Generating More Audios for End-to-End Spoken Language Understanding

Xuxin Cheng and **Yuexian Zou**
School of ECE, Peking University, China
chengxx@stu.pku.edu.cn, zouyx@pku.edu.cn

Abstract

End-to-end spoken language understanding (SLU) aims to directly capture the comprehensive semantics from the given spoken utterance without generating transcripts. Since transcripts may not always be available, Textless SLU is attracting increasing attention, which eliminates the need for transcripts but usually does not perform as well as SLU models trained with transcripts. In this paper, we focus on the scenarios where the transcripts are not available and propose GMA-SLU to generate more audio according to the labels. In order to solve the modality gap between text and audio, two different language models are built, and discrete tokens are utilized as a bridge, where the first language model utilizes labels to generate the semantic tokens and the second language model uses these semantic tokens and the acoustic tokens of source audios to obtain the synthetic audios. All experiments are conducted on the monolingual SLU dataset SLURP and the multilingual SLU dataset MINDS-14. Experimental results show that our method outperforms the previous best Textless End-to-end SLU models and can obtain a comparable performance with these models trained with the assistance of the corresponding transcripts.

1 Introduction

Spoken Language Understanding (SLU) focuses on comprehending the spoken utterances and generating relevant predictions, which is widely used in the personal assistants, spoken dialogue systems, and recent voice-controlled devices [Wang *et al.*, 2005; Cheng *et al.*, 2023a; Zhu *et al.*, 2023]. As demonstrated in Figure 1, for the audio input, SLU typically contains two subtasks: Slot Filling (SF) and Intent Detection (ID). SF is a sequence labeling task [Tur and De Mori, 2011], aiming to assign the slot to each token in spoken utterance, and ID is a classification task [Cheng *et al.*, 2023c], aiming to predict the intent label of the entire spoken utterance.

Traditional SLU approaches integrate an automatic speech recognition (ASR) model and a natural language understanding (NLU) model within the two-step pipelines [Hakkani-Tür *et al.*, 2006; Morbini *et al.*, 2012]. However, the prediction of ASR may contain errors and lose the prosodic information which is beneficial for NLU. Besides, cascaded systems usually have higher latency, which is not conducive to deploying models in real situations. For the reason, end-to-end SLU has attracted increasing attention in the recent years [Chen *et al.*, 2018; Chung *et al.*, 2021]. For end-to-end SLU, the predicted results are directly generated from input audio without generating the intermediate transcripts.

Due to the modality gap between audio representations and text embeddings [Mai *et al.*, 2020], training end-to-end SLU models directly is much more challenging than training cascade SLU models. To address the issue, a mainstream method is to train end-to-end SLU models with the assistance of transcripts. [Seo *et al.*, 2022] designs junctional representation to leverage the transcripts as the interface of the ASR model and the NLU model. [Ma *et al.*, 2023b] adopts knowledge distillation and leverages the transcripts to train an NLU model as the teacher model. However, owing to the exorbitant expenses of collecting the transcripts, transcripts are not always available. In addition, there are still thousands of unwritten languages in the world, it is impractical to obtain their corresponding transcripts [Zhang *et al.*, 2021]. Therefore, designing methods to train end-to-end SLU models without utilizing the transcripts is becoming an important research direction.

In the absence of transcripts, audio and their corresponding labels should be utilized more effectively to train the superior end-to-end SLU models. Motivated by the recent success of data augmentation in many other tasks [Fang and Feng, 2023; Cheng *et al.*, 2023b; Wang *et al.*, 2023b], we decide to generate more audios according to the labels. However, as a cross-modality task, directly utilizing labels to generate more audio will face two major difficulties:

(1) **Modeling difficulty.** Due to the modality gap between textual labels and audio inputs, training a single model to directly generate audios based on the labels is challenging. Motivated by the success of discrete tokens in various tasks [Hsu *et al.*, 2021; Zhang *et al.*, 2023a; Wang *et al.*, 2023a], we propose to employ the discrete tokens as a bridge. We separately train two different language models, where the first language model is leveraged to transform the labels to semantic tokens, and the second language model uses the semantic tokens and acoustic tokens of audio inputs to obtain the generated acoustic tokens. By using the two-stage generation method, we can effectively address the challenges in modeling.

(2) **Data scarcity.** By contrast, SLU dataset is relatively

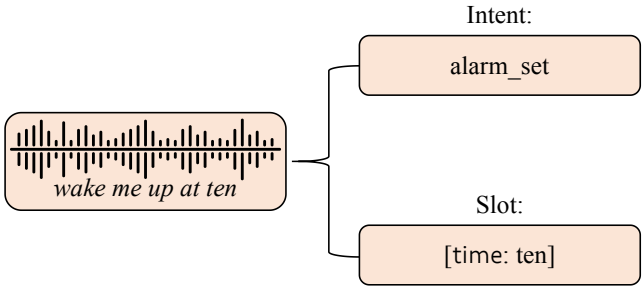

Figure 1: An illustration of Spoken Language Understanding, which includes two subtasks: intent detection and slot filling.

small [Bastianelli *et al.*, 2020; Gerz *et al.*, 2021]. To achieve the high performance without relying on transcripts for training, it is helpful to generate more audio data. As a result, we design multiple prompts for the first model, enabling the generation of more semantic tokens that align with specified slots and intents. Acoustic tokens from different source audios are fed into the second language model to enhance the generation of audios with the diverse speaker characteristics and thereby further expand the training data.

In this paper, we propose GMA-SLU to generate more audios based on the labels. All experiments are conducted on a monolingual SLU benchmark dataset SLURP [Bastianelli *et al.*, 2020] and the multilingual SLU dataset MINDS-14 [Gerz *et al.*, 2021]. Experimental results show that our method surpasses the previous best Textless end-to-end SLU models and achieves a comparable performance with the models trained with the assistance of transcripts. Further analyses also verify the advantages and effectiveness of our proposed method.

To sum up, the contributions of our method are three-fold:

- We propose the framework GMA-SLU, which generates more audios based on labels to enhance end-to-end SLU.
- We separately train two language models to tackle modeling difficulty and data scarcity.
- Experiment results on two benchmark datasets show that our proposed model surpasses the previous best model.

2 Related Work

2.1 Spoken Language Understanding

SLU aims at comprehending and interpreting the spoken language. Cascaded SLU methods work on the ASR transcripts, where solving the error propagation poses a significant challenge [Lee *et al.*, 2012; Chang and Chen, 2022]. Recent end-to-end SLU approaches have gained attention, especially with the performance gap compared to cascaded SLU systems being mitigated in many cases due to the rich knowledge of pre-trained models [Serdyuk *et al.*, 2018; Haghani *et al.*, 2018; Wang *et al.*, 2021]. However, most of previous SLU methods use transcripts for training, which are not always available. In our study, we explore leveraging the audio input more effectively to maintain the high performance.

2.2 Discrete Tokens

Discrete tokens are widely used in various speech processing tasks [Borsos *et al.*, 2022; Rubenstein *et al.*, 2023; Dong *et al.*, 2023b]. Current discrete tokens can be broadly classified into two types: semantic tokens [Baevski *et al.*, 2020; Hsu *et al.*, 2021] and acoustic tokens [Zeghidour *et al.*, 2021; Borsos *et al.*, 2022; Garcia *et al.*, 2023]. Semantic tokens are derived from pre-trained models that use masked language modeling as their training objective [Vaessen and Van Leeuwen, 2022]. These tokens primarily capture the content information in the speech while neglecting the paralinguistic aspects [Polyak *et al.*, 2021]. Acoustic tokens can be obtained from neural audio codecs with reconstruction as their training objective, aiming to capture all facets of information, including timbre, content, prosody, and recording conditions [Défossez *et al.*, 2022]. In our approach, we use semantic tokens as the bridge of the two language models to tackle the modality gap. Besides, we use acoustic tokens from different source audios to generate more audios with different speaker characteristics.

2.3 Language Models

Recently, natural language capabilities have obtained significant advancements via language modeling. Language modeling encompasses the approaches to predict the corresponding next tokens in the utterance or those masked spans [Devlin *et al.*, 2019]. [Brown *et al.*, 2020] introduces a novel large language model with 175 billion parameters, demonstrating the robust performance in many tasks in zero-shot, one-shot, and few-shot scenarios. [Ma *et al.*, 2023a] finds that existing language models are not effective few-shot information extractors and proposes the novel adaptive paradigm to leverage the strengths of language models effectively. In our approach, we design two language models to generate more pseudo audios.

3 Method

In this section, we introduce the problem definition (§3.1) and the model architecture (§3.2). The audio generation contains three parts, including discrete tokens generation (§3.3), label-to-token language model (§3.4), and token-to-audio language model (§3.5). Finally, we present the data filtration (§3.6) and the training strategy (§3.7) of our method. Note that these two language models are trained from scratch, and we also report the results obtained by supervised fine-tuning the recent large language models in Sec. 5.7 for reference.

3.1 Problem Definition

Given the audio $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$, where m is the length of audio waveform $\boldsymbol{x}$, cascade SLU methods first transform $\boldsymbol{x}$ to transcript $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ and then predict the intent $\boldsymbol{o}^I$ and the slot $\boldsymbol{o}^S = \left(o_1^S, o_2^S, \ldots, o_n^S\right)$, where n is the length of transcript $\boldsymbol{y}$. End-to-end SLU methods directly predict $\boldsymbol{o}^I$ and $\boldsymbol{o}^S$ based on $\boldsymbol{x}$ without generating $\boldsymbol{y}$. It has been verified that utilizing transcript $\boldsymbol{y}$ for auxiliary training could improve the performance of End-to-end SLU methods. However, considering that transcripts are not always available for training, we decide to train the Textless end-to-end SLU model without using transcripts in this paper unless otherwise specified.

3.2 Model Architecture

Following previous work [Wang *et al.*, 2021], our framework GMA-SLU consists of an acoustic encoder to generate representations for the audio input and two decoders for ID and SF,

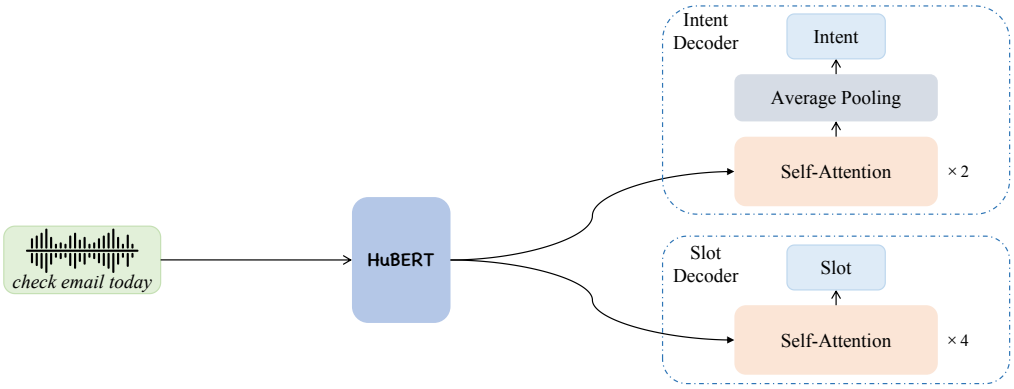

Figure 2: Overview of the model architecture, including the HuBERT as the acoustic encoder and two decoders for ID and SF, respectively.

respectively. For a fair comparison, we use the HuBERT [Hsu *et al.*, 2021] as the acoustic encoder. For ID, the decoder consists of two self-attention layers [Vaswani *et al.*, 2017] and the following average pooling layer. For SF, the decoder consists of four self-attention layers. The overview of model architecture is illustrated in Figure 2.

The training objectives of ID and SF are as follows:

$$\mathcal{L}_I \triangleq -\sum_{i=1}^{n_I} \hat{\boldsymbol{y}}^{i,I} \log\left(\boldsymbol{o}^{i,I}\right) \tag{1}$$

$$\mathcal{L}_S \triangleq -\sum_{j=1}^{n}\sum_{i=1}^{n_S} \hat{\boldsymbol{y}}_j^{i,S} \log\left(\boldsymbol{o}_j^{i,S}\right) \tag{2}$$

where $\hat{\boldsymbol{y}}^{i,I}$ and $\hat{\boldsymbol{y}}_j^{i,S}$ denote the gold intent label and gold slot label, respectively, n_I denotes the number of the intent labels, and n_S denotes the number of the slot labels.

The final training objective $\mathcal{L}$ is as follows:

$$\mathcal{L} = \alpha\mathcal{L}_I + (1-\alpha)\mathcal{L}_S \tag{3}$$

where α is the coefficient balancing the two subtasks.

3.3 Discrete Tokens Generation

We utilize two kinds of discrete tokens to train language models, including semantic tokens and acoustic tokens.

We use a pre-trained HuBERT[1] [Hsu *et al.*, 2021] to generate semantic tokens. HuBERT produces continuous representations at a rate of 50Hz for the audio. We adopt the K-means clustering algorithm to continuous representations of the audio input and transform these continuous representations into their corresponding cluster indices subsequently. These cluster indices are regarded as semantic tokens. By this approach, the audio input $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ is converted to semantic tokens $\boldsymbol{z} = (z_1, z_2, \ldots, z_T)$, where $z_t \in \{0, 1, ..., K-1\}$, z_t denotes any token in $\boldsymbol{z}$, K denotes the number of clusters, and $T = \lfloor \frac{m}{320} \rfloor$ denotes the number of frames. Through using semantic tokens, the modeling difficulty could be alleviated.

We utilize a pre-trained EnCodec[2] [Défossez *et al.*, 2022] to generate the acoustic tokens. The EnCodec model is a simple streaming and convolutional-based encoder-decoder architecture with a sequential modeling component utilized to the latent representation, both on the encoder side and decoder side, and the audio input is sampled at 24 kHz. Acoustic tokens can preserve the speech information more effectively than semantic tokens, so we decide to utilize them to assist in preserving the speech information of speakers.

3.4 Label-to-Token Language Model

As demonstrated in the left part of Figure 3, we train the label-to-token language model to generate semantic tokens according to the labels. Motivated by the recent success of the reduction strategy [Lee *et al.*, 2022], the repeating semantic tokens are merged to obtain the reduced semantic tokens. In order to facilitate the generation of semantic tokens, we design several prompts to construct the training corpus for natural language based on the semantic tokens and labels. For example, a simple prompt can be like: "*Generate the utterance whose intent is **[I_label]** and slot is **[S_label]**: **[semantic_token]***", where ***[I_label]*** will be replaced with the intent label, ***[S_label]*** will be replaced with the slot label, and ***[semantic_token]*** will be replaced with the reduced semantic tokens. More prompts are demonstrated in Table 1. Via using the prompts, the language model implicitly improves the alignment within the representation space between semantic tokens and labels. It should be noted that since the MINDS-14 dataset only includes the intent labels, the portion containing slot labels will be removed from the example prompts, and the processed prompt can be like: "*Generate the utterance whose intent is **[I_label]**: **[semantic_token]***". During the training stage, semantic tokens of the source audios are utilized, and during the inference stage, the generated semantic tokens are fed into the token-to-audio language model to generate the pseudo audios. The prompt is randomly selected during both training and inference.

3.5 Token-to-Audio Language Model

As illustrated in the right part of Figure 3, we train the token-to-audio language model to generate pseudo audios. Its input contains three parts, where the first part is the acoustic tokens of another randomly selected audio from the training data, the second part is the semantic tokens obtained by expanding the semantic tokens from the label-to-token language model, and the third part is the corresponding acoustic tokens.

[1]https://github.com/facebookresearch/fairseq/tree/main/examples/hubert

[2]https://github.com/facebookresearch/encodec

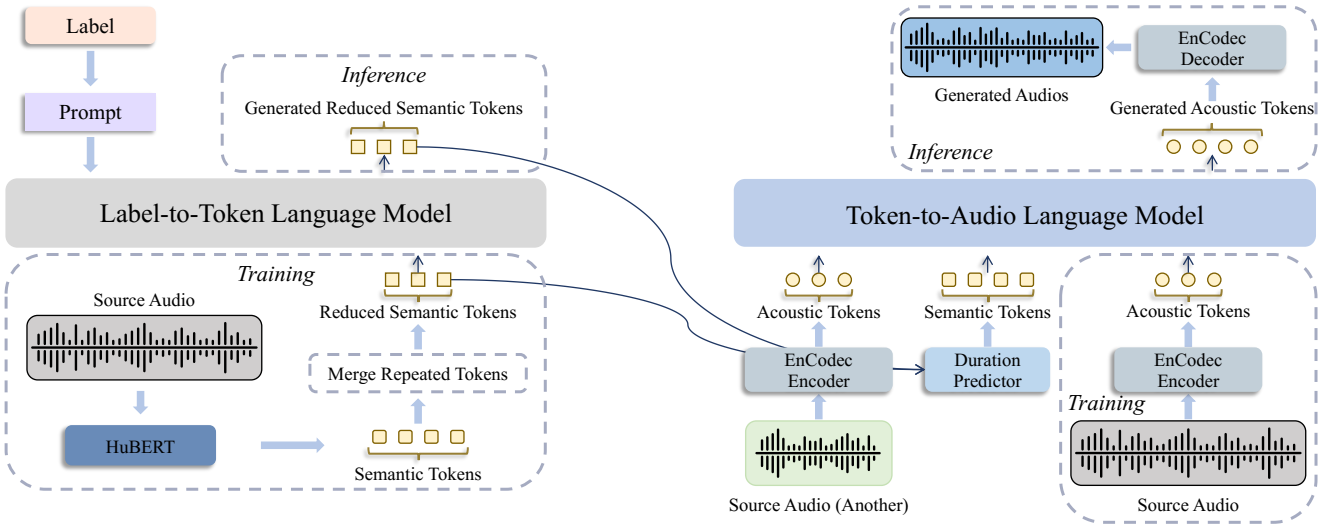

Figure 3: Overview of the two language models. Semantic tokens are used as the bridge between these two language models. During training, semantic tokens of source audios are used by the label-to-token language model, and acoustic tokens of source audios are used by the token-to-audio language model. During inference, the generated reduced semantic tokens by the label-to-token language model are then fed into the token-to-audio language model and the generated acoustic tokens by the token-to-audio language model are fed into the EnCodec decoder.

Data: <**I_label, S_label, semantic_token**>

Prompt 1:	*Generate the utterance whose intent is **[I_label]** and slot is **[S_label]**: **[semantic_token]***
Prompt 2:	*Generate the utterance whose slot is **[S_label]** and intent is **[I_label]**: **[semantic_token]***
Prompt 3:	*Generate a statement with the intent as **[I_label]** and slot as **[S_label]** : **[semantic_token]***
Prompt 4:	*Generate a statement with the slot as **[S_label]** and intent as **[I_label]**: **[semantic_token]***
Prompt 5:	*Produce an expression containing the slot **[S_label]** and the intent **[I_label]**: **[semantic_token]***
Prompt 6:	*Produce an expression containing the intent **[I_label]** and the slot **[S_label]**: **[semantic_token]***
Prompt 7:	*Craft a phrase with the slot labeled as **[S_label]** and the intent labeled as **[I_label]**: **[semantic_token]***
Prompt 8:	*Craft a phrase with the intent labeled as **[I_label]** and the slot labeled as **[S_label]**: **[semantic_token]***
Prompt 9:	*Construct a sentence denoting the slot as **[S_label]** and the intent as **[I_label]**: **[semantic_token]***
Prompt 10:	*Construct a sentence denoting the intent as **[I_label]** and the slot as **[S_label]** : **[semantic_token]***

Table 1: Prompts to construct the training data for the label-to-token language model.

Considering that the semantic tokens obtained by the label-to-token language model are reduced, we introduce a duration predictor [Ren *et al.*, 2020] to predict the duration of semantic tokens. The predictor consists of two 1D-convolutional layers with ReLU activation [Glorot *et al.*, 2011], each succeeded by layer normalization and the dropout layer. Additionally, there is a linear layer to project the hidden states into the duration. Given a predicted duration vector $\boldsymbol{d} = (d_1, d_2, \ldots, d_{T^*})$ and the ground truth $\hat{\boldsymbol{d}} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{T^*})$, the training objective $\mathcal{L}_D$ of the duration predictor is as follows:

$$\mathcal{L}_D = \frac{1}{T^*} \sum_{t=1}^{T^*} (\log(1 + d_t) - \log(1 + \hat{d}_t))^2 \qquad (4)$$

By using the duration vector, reduced semantic tokens can be expanded via repeating each semantic token. For instance, before being used as the input of the label-to-token language model, the semantic tokens (1, 2, 2, 3, 3, 4, 4) collapse to the reduced semantic tokens (1, 2, 3, 4) and the duration vector is (1, 2, 2, 2). Via predicting the duration, the reduced semantic tokens can be converted into original semantic tokens. During the training stage, ground truth duration is applied, and during the inference stage, the predicted duration is applied.

The acoustic tokens are generated by feeding the source audios into the encoder of the EnCodec model. Since the acoustic tokens can preserve useful speech information more effectively, we use them to assist in generating audios that matches the speech information of the speakers in the training set more accurately. To solve data scarcity, we randomly select another source audio from the training set to obtain acoustic tokens as the first part of the input. During the whole training stage, the third part is obtained from these source audios, and during inference, the generated acoustic tokens are fed into the decoder of the EnCodec model to synthesize the audios.

3.6 Data Filtration

In order to filter out the low-quality generated audios, we train an additional SLU model to predict the intent and slot of the generated audios, whose model architecture is the same as the architecture in Sec 3.2. We assess the intent accuracy and the slot SLU-F1 of the generated audios. Only the top $\rho\%$ audios are kept based on the sum of intent accuracy and slot SLU-F1. Similarly, since only intent labels are available in MINDS-14 dataset, only intent accuracy is considered for data filtration.

3.7 Training Strategy

Though we apply the data filtration, the generated audios still contain the noise. Therefore, to dominate the training process with the real audios, we first pre-train the SLU model utilizing generated audios, followed by fine-tuning on real audios.

4 Experiments

4.1 Datasets and Metrics

We conduct all the experiments on a monolingual SLU benchmark dataset SLURP[3] [Bastianelli *et al.*, 2020] and the multilingual SLU dataset MINDS-14[4] [Gerz *et al.*, 2021]. SLURP dataset constitutes the repository encompassing 72.2k real audio recordings and 69.3k synthetic audio for a broad range of speech commands, each encapsulating brief interactions with the home assistant, which is considered the most challenging SLU dataset due to its lexical complexity. SLURP is meticulously annotated across 3 semantic layers, including scenario, action, and entities, where a pair (scenario, action) defines an intent. Unlike SLURP dataset focuses on English utterances, MINDS-14 is a multilingual SLU dataset for the banking scenarios with 14 distinct intents in 14 languages. Each language incorporates approximately 600 utterances. We report the results of 4 languages, including en-US, fr-FR, pl-PL, and ko-KR with a 30-20-50% train-dev-test split following the previous work for a fair comparison [Conneau *et al.*, 2022].

Following [Dong *et al.*, 2023a; Peng *et al.*, 2022; Conneau *et al.*, 2022], for SLURP, we evaluate the performance of slot filling with SLU-F1 score, intent detection with accuracy, and for MINDS-14, we evaluate the performance of intent detection in different languages with accuracy.

4.2 Implementation Details

We utilize the pre-trained HuBERT[5] [Hsu *et al.*, 2021] following the base configuration as the acoustic encoder. Following previous work [Wang *et al.*, 2021], the batch size is set to 16. We apply the Adam optimizer [Kingma and Ba, 2015], and 4k warm-up updates to optimize parameters, where the learning rate is increased from 4e-4 to 2e-3. If the loss on *dev* set does not decrease for 5 epochs, the training process will early-stop to avoid overfitting. For all experiments, we choose the model that achieves the best performance on the *dev* set and evaluate it on the *test* set. The weight α is set to 0.9. For the semantic tokens, we utilize the pre-trained quantized model[6] where the number of clusters K is 100 to convert the audios to semantic tokens. During the data filtration stage, we set the threshold ρ to 80. All experiments are conducted at an Nvidia V100.

4.3 Baselines

SLURP We compare our GMA-SLU with four SLU baselines trained without transcripts, including MTL-SLT [Huang *et al.*, 2022], Speech-Brain [Ravanelli *et al.*, 2021], Branchformer [Peng *et al.*, 2022], and HuBERT SLU [Wang *et al.*, 2021]. We also report the results of SLU models trained with the assistance of transcripts for reference, including CTI [Seo *et al.*, 2022], CIF-PT [Dong *et al.*, 2023a] with the Conformer [Gulati *et al.*, 2020] as the acoustic encoder and CIF-PT with the pre-trained Data2vec [Baevski *et al.*, 2022] as the acoustic encoder. In addition, we also report the results of recent large language models for reference, including ChatGPT [OpenAI, 2023] and SpeechGPT [Zhang *et al.*, 2023a]. Since the pre-trained models applied by different SLU models might be different, we list the pre-trained models in Table 2 for reference.

MINDS-14 We compare our GMA-SLU with LaBSE [Gerz *et al.*, 2021] and XLSR [Lozhkov, 2022]. In addition, we also report the results of ChatGPT for reference. LaBSE performs SLU in a cascade manner and the input is the ASR transcripts. BERT [Devlin *et al.*, 2019] is applied as its pre-trained model. XLSR is trained without the assistance of transcripts and performs SLU in an end-to-end manner as GMA-SLU. Wav2vec 2.0 [Baevski *et al.*, 2020] is the pre-trained model of XLSR. The performance of ChatGPT is from [He and Garner, 2023].

5 Results and Analysis

5.1 Results on SLURP Dataset

Experimental results on SLURP dataset are shown in Table 2, from which we have the following observations:

(1) Our approach achieves the consistent improvements in slot filling and intent detection. Specifically, GMA-SLU surpasses the previous best model training without transcripts by 2.01% SLU-F1 and 1.59% accuracy. This improvement could be attributed to the proposed audio generation strategy, which can enlarge the training set and improve the robustness.

(2) Another encouraging result is that our method achieves the comparable performance with the models trained with the assistance of transcripts, which outperforms CTI and CIF-PT with Conformer as the acoustic encoder. This result indicates that generating more audios through our approach can alleviate the reliance on transcripts, which is particularly meaningful in scenarios where transcripts are unavailable.

(3) Following the previous work [He and Garner, 2023], we use ASR transcripts to evaluate the performance of ChatGPT and SpeechGPT. In this evaluation, these models are provided with 20 examples and prompted to adapt to ASR errors. However, though these large language models have shown the superiority in few-shot learning and zero-shot learning, there is still a performance gap between them and our approach. This result implies that large language models might encounter difficulties in understanding spoken utterances. As a result, it is still an essential task to establish a robust SLU framework and it warrants further exploration and investigation.

5.2 Results on MINDS-14 Dataset

Table 3 shows the experimental results on MINDS-14 dataset. We can obviously observe that the proposed approach outperforms previous models in all languages, which further verifies the effectiveness of our approach.

5.3 Ablation Study

To verify the benefits of our approach from various angles, we undertake ablation studies on SLURP and MINDS-14, whose

[3]https://github.com/pswietojanski/slurp

[4]https://huggingface.co/datasets/PolyAI/minds14

[5]https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

[6]https://dl.fbaipublicfiles.com/textless_nlp/gslm/hubert/km100/km.bin

Model	Pre-trained Model	Slot (SLU-F1) ↑	Intent (Acc) ↑
Training w/o Transcripts			
MTL-SLT [Huang *et al.*, 2022]	LAS [Chan *et al.*, 2016] + BART [Lewis *et al.*, 2020]	74.49	83.10
Speech-Brain [Ravanelli *et al.*, 2021]	wav2vec 2.0 [Baevski *et al.*, 2020]	74.62	85.34
Branchformer [Peng *et al.*, 2022]	-	77.70	88.10
HuBERT SLU [Wang *et al.*, 2021]	HuBERT [Hsu *et al.*, 2021]	78.92	89.38
Training w/ Transcripts			
CTI [Seo *et al.*, 2022]	wav2vec 2.0 [Baevski *et al.*, 2020] + RoBERTa [Liu *et al.*, 2019]	74.66	86.92
CIF-PT [Dong *et al.*, 2023a]	-	78.67	89.60
CIF-PT [Dong *et al.*, 2023a]	Data2vec [Baevski *et al.*, 2022]	81.63	91.32
Large Language Models			
ChatGPT [OpenAI, 2023]	-	62.76	73.96
SpeechGPT [Zhang *et al.*, 2023a]	-	61.23	72.61
GMA-SLU w/o Reduced Strategy	HuBERT [Hsu *et al.*, 2021]	79.68 (↓1.25)	90.39 (↓0.58)
GMA-SLU w/o Different Acoustic Tokens	HuBERT [Hsu *et al.*, 2021]	79.45 (↓1.48)	90.13 (↓0.84)
GMA-SLU w/o Data Filtration	HuBERT [Hsu *et al.*, 2021]	79.57 (↓1.36)	90.24 (↓0.73)
GMA-SLU w/o Two-Stage Training Strategy	HuBERT [Hsu *et al.*, 2021]	79.82 (↓1.11)	90.55 (↓0.42)
GMA-SLU (ours)	HuBERT [Hsu *et al.*, 2021]	**80.93**†	**90.97**†

Table 2: SLU-F1 of slot filling and accuracy (Acc) of intent detection on the SLURP dataset. '†' denotes our GMA-SLU achieves statistically significant improvements over baselines with $p < 0.01$. 'Training w/o Transcripts' indicates the SLU model is trained without transcripts and 'Training w/ Transcripts' indicates the SLU model is trained with the assistance of transcripts. 'Pre-trained Model' indicates the corresponding pre-trained model used by the SLU model. Considering that the transcripts are not always available, GMA-SLU is trained **without** transcripts.

Model	en-US	fr-FR	pl-PL	ko-KR
LaBSE [Gerz *et al.*, 2021]	95.1	93.1	89.2	91.4
XLSR [Lozhkov, 2022]	93.3	94.4	91.5	86.5
ChatGPT (0-shot)	95.4	97.4	90.0	89.2
ChatGPT (1-shot)	97.9	99.3	96.1	90.5
GMA-SLU w/o RS	96.6 (1.6↓)	98.6 (0.9↓)	95.2 (1.1↓)	91.3 (0.5↓)
GMA-SLU w/o DAT	96.1 (2.1↓)	97.8 (1.7↓)	94.4 (1.9↓)	90.5 (1.3↓)
GMA-SLU w/o DF	96.3 (1.9↓)	98.2 (1.3↓)	94.7 (1.6↓)	90.9 (0.9↓)
GMA-SLU w/o TSTS	96.8 (1.4↓)	98.9 (0.6↓)	95.5 (0.8↓)	91.5 (0.3↓)
GMA-SLU (ours)	**98.2**†	**99.5**†	**96.3**†	**91.8**†

Table 3: Intent detection results for different languages in MINDS-14 dataset. '†' denotes GMA-SLU achieves the statistically significant improvements over baselines with $p < 0.01$.

outcomes are shown in the lower part of Table 2 and Table 3.

Effect of Reduced Strategy We apply the reduced strategy to merge the repeated semantic tokens in the proposed label-to-token language model and utilize the duration predictor to expand the reduced semantic tokens in the token-to-audio language model. To evaluate the superiority of the reduced strategy, we conduct an ablation experiment where we remove the reduced strategy, and the original semantic tokens are used as the bridge between these two proposed language models. We refer this experiment to *w/o Reduced Strategy* in Table 2 and *w/o RS* in Table 3. We observe the significant decreases in all the metrics across the two datasets, affirming that the reduced strategy makes the positive contribution to our method, which aligns with past observations in [Lee *et al.*, 2022]. We believe it is because reduced strategy could make the semantic tokens resemble text more closely and simplify the whole generation process due to the shortened length of semantic tokens.

Effect of Different Acoustic Tokens In the token-to-audio language model, acoustic tokens of another audio in the training set are used as the first part of the input. To verify that it is more effective to use different acoustic tokens, we remove it and retain only the second and third parts of the input. We refer the experiment to *w/o Different Acoustic Tokens* in Table 2 and *w/o DAT* in Table 3. We could clearly discern that the absence of the first part leads to the decreased performance. We believe the reason is that applying the first part can solve *data scarcity* issue more effectively. Besides, via incorporating the acoustic tokens as the first part of the inputs, the acoustic tokens of the third part could align more closely with the established rules of acoustic tokens during the generation process.

Effect of Data Filtration To filter out the low-quality generated audios, we first perform the data filtration before training the SLU model. To verify the superiority of data filtration, we remove it and refer it to *w/o Data Filtration* in Table 2 and *w/o DF* in Table 3. It is evident that the performance declines after removing the data filtration, which suggests that the data filtration indeed could improve the performance. It is because even the two language models are meticulously designed, the noise in the generated audios is still inevitable.

Effect of Two-Stage Training Strategy We train the SLU model using a two-stage training strategy to alleviate the negative effect of the noise in generated audios. In order to evaluate its effectiveness, we remove it and refer this ablation experiment to *w/o Two-Stage Training Strategy* and *w/o TSTS* in Table 2 and Table 3, respectively. We could observe a decline in performance on these two datasets, which substantiates that the two-stage training strategy is effective.

	Text:	call	the nearest	thai	restaurant	with delivery
Ref.	Intent:	takeway_order				
	Slot:			food_type	bussiness_type	order_type
HuBERT SLU	Intent:	*cooking_order*				
	Slot:			*coffee_type*	bussiness_type	order_type
GMA-SLU	Intent:	takeway_order				
	Slot:			food_type	bussiness_type	order_type

Table 4: Case study of HuBERT SLU and GMA-SLU on SLURP dataset. Text in *italic* denotes the incorrect predictions.

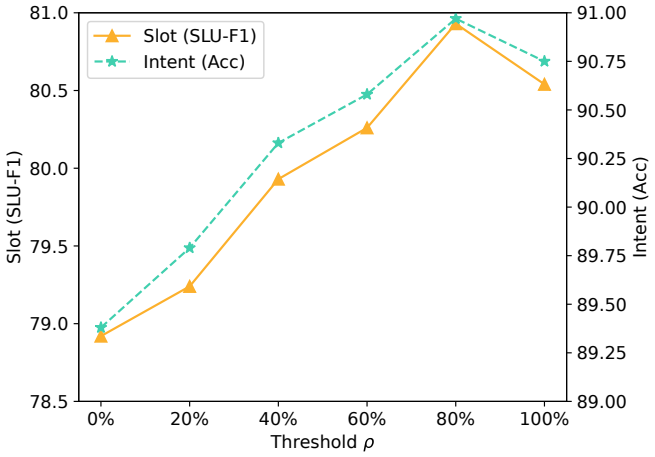

Figure 4: SLU-F1 of slot filling and accuracy (Acc) of intent detection on the SLURP dataset with different threshold ρ.

Model	Slot (SLU-F1) ↑	Intent (ACC) ↑
SpeechTokenizer [Zhang *et al.*, 2023b]	81.25	91.23
TiCodec [Ren *et al.*, 2023]	80.98	91.29
HiFi-Codec [Yang *et al.*, 2023]	81.15	91.18
GMA-SLU (ours)	80.93	90.97

Table 5: SLU-F1 of slot filling and accuracy (Acc) of intent detection on the SLURP dataset with different audio codecs.

Model	Slot (SLU-F1) ↑	Intent (ACC) ↑
Llama 2 [Touvron *et al.*, 2023]	83.46	92.51
SpeechGPT [Zhang *et al.*, 2023a]	86.72	94.53
LTU-AS [Gong *et al.*, 2023]	87.35	94.32
GMA-SLU (ours)	80.93	90.97

Table 6: SLU-F1 of slot filling and accuracy (Acc) of intent detection on the SLURP dataset with different large language models. For simplicity, we only use the large language models as the pre-trained model of the label-to-token language model.

5.4 Impact of Data Filtration Ratio

When performing the data filtration, it is important to choose the appropriate threshold ρ. To investigate how ρ affects SLU performance, we constrain ρ in [0, 20, 40, 60, 80, 100] on the SLURP, whose results are shown in Figure 3. When $\rho = 100$, data filtration is equivalent to being removed, which indicates that all generated audios are retrained, and when $\rho = 0$, our method degrades to HuBERT SLU [Wang *et al.*, 2021]. From the results, we observe that our method achieves the best performance at $\rho = 80\%$. Besides, when $\rho = 100\%$, the performance is still relatively high, which verifies that the generated audios maintain a relatively high level of quality.

5.5 Case Study

To further showcase the effectiveness of our approach in handling SLU, we provide a case study on SLURP dataset applying HuBERT SLU and our GMA-SLU. As shown in Table 4, HuBERT SLU predicts the scenario of intent and the slots of some tokens incorrectly, whereas our model can predict them accurately. This result substantiates that the generated audios indeed has the potential to enhance the performance.

5.6 Effect of Different Audio Codecs

Recently, there has been an increasing amount of research on audio codecs, and more and more speech tokenizers with high performance are proposed. To verify that our approach is also adapted to the advanced tokenizers, we replace the EnCodec with SpeechTokenizer [Zhang *et al.*, 2023b], TiCodec [Ren *et al.*, 2023], and HiFi-Codec [Yang *et al.*, 2023], respectively. The corresponding results on SLURP are demonstrated in Table 5. We can observe that through employing these advanced speech tokenizers, the corresponding performance can be further enhanced, underscoring the generality of our method.

5.7 Effect of Large Language Model

In this study, we train both two language models from scratch. In order to explore the effect of recent large language models, we perform supervised fine-tuning to the large language models to obtain the label-to-token language model. As shown in Table 6, we observe that the recent large language models can indeed enhance the performance of SLU. We attribute this improvement to abundant semantic knowledge in large language models. Besides, SpeechGPT [Zhang *et al.*, 2023a] and LTU-AS [Gong *et al.*, 2023] can perform better than Llama 2 [Touvron *et al.*, 2023]. We believe this is because SpeechGPT and LTU-AS are meticulously designed for audios, which enables the generation of more accurate semantic tokens.

6 Conclusion

In this paper, we propose the framework GMA-SLU for SLU, which utilizes two language models to generate more audios as additional training data. Experiments on two datasets show that our model surpasses the previous best Textless SLU models and obtains a comparable performance with recent end-to-end SLU models trained with the assistance of transcripts. We believe that our approach is generalizable to more fully utilize limited annotated data and thus improve the performance.

Acknowledgments

We thank all the reviewers for their insightful comments. This paper was partially supported by NSFC (No: 62176008).

References

[Baevski *et al.*, 2020] Alexei Baevski, Yuhao Zhou, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proc. of NeurIPS*, 2020.

[Baevski *et al.*, 2022] Alexei Baevski, Wei-Ning Hsu, et al. data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proc. of ICML*, 2022.

[Bastianelli *et al.*, 2020] Emanuele Bastianelli, Andrea Vanzo, et al. SLURP: A spoken language understanding resource package. In *Proc. of EMNLP*, 2020.

[Borsos *et al.*, 2022] Zalán Borsos, Raphaël Marinier, et al. Audiolm: a language modeling approach to audio generation, 2022.

[Brown *et al.*, 2020] Tom B. Brown, Benjamin Mann, et al. Language models are few-shot learners. In *Proc. of NeurIPS*, 2020.

[Chan *et al.*, 2016] William Chan, Navdeep Jaitly, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proc. of ICASSP*, 2016.

[Chang and Chen, 2022] Ya-Hsin Chang and Yun-Nung Chen. Contrastive Learning for Improving ASR Robustness in Spoken Language Understanding. In *Proc. of Interspeech*, 2022.

[Chen *et al.*, 2018] Yuan-Ping Chen, Ryan Price, et al. Spoken language understanding without speech recognition. In *Proc. of ICASSP*, 2018.

[Cheng *et al.*, 2023a] Xuxin Cheng, Bowen Cao, et al. Mllmcl: Mutual learning and large-margin contrastive learning for improving asr robustness in spoken language understanding. In *Proc. of ACL Findings*, 2023.

[Cheng *et al.*, 2023b] Xuxin Cheng, Qianqian Dong, et al. M 3 st: Mix at three levels for speech translation. In *Proc. of ICASSP*, 2023.

[Cheng *et al.*, 2023c] Xuxin Cheng, Zhihong Zhu, et al. Mrrl: Modifying the reference via reinforcement learning for non-autoregressive joint multiple intent detection and slot filling. In *Proc. of EMNLP Findings*, 2023.

[Chung *et al.*, 2021] Yu-An Chung, Chenguang Zhu, et al. SPLAT: Speech-language joint pre-training for spoken language understanding. In *Proc. of NAACL*, 2021.

[Conneau *et al.*, 2022] Alexis Conneau, Ankur Bapna, et al. XTREME-S: Evaluating Cross-lingual Speech Representations. In *Proc. of Interspeech*, 2022.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 2019.

[Dong *et al.*, 2023a] Linhao Dong, Zhecheng An, et al. CIF-PT: Bridging speech and text representations for spoken language understanding via continuous integrate-and-fire pre-training. In *Proc. of ACL Findings*, 2023.

[Dong *et al.*, 2023b] Qianqian Dong, Zhiying Huang, et al. Polyvoice: Language models for speech to speech translation, 2023.

[Défossez *et al.*, 2022] Alexandre Défossez, Jade Copet, et al. High fidelity neural audio compression, 2022.

[Fang and Feng, 2023] Qingkai Fang and Yang Feng. Back translation for speech-to-text translation without transcripts. In *Proc. of ACL*, 2023.

[Garcia *et al.*, 2023] Hugo Flores Garcia, Prem Seetharaman, et al. Vampnet: Music generation via masked acoustic token modeling. *ArXiv preprint*, 2023.

[Gerz *et al.*, 2021] Daniela Gerz, Pei-Hao Su, et al. Multilingual and cross-lingual intent detection from spoken data. In *Proc. of EMNLP*, 2021.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, et al. Deep sparse rectifier neural networks. In *Proc. of AISTATS*, 2011.

[Gong *et al.*, 2023] Yuan Gong, Alexander H Liu, et al. Joint audio and speech understanding. In *Proc. of ASRU*, 2023.

[Gulati *et al.*, 2020] Anmol Gulati, James Qin, et al. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. of INTERSPEECH*, 2020.

[Haghani *et al.*, 2018] Parisa Haghani, Arun Narayanan, et al. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[Hakkani-Tür *et al.*, 2006] Dilek Hakkani-Tür, Frédéric Béchet, et al. Beyond asr 1-best: Using word confusion networks in spoken language understanding. *Computer Speech & Language*, 2006.

[He and Garner, 2023] Mutian He and Philip N. Garner. Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding. In *Proc. of Interspeech*, 2023.

[Hsu *et al.*, 2021] Wei-Ning Hsu, Benjamin Bolte, et al. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[Huang *et al.*, 2022] Zhiqi Huang, Milind Rao, et al. MTL-SLT: Multi-task learning for spoken language tasks. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, 2022.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

[Lee *et al.*, 2012] Hung-yi Lee, Chia-ping Chen, et al. Integrating recognition and retrieval with relevance feedback for spoken term detection. *IEEE transactions on audio, speech, and language processing*, 2012.

[Lee *et al.*, 2022] Ann Lee, Peng-Jen Chen, et al. Direct speech-to-speech translation with discrete units. In *Proc. of ACL*, 2022.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*, 2020.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, et al. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, 2019.

[Lozhkov, 2022] Anton Lozhkov. Hugging Face: anton-l/xtreme_s_xlsr_300m_minds14. https://huggingface.co/anton-l/xtreme_s_xlsr_300m_minds14, 2022.

[Ma *et al.*, 2023a] Yubo Ma, Yixin Cao, et al. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Proc. of EMNLP Findings*, 2023.

[Ma *et al.*, 2023b] Yukun Ma, Trung Hieu Nguyen, et al. Auxiliary pooling layer for spoken language understanding. In *Proc. of ICASSP*, 2023.

[Mai *et al.*, 2020] Sijie Mai, Haifeng Hu, et al. Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion. In *Proc. of AAAI*, 2020.

[Morbini *et al.*, 2012] Fabrizio Morbini, Kartik Audhkhasi, et al. A reranking approach for recognition and classification of speech input in conversational dialogue systems. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012.

[OpenAI, 2023] OpenAI. Chatgpt. https://chat.openai.com, 2023.

[Peng *et al.*, 2022] Yifan Peng, Siddharth Dalmia, et al. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *Proc. of ICML*, 2022.

[Polyak *et al.*, 2021] Adam Polyak, Yossi Adi, et al. Speech resynthesis from discrete disentangled self-supervised representations. In *Proc. of INTERSPEECH*, 2021.

[Ravanelli *et al.*, 2021] Mirco Ravanelli, Titouan Parcollet, et al. Speechbrain: A general-purpose speech toolkit. *ArXiv preprint*, 2021.

[Ren *et al.*, 2020] Yi Ren, Chenxu Hu, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *Proc. of ICLR*, 2020.

[Ren *et al.*, 2023] Yong Ren, Tao Wang, et al. Fewer-token neural speech codec with time-invariant codes, 2023.

[Rubenstein *et al.*, 2023] Paul K. Rubenstein, Chulayuth Asawaroengchai, et al. Audiopalm: A large language model that can speak and listen, 2023.

[Seo *et al.*, 2022] Seunghyun Seo, Donghyun Kwak, et al. Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding. In *Proc. of ICASSP*, 2022.

[Serdyuk *et al.*, 2018] Dmitriy Serdyuk, Yongqiang Wang, et al. Towards end-to-end spoken language understanding. In *Proc. of ICASSP*, 2018.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, 2023.

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

[Vaessen and Van Leeuwen, 2022] Nik Vaessen and David A Van Leeuwen. Fine-tuning wav2vec2 for speaker recognition. In *Proc. of ICASSP*, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, et al. Attention is all you need. In *Proc. of NeurIPS*, 2017.

[Wang *et al.*, 2005] Ye-Yi Wang, Li Deng, et al. Spoken language understanding. *IEEE Signal Processing Magazine*, 2005.

[Wang *et al.*, 2021] Yingzhi Wang, Abdelmoumene Boumadane, et al. A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding. *ArXiv preprint*, 2021.

[Wang *et al.*, 2023a] Chengyi Wang, Sanyuan Chen, et al. Neural codec language models are zero-shot text to speech synthesizers, 2023.

[Wang *et al.*, 2023b] Qing Wang, Jun Du, et al. A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[Yang *et al.*, 2023] Dongchao Yang, Songxiang Liu, et al. Hifi-codec: Group-residual vector quantization for high fidelity audio codec, 2023.

[Zeghidour *et al.*, 2021] Neil Zeghidour, Neil, Luebs, et al. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[Zhang *et al.*, 2021] Chen Zhang, Xu Tan, et al. Uwspeech: Speech to speech translation for unwritten languages. In *Proc. of AAAI*, 2021.

[Zhang *et al.*, 2023a] Dong Zhang, Shimin Li, et al. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Proc. of EMNLP Findings*, 2023.

[Zhang *et al.*, 2023b] Xin Zhang, Dong Zhang, et al. Speechtokenizer: Unified speech tokenizer for speech large language models, 2023.

[Zhu *et al.*, 2023] Zhihong Zhu, Xuxin Cheng, et al. Towards unified spoken language understanding decoding via label-aware compact linguistics representations. In *Proc. of ACL Findings*, 2023.