Integration of Pre-trained Networks with Continuous Token Interface For End-to-End Spoken Language Understanding

(Seunghyun Seo[1,2*], Donghyun Kwak[1*], Bowon Lee[2])
[1]Clova AI, NAVER Corp. [2]INHA University.

ICASSP

May 2022

* These authors contributed equally

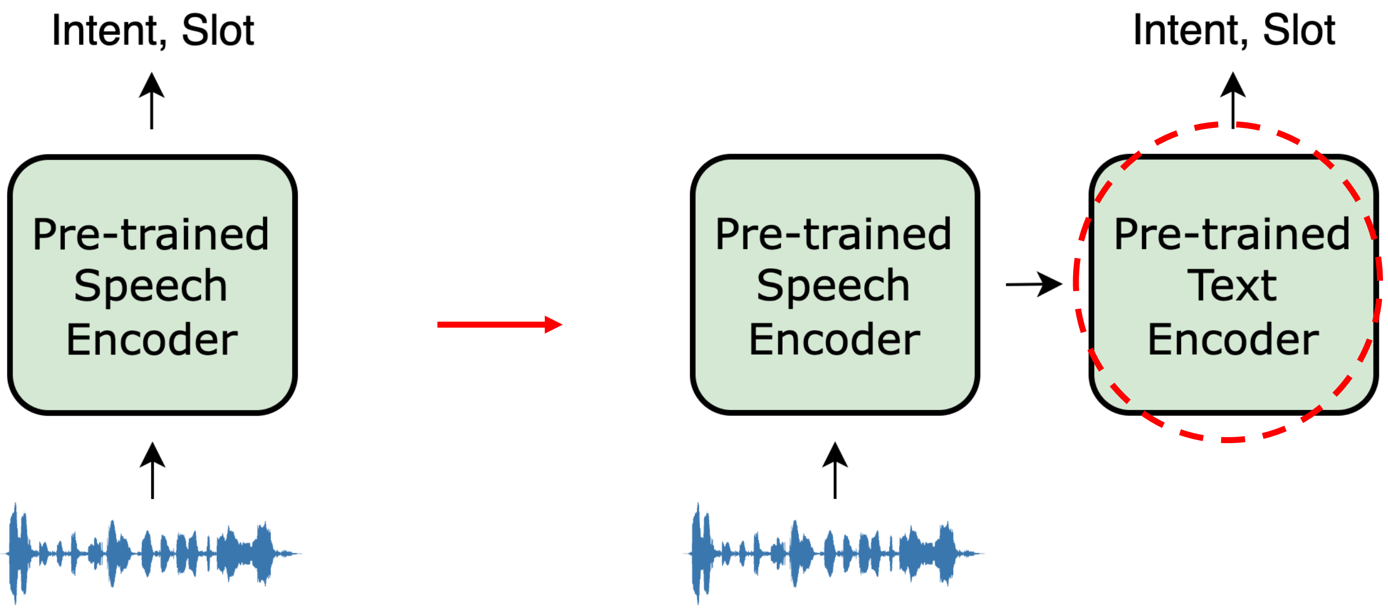
Highlight

1. **SOTA End-to-End (E2E) SLU Model** on challenging SLU dataset, **SLURP**

   - Intent Classification (Accuracy Score)

   - Slot Filling (SLU-F1 Score)

2. **Integration** of two pre-trained modules (ASR, NLU Networks) **with proposed interface, CTI (E2E)**

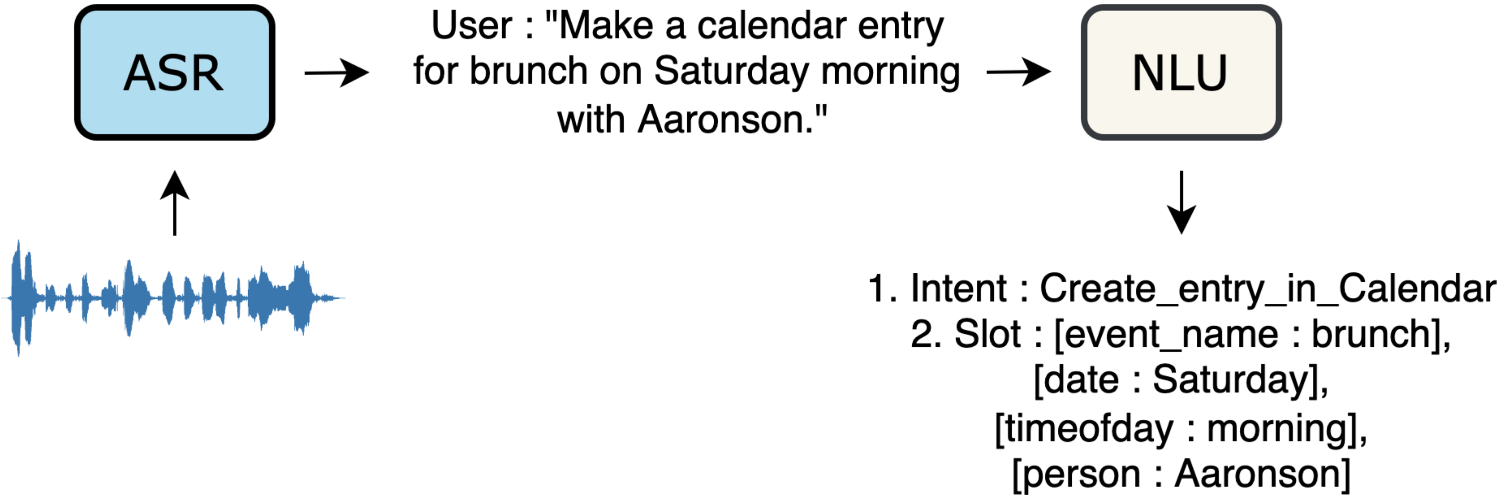


(a) Common E2E SLU

(b) Proposed E2E SLU

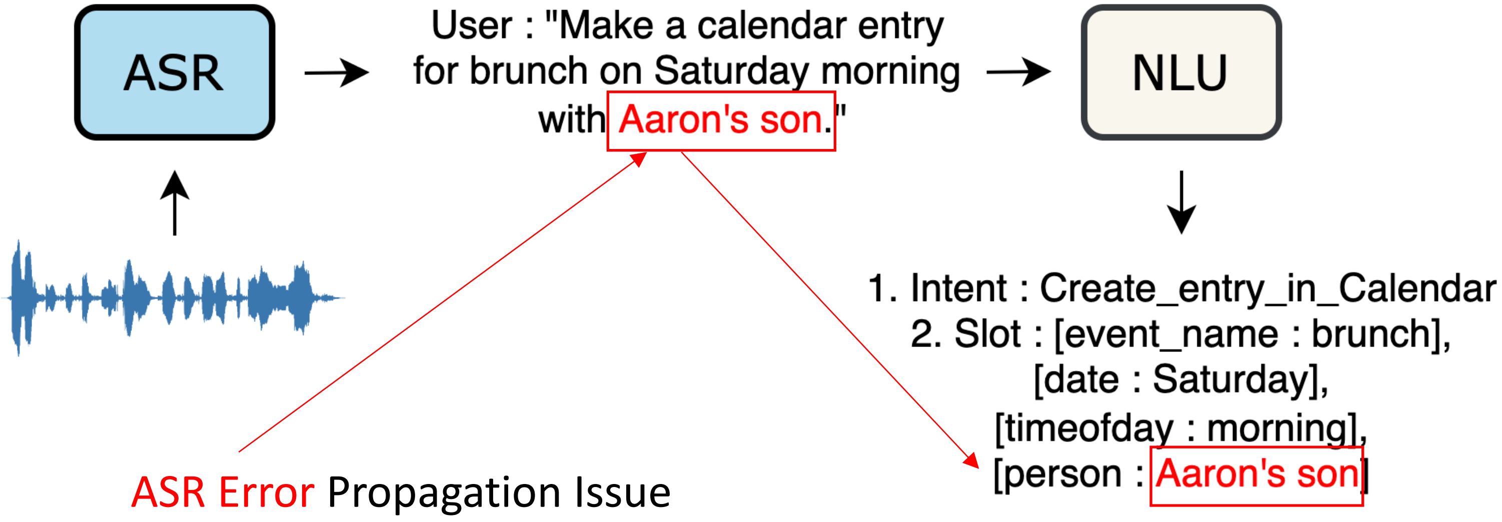
Motivation

Conventional SLU →

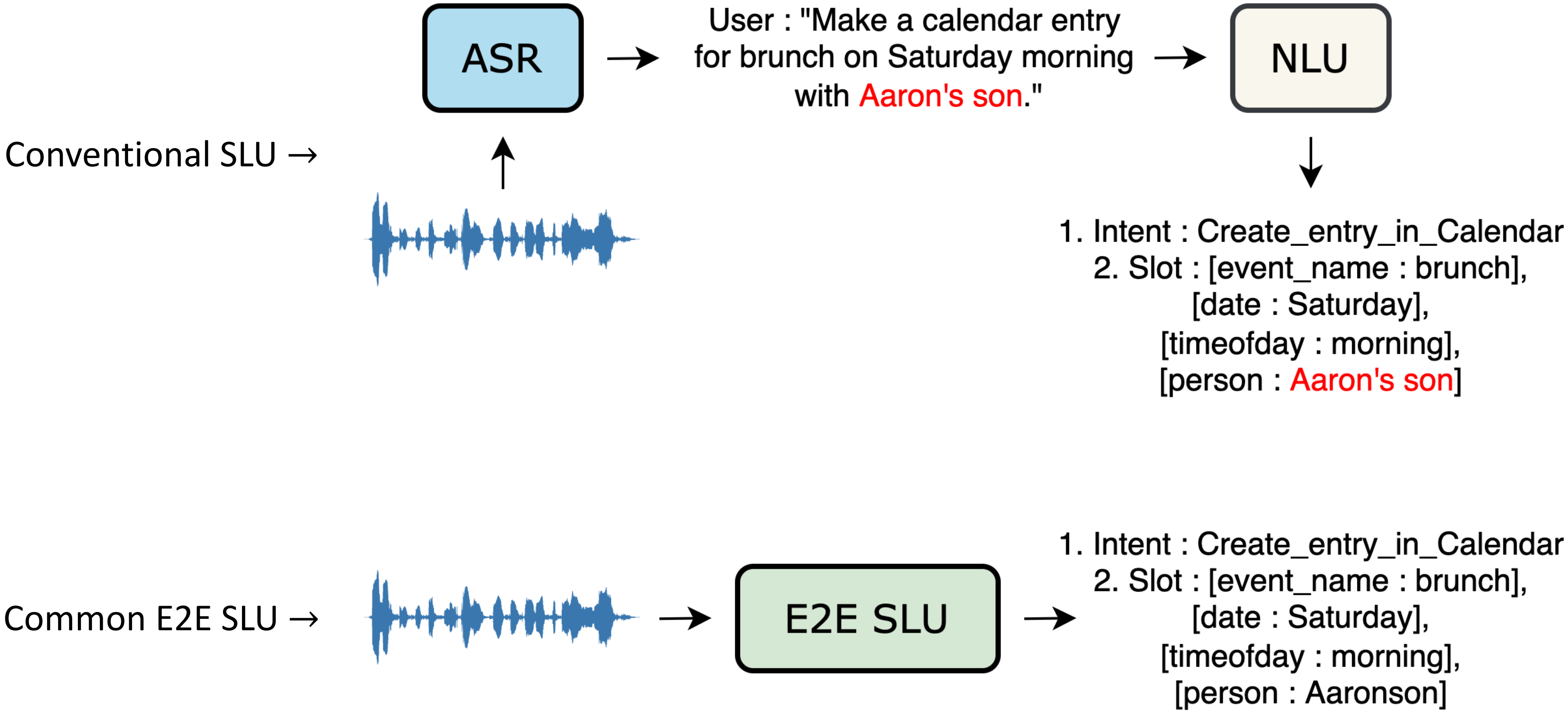


Motivation

Conventional SLU →



Motivation

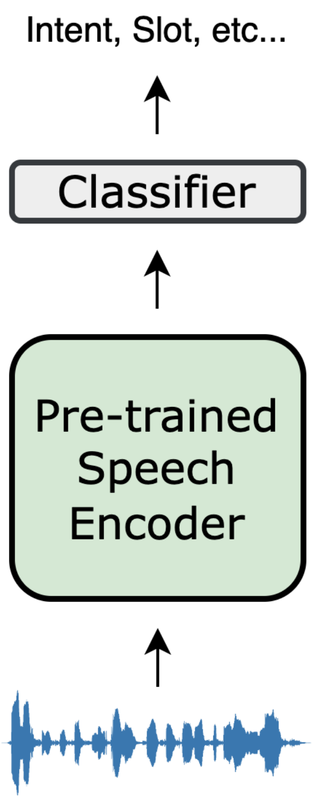


Conventional SLU →
ASR
User : "Make a calendar entry for brunch on Saturday morning with Aaron's son."
NLU
1. Intent : Create_entry_in_Calendar
2. Slot : [event_name : brunch], [date : Saturday], [timeofday : morning], [person : Aaron's son]
Common E2E SLU →
E2E SLU
1. Intent : Create_entry_in_Calendar
2. Slot : [event_name : brunch], [date : Saturday], [timeofday : morning], [person : Aaronson]

Motivation

Common E2E SLU Approaches

Use Pre-trained Speech Encoder



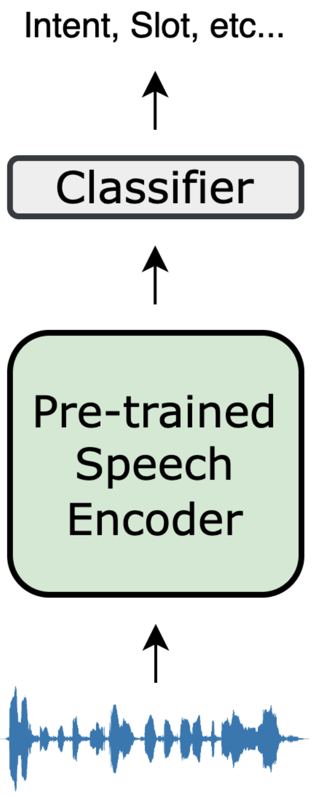
Motivation

Common E2E SLU Approaches

Use Pre-trained Speech Encoder

-> Poor Linguistic Generalization

(lack of NLU knowledge)



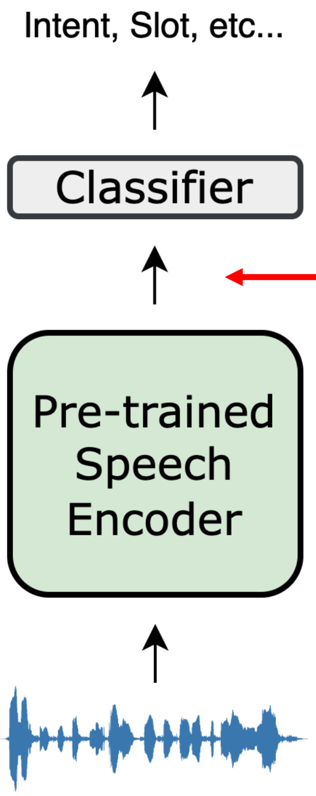
Motivation

Common E2E SLU Approaches

Use Pre-trained Speech Encoder

-> Poor Linguistic Generalization

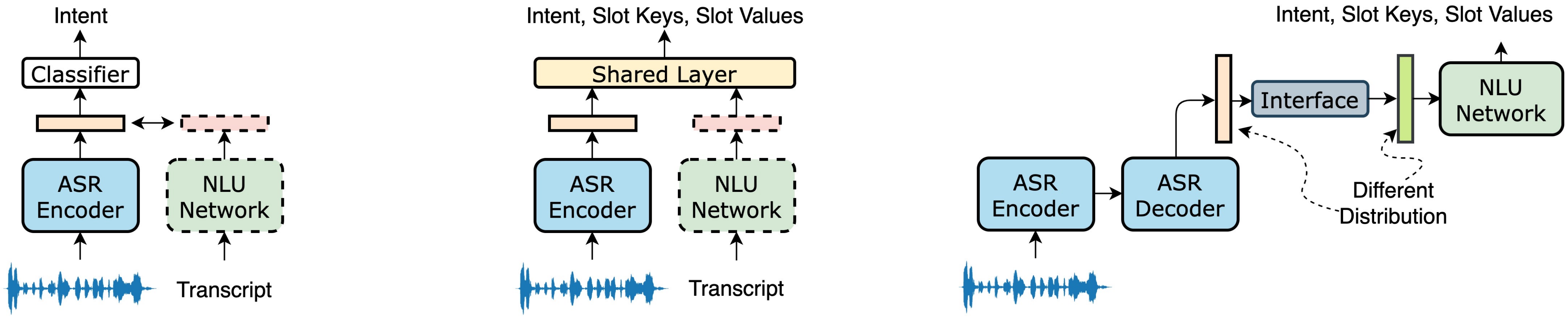
(lack of NLU knowledge)



Pre-trained NLU’s Knowledge (e.g. BERT)

How?

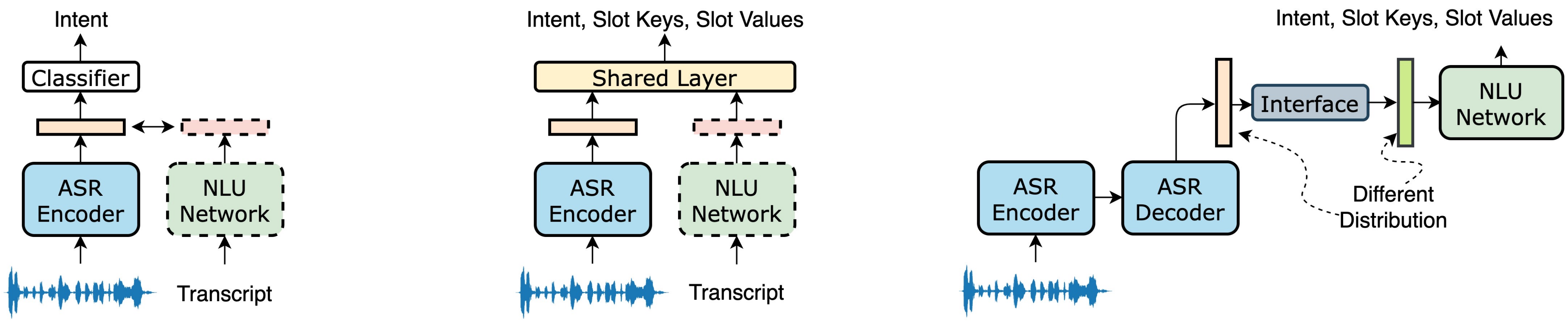
Related Works



(a) Knowledge Distillation (b) Cross-Modal Shared Embedding (c) Network Integration with Interface

- Several E2E SLU Models are designed to utilize contextual semantic information

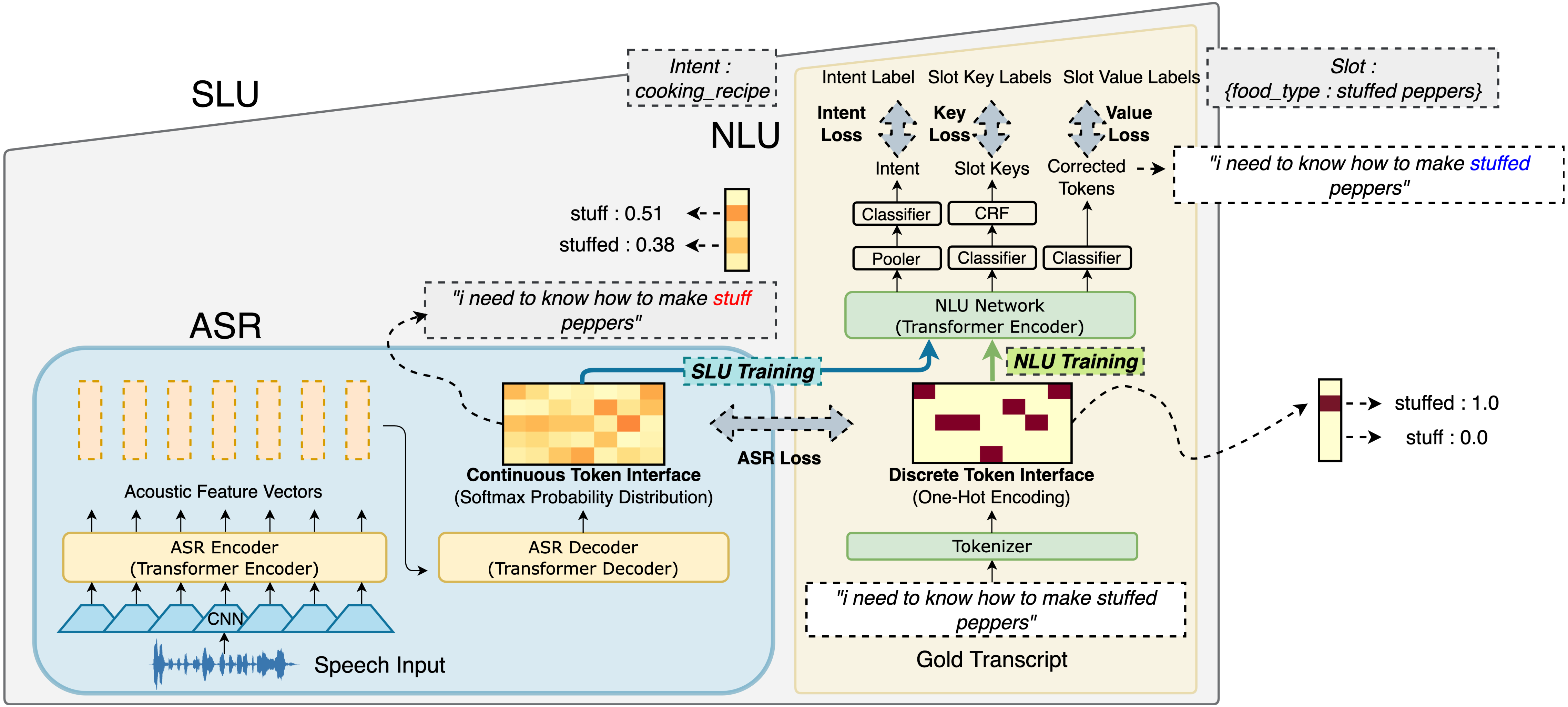
Related Works



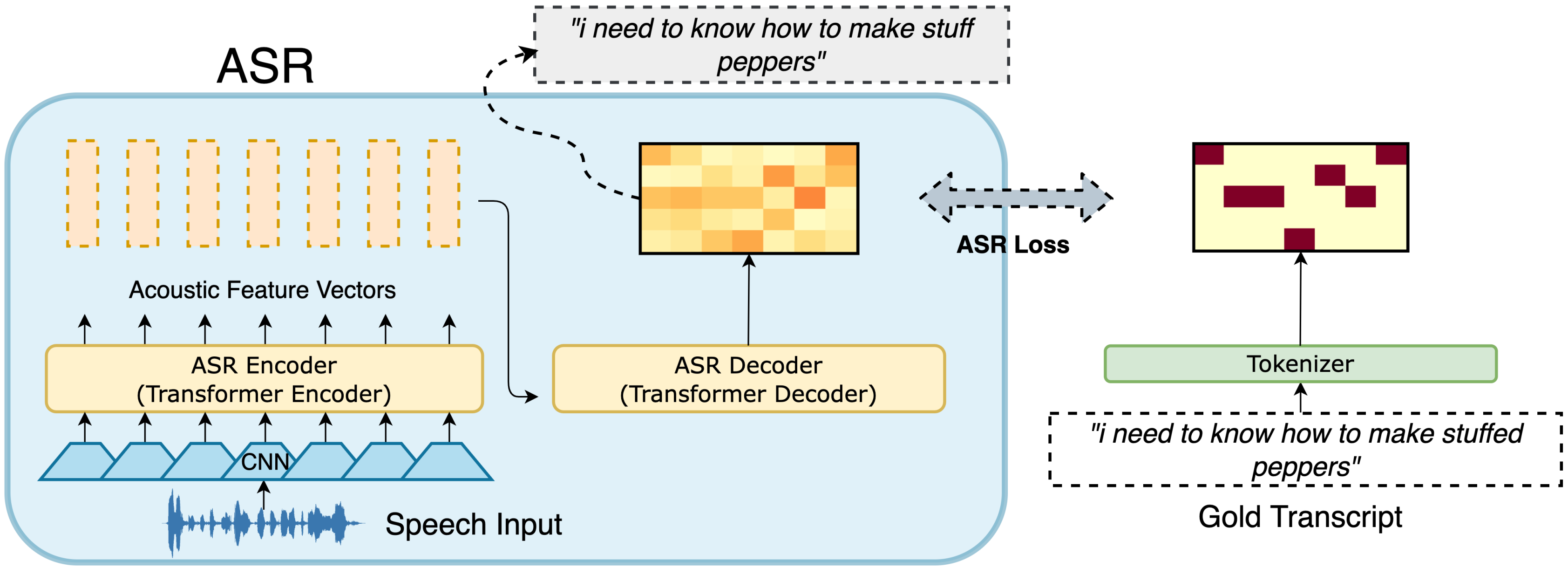
(a) Knowledge Distillation (b) Cross-Modal Shared Embedding (c) Network Integration with Interface

- Several E2E SLU Models are designed to utilize contextual semantic information
    - It might lose Pre-trained NLU's Information

Proposed Method

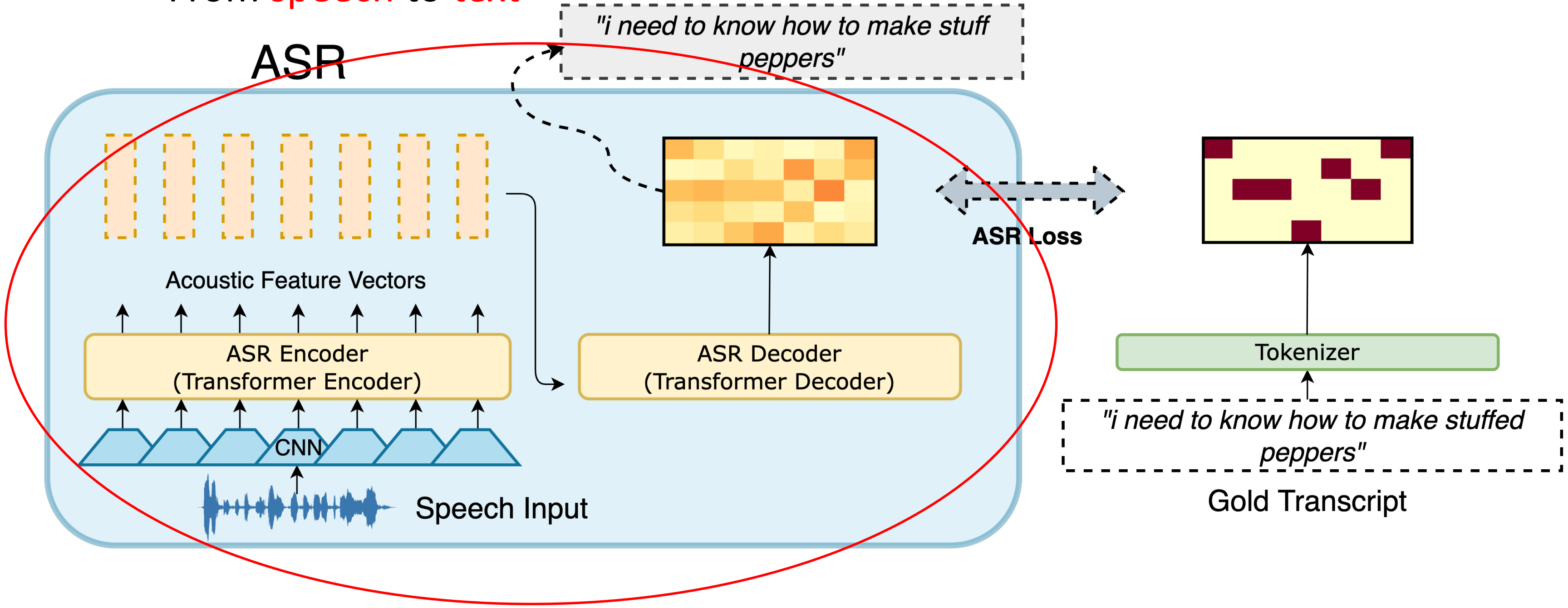


Proposed Method (ASR)

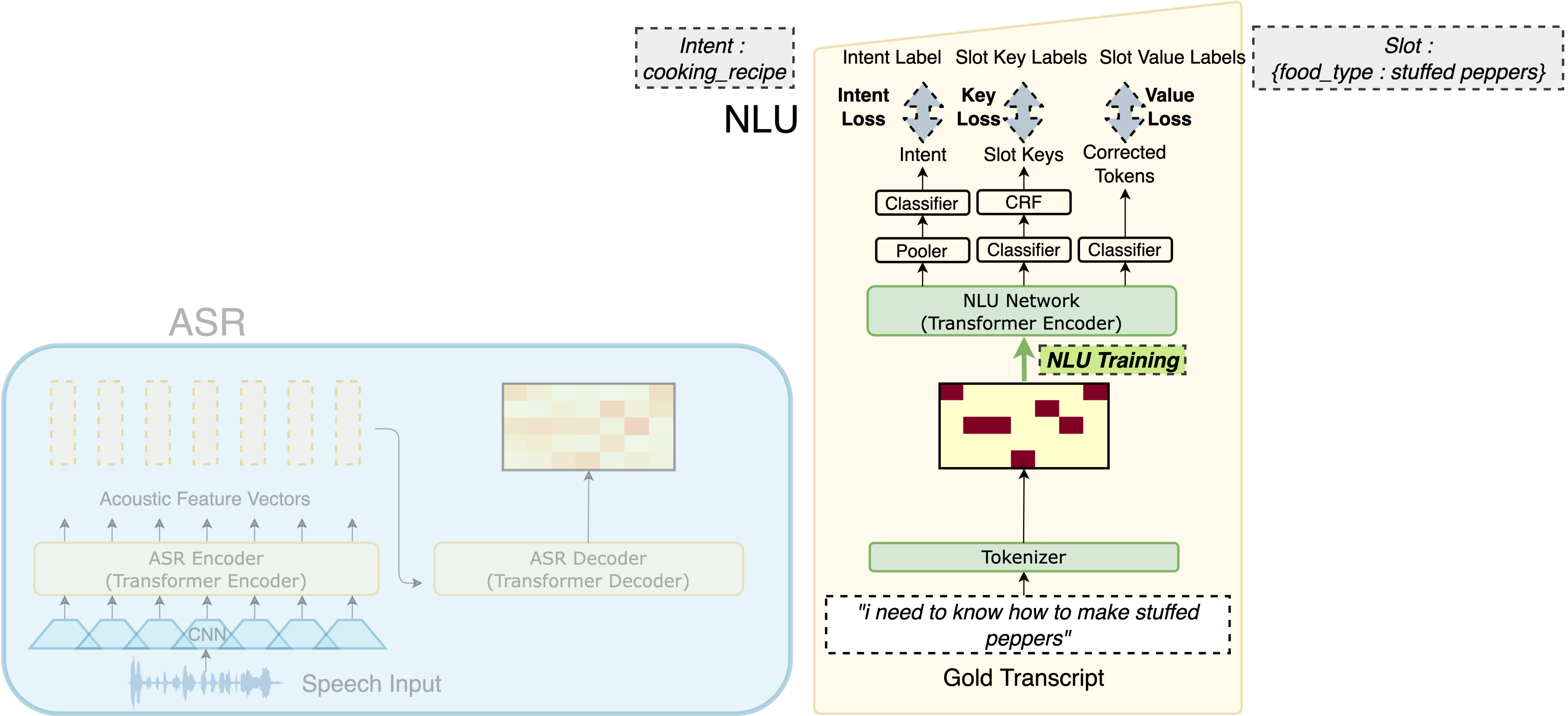


Proposed Method (ASR)

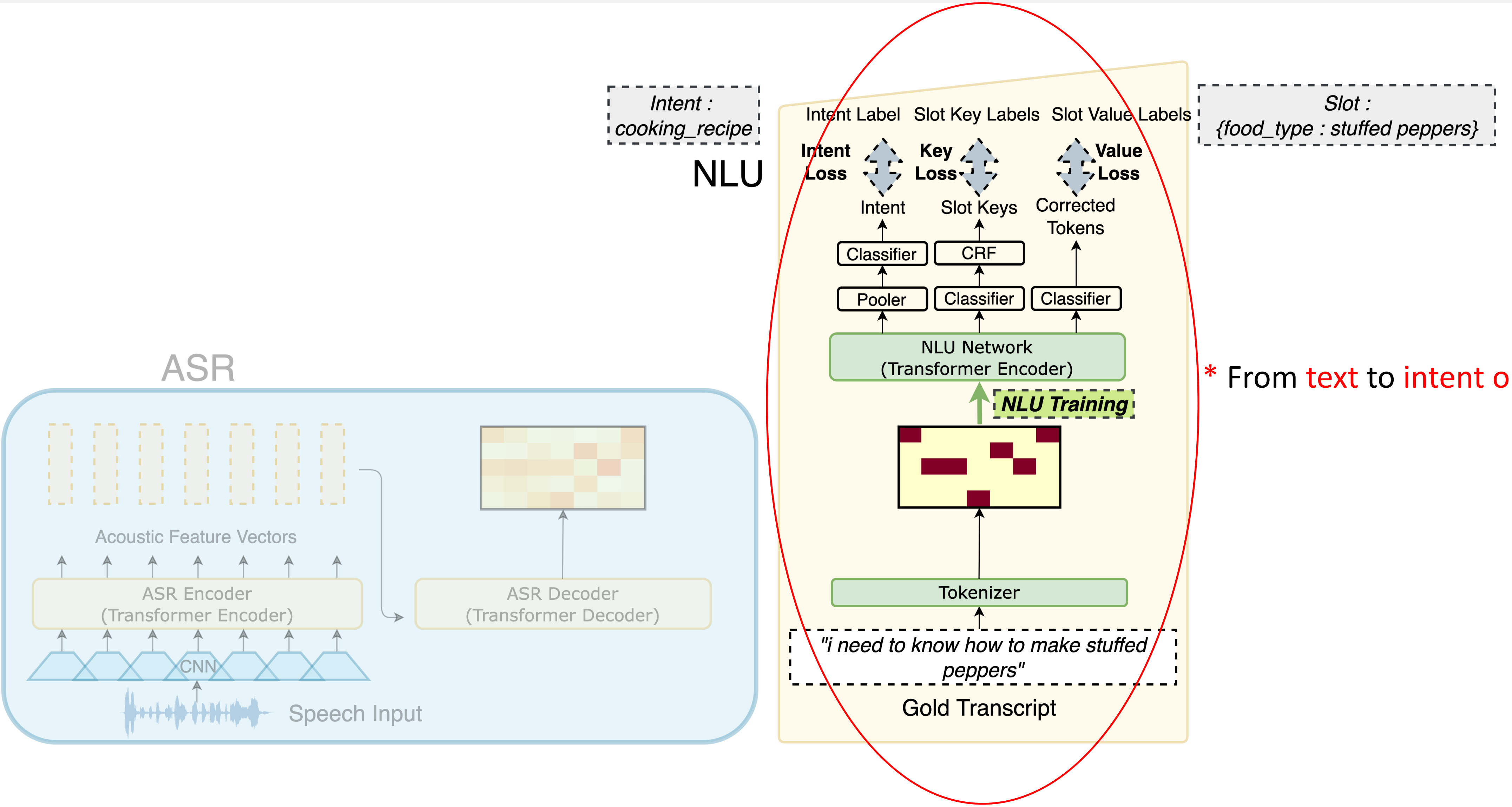
* From speech to text



Proposed Method (ASR+NLU)

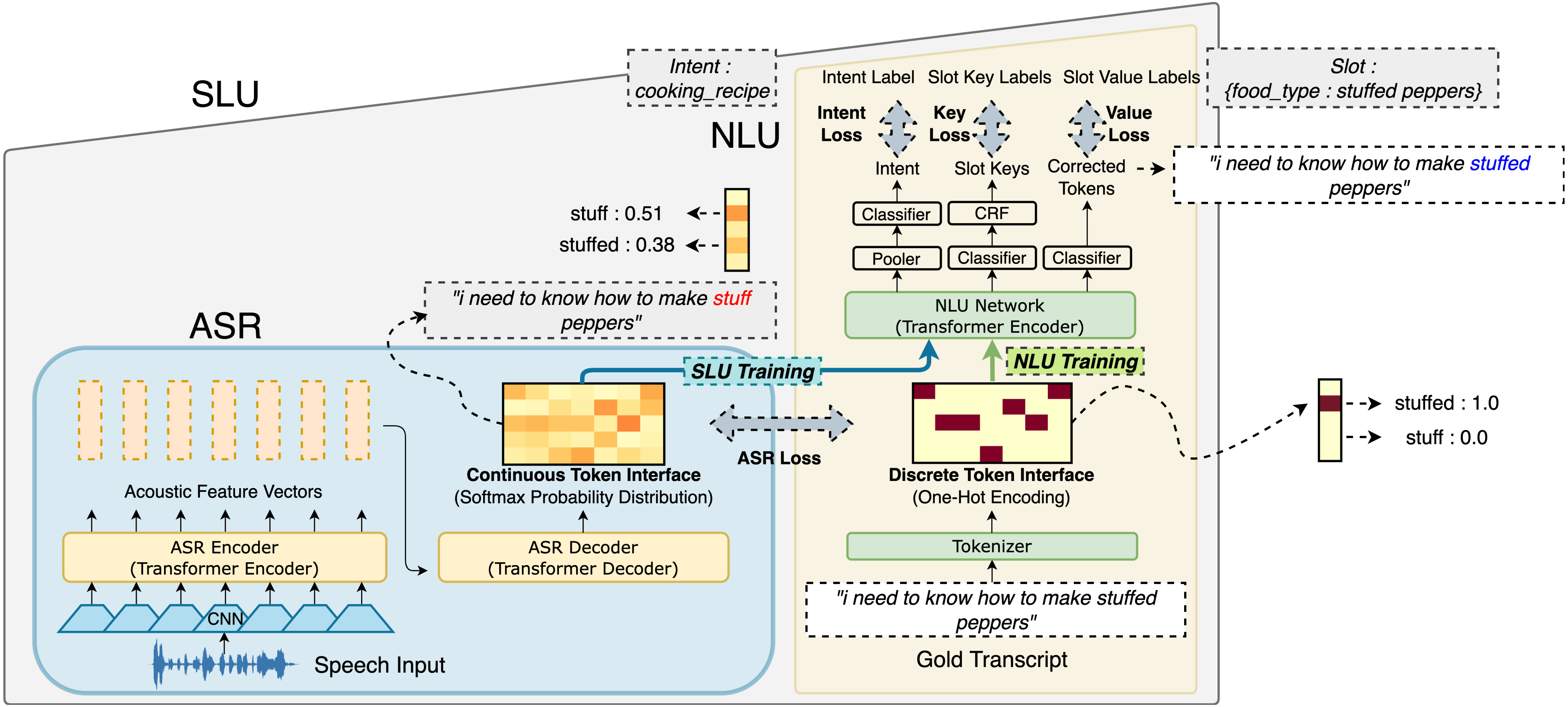


Proposed Method (ASR+NLU)

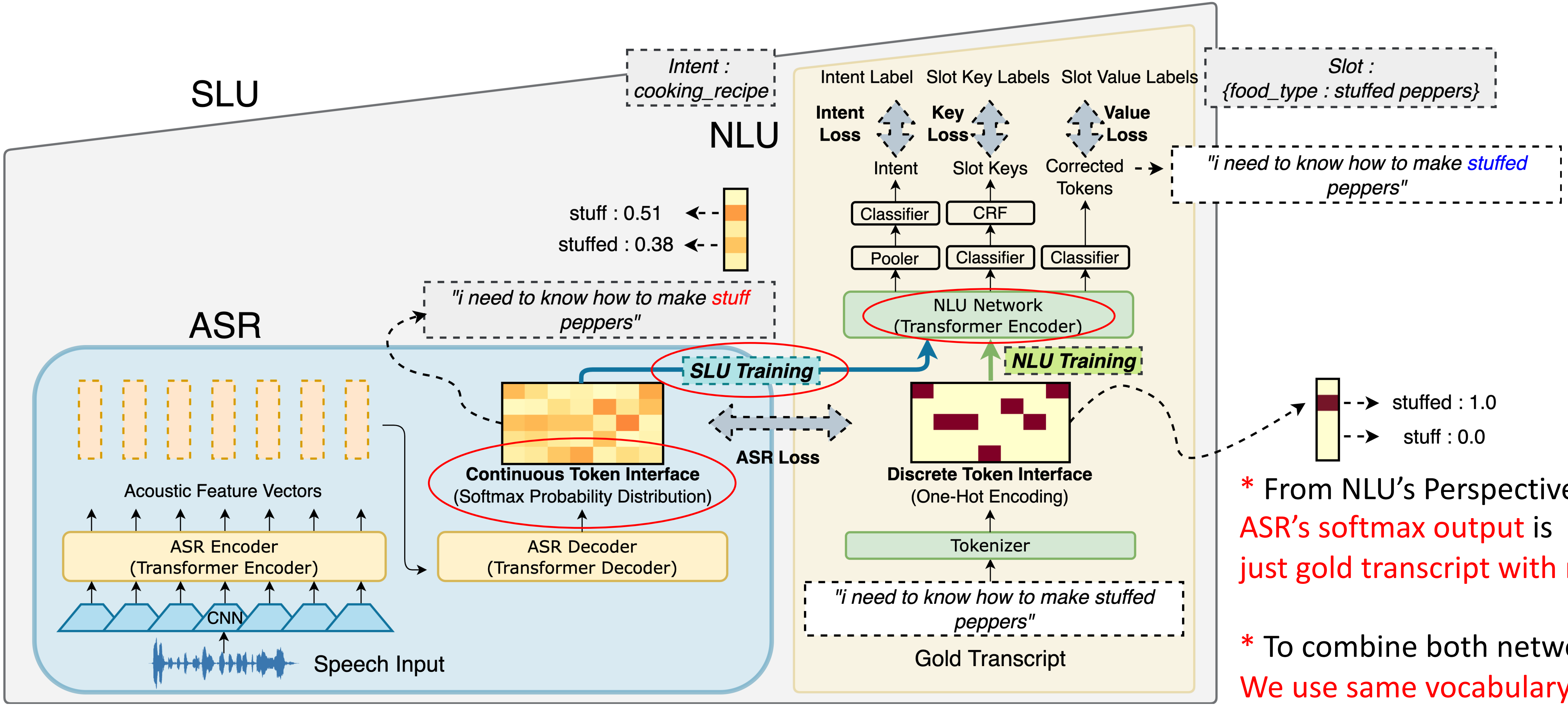


* From text to intent or slot

Proposed Method (ASR+NLU+SLU)



Proposed Method (ASR+NLU+SLU)



* From NLU's Perspective, ASR's softmax output is just gold transcript with noise

* To combine both network, We use same vocabulary for ASR and NLU Modules

Experimental Setup

- Dataset
    - For SLU : SLURP
    - For ASR Module Finetuning : Librispeech

Why SLURP ?

- SLURP is more challenging than other SLU datasets !

	FSC	SNIPS	SLURP	SLURP-synth
Speakers	97	69	**177**	34
Audio files	30,043	5,886	**72, 277**	69,253
-Close range	30,043	2,943	34,603	-
-Far range	-	2,943	37,674	-
Audio/Sentence	121.14	2.02	4.21	3.87
Distinct Trigrams (Lex)	250	5,703	**50, 422**	45,631
Unique Intents	31	6	93	93
Unique Slots	16	1,348	**5, 613**	4619

Why SLURP ?

- SLURP is more challenging than other SLU datasets !

	FSC	SNIPS	SLURP	SLURP-synth
Speakers	97	69	**177**	34
Audio files	30,043	5,886	**72,277**	69,253
-Close range	30,043	2,943	34,603	-
-Far range	-	2,943	37,674	-
Audio/Sentence	121.14	2.02	4.21	3.87
Distinct Trigrams (Lex)	250	5,703	**50,422**	45,631
Unique Intents	31	6	93	93
Unique Slots	16	1,348	**5,613**	4619

- Even SLU Network without NLU knowledge achieves SOTA on FSC

Model (E2E SLU)	Input	Dev	Test
Lugosh et al. [3]	Speech	-	98.8
Kim et al. [6]	Speech	97.8	99.7
Qian et al. [26]	Speech	-	99.7
Wav2Vec2.0-Classifier (Ours)	Speech	**98.9**	**99.7**

Experimental Setup

- Dataset
    - For SLU : SLURP
    - For ASR Module Finetuning : Librispeech
- Model Architecture Details
    - Transformer Seq2Seq for ASR Module (Target Vocab : gpt2 BPE tokenizer)
        - Pre-trained Wav2Vec 2.0 Base for Encoder
    - Transformer Encoder for NLU Module (Source Vocab : gpt2 BPE tokenizer)
        - Pre-trained RoBERTa Base

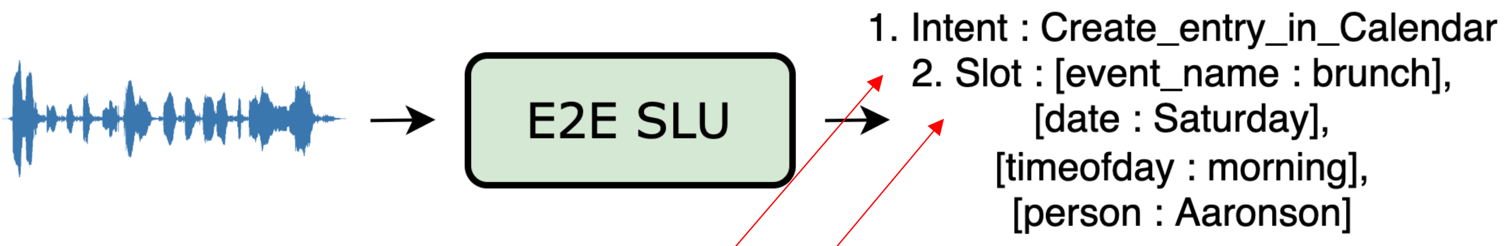
Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Metrics



For Intent Classification Task : Classification Accuracy

For Slot Filling Task : SLU-F1 (proposed in SLURP paper)

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

↑ Baselines

↓ Proposed

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

NLU is practical

Upper Bound

Text → Intent, Slot

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

SLU (Inference) is
Conventional SLU
Speech → Intent, Slot

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

ASR → NLU
is Conventional SLU
with DTI (argmax)

ASR ⇒ NLU
Is Conventional SLU
with CTI (softmax)

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Speech Encoder
+ Linear Classifier
(no NLU knowledge)

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Similar to our proposed SLU model but integrated with Gumbel Softmax module

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

↑ Baselines

↓ Proposed

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Proposed model with and without NLU loss
→ can improve model with text-only data

Results

Model Type	Model	Intent	SLU-F1
NLU	NLU [13]	84.84	-
	NLU (Ours)	87.73	84.34
SLU (Inference Only)	ASR→NLU [13]	78.33	70.84
	ASR→NLU (Ours)	80.37	70.23
	ASR⇒NLU (Ours)	81.17	70.20
SLU (E2E Train)	Wav2Vec2.0-Classifier (Ours)	76.6	-
	Gumbel-Interface (A+S+N) [11]	82.10	70.55
	CTI (A+S)	82.39	70.61
	CTI (A+S+N)	**82.93**	**71.12**
	CTI (A+S+N) + Extra data (Text-only)	84.34	71.08
	CTI (A+S+N) + Extra data (All)	**86.92**	**74.66**

Training with more data (synthetic speech data, SLURP-Synth) makes improvement

Summary

1. Integrating two pre-trained networks with CTI achieves SOTA on SLURP dataset (IC and SLU-F1 scores)

2. With CTI, we can train each component of the SLU network independently, even after integration.

3. Future work : better strategy for pre-training NLU

(e.g. by recovering some tokens corrupted by acoustic noise)

Thanks!

Q & A