-trained models achieve superior results, especially more than 30% gain on Spoken SQuAD, proving the effectiveness of pre-training.

Secondly, the inclusion of language module and the alignment task during pre-training is very beneficial. For instance, on CMU-MOSEI, **SPLAT-**

[1] Li et al. (2018) used Google text-to-speech to generate the spoken version of the articles in SQuAD.

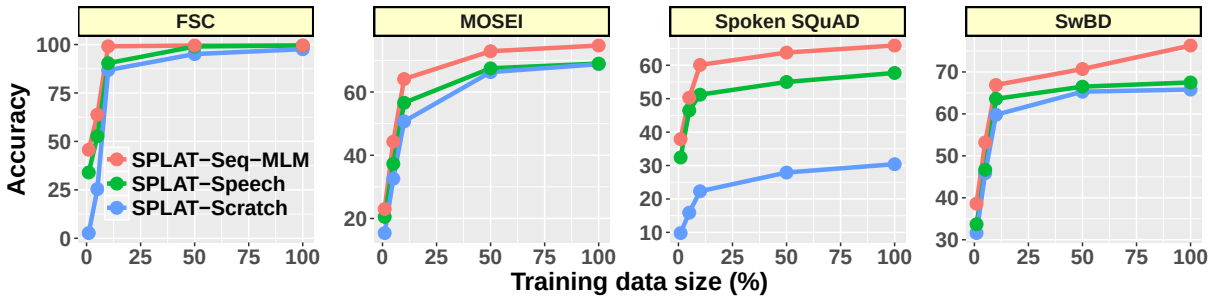

Figure 3: Performance on downstream tasks with varying training data sizes. All numbers are an average of three runs, of which variances are negligibly small and not included.

Seq-MLM outperforms **SPLAT-Speech** by 5.7%, and outperforms several baseline systems from recent literature. We argue that as SLU tasks require the model to interpret acoustic signals and their underlying semantics, the language module will guide the speech module towards a mutual understanding of both modalities via our alignment task.

Thirdly, updating the language module using MLM during pre-training is helpful. Although the language module has been initialized with BERT, adaptation to the speech domain can help with semantic understanding in the downstream task.

Types of alignment Comparing **SPLAT-Seq** against **SPLAT-Tok**, we find that sequence-level alignment outperforms token-level alignment on all four tasks, although the latter is supposed to learn more fine-grained multi-modal representations. We leave the investigations of reasons for such phenomenon and more advanced token-level alignment approaches for future work.

Low-resource scenario We experiment with a version of SPLAT that uses only 1 hour of transcribed speech randomly sampled from the LibriSpeech train-clean-360 subset for aligning speech and language modules, denoted as **SPLAT-Seq-MLM 1-hour**. The language module of **SPLAT-Seq-MLM 1-hour**—after being initialized with $\text{BERT}_{\text{BASE}}$—is trained on the 1-hour LibriSpeech transcripts before minimizing the alignment loss. It achieves comparable results with the best variant **SPLAT-Seq-MLM**: same accuracy on FSC, 0.5% less on SwBD, and 0.6% less on Spoken SQuAD. This shows that with a small amount of labeled speech data, our pre-training framework can achieve good results on downstream tasks.

5.2 Robustness to the size of downstream training data

As human labeling is time-consuming and labor-intensive, the amount of labeled training data for downstream tasks is often small and insufficient. In this section, we show that with effective pre-training, the model will be less dependent on the amount of downstream labeled data.

We randomly sample 50%, 10%, 5%, and 1% of the training data in the downstream tasks, and evaluate the performance of different variants of SPLAT when fine-tuned on the sampled data.

Figure 3 shows the performance on all four downstream tasks with varying training data sizes. We observe that among the variants, **SPLAT-Seq-MLM** is least sensitive to training data sizes. For instance, in FSC, with only 10% of the training data, its accuracy only drops 0.4 points. In comparison, both **SPLAT-Scratch** and **SPLAT-Speech** drops about 10 points. And the gaps are in general larger when the size of training data further shrinks. Therefore, our proposed joint pre-training of speech and language modules can help the model quickly adapt to downstream tasks given a modest amount of training data.

5.3 The geometry of the speech latent space before and after alignment

So far we have empirically demonstrated the effectiveness of SPLAT for learning multi-modal speech-language representations that are useful in various SLU tasks. Here we further show that our sequence-level alignment loss (Equation 2) can help project two speech utterances that have similar textual embeddings to nearby points in the speech latent space.

Recall that we use the embedding of the first token/feature to represent an utterance and conduct sequence-level alignment (Equation 2). Sup-

Table 4: Average cosine similarity between all pairs of speech embeddings (S_{avg}), and the average cosine similarity between a speech embedding $\boldsymbol{s}_1^{(p)}$ and that of an utterance whose textual embedding is closest to the corresponding textual embedding $\boldsymbol{t}_1^{(p)}$ ($S_{closest}$).

Model	S_{avg}	$S_{closest}$
SPLAT-Speech	0.136	0.238
SPLAT-Seq	0.144	0.781
SPLAT-Seq-MLM	0.148	0.829

pose $\boldsymbol{t}_1^{(p)}$ and $\boldsymbol{s}_1^{(p)}$ correspond to the textual and speech embeddings of the first utterance by SPLAT and $\boldsymbol{t}_1^{(q)}$ and $\boldsymbol{s}_1^{(q)}$ correspond to the embeddings of the second utterance. Then, if $\boldsymbol{t}_1^{(p)} \approx \boldsymbol{t}_1^{(q)}$, our SPLAT model trained with the sequence-level alignment loss will produce $\boldsymbol{s}_1^{(p)} \approx \boldsymbol{s}_1^{(q)}$.

We use the dev-clean subset from the LibriSpeech corpus for the analysis. First, we compute the average pairwise cosine similarity between the utterances of all speech embeddings:

$$S_{avg} = \frac{1}{K(K-1)/2} \sum_{p=2}^{K} \sum_{q=1}^{p-1} \text{cossim}(\boldsymbol{s}_1^{(p)}, \boldsymbol{s}_1^{(q)}), \tag{4}$$

where K is the number of utterances in dev-clean.

Next, for each utterance with its speech and textual embeddings denoted as $\boldsymbol{s}_1^{(p)}$ and $\boldsymbol{t}_1^{(p)}$ respectively, we first use $\boldsymbol{t}_1^{(p)}$ to retrieve the utterance with the most similar textual embedding $\boldsymbol{t}_1^{(q^*)}$, i.e., $q^* = \text{argmax}_{1 \le q \le K, q \ne p} \text{cossim}(\boldsymbol{t}_1^{(p)}, \boldsymbol{t}_1^{(q)})$. We then compute the cosine similarity between $\boldsymbol{s}_1^{(p)}$ and $\boldsymbol{s}_1^{(q^*)}$ and take the average of such value over all utterances in dev-clean:

$$S_{closest} = \frac{1}{K} \sum_{p=1}^{K} \text{cossim}(\boldsymbol{s}_1^{(p)}, \boldsymbol{s}_1^{(q^*)}). \tag{5}$$

We show the S_{avg} and $S_{closest}$ of embeddings produced by **SPLAT-Speech**, **SPLAT-Seq**, and **SPLAT-Seq-MLM** in Table 4.

We see that S_{avg} is approximately the same for all model variants. However, $S_{closest}$, the average similarity between the speech embeddings of two linguistically similar utterances, increases from 0.238 to 0.781 after aligning the speech and language modules, and further increases to 0.829 after adapting the language module on LibriSpeech transcripts with MLM before the alignment. Overall, SPLAT can make a pair of semantically similar utterances to have much closer speech embeddings, compared with other random pairs of utterances.

These results demonstrate that via an cross-modal alignment loss as simple as Equation 2, SPLAT can effectively transfer knowledge from the language module to the speech module to capture both acoustic and linguistic information of speech utterances.

6 Conclusions

Spoken language understanding (SLU) tasks require an understanding of the input audio signal and its underlying semantics. In this paper, we present a novel speech-language joint pre-training framework, SPLAT, to carry out both speech and language understanding tasks during pre-training. Besides a self-supervised training on the speech and language modules, we propose two methods to align the semantic representations from both modules using a modest amount of labeled speech data. The speech module can quickly adapt to downstream tasks and achieve superior results on various SLU datasets including intent detection, dialog act classification, spoken sentiment analysis, and spoken question answering. This joint pre-training also makes the model less sensitive to the amount of labeled training data in downstream domains.

For future work, we plan to integrate automatic speech recognition and natural language generation into our framework to achieve good results on spoken language generation tasks.

References

Alexei Baevski, Steffen Schneider, and Michael Auli. 2020a. vq-wav2vec: Self-supervised learning of discrete speech representations. In *ICLR*.

Alexei Baevski, Yuhao Zhou, Abdel-rahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*.

Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya. 2013. Easy contextual intent prediction and slot detection. In *ICASSP*.

Sasha Calhoun, Jean Carletta, Jason Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. 2010. The NXT-format Switchboard Corpus: A rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4):387–419.

Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore. 2018. Spoken language understanding without speech recognition. In *ICASSP*.

Yung-Sung Chuang, Chi-Liang Liu, and Hung-Yi Lee. 2020. SpeechBERT: Cross-modal pre-trained language model for end-to-end spoken question answering. In *Interspeech*.

Yu-An Chung and James Glass. 2020a. Generative pre-training for speech with autoregressive predictive coding. In *ICASSP*.

Yu-An Chung and James Glass. 2020b. Improved speech representations with multi-target autoregressive predictive coding. In *ACL*.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019a. An unsupervised autoregressive model for speech representation learning. In *Interspeech*.

Yu-An Chung, Hao Tang, and James Glass. 2020. Vector-quantized autoregressive predictive coding. In *Interspeech*.

Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and RJ Skerry-Ryan. 2019b. Semi-supervised training for improving data efficiency in end-to-end speech synthesis. In *ICASSP*.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. 2019c. Towards unsupervised speech-to-text translation. In *ICASSP*.

Pavel Denisov and Ngoc Thang Vu. 2020. Pretrained semantic speech embeddings for end-to-end spoken language understanding via cross-modal teacher-student learning. In *Interspeech*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Nathan Duran and Steve Battle. 2018. Probabilistic word association for dialogue act classification with recurrent neural networks. In *EANN*.

Deepanway Ghosal, Md Shad Akhtar, Dushyant Chauhan, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Contextual inter-modal attention for multi-modal sentiment analysis. In *EMNLP*.

Kyle Gorman, Jonathan Howell, and Michael Wagner. 2011. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. 2018. From audio to semantics: Approaches to end-to-end spoken language understanding. In *SLT*.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018. ODSQA: Open-domain spoken question answering dataset. In *SLT*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Chia-Hsuan Li, Szu-Lin Wu, Chi-Liang Liu, and Hung-Yi Lee. 2018. Spoken SQuAD: A study of mitigating the impact of speech recognition errors on listening comprehension. In *Interspeech*.

Shaoshi Ling and Yuzong Liu. 2020. DeCoAR 2.0: Deep contextualized acoustic representations with vector quantization. *arXiv preprint arXiv:2012.06659*.

Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff. 2020. Deep contextualized acoustic representations for semi-supervised speech recognition. In *ICASSP*.

Alexander H. Liu, Yu-An Chung, and James Glass. 2020a. Non-autoregressive predictive coding for learning speech representations from local dependencies. *arXiv preprint arXiv:2011.00406*.

Andy T. Liu, Shang-Wen Li, and Hung-Yi Lee. 2020b. TERA: Self-supervised learning of transformer encoder representation for speech. *arXiv preprint arXiv:2007.06028*.

Andy T. Liu, Shu-Wen Yang, Po-Han Chi, Po-Chun Hsu, and Hung-Yi Lee. 2020c. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP*.

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*.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. In *Interspeech*.

Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Yannick Estève, and Laurent Besacier. 2020. Investigating self-supervised pre-training for end-to-end speech translation. In *Interspeech*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. LibriSpeech: An ASR corpus based on public domain audio books. In *ICASSP*.

Daniel Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Cubuk, and Quoc Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*.

Yao Qian, Rutuja Ubale, Vikram Ramanaryanan, Patrick Lange, David Suendermann-Oeft, Keelan Evanini, and Eugene Tsuprun. 2017. Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system. In *ASRU*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Suman Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In *Interspeech*.

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages. In *ICASSP*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *Interspeech*.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *ICASSP*.

Xingchen Song, Guangsen Wang, Zhiyong Wu, Yiheng Huang, Dan Su, Dong Yu, and Helen Meng. 2020. Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks. In *Interspeech*.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Weiran Wang, Qingming Tang, and Karen Livescu. 2020. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In *ICASSP*.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Weinberger, and Yoav Artzi. 2020a. BERTScore: Evaluating text generation with BERT. In *ICLR*.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. In *AAAI*.

Chenguang Zhu, Ziyi Yang, Robert Gmyr, Michael Zeng, and Xuedong Huang. 2019. Make lead bias in your favor: A simple and effective method for news summarization. *arXiv preprint arXiv:1912.11602*.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. SDNet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*.