

CIF-PT: Bridging Speech and Text Representations for Spoken Language Understanding via Continuous Integrate-and-Fire Pre-Training

Linhao Dong*, Zhecheng An*, Peihao Wu, Jun Zhang, Lu Lu, Zejun Ma

ByteDance AI Lab

{donglinhao, anzhecheng, wupeihao, zhangjun.jarry,
lulu.0314, mazejun}@bytedance.com

Abstract

Speech or text representation generated by pre-trained models contains modal-specific information that could be combined for benefiting spoken language understanding (SLU) tasks. In this work, we propose a novel pre-training paradigm termed Continuous Integrate-and-Fire Pre-Training (CIF-PT). It relies on a simple but effective frame-to-token alignment: continuous integrate-and-fire (CIF) to bridge the representations between speech and text. It jointly performs speech-to-text training and language model distillation through CIF as the pre-training (PT). Evaluated on SLU benchmark SLURP dataset, CIF-PT outperforms the state-of-the-art model by 1.94% of accuracy and 2.71% of SLU-F1 on the tasks of intent classification and slot filling, respectively. We also observe the cross-modal representation extracted by CIF-PT obtains better performance than other neural interfaces for the tasks of SLU, including the dominant speech representation learned from self-supervised pre-training.

1 Introduction

Spoken language understanding (SLU) plays a key role in speech interaction systems such as spoken dialogue systems, voice assistants, automated calling robots, etc. It focuses on extracting key information and making predictions from audio signals of human speech (Wang et al., 2005; Tur and Mori, 2011). Traditional methods decompose SLU into two cascading tasks: automated speech recognition (ASR) and natural language understanding (NLU), where audio signals are first transcribed into texts, and then processed by a text-based language understanding model. In the cascading scheme, the errors of ASR module will be accumulated in the NLU module and degrade the final performance. Moreover, predicted text of ASR module may not be the ideal interface for the language understanding task. For example, acoustic information such

as intonation and pitch that may be helpful for understanding tasks are lost after ASR. To tackle the problems above, recent researches employ end-to-end approaches for SLU (Serdyuk et al., 2018; Haghani et al., 2018; Chung et al., 2021; Arora et al., 2022), where the language understanding is directly performed from audio signals without explicitly utilizing predicted text of ASR.

For text-based language understanding tasks, pre-trained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b) and GPT (Radford et al., 2019) have achieved remarkable success. These models utilize self-supervised pre-training on large-scale unlabeled corpora to learn contextual representations in token or sentence level with rich syntactic and semantic knowledge (Liu et al., 2019a), which significantly benefit downstream tasks such as NLU during fine-tuning. This self-supervised pre-training fashion has been extended into the representative learning on speech. Researches such as wav2vec (Baevski et al., 2020), HuBERT (Hsu et al., 2021) and data2vec (Baevski et al., 2022a) focus on learning better frame-level contextual representations using unlabeled speech data, to improve the performance of ASR as well as other speech processing tasks. For end-to-end SLU, these self-supervised speech models have been proven to be powerful backbones on learning semantic representations (Wang et al., 2021; Arora et al., 2022).

The self-supervised pre-training methods for speech mainly focus on leveraging speech data to model acoustic information (Chung et al., 2021) on the frame level, while pre-trained language models work on higher token or sentence levels to encode linguistic knowledge (Liu et al., 2019a). These two kinds of representation could be combined for better benefiting downstream tasks such as SLU. The combination of speech and text representations can be performed by jointly pre-training on data of the two modalities (Chuang et al., 2020),

*Equal contribution.

or distilling one pre-trained representations into another (Kim et al., 2021). In either way the frame-level speech representation needs to be aligned with the token-level textual representation. Frame-to-token alignment methods such as forced alignment has been applied to speech-text joint pre-training (Chuang et al., 2020). However, these alignment methods mainly rely on external models or rules, and can only generate hard alignment mapping that can not be updated in end-to-end training. On the other hand, aligning frames and tokens through cross-attention (Arora et al., 2022; Zhu et al., 2022) suffers from high complexity and lack of token timestamps that synchronized to frames.

The frame-to-token alignment also plays a critical role in ASR systems. Various works, such as Connectionist Temporal Classification (CTC) (Graves et al., 2006), Listen, Attend and Spell (LAS) (Chan et al., 2016), RNN Transducer (RNN-T) (Graves, 2012) and Continuous Integrate-and-Fire (CIF) (Dong and Xu, 2020), focus on bringing effective alignment methods for better speech recognition performance. Among these works, the CIF alignment, which explicitly aggregates frame-level speech representations into token-level, is adopted in our work to combine with text representation. Specifically, we propose a novel pre-training paradigm: Continuous Integrate-and-Fire Pre-Training (CIF-PT) for end-to-end SLU. Two pre-training tasks are included in CIF-PT: the first task is speech-to-text modeling (Wang et al., 2020) with CIF alignment. In this work, ASR task that transcribes speech to text is applied. The second task is language model distillation (LMD). Since the integrated speech representation by CIF is at token-level, token-level distillation from a pre-trained language model can be performed to inject text-based linguistic knowledge into the representation. Through the joint pre-training of the two tasks, CIF-PT is able to generate representations with information from both speech and text modalities.

We examine our CIF-PT methods in downstream SLU tasks including intent classification and slot filling. On SLU benchmark SLURP (Bastianelli et al., 2020) dataset, the end-to-end SLU model with CIF-PT outperforms the state-of-the-art model by 1.94% of accuracy and 2.71% of SLU-F1 on the tasks of intent classification and slot filling, respectively. The cross-modal representation extracted by CIF-PT also shows its competitiveness in comparison of other neural interfaces (Rao et al., 2020;

Raju et al., 2022) utilized in SLU. The obtained results and a series of experiments including ablation study and the pre-training on out-of-domain data demonstrate the effectiveness and generalization of CIF-PT.

2 Related Works

End-to-End SLU Various works extend models originally designed for ASR into the field of SLU. Peng et al. (2022) propose Branchformer as an alternative to Conformer (Gulati et al., 2020), and show performance gains in SLU as well as ASR. Huang et al. (2022) jointly train ASR and SLU as multitasks to exploit shared knowledge from different tasks. Seo et al. (2022) use the probability distribution output of ASR model as continuous token interface (CTI) for downstream NLU. Self-supervised representative learning on speech data provides powerful backbones such as wav2vec 2.0 (Baeovski et al., 2020), HuBERT (Hsu et al., 2021) for SLU. Arora et al. (2022) propose ESPnet-SLU and analyze the performance of HuBERT encoder pre-trained with ASR as feature extractor for SLU. Wang et al. (2021) perform partial fine-tuning and entire fine-tuning on pre-trained wav2vec 2.0 and HuBERT on SLU tasks.

Cross-Modal Pre-training for SLU In order to exploit information from speech and text for SLU, jointly pre-training on both of speech and text data has been proposed. SpeechBERT (Chuang et al., 2020) extends the masked language model (MLM) pre-training from BERT into the mixture of audio and text data. In SPLAT (Chung et al., 2021), a speech module and a language module are jointly pre-trained with token-level and sentence-level alignment. Another branch of researches focus on knowledge distillation from pre-trained language model into pre-trained speech encoder. Kim et al. (2021) utilize BERT as a teacher to perform sentence-level knowledge distillation at the pre-training stage and target-specific distillation during fine-tuning. Zhu et al. (2022) introduce cross-attention between text and speech and perform distillation on the attention heads for knowledge transferring.

Frame-to-Token Alignment in SLU In SpeechBERT (Chuang et al., 2020), forced alignment based on external ASR engine is used to train the initial phonetic-semantic joint embedding. Chung et al. (2021) adopt a heuristic alignment approach

in SPLAT, where alignment scores is computed by the cosine similarity between the output embeddings of the pre-trained speech and text models. The cross-attention alignment is introduced in (Zhu et al., 2022) to capture the interactions between text tokens and speech frames. For SpeechT5, since the pre-training does not strictly rely on audio-text pair data, (Ao et al., 2022) adopt shared codebook for speech and text representation and a diversity loss to encourage the alignment in latent space.

3 Method

In this section, we present the architecture of our proposed continuous integrate-and-fire pre-training (CIF-PT) method for SLU. As shown in Figure 1, our end-to-end SLU models go through two stages: CIF-PT and SLU training.

During CIF-PT, we employ two pre-training tasks: ASR training with CIF alignment and token-level language model distillation (LMD). These two tasks help the model learn contextual representation of the speech features aligned to the tokens with high level linguistic knowledge. After CIF-PT, the pre-trained parameters including the speech encoder and CIF part are used for downstream SLU tasks such as intent classification and slot filling.

3.1 ASR training with CIF Alignment

As shown in Figure 1(a), the structure of CIF-based ASR model includes three parts: speech encoder, CIF part, and the corresponding decoder. For an input speech utterance, it is first processed into a sequence of frames $\mathbf{x} = [x_1, x_2, \dots, x_{T'}]$ with length T' via speech feature extractor (e.g. mel-filter bank, convolutional front-end (Baevski et al., 2020)), where x_t is the feature vector of the t th frame. The speech encoder converts the frame-level input vector into frame-level hidden states:

$$\mathbf{h} = [h_1, h_2, \dots, h_T] = \text{enc}([x_1, x_2, \dots, x_{T'}])$$

CIF part follows the speech encoder to convert the frame-level hidden states \mathbf{h} into token-level speech representations \mathbf{c} . We follow the CIF setup from Dong and Xu (2020), which is briefed as follows. At first, the encoded hidden states $\mathbf{h} = [h_1, h_2, \dots, h_T]$ are fed into a weight estimator module to calculate a series of weights $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_T]$. The weights $\boldsymbol{\alpha}$ and the frame-level hidden states \mathbf{h} are input to CIF to obtain $\mathbf{c} = [c_1, c_2, \dots, c_i, \dots, c_N]$, where N is the

number of total tokens. Each token-level representation c_i is a linear combination of frame-level representations $\{h_t\}$. At each frame step t , the weight α_t added to an accumulated weight $\alpha_i^a \leftarrow \alpha_i^a + \alpha_t$, and the frame-level hidden state h_t is integrated into token-level representation $c_i \leftarrow c_i + \alpha_t h_t$, until the accumulated weight α_i^a exceeds a threshold β . When α_i^a exceeds β , the weight of the boundary hidden state is divided into two parts $\alpha_t = \alpha_{t1} + \alpha_{t2}$, to ensure the accumulated weight for each token is exactly β , and the second part α_{t2} is accumulated to the next token representation. In such way, the frame-level hidden states are integrated into token-level representation, which not only reduces the redundancy of speech information but also reduces computation complexity when used for the subsequent ASR decoder and downstream understanding tasks.

We use the autoregressive ASR decoder in (Dong and Xu, 2020). It accepts previous token y_{i-1} and the integrated c_i from CIF part as inputs, and autoregressively predicts the token output distribution for each c_i . The CIF-based encoder-decoder model is trained with a cross entropy (CE) loss in a teacher-forcing manner:

$$\mathcal{L}_{\text{CE}} = \sum_{i=1}^N \log p(y_i | y_{<i}, c_i).$$

Optionally, \mathcal{L}_{CTC} can be applied on the frame-level hidden states \mathbf{h} to be jointly trained. The quantity loss \mathcal{L}_{QUA} is to supervise the CIF part to predict the quantity of tokens closer the number of target tokens:

$$\mathcal{L}_{\text{QUA}} = \left| \sum_{i=1}^T \alpha_i - N \right|.$$

The final CIF loss is the weighted sum of three:

$$\mathcal{L}_{\text{CIF}} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{CTC}} + \lambda_2 \mathcal{L}_{\text{QUA}}. \quad (1)$$

3.2 Language Model Distillation

Since the speech representation c_i integrates speech information into the token-level, we use a pre-trained BERT model as a knowledge distillation teacher to inject textual knowledge into speech representation. Let $\mathbf{x} = \{x_i\}_{i=1}^T$ be the speech frame sequence and $\mathbf{y} = \{y_i\}_{i=1}^N$ be the corresponding transcript token sequence. As shown in Figure 1, \mathbf{x} is encoded into speech feature $\{c_i\}_{i=1}^N$ by the speech encoder and the CIF part. \mathbf{y} is encoded by BERT into contextual representation vectors $\{h_i^t\}_{i=1}^N$. Since $\{c_i\}$ are aligned to tokens, directly token-level knowledge distillation can be

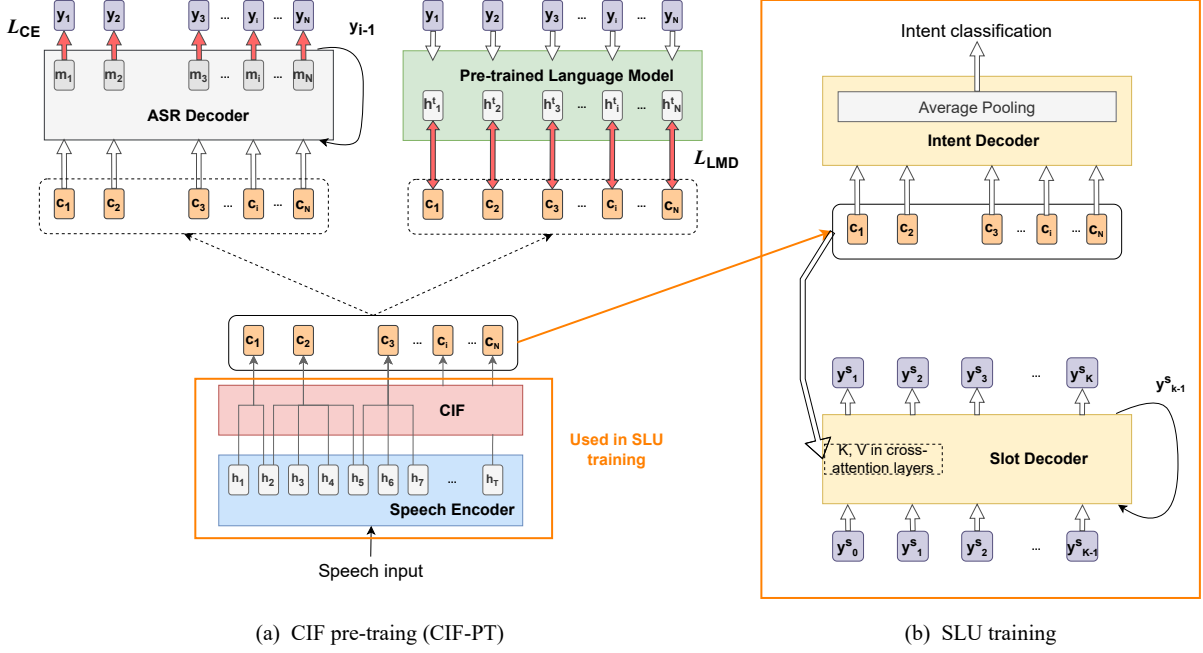


Figure 1: Architecture of our end-to-end SLU model with CIF-PT: (a) shows the procedure of CIF-PT including the ASR task with CIF alignment and token-level language model distillation; (b) shows the model structure of SLU decoder used for SLU training, including intent decoder and slot decoder for IC and SF, respectively.

performed to make the speech representation close to the contextual representation brought by BERT, thus forming a cross-modal representation.

We consider three types of language model distillation (LMD) loss in our paper, MSE loss, smoothed L1 loss and contrastive loss. Using BERT hidden output h_i^t as target, the MSE loss of c_i is $\mathcal{L}_{\text{LMD}}^{\text{MSE}}(h_i^t, c_i) = \|h_i^t - c_i\|^2$. The smoothed L1 loss is proposed in (Baevski et al., 2022a), where a γ is used to control the transition from a squared loss to an L_1 loss, i.e.

$$\mathcal{L}_{\text{LMD}}^{\text{SL1}}(h_i^t, c_i) = \begin{cases} \frac{1}{2}(h_i^t - c_i)^2/\gamma & |h_i^t - c_i| \leq \gamma \\ (|h_i^t - c_i| - \frac{1}{2}\gamma) & \text{otherwise.} \end{cases}$$

The contrastive loss encourage c_i to be closer to h_i^t than other c' sampled from an in-batch negative set \mathcal{N}_c .

$$\mathcal{L}_{\text{LMD}}^{\text{cont}}(h_i^t, c_i) = \frac{\exp[\text{sim}(h_i^t, c_i)/\tau]}{\sum_{c' \in \mathcal{N}_c} \exp[\text{sim}(h_i^t, c')/\tau]},$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity function and τ is the temperature scalar.

The LMD task is trained simultaneously with CIF-based ASR training as multitasks, which forms the training loss \mathcal{L} of CIF-PT as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CIF}} + \lambda \mathcal{L}_{\text{MLD}}. \quad (2)$$

3.3 Spoken Language Understanding

After CIF-PT, the pre-trained speech encoder and CIF part convert speech input into the sequence of cross-modal representation $\{c_i\}$, which is used for downstream SLU training. We evaluate our pre-trained model on SLU tasks of intent classification and slot filling. The corresponding intent decoder and slot decoder are shown in Figure 1(b).

For intent classification, $\{c_i\}_{i=0}^N$ is fed into additional Transformer layers to generate task specific decoder states. We use the average of decoder state on all position as the utterance representation for intent prediction through a linear projection.

The slot filling task is performed in a sequence generation style. The slot types and slot values are concatenated as targets $\{y_i^s\}$ to train a sequence-to-sequence model, i.e. “[SEP] slot_type1 slot_value1 [SEP] slot_type2 slot_value2”. The slot decoder consists of Transformer decoder layers where the sequence of c_i is used as the key and value of the cross-attention layer. We train the encoder-decoder to generate slot target sequence $\{y_i^s\}_{i=0}^K$ with teacher-forcing.

4 Experimental Setup

4.1 Dataset and Preprocessing

We conduct experiments on the dataset of SLURP (Bastianelli et al., 2020), which is currently the

largest SLU benchmark and is also linguistically more diverse than other datasets. It is collected for developing an in-home personal robot assistant. The train, development and test sets split in the SLURP paper are used for the training and evaluation of our methods. In addition to use the in-domain SLURP data for pre-training, we also introduce the Librispeech (Panayotov et al., 2015) dataset that contains 960 hours of speech derived from audiobooks as the out-domain pre-training dataset, which is only used in Section 5.4. All speech data is re-sampled or kept at 16 kHz, and all text data is converted into a sequence of subword units by the subword-nmt (Sennrich et al., 2016) toolkit¹. Specifically, we generate 10706 subword units by performing 36000 merge operations on the training set of Librispeech datasets, and use the learned BPE as the only tokenizer for text of all datasets.

4.2 Model Configuration

In this part, we detail the model structure and configuration utilized in our experiments. All the models are implemented using (Paszke et al., 2019):

Encoder we use two types of speech encoder which are denoted as conformer and data2vec in subsequent experiments. For the encoder of conformer, it consists of a two-layer convolutional front-end and 15-layer conformer blocks (Li et al., 2021). It applies a 8-time temporal down-sampling similar to (Dong et al., 2019). The hidden size in the conformer block uses 400. For the encoder of data2vec, it follows the official data2vec-large configuration (Baevski et al., 2022b) and uses the released model² from (Wolf et al., 2020). For the text encoder that provides text representation in CIF-PT, we follow the BASE configuration of BERT (Devlin et al., 2019) and use our learned BPE tokenizer to perform pre-training on the English Wikipedia corpus.

CIF part we follow the implementation of weight estimator and CIF calculator in (Dong and Xu, 2020). The channel number in convolutional layer keeps the same as the hidden size in decoder. The threshold β during CIF calculation is set to 1.0. The corresponding scaling strategy and tail handling methods are also used.

¹<https://github.com/rsennrich/subword-nmt>

²<https://huggingface.co/facebook/data2vec-audio-large>

	IC (Acc.)	SF (SLU-F1)
MTL-SLT (Huang et al., 2022)	83.10%	74.49%
Speech-Brain (Ravanelli et al., 2021)	85.34%	74.26%
ESPNET-SLU (Arora et al., 2022)	86.30%	71.90%
CTI (Seo et al., 2022)	86.92%	74.66%
Branchformer (Peng et al., 2022)	88.10%	77.70%
Hubert SLU (Wang et al., 2021)	89.38%	78.92%
CIF-PT (Conformer encoder)	89.60%	78.67%
CIF-PT (Data2vec encoder)	91.32%	81.63%

Table 1: Comparison with the published results on SLU benchmark (SLURP), including two tasks: intent classification (IC) and slot filling (SF). Our CIF-PT method uses the result of M0, M1 in Table 2, respectively. Both are pre-trained and fine-tuned only on the SLURP.

Decoder we use three types of decoder in our experiments, including the ASR decoder for speech-to-text training in CIF-PT, the down-streaming intent decoder for IC and slot decoder for SF. For ASR decoder, it uses the original autoregressive decoder (Dong and Xu, 2020) with 2-layer self-attention networks (SANs, also known as transformer encoder layers (Vaswani et al., 2017)). The hidden size is 400 when the encoder uses conformer and 512 for data2vec. For intent decoder, it uses 2-layer SANs and a following average pooling layer. For slot decoder, it uses 4-layer SANs for the tag-based slot decoder and uses 4-layer transformer decoder layers (with additional cross-attention layer) for the generation-based slot decoder. Without specific statement, the generation-based slot decoder is used by default. The hidden size keeps the same as ASR decoder for the two types of SLU decoder.

4.3 Training and Evaluation

We use an AdamW (Loshchilov and Hutter, 2018) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and weight decay of $1e-5$. During CIF pre-training, we warm up the learning rate for the first 4% of updates to a peak of $1e-3$ and keep it constant in the later 64% of updates, then linearly decay it to $1e-4$. The number of total training steps is 80k. We set the weight of CTC loss $\lambda_1 = 0.5$, and the weight of quantity loss $\lambda_2 = 1.0$. The hyper-parameter of LMD loss is explored in section 5.2. During SLU training, we follow the Noam scheduler (Vaswani et al., 2017) with 1600 warm-up steps and peak learning rate of $5e-4$. The number of total training steps is 32k.

After training, we first perform model average on the last 10 checkpoints for all models and then use the averaged model for evaluation. We follow

(Model Id.)	Method	Speech Encoder	Intent Classification (Acc.)	Slot Filling (SLU-F1)
M0	CIF-PT	Conformer	89.60%	78.67%
M1	CIF-PT	Data2vec	91.32%	81.63%
On the importance of CIF-PT				
M2	M0 w/o any PT	Conformer	86.43% (-3.17%)	72.51% (-6.16%)
M3	+ triple steps	Conformer	87.28% (-2.32%)	74.92% (-3.75%)
M4	+ CTC-PT	Conformer	86.41% (-3.19%)	75.87% (-2.80%)
On the importance of language model distillation (LMD)				
M5	M0 w/o LMD	Conformer	88.31% (-1.29%)	77.84% (-0.83%)
M6	M1 w/o LMD	Data2vec	91.18% (-0.14%)	81.02% (-0.61%)
On the importance of CIF alignment (all w/o language model distillation)				
M7	M3 w/o CIF	Data2vec	90.36% (-0.96%)	79.29% (-2.34%)
M8	+CTC-PT	Data2vec	90.63% (-0.69%)	80.31% (-1.32%)

Table 2: Ablation study on the proposed CIF-PT. For fair comparison, models in this table use the same structure of SLU decoder. For M7 where CIF is ablated, it directly passes the frame-level outputs of speech encoder to the SLU decoder. For M8, it follows the model structure of M7 but performs CTC Pre-Training (CTC-PT) on ASR tasks before training on SLU. The model structure of M4 is similar to M8 except using conformer as its speech encoder. All models are pre-trained and fine-tuned only on the SLURP data.

the metric of accuracy and SLU-F1 (Bastianelli et al., 2020) to evaluate the models on task of IC and SF, respectively. During the inference of SF task, we perform beam search with beam width 10 and a temperature scalar of 1.25. All experimental results are averaged at least 2 runs.

5 Results and Analysis

5.1 Main Results

To verify the effectiveness of our proposed methods, we first conduct three sets of experiments to explore the importance of designs in CIF-PT. The main results are summarized in Table 2.

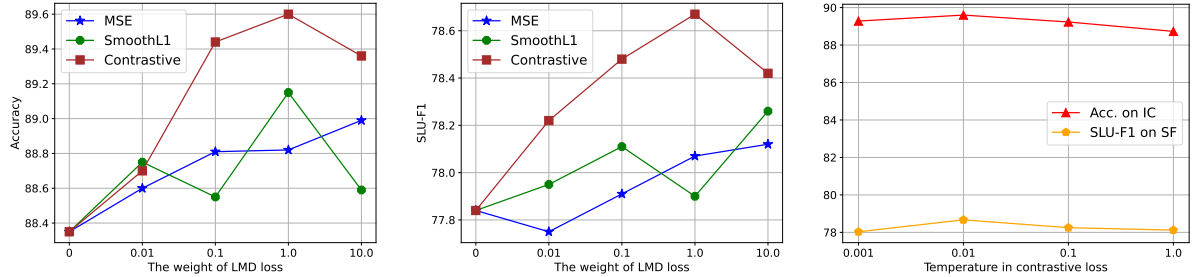
The first two rows of Table 2 show the performance of our end-to-end SLU models using CIF-PT. Consistent with our expectation, the model M1 with the self-supervised data2vec encoder obtains better results than the model M0 with conformer encoder on both tasks. We also compare the performance of our methods with the published results. As shown in Table 1, the model with CIF-PT (M1 in Table 2) achieves state-of-the-art result on both of IC and SF tasks. The performance advantages on the task of SF reaches 2.71% SLU-F1. We suspect that the cross-modal representation extracted by CIF-PT contains more language knowledge that benefits more to SF, which needs to predict the slot key and speech content simultaneously. It is worthy to mention that the model M0 with conformer encoder also achieves competitive performance, which is even superior or comparable to the published strong models (Wang et al., 2021; Seo

et al., 2022) with self-supervised speech encoder.

For the model of M2 in Table 2, we ablate CIF-PT utilized in the model M0 and conduct a joint training of ASR and SLU tasks from scratch. The results show that ablating CIF-PT leads to a large performance degradation on both SLU tasks. Since CIF-PT consumes extra pre-training steps, we suspect the total training step maybe a factor of the performance gap. Therefore, we increase the training step to triple (from 32k to 96k) to obtain the model M3. The performance gap is narrowed but the model M0 with CIF-PT still has a certain performance advantage over model M3 with longer SLU training.

For the model of M5 and M6 in Table 2, we ablate language model distillation (LMD) utilized in CIF-PT. During pre-training, we find applying LMD bring 3.9% (14.83 \rightarrow 14.25) relative WER reduction on the model with conformer encoder. During SLU training, we also observe the introduced LMD methods boosts the performance improvements on the two tasks in Table 2. For the reason of the smaller performance improvements of data2vec encoder, it may be that the model with data2vec encoder itself has strong modeling power and already learns effective pattern and textual knowledge, so that the injected textual knowledge can only be helpful for fewer evaluation samples.

We also compare the cross-modal representation extracted by CIF-PT with the speech representation derived from self-supervised learning. For the model M7 in Table 2, we ablate the frame-to-token CIF alignment in SLU models and directly pass



(a) Comparison of LMD methods on IC (b) Comparison of LMD methods on SF (c) Scalar τ in contrastive LMD

Figure 2: (a) and (b) depict the performance fluctuation of different LMD methods on the two SLU tasks as the weight λ of LMD loss changes. (c) depicts the performance fluctuation of the contrastive LMD method as the temperature scalar τ changes.

the frame-level speech representation extracted by data2vec to the SLU decoder. Although achieving competitive results, model M7 could achieve further improvements after combining with CIF. We suspect the reason is two-folds: 1) CIF performs frame-to-text mapping that integrates relevant speech/semantic information, thus able to remove information redundancy in adjacent frames, 2) CIF-PT bridges the speech representation and text representation through ASR training and LMD, thus providing more textual knowledge that benefits SLU performance. To further verify our hypothesis, we introduce CTC-based ASR pre-training (CTC-PT) before the training of SLU model. Results show that CTC-PT provides improvements on SLU tasks (M7 \rightarrow M8, M2 \rightarrow M4), but it still has gap from CIF-PT. Above observations demonstrate the effectiveness of CIF-PT.

5.2 Comparison on Language Model Distillation

In this part, we compare different language model distillation (LMD) methods applied in CIF-PT. From the Figure 2 we get three observations: (1) All LMD methods provide positive effects on SLU performance in most cases, except for one outlier uses MSE loss with a weight of 0.01 on SF. The degradation disappears as the loss weight increases; (2) The contrastive LMD method shows better quality on both SLU tasks than the other two methods. We suspect the reason is contrastive distillation with proper temperature scalar mainly focuses on distinguishing hard negatives, instead of forcing the representation to be consistent like MSE. This helps the extracted representation retain speech and language information at the same time, which may benefit SLU modeling; (3) Different temperature

scalar in contrastive LMD method has effects on down-streaming SLU tasks, with a τ value of 0.01 producing the best results on both SLU tasks.

5.3 Comparison on Neural Interfaces

We have compared the token-level representation c_i extracted by CIF-PT with frame-level speech representations in section 5.1. In this part, we continue to compare c_i with other popular token-level neural interfaces (or representations) summarized in (Raju et al., 2022), including hidden interface m_i , posterior interface p_i , tied embedding interface e_i and the combinations. For fair comparison, we give up using LMD loss in CIF-PT which benefits c_i . The results are shown in Table 3.

On the task of IC, we find all token-level neural interfaces achieve comparable accuracy. This may be because these interfaces contain close information that is useful for IC, and the pooling operation in intent decoder further reduce the discrimination between representations. The combination of c_i and e_i achieves the best performance. We suspect this is because they are located at the beginning (c_i) and end (e_i) of ASR decoder respectively, so they may have a large information difference and complementarity.

On the task of SF, we observe a relatively large differentiation among these neural interfaces. On the model using generation-based slot decoder, c_i obtains the best performance, while other interfaces have a certain performance gap in comparison. This phenomenon can be understood as c_i , which is sourced from pure speech inputs, contains more original and comprehensive speech information. It can provide sufficient information for the calculation of cross-attention in the slot decoder. In contrast, the other interfaces are all calculated

Interfaces	IC (Acc.)	SF (SLU-F1)	
		generation	tag
c_i	88.31%	77.84%	71.68%
m_i	88.26%	68.24%	74.21%
e_i	88.25%	65.42%	74.42%
p_i	88.30%	54.04%	72.94%
c_i, m_i	88.48%	74.94%	74.61%
c_i, e_i	88.79%	76.22%	74.52%

Table 3: Comparison with other token-level neural interfaces summarized in (Raju et al., 2022). Here, c_i represents the cross-modal representations extracted by CIF-PT. m_i represents the output of ASR decoder (Hidden Interfaces). p_i represents the posterior predicted by ASR decoder (Posterior Interface). e_i represent the token embedding of ASR’s one-best token sequence (Tied Embedding Interface). Generation and tag in the table describe two types of slot decoder used in our model, we detail their structures in Section A.2.

via the autoregressive ASR decoder, thus the information may be biased to a certain hypothesis with errors in inference. In addition, using c_i as the interface can also avoid the mismatch between the teacher-forcing inputs and predicted inputs in inference.

Interestingly, on the model using tag-based slot decoder, c_i performs inferior to other neural interfaces. Since the tag-based slot model predicts slot key for each token of the one-best ASR hypothesis, the neural interfaces m_i, p_i, e_i that are updated synchronously with the ASR decoding could provide closer slot prediction for the final ASR hypothesis. The original speech information provided by c_i can also provide supplements to these interfaces, and the best performance is obtained by the combination of c_i and m_i .

Between two types of slot decoder, the model with generation-based slot decoder is superior to the tag-based slot decoder, we believe this is because generation-based decoder utilize the bi-directional contextual information from full sequence, which makes it have higher ceiling in the prediction of slot information. In contrast, tag-based decoder could only use the uni-directional information that is limited by the autoregressive ASR decoder. However, this characteristic makes the tag-based model suitable for the application scenario with low-latency.

Model	IC (Acc.)	SF (SLU-F1)
Slurp-Frozen	89.60%	78.67%
Slurp-Unfrozen	88.84%	78.08%
LS-Frozen	80.65%	64.02%
LS-Unfrozen	90.65%	79.74%

Table 4: Comparison on the out-of-domain data. In the column of model, Slurp and LS before the dash represent the utilized pre-trained dataset, LS represents Librispeech. Frozen and Unfrozen represent the state of pre-trained parameters in SLU training.

5.4 Comparison on Out-of-domain Data

In above experiments, CIF pre-training is performed on the in-domain SLURP dataset. During SLU training, the pre-trained parameters are kept frozen (‘Slurp-Frozen’ in Table 4) and only the part of SLU decoder is trained. In this part, we first explore unfreezing the pre-trained parameters during SLU training (‘Slurp-Unfrozen’ in Table 4). Specifically, we hold the pre-trained parameters frozen in the first half of training, and then make the model entirely trained by performing joint training of ASR and SLU tasks. Results show that unfreezing pre-trained parameters leads to slight performance degradation. We suspect it is because the textual knowledge injected by LMD suffers catastrophic forgetting during SLU training. But the result achieved by ‘Slurp-Unfrozen’ is still better than the model using the frozen pre-trained model without LMD (model M5 in Table 2).

We also conduct experiments on an out-of-domain pre-training dataset (Librispeech) to explore its effects on the final SLU performance. Consistent with our expectations, freezing the parameters pre-trained on out-of-domain data (‘LS-Frozen’ in Table 4) leads to a large performance degradation on SLU tasks of SLURP. When unfreezing these pre-trained parameters (‘LS-Unfrozen’ in Table 4) during SLU fine-tuning, the model obtains a noticeable performance boost, and even outperforms the model achieved on slurp dataset. This partly reflects the good generalization and the potential on transfer learning of our proposed CIF-PT method.

6 Conclusion

In this work, we propose a new pre-training paradigm: Continuous Integrate-and-Fire Pre-Training (CIF-PT) for end-to-end SLU. CIF serves as a bridge connecting speech and text modality: on the one hand, it integrates speech representation

into token-level through its frame-to-token alignment ability learned from ASR pre-training task. On the other hand, it support one-to-one transfer of the textual knowledge into the integrated token-level speech representation via the pre-training of language model distillation. After CIF-PT, we obtain a cross-model representation that is used as neural interface into down-streaming SLU tasks.

Evaluated on the largest SLU benchmark of SLURP, CIF-PT creates new state-of-the-art result on both of IC and SF tasks. We further validate the effectiveness and generalization of CIF-PT by a series of experiments including ablation study and the pre-training on out-of-domain data. We also observe the cross-modal representation extracted by CIF-PT shows its competitiveness in comparison with other neural interfaces on SLU. We believe that CIF-PT has the potential to better encode long-form speech content (e.g. spoken paragraph) through its language model distillation, and will explore to combine it with LLM methods like ChatGPT to further empower spoken language understanding (SLU) systems.

7 Limitation

In the process of conducting experiments, we find our method has some limitations. First, CIF-PT needs to be performed on the dataset with speech-text pair. For some small-scale dataset that only contains speech and SLU labels, our method needs to use external ASR dataset to conduct the pre-training, leading to the increase of complexity of model building. In addition, in CIF-PT, we need to ensure that the tokenizer of the pre-trained language model is consistent with the tokenizer in the ASR task. However, there is usually a gap between the two in terms of vocabulary size. In consideration of performance, it is necessary to modify the tokenzier of one or both sides.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, et al. 2022. *Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738.

Siddhant Arora, Siddharth Dalmia, Pavel Denisov, Xuan-kai Chang, Yushi Ueda, Yifan Peng, Yuekai Zhang, Sujay Kumar, Karthik Ganesan, Brian Yan, et al.

2022. *Espnet-slu: Advancing spoken language understanding through espnet*. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7167–7171. IEEE.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022a. *data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language*. In *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, Maryland, USA.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022b. *Data2vec: A general framework for self-supervised learning in speech, vision and language*. *arXiv preprint arXiv:2202.03555*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. *wav2vec 2.0: A framework for self-supervised learning of speech representations*. volume 33, pages 12449–12460.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. *Slurp: A spoken language understanding resource package*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7252–7262.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. *Listen, attend and spell: A neural network for large vocabulary conversational speech recognition*. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

Yung Sung Chuang, Chi Liang Liu, Hung Yi Lee, and Lin Shan Lee. 2020. *SpeechBERT: An audio-and-text jointly learned language model for end-to-end spoken question answering*. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2020-Octob:4168–4172*.

Yu-An Chung, Chenguang Zhu, and Michael Zeng. 2021. *Splat: Speech-language joint pre-training for spoken language understanding*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1897–1907.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Linhao Dong, Feng Wang, and Bo Xu. 2019. *Self-attention aligner: A latency-control end-to-end model*

- for asr using self-attention network and chunk-hopping. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5656–5660. IEEE.
- Linhao Dong and Bo Xu. 2020. [CIF: Continuous integrate-and-fire for end-to-end speech recognition](#). In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083. IEEE.
- Alex Graves. 2012. [Sequence transduction with recurrent neural networks](#). In *the International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning*.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. [Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks](#). In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- Anmol Gulati, James Qin, Chung Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. [Conformer: Convolution-augmented transformer for speech recognition](#). In *Proceedings of the 2020 Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 5036–5040.
- 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 *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- Zhiqi Huang, Milind Rao, Anirudh Raju, Zhe Zhang, Bach Bui, and Chul Lee. 2022. [Mtl-slt: Multi-task learning for spoken language tasks](#). In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 120–130.
- Seongbin Kim, Gyuwan Kim, Seongjin Shin, and Sangmin Lee. 2021. [Two-stage textual knowledge distillation for end-to-end spoken language understanding](#). volume 2021-June, pages 7463–7467.
- Bo Li, Anmol Gulati, Jiahui Yu, Tara N Sainath, Chung-Cheng Chiu, Arun Narayanan, Shuo-Yiin Chang, Ruoming Pang, Yanzhang He, James Qin, et al. 2021. A better and faster end-to-end model for streaming asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5634–5638. IEEE.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. [Linguistic knowledge and transferability of contextual representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#).
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. 2022. [Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding](#). In *International Conference on Machine Learning*, pages 17627–17643. PMLR.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Anirudh Raju, Milind Rao, Gautam Tiwari, Pranav Dheram, Bryan Anderson, Zhe Zhang, Chul Lee, Bach Bui, and Ariya Rastrow. 2022. [On joint training with interfaces for spoken language understanding](#). In *Interspeech 2022*.
- Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rastrow. 2020. Speech to semantics: Improve asr and nlu jointly via all-neural interfaces. *arXiv preprint arXiv:2008.06173*.
- Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. 2021. [Speechbrain: A general-purpose speech toolkit](#). *arXiv preprint arXiv:2106.04624*.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics (ACL).
- Seunghyun Seo, Donghyun Kwak, and Bowon Lee. 2022. Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7152–7156. IEEE.
- Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. [Towards end-to-end spoken language understanding](#). In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758.
- 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 N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Advances in neural information processing systems*, 30.
- Changan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39.
- Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding. *IEEE Signal Processing Magazine*, 22(5):16–31.
- Yingzhi Wang, Abdelmoumene Boumadane, and Abdelwahab Heba. 2021. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *arXiv preprint arXiv:2111.02735*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.
- Yi Zhu, Zexun Wang, Hang Liu, Peiyang Wang, Mingchao Feng, Meng Chen, and Xiaodong He. 2022. Cross-modal transfer learning via multi-grained alignment for end-to-end spoken language understanding. *Proc. Interspeech 2022*, pages 1131–1135.

A Appendix

A.1 Computational Experiments

The total parameters for our SLU model with conformer encoder is 95.07 M. It costs 10.1 hours and 12.0 hours for CIF-PT and SLU fine-tuning on 8 A100 GPUs, respectively. The batch size of both stages is set to 30000 frames on each GPU. For our CIF SLU model with data2vec encoder, it has 357.50M parameters and needs 23.0 hours and 7.5 hours to finish CIF-PT and SLU fine-tuning, the corresponding batch size for the two stages is set to 1.2M and 1.6M samples, respectively.

A.2 Details of Model Structure

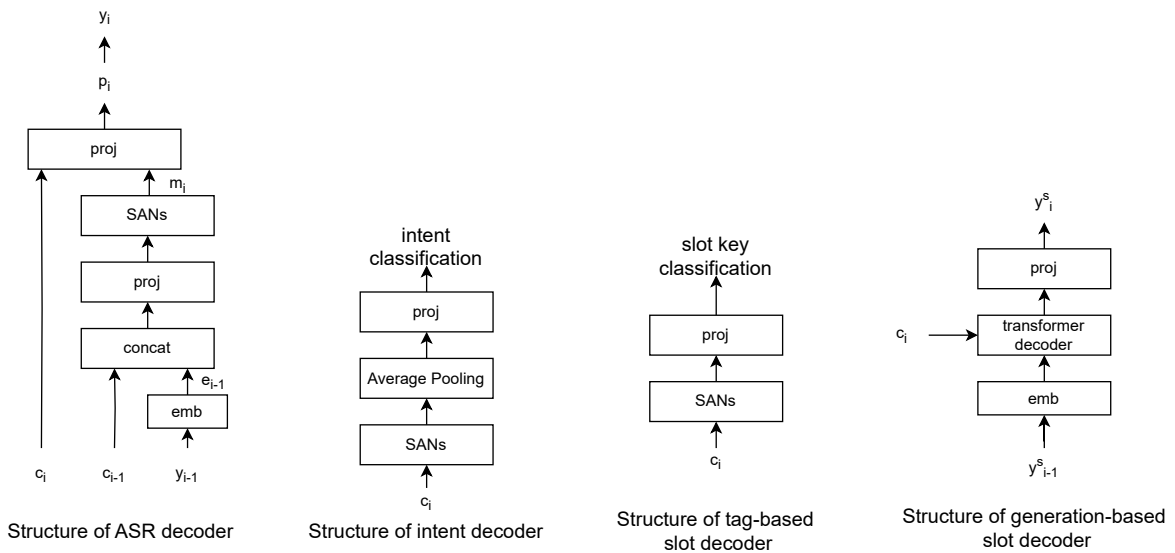


Figure 3: Model structure of our ASR decoder and SLU decoders. Different neural interfaces are depicted in the ASR decoder. The details of tag-based slot decoder and generation decoder are also included in this figure. c_i in this figure could be replaced by other interfaces, which are investigated in Table 3.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
In section 7
- A2. Did you discuss any potential risks of your work?
We believe there is no risk in our work. We only use the open-resource codes and publicly release data from the community, and no misuse of any resource and no other stakeholders involved in our work.
- A3. Do the abstract and introduction summarize the paper's main claims?
In Abstract and Introductions section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

in Results and Analysis Section 5

- B1. Did you cite the creators of artifacts you used?
In Experimental Setup Section 4, and Results and Analysis Section 5
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We all use open-resource code an public resource in our work and cite them correctly. We will discuss and state the license used when publicly releasing our code in more detail.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In Results and Analysis Section 5
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
in Experimental Setup Section 4 we reported the used public dataset and their resource. No anonymous problem of data our work.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
all artifacts in our work are publicly available and frequently used. We have cited them clearly.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
in Experimental Setup Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C Did you run computational experiments?

In Experimental Setup Section4, Results and Analysis Section5 and Appendix A

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

In Appendix A

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

In Experimental Setup Section4, Results and Analysis Section5

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

In Experimental Setup Section4

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

In Experimental Setup Section4

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.