UniverSLU: Universal Spoken Language Understanding for Diverse Tasks with Natural Language Instructions

Siddhant Arora\textsuperscript{1}, Hayato Futami\textsuperscript{2}, Jee-woon Jung\textsuperscript{1}, Yifan Peng\textsuperscript{1}, Roshan Sharma\textsuperscript{1}\textsuperscript{*}, Yosuke Kashiwagi\textsuperscript{2}, Emiru Tsunoo\textsuperscript{2}, Karen Livescu\textsuperscript{3}, Shinji Watanabe\textsuperscript{1}

\textsuperscript{1} Carnegie Mellon University, USA
\textsuperscript{2} Sony Group Corporation, Japan
\textsuperscript{3} Toyota Technological Institute at Chicago
\{siddhana}@cs.cmu.edu

Abstract

Recent studies leverage large language models with multi-tasking capabilities, using natural language prompts to guide the model’s behavior and surpassing performance of task-specific models. Motivated by this, we ask: can we build a single model that jointly performs various spoken language understanding (SLU) tasks? We start by adapting a pre-trained automatic speech recognition model to additional tasks using single-token task specifiers. We enhance this approach through instruction tuning, i.e., finetuning by describing the task using natural language instructions followed by the list of label options. Our approach can generalize to new task descriptions for the seen tasks during inference, thereby enhancing its user-friendliness. We demonstrate the efficacy of our single multi-task learning model “UniverSLU” for 12 speech classification and sequence generation task types spanning 17 datasets and 9 languages. On most tasks, UniverSLU achieves competitive performance and often even surpasses task-specific models. Additionally, we assess the zero-shot capabilities, finding that the model generalizes to new datasets and languages for seen task types.

1 Introduction

Spoken language understanding (SLU) refers to the task of inferring semantic meaning or linguistic structure from spoken utterances. Like prior work (Chang et al., 2023), we use the term broadly to refer to understanding content (Warden, 2018), speaker characteristics (Hechmi et al., 2021), paralinguistics (Busso et al., 2008) and semantics (Bastianelli et al., 2020) directly from speech.\textsuperscript{1} In this work, we define a task as a specific combination of task type and dataset. A task type refers to the type of SLU operation being carried out, like intent classification (IC) or emotion recognition (ER). For instance, performing the same task type (IC) on the SNIPS (Saade et al., 2018) and FSC (Lugosch et al., 2019) datasets are different tasks.

SLU tasks are traditionally approached with “task-specific” models (Meeus et al., 2023), i.e. having a separate model for each task. On the other hand, the use of multi-task learning (MTL) approaches in natural language processing (NLP) (Chen et al., 2021; Collobert and Weston, 2008) and speech processing (Zhang et al., 2023; Radford et al., 2022) has garnered significant interest. Such approaches offer cost-efficiency by eliminating the need for separate models for each task. MTL has also been employed for SLU tasks (Arora et al., 2022a,b; Huang et al., 2022), primarily aiming to improve the performance of tasks with limited labeled data using an auxiliary automatic speech recognition (ASR) objective.

The rise of large language models (LLMs) (Radford et al., 2019) has sparked interest in MTL, where the selection of appropriate “prompts” (Liu et al., 2023) enables manipulation of the model’s behavior to perform specific tasks without additional task-specific parameters. LLMs fine-tuned with natural language instructions (Wei et al., 2022) describing the task show strong zero-shot performance. Building upon LLM’s success in solving various NLP tasks, there is growing interest in leveraging the “pre-train, prompt, and predict” (Liu et al., 2023) paradigm.

Prompt-based paradigms have also been explored for speech processing tasks (Gao et al., 2022; Peng et al., 2023a). SpeechPrompt (Chang et al., 2022, 2023) employs trainable task-specific prompt vectors with generative spoken language models (GSLMs) (Lakhotia et al., 2021). However, this model still lags behind task-specific baselines in most speech classification tasks and par-

\textsuperscript{1}Additionally, we also include audio understanding tasks. Table 7 in Appendix briefly outlines the SLU tasks addressed in this study.

\textsuperscript{*}Author is now at Google.
particularly struggles on sequence generation tasks. Recently, there has been work on instruction tuning for speech processing tasks (Lai et al., 2023). However, these approaches mainly concentrate on ASR and related tasks and are often not able to perform diverse SLU tasks.

Motivated by the potential of prompt-based MTL, there has been interest in building “universal” ASR models (Zhang et al., 2023; Li et al., 2021) that can perform ASR across many languages. We extend this work by building a “universal” SLU model that can match or even outperform state-of-the-art (SOTA) task-specific SLU models. We leverage the Whisper (Radford et al., 2022) model, which has been pre-trained on labeled data for ASR and speech translation (ST) tasks using only 2 task specifiers, namely ⟨transcribe⟩ and ⟨translate⟩ respectively. We first generalize the Whisper model to a wide range of SLU tasks by utilizing single token task specifiers, as shown in Fig. 1a, drawing inspiration from Whisper’s pre-training formulation. We show that unlike the most comparable prior multi-task approach, SpeechPrompt v2 (Chang et al., 2023), our MTL model can outperform or achieve competitive performance to task-specific baselines across many classification and sequence generation tasks.

However, the use of single-token task specifiers not only constrains our model’s ability to handle variations in input format, affecting its user-friendliness, but also prevents it from performing unseen tasks in a zero-shot manner. Hence, we experiment with utilizing natural language instruction (Wei et al., 2022), such as “Classify speech-based commands” that describe each task as a prompt (Fig. 1b). For classification tasks, we include a list of options within the instruction. Our approach generalizes to new paraphrases describing the same task, enhancing the model’s user-friendliness. We assess our model’s ability to handle unseen tasks, covering new datasets and languages for the known task types, as well as fully unseen task types. This ability is achieved by conducting inference using a task type description accompanied by the list of options specific to the given dataset as the instruction. Our model surpasses random and majority baselines in zero-shot inference on unseen datasets and languages, although it does not match supervised performance.

Our experiments encompass 10 speech classification and 2 sequence generation task types, covering 17 publicly available datasets based on 9 languages. Our proposed approach using natural language instruction advances the SOTA on 10 classification tasks and achieves competitive performance to task-specific baselines on sequence generation tasks. We will make our source code and models publicly available as part of the ESPnet-SLU toolkit2 (Arora

---

2https://github.com/espnet/espnet

---

Figure 1: Schematics of our discrete prompt-based MTL formulation. Our architecture comprises an encoder-decoder architecture pre-trained using OpenAI’s Whisper model, as detailed in Sec. 4. The figure illustrates the sequence of tokens used as prompts and predicted by the decoder during inference. We explore the use of single-token task specifiers (System | Task specifiers: SOP, SOT, NT, and TRANS) to a wide range of SLU tasks by utilizing single Whisper tokens that specify start-of-prev, start-of-transcript, no-timestamps, and transcribe, respectively.
et al., 2022a) for reproducibility. The key contributions of our work are:

- We introduce a novel approach for multi-task SLU that leverages human-interpretable instructions combined with a list of option labels as prompts. We demonstrate that our method can generalize to unseen paraphrases.
- Our proposed approach can generalize effectively to new datasets and languages for seen task types in a zero-shot manner.
- We build a single MTL model across 17 SLU datasets that performs competitively or outperforms task-specific models on most tasks.

2 Related Work

Multi-task Learning (MTL) (Chen et al., 2021; Caruana, 1997) concurrently trains a model to perform multiple tasks, allowing the model to learn generalizable features instead of task-specific ones. MTL has been extensively explored in various NLP settings (Wu et al., 2019; Zhao et al., 2019; Zhuang and Liu, 2019; Pfieffer et al., 2020), ranging from similar task types (de Souza et al., 2015) to the same task type across multiple domains and languages (Yang and Hospedales, 2015). MTL can surpass task-specific models in certain scenarios, especially for tasks with limited labeled data (Chan et al., 2021). Similar investigations have been conducted in speech processing, such as adding auxiliary tasks in ASR (Prabhavalkar et al., 2023; Kim et al., 2017; Pironkov et al., 2016; Toshniwal et al., 2017) and for other tasks (Anastasopoulos and Chiang, 2018; Zheng et al., 2022; Cai et al., 2021; Sigtia et al., 2020), primarily using an auxiliary ASR objective. However, these approaches covered very few SLU benchmarks, and we extend this line of research by MTL across 17 SLU datasets.

LLMs for speech processing: Recently, there has been significant interest in enhancing LLMs with speech processing and understanding capabilities (Gong et al., 2023b; Rubenstein et al., 2023; Fathullah et al., 2023a; Maiti et al., 2023). These approaches typically involve multi-tasking on a wide range of speech processing tasks, demonstrating robust performance across multiple benchmarks. LauraGPT (Wang et al., 2023) is one such approach that uses Whisper-style task specifiers with Qwen (Bai et al., 2023) to handle various speech processing and generation tasks. Some works have explored preserving LLM capabilities, such as understanding “instructions” to perform tasks in a zero-shot manner (Gong et al., 2023a; Tang et al., 2023; Chu et al., 2023; Fathullah et al., 2023b). LTU-AS (Gong et al., 2023a) concatenates text instruction with Whisper output and feeds them into Vicuna LLM (Chiang et al., 2023) to perform a range of speech and audio tasks. SALMONN (Tang et al., 2023) employs both speech and audio representation into Vicuna LLM for “instruction tuning” across various tasks.

However, while these efforts primarily concentrate on ASR (Yu et al., 2023; Lakomkin et al., 2023; Fathullah et al., 2023a) and ST (Chen et al., 2023), our approach focuses on a diverse array of SLU tasks (see Sec. 6.2 and A.4). Furthermore, our novel approach, involving the addition of the list of options in the instruction, not only improves performance but also enhances model generalization to new datasets and languages for the trained task types. Our results in Sec. 6.2 demonstrate our ability to outperform LLM-based approaches on the subset of SLU tasks they are trained on. Moreover, we distinguish ourselves as the first to exhibit instruction tuning capabilities with a decoder of speech foundation model to the best of our knowledge, that has the important advantage of using much fewer parameters and lower computation.

3 Problem Formulation

Our objective is to build a prompt-based MTL framework capable of performing various speech classification and sequence generation tasks. Our model takes a sequence of speech features, denoted $X$, as input for all downstream tasks. For a specific downstream task $r$, we aim to predict the corresponding $O^r$-length label sequence $Y^r = \{y^r_o \in L^r | o = 1, \ldots, O^r\}$, where $L^r$ represents the label set for that particular task. Table 7 in the Appendix provides a comprehensive description of the diverse SLU tasks covered in this work, including example label sequences $Y^r$ for each task. For classification tasks, all label sequences have a fixed length of $O^r = 1$. To achieve our goal, the prompting paradigm introduces a task-specific prompt $S^r$. Using maximum a posteriori (MAP) decision theory, the MTL model estimates $\hat{Y}^r$ as:

$$\hat{Y}^r = \arg\max_{Y^r} P(Y^r|X, S^r) \quad (1)$$

Prior studies like SpeechPrompt (Chang et al., 2022, 2023) have modelled the prompting formulation using GSLMs. They preprend the embedding
Table 1: Example prompts for different task types, datasets, and languages, with both task specifiers (Eq. 6) and natural language phrases (Eq. 7) as discussed in Sec. 4.1 and 4.2 respectively. Red bold values indicate task specifiers \( (S_{\text{task\_type}}, S_{\text{lang}}, S_{\text{data}}) \) in Eq. 2 and blue bold values indicate natural language instruction \( (I') \) in Eq. 4. SOP, SOT, NT and TRANS are standard Whisper tokens that specify start-of-prev, start-of-transcript, no-timestamps and transcribe respectively.

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Dataset</th>
<th>Language</th>
<th>Task Specifier ( (S') )</th>
<th>Generated Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Command</td>
<td>Google SC v1</td>
<td>En</td>
<td>SOT ((\text{en})) (\text{scr}) NT</td>
<td>SOP Classify speech-based commands. The options are (0.&quot;\text{down}&quot;), (\ldots) SOT ((\text{en})) TRANS NT</td>
</tr>
<tr>
<td>Recognition</td>
<td>Lithuanian SC</td>
<td>Lt</td>
<td>SOT ((\text{lt})) (\text{scr}) (\text{lt_scr}) NT</td>
<td>SOP Classify speech-based commands. The options are (0.&quot;\text{iki}&quot;), (\ldots) SOT ((\text{lt})) TRANS NT</td>
</tr>
<tr>
<td>Fake Speech Detection</td>
<td>ASVspoof</td>
<td>En</td>
<td>SOT ((\text{en})) (\text{fsd}) NT</td>
<td>SOP Tell apart speech synthesis and voice conversion from authentic speech. The options are (0.&quot;\text{bonafide}&quot;), 1.&quot;\text{spoof}&quot;. SOT ((\text{en})) TRANS NT</td>
</tr>
<tr>
<td>Accent Classification</td>
<td>AccentDB</td>
<td>En</td>
<td>SOT ((\text{en})) (\text{accent_rec}) (\text{accentdb}) NT</td>
<td>SOP Accent identification in speech. The options are (0.&quot;\text{telugu}&quot;), (\ldots) SOT ((\text{en})) TRANS NT</td>
</tr>
<tr>
<td>Audio Classification</td>
<td>ESC-50</td>
<td>(X)</td>
<td>SOT ((\text{audio})) (\text{auc}) ((\text{esc50})) NT</td>
<td>SOP Specify the type of environmental noise in the audio. The options are (0.&quot;\text{audio_class:17}&quot;), (\ldots) SOT ((\text{audio})) TRANS NT</td>
</tr>
</tbody>
</table>

(embedding layer \(\text{Emb}(\cdot)\)) of discrete tokens \(U\) with continuous task-specific prompt vectors \(P^r\). These vectors are learned for each task by maximizing the posterior distribution \(P(Y^r|\text{Emb}(U), P^r)\).

In our approach, instead of relying on a pre-trained model solely for generating discrete tokens for GSLM, we fine-tune an entire pre-trained speech foundation model. Furthermore, we use discrete prompts corresponding to a sequence of tokens. Starting from a natural extension of the pre-training formulation in speech foundation models like Whisper (Radford et al., 2022), we first investigate using single token “task specifiers” as prompts. During pre-training, Whisper utilizes only two task specifiers, namely \(\langle\text{transcribe}\rangle\) and \(\langle\text{translate}\rangle\), to facilitate MTL across ASR and ST tasks, respectively. We experiment with the use of a single token as specifier \(S_{\text{task\_type}}\) for each task type, \(S_{\text{lang}}\) for each language, and \(S_{\text{data}}\) for each dataset as shown in Fig. 1a to represent \(S'\) (Eq. 1):

\[ S' = (S_{\text{task\_type}}, S_{\text{lang}}, S_{\text{data}}) \]  

We then propose to incorporate human-interpretable natural phrases to enhance the attractiveness of our model for human interaction, as shown in the example of Fig. 1b. We use a probabilistic formulation to provide a new training objective, which handles various natural phrase prompts. Let \(I'\) be the natural language instruction describing downstream task \(r\). We can model the posterior distribution in Eq. 1 as:

\[ P(Y^r|X, S') = \sum_{I'} P(Y^r|X, S', I')P(I'|X, S'), \]  

where we assume conditional independence of \(Y^r|X, I'\) from \(S'\) and \(I'|S'\) from \(X\) to simplify Eq. 3. With the sampling approximation, Eq. 3 is rewritten as

\[ P(Y^r|X, S') = \sum_{I'} P(Y^r|X, I') \]  

The samples of natural language task descriptions \(I'\) are generated by an LLM that models the distribution \(P(I'|S')\), as discussed in Sec. 4.2. By using this training scheme, the model can accept various natural phrase prompts provided by the user as a way of describing \(S'\) (Eq. 2). Unlike prior studies (Chang et al., 2022, 2023), our approach allows for potential generalization to various paraphrases \(I'\) that specify the same task, thereby enhancing the user-friendliness and adaptability of our model for human interaction.

2757
4 Prompt-Based Multi-Tasking

To achieve the formulation described in Eq. 4, this work proposes a prompt-based MTL training format, as illustrated in Fig. 1b. The input speech \( X \) is passed through an encoder-decoder architecture. We prepend instruction \( I' \) to the preceding tokens generated by the decoder (shown as “instruction tuning” in Fig. 1b and described in further detail in Sec. 4.2). The likelihood of the entire label sequence i.e. \( P(Y^r | X, I') \) in Eq. 4, can be computed using task-specific decoder representations \( h^r_o \) as:

\[
P(Y^r | X, I') = \prod_{o=1}^{O^r} \text{Softmax}(\text{Out}(h^r_o))
\]

where \( \text{Out}(\cdot) \) denotes a linear layer that maps the decoder hidden representation \( h^r_o \) to vocabulary \( \mathcal{V} \). The vocabulary \( \mathcal{V} \) is the union of the label sets \( \mathcal{L}^r \) for all the tasks. In this work, both the encoder and decoder are pre-trained using a large speech foundation model (Radford et al., 2022).

4.1 Prompting with task specifier

As discussed in Sec. 3, we also experiment with training our MTL model using task specifiers as shown in Fig. 1a. The prompt is constructed based on Eq. 2 as:

\[
\text{Prompt} = \text{SOT}(\text{lang})\langle\text{task}\rangle(\text{dataset})\text{NT} \tag{6}
\]

where \( \langle\text{lang}\rangle \) is the language specifier \( S^\text{lang} \) (Eq. 2) from a set of language tags, \( \langle\text{task}\rangle \) is the task type specifier \( S^\text{task}_\text{type} \) (Eq. 2) from the token set \{scr, ic, ner, \ldots\} (denoting speech command recognition, intent classification, named entity recognition, \ldots respectively as further described in Sec. 5.1), \( \langle\text{dataset}\rangle \) is a dataset specifier \( S^\text{data} \) in the set \{google_scr, asvspoof, \ldots\} (See Sec. A.2), and \( \text{SOT} \) and \( \text{NT} \) represent the standard Whisper tokens specifying start-of-transcript and no-timestamps.

We provide example prompts using task specifiers for some sample tasks in Table 1. Every decoded text sequence must begin with a SOT token. The model is then prompted using \( S^\text{lang} \) as in the pre-training setup (Radford et al., 2022), and we also fine-tune the model to predict \( S^\text{lang} \) for language identification. In addition to the 999 languages already considered during Whisper pre-training, we add another language tag, \( \langle\text{audio}\rangle \), to enable audio-related tasks such as audio classification. If there is no speech in the audio segment and the model is not prompted using the \( \langle\text{audio}\rangle \) tag, the model has been pre-trained to predict “no-speech” and finish decoding. We fine-tune the model for voice activity detection similar to this pre-training setup. The next token is the task type specifier \( S^\text{task}_\text{type} \), and we add a new token to the vocabulary of the pre-trained model for each SLU task type. We further add a new token \( S^\text{data} \) for each dataset since the label set \( \mathcal{L}^r \) (see Sec. 3) can differ between datasets. For example, in the case of IC, the label (deactivate_lights_none) in (Lugosch et al., 2019) indicates the same intent as (switch-lightoff) in (Saade et al., 2018). Finally, to simplify fine-tuning, we specify that the model should not predict time stamps. The model then uses this prompt to predict the class or label sequence for the given SLU task.

4.2 Prompting with natural language instructions

During training and inference, we query ChatGPT, the gpt-3.5-turbo version, to sample natural language paraphrases \( I' \) (in Eq. 4) for the given task specifier \( S^r \). The authors then manually examine these paraphrases and select those believed to be representative of what a user might provide as a task description. We incorporate instructions \( I' \) as previous context text as shown in Fig. 1b. The prompt for the model trained with natural language instructions is:

\[
\text{Prompt} = \text{SOP}(\text{instruction})\text{SOT}(\text{lang})\text{TRANS NT} \tag{7}
\]

where \( \langle\text{instruction}\rangle \) is a natural language instruction \( I' \), \( \langle\text{lang}\rangle \) is the language specifier as in Eq. 6, and the remaining are required standard whisper tokens (SOP and TRANS specify start-of-prev and transcribe). For classification tasks, we augment the instruction \( I' \) with a list of possible options and instruct the model to predict the option number, such as: “[SOP Classify speech-based commands. The options are 0."go", 1."down", 2."silence", \ldots SOT (en) TRANS NT 1]”. As in Sec. 4.1, we fine-tune the model on the pre-training task types, such as language identification and voice activity.

To achieve strong performance, we believe that it will be useful to leverage a strong speech foundation model like Whisper that has been trained on large amounts of ASR data. Although in this work, we only experiment with fine-tuning Whisper model, our approach can generalize to incorporating any speech foundation model (Peng et al., 2023b).

---

3To achieve strong performance, we believe that it will be useful to leverage a strong speech foundation model like Whisper that has been trained on large amounts of ASR data. Although in this work, we only experiment with fine-tuning Whisper model, our approach can generalize to incorporating any speech foundation model (Peng et al., 2023b).

4https://chat.openai.com/https://platform.openai.com/docs/models/gpt-3-5-turbo

5More details in Sec. A.5.
Table 2: Accuracy (%) of the MTL model trained using natural language phrases as prompts on FSC, IEMOCAP and Google Speech Commands dataset.

We also provide example prompts using natural instructions for some tasks in Table 1. Every decoded text sequence must begin with an SOP token. This is followed by natural language instruction followed by a list of possible options. We mask out the training loss over the instruction text and train the model to predict all other tokens. The next token is SOT followed by a language specifier $S_{lang}$ and TRANS as in its pre-training setup (Radford et al., 2022). Finally, the model predicts the option number of SLU class or label sequence for the given SLU task.

We further perform an ablation study where instead of incorporating instruction as previous context text, we replace the “TRANS” token in Eq. 7 with the instruction phrase, such that

$$\text{Prompt} = \text{SOT} \langle \text{lang} \rangle \langle \text{instruction} \rangle \text{NT} \quad (8)$$

An example model prompt sequence would be “[SOT (en) Classify speech-based commands. The options are 0."go", 1."down", 2."silence", ..., NT]”. We evaluate the efficacy of various ways of incorporating instructions in Sec. 6.1.

5 Experiments

5.1 Datasets and Tasks

We conduct experiments on 18 tasks over 17 publicly available SLU datasets spanning 9 languages and 12 task types, including 10 classification task types (IC, speech command recognition (SCR), language identification (LID), fake speech detection (FSD), emotion recognition (ER), accent classification (AcC), sarcasm detection (SD), gender identification (GID), voice activity detection (VAD), and audio classification (AuC)) as well as 2 sequence generation task types (named entity recognition (NER) and semantic parsing (SP)). We use additional evaluation datasets for testing zero-shot performance in a new dataset and language, as well as for an unseen task type, dialogue act classification (DAC). See Sec. A.2 in the Appendix for task type and dataset descriptions. We will make all our data preparation publicly available as part of the ESPnet-SLU toolkit (Arora et al., 2022a) for reproducibility.

5.2 Baselines

We compare our approach with SOTA task-specific baselines from prior work (Arora et al., 2022a; Vygon and Mikhailovskiy, 2021; Tian and Gorinski, 2020; Yen et al., 2021; Bermuth et al., 2022; Shor et al., 2022; Ahamad et al., 2020; Castro et al., 2019; Ray et al., 2022; Hechmi et al., 2021; Jia et al., 2021; Chen et al., 2022) by reporting the performance from the original papers (“SOTA” in Table 3). We also report the performance achieved by SpeechPrompt v2 (Chang et al., 2023) (“SpeechPrompt v2” in Table 3) to quantify the impact of our proposed modifications to prior MTL approaches based on speech prompting, as discussed in Sec. 3.

5.3 Experimental Setups

All models are built using the ESPnet-SLU toolkit (Arora et al., 2022a). We adopt the evaluation metrics used in prior work.

As a proof of concept, we initially train an MTL model (refer to Sec. 4.2) employing natural language instructions $I'$ (in Eq. 4) as prompts for three tasks using the FSC (IC), IEMOCAP (ER), and Google SC v1 (SCR) datasets. We explore various approaches for incorporating natural language phrases, as discussed in Sec. 4.2. When the list of options is included in the instruction, we introduce a random shuffling of the order of options during training. Our preliminary investigations (not...
Table 3: Performance of UniverSLU employing both task specifiers (Eq. 6) and natural language instructions (Eq. 7) on both classification and sequence generation tasks, compared to both task-specific SOTA performance (including Whisper task-specific baselines in Tab. 6 in Appendix) and SpeechPrompt v2 (Chang et al., 2023).

Subsequently, we perform multitasking on the same 14 speech classification datasets as SpeechPrompt v2 (Chang et al., 2023), utilizing both single-token task specifiers (Eq. 6) and natural language instructions (Eq. 7). We refer to these models as UniverSLU-14 Task Specifier and UniverSLU-14 Natural Phrase, respectively. We then expand the training tasks to include sequence generation task types such as NER and SP. This enhanced MTL model, trained on a total of 17 datasets, is denoted UniverSLU-17. To enhance performance on low-resource and para-linguistic datasets, we upsample (see Sec. A.3) the Lithuanian and Arabic speech command datasets, ESC-50, SNIPS, sarcasm detection, and emotion recognition datasets. All model, training, and inference parameters are selected based on validation performance. Further details are provided in Sec. A.3.

6 Results and Discussion

6.1 Using Natural Language Instructions

Table 2 shows the performance of our MTL model trained with natural language phrases instead of task specifiers as discrete prompts. As detailed in Sec. 4.2, we experiment with various methods of incorporating instructions. We initially report results on seen prompts—natural language instructions encountered during training. In the scenario where the “TRANS” token is replaced with natural instructions (Eq. 8), including the list of options in the instruction proves beneficial for performance improvement. Incorporating the natural language instruction as a previous text token (Eq. 7) yields the best performance, closely resembling that achieved using task specifiers (Eq. 6).

To assess the generalizability of our approach, we evaluate the model on natural language phrases not included in training. We average the results obtained from 5 unseen natural language phrases. Our findings suggest that the model generalizes effectively to unseen natural language phrases, exhibiting performance comparable to that with seen prompts. Additionally, we test the model
on two different orders of options in the instruction and report the average performance, noting stable results regardless of option ordering.

This investigation underscores the potential of using natural language phrases as prompts. Results support our decision to depart from continuous prompts used in prior work (Chang et al., 2023), opting for natural language instructions to create a more user-friendly model capable of handling diverse paraphrases.

6.2 More tasks: UniverSLU

We present the performance of our UniverSLU Task Specifier, UniverSLU Natural Phrase, Speech-Prompt v2, and the SOTA models on these benchmarks in Table 3. Our results demonstrate that the UniverSLU-14 Task Specifier outperforms prior speech prompting methods on 11 out of 14 tasks and improves the SOTA on 9 out of 14 tasks. UniverSLU-14 Natural Phrase achieves performance comparable to UniverSLU-14 Task Specifier across most tasks, with the added benefit of generalizing to new paraphrases of instructions, as discussed in Sec. 6.1. Our UniverSLU-17 Task Specifier model, trained on a combination of speech classification and sequence generation tasks, demonstrates comparable or superior performance compared to the SOTA on 11 tasks. It also achieves competitive performance compared to task-specific baselines on sequence generation tasks, in contrast to previous approaches (Chang et al., 2022) that face challenges in handling sequence generation tasks. We observe that our upsampling approach proves to be highly effective, significantly improving the performance on low-resource datasets such as Arabic speech command recognition. Finally, our UniverSLU-17 Natural Phrase model also outperforms the SOTA on 10 tasks and again achieves comparable performance to the UniverSLU-17 Task Specifier model on most tasks. These results demonstrate our model’s capability to perform very well, exceeding SOTA results even on tasks with limited fine-tuning data, such as AcC, which only involves 20 hours of fine-tuning data. We hypothesize that this strong performance can be attributed to both the model’s pre-training as well as our multi-task learning setup.

However, UniverSLU-14 Natural Phrase and UniverSLU-17 Natural Phrase fail to perform better than the random baseline on the audio classification (AuC) task, presumably because Whisper has not been trained on non-speech tasks. Previous work on LTU-AS (Gong et al., 2023a) has shown the capability of Whisper to follow instructions for performing audio-related tasks. However, it’s important to note that LTU-AS was fine-tuned on a substantial dataset comprising 845,000 audio clips, while our fine-tuning dataset, ESC-50, is much smaller, consisting of only 1600 training clips. Based on these results, we hypothesize that pre-training plays a crucial role in effective instruction tuning, particularly when dealing with a small fine-tuning dataset.

Comparison to recent LLM-based approaches: As outlined in Sec. 2, several recent works aim to enhance LLMs with speech processing capabilities. Although the majority of these works do not specifically target SLU tasks, some (Gong et al., 2023a; Tang et al., 2023; Wang et al., 2023) are fine-tuned on a subset of the SLU tasks considered in this work, and we show their comparison with UniverSLU in Table 4.6 UniverSLU-17 outperforms all of these approaches on these tasks, except for AuC where LTU-AS excels due to fine-tuning on a larger audio dataset. We also use SALMONN’s publicly available checkpoint (Tang et al., 2023) and report its zero-shot performance on other UniverSLU tasks in Sec. A.4, where it is observed to lag behind UniverSLU models. In conclusion, our MTL approach exhibits substantial advancements over task-specific models as well as previous MTL approaches, including models that deploy LLMs.

6.3 Zero Shot Performance

To assess the generalization capabilities of our multi-task model, we initially evaluate its performance on unseen tasks that assess its ability to handle seen task types on new datasets and languages. For this analysis, we employ the SNIPS and KSU_Emotions7 (Arabic Emotion Recognition) datasets. During inference, similarly to the training process, we include the possible label options in the instruction prompts. We also show the performance of random and majority baselines, obtained by randomly (uniformly) selecting answers and selecting the most common answer, respectively. Our findings in Table 5 indicate that UniverSLU-14 Natural Phrase consistently outperform...
Table 4: Results comparing UniverSLU with recently proposed LLMs for speech processing.

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Metric</th>
<th>Dataset</th>
<th>LauraGPT (2023)</th>
<th>LTU-AS (2023a)</th>
<th>SALMONN (2023)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>Acc ↑</td>
<td>SLURP</td>
<td>87.87 x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ER</td>
<td>Acc ↑</td>
<td>IEMOCAP</td>
<td>65.2 x</td>
<td>69.0 x</td>
<td></td>
</tr>
<tr>
<td>GID</td>
<td>F1 ↑</td>
<td>VoxCeleb1</td>
<td>90.8 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AuC</td>
<td>Acc ↑</td>
<td>ESC-50</td>
<td>80.8 x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NER</td>
<td>SLU F1 ↑</td>
<td>SLURP</td>
<td>73.45 x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Performance of MTL models trained using natural language phrases as prompts on an unseen dataset, language and task type in a zero-shot manner. We include random and majority baselines for comparison and report the performance of supervised models (Table 6 for SNIPS, 2021 for KSU Emotions, 2023 for DAC).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Language</th>
<th>Task Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Snips</td>
<td>KSU</td>
<td>Daily</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>Emotions</td>
<td>Talks</td>
</tr>
<tr>
<td></td>
<td>F1 ↑</td>
<td>ER</td>
<td>DAC</td>
</tr>
<tr>
<td>SOTA (supervised)</td>
<td>96.3</td>
<td>86.8</td>
<td>61.1</td>
</tr>
<tr>
<td>Random</td>
<td>19.7</td>
<td>25.0</td>
<td>25.0</td>
</tr>
<tr>
<td>Majority</td>
<td>6.4</td>
<td>28.5</td>
<td>45.5</td>
</tr>
<tr>
<td>UniverSLU-14 (Natural Phrase)</td>
<td>44.6</td>
<td>38.8</td>
<td>17.6</td>
</tr>
<tr>
<td>UniverSLU-17 (Natural Phrase)</td>
<td>x</td>
<td>41.7</td>
<td>19.3</td>
</tr>
</tbody>
</table>

7 Conclusion

We introduce an “instruction tuning” approach that utilizes human-interpretable natural language instructions, combined with a list of option labels as prompts, to fine-tune a speech foundation model on SLU benchmarks. Our experiments demonstrate the model’s ability to understand diverse expressions as instructions during inference. Leveraging this approach, we train a single MTL model named UniverSLU, capable of performing 12 SLU task types, encompassing both classification and sequence generation tasks. UniverSLU not only provides cost efficiency but also achieves competitive performance compared to task-specific baselines, outperforming SOTA models on several SLU benchmarks. Additionally, we find that the model has some generalization ability on new datasets and languages for seen task types.

However, a notable performance gap exists when compared to supervised topline models, suggesting substantial room for improvement.

Furthermore, we explore the model’s capability to perform a completely new task type—Dialogue Act Classification (DAC)—that was not encountered during training. We observe in Table 5 that Univer-SLU models currently fail to generalize to this unseen task type.

We also calculate the following rate proposed in (Tang et al., 2023), which measures the percentage of times the instruction-tuned model successfully follows the instructions. We observe that it is 100% for UniverSLU on all tasks, indicating that it always predicts one of the label options. This preliminary investigation provides insight into the model’s generalization ability to unseen tasks in a zero-shot manner.

While our current approach faces challenges in generalizing to new task types, it demonstrates strong generalization to new paraphrases describing known tasks (Sec. 6.1) and achieves non-trivial performance on new datasets and languages with entirely different label sets in a zero-shot scenario. Thus, we believe our approach possesses some instruction-following capabilities. Prior work, exemplified by instruction tuning models like FLAN (Wei et al., 2022), has shown that generalization improves with an increase in the size of the instruction tuning dataset. We plan to explore this further in future work, along with integrating an LLM-based decoder and implementing few-shot inference techniques.
Limitations

Our instruction tuning approach relies on incorporating the list of option labels in the prompt. A potential limitation arises if the option list becomes too long, as it may surpass the token limit of the decoder. We plan to delve deeper into this aspect in future work. Additionally, Table 5 illustrates that our Univer-SLU model currently encounters challenges in generalizing to new, unseen task types. This may be a limitation of Whisper-style models and could perhaps be addressed with the integration of an LLM-based decoder. It may also be that zero-shot generalization is too challenging for certain unseen tasks, requiring at least a few-shot inference.

Broader Impact and Ethics

With our UniverSLU model, we strive to build a single MTL model capable of doing multiple SLU tasks, which saves compute as we do not need separate models for each task. Our approach effectively uses speech foundation models and hence can perform well without requiring large amounts of task-specific labeled data for instruction tuning. Furthermore, we show that our model can perform well in a zero-shot manner for new datasets and languages, which is very useful to ensure speech technologies are available to a wide range of users across various domains. Additionally, we adhere to the ACL Ethics Policy. Our experiments are based on open-source datasets that are widely used in prior work and to our knowledge, there is no violation of privacy in these datasets.

Acknowledgement

Experiments of this work used the Bridges2 system at PSC and Delta system at NCSA through allocations CIS210014 and IRI120008P from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, supported by National Science Foundation grants #2138259, #tel:2138286, #tel:2138307, #tel:2137603, and #tel:2138296.

References


A Appendix

A.1 Comparison to Task-Specific Baseline

Since most of these baselines do not incorporate Whisper, we train “task-specific” baselines (see Table 6) by fine-tuning Whisper on selected datasets, following the approach in (Arora et al., 2023), to better understand the efficacy of training multiple tasks simultaneously. Table 6 presents the results of task-specific baselines and the MTL model (UniverSLU-3) trained with task specifiers (Sec. 4.1) on selected speech classification and sequence generation tasks, namely IC, NER, and SP tasks. We start by comparing the task-specific baseline, which involves fine-tuning the Whisper model, with end-to-end SLU models in ESPNetSLU (Arora et al., 2022a) (rows 1 and 2). Results show that fine-tuning Whisper is beneficial for SLU tasks and yields strong baselines.

MTL framework UniverSLU-3 that incorporates task specifiers $S_{\text{lang}}$, $S_{\text{task_type}}$, $S_{\text{data}}$ (Eq. 2 and 6) achieves competitive performance to the task-specific baselines. We additionally investigate training the MTL model using only $S_{\text{task_type}}$ (w/ Task Type Specifier in Table 3) (e.g., “$\langle\text{en}\rangle$ $\langle\text{scr}\rangle$ NT down”) and both the $S_{\text{task_type}}$ and $S_{\text{data}}$ prompts (w/ Data Specifier in Table 3) (e.g., “$\langle\text{en}\rangle$ $\langle\text{scr}\rangle$ $\langle\text{google}\_\text{scr}\rangle$ NT down” same as Eq. 6). We observe similar performance with and without using the dataset specifier ($S_{\text{data}}$ in Eq. 6) as an additional prompt. Based on these findings, we conclude that the multi-tasking model achieves comparable performance to task-specific baselines for both classification and sequence generation tasks. We can also significantly reduce the number of trainable parameters since we now have a single MTL model instead of multiple models, making it cost-efficient.

A.2 Data and Task Type

Intent Classification (IC) aims to recognize the intent from a user’s command in order to take appropriate action. We leverage 2 publicly available corpora, i.e., SNIPS and FSC. The SNIPS corpus (Saade et al., 2018) (CC-BY-SA license) consists of 1,660 conversations with in-house SA license assistant. For our experiments, we utilize a random split following the approach outlined in (Agrawal et al., 2020). The Fluent Speech Commands
<table>
<thead>
<tr>
<th>Model</th>
<th>SNIPS</th>
<th>SLURP</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IC F1↑</td>
<td>IC Acc↑</td>
<td>NER SLU F1↑</td>
</tr>
<tr>
<td>ESPnet-SLU</td>
<td>91.7</td>
<td>87.4</td>
<td>77.7</td>
</tr>
<tr>
<td>Whisper task-specific</td>
<td><strong>96.3</strong></td>
<td>89.6</td>
<td>79.7</td>
</tr>
</tbody>
</table>

**Task-specific baselines (models tuned for each task)**

<table>
<thead>
<tr>
<th>UniverSLU-3 (single model covers all tasks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/ Task TypeSpecifier $S_{\text{task_type}}$</td>
</tr>
<tr>
<td>w/ DatasetSpecifier $S_{\text{data}}$</td>
</tr>
</tbody>
</table>

Table 6: Results showing performance of prompt based MTL model UniverSLU-3 trained using task specifiers on selected datasets.

Named Entity Recognition (NER) refers to the task of labeling the spoken tokens in a user’s command with associated entities and recognizing mentions of these entities. It is a sequence generation task. The SLURP corpus (Bastianelli et al., 2020) (CC BY-NC 4.0 license) is an open source NER corpus consisting of 57.4 hours of single-turn conversations with a home assistant, making it relevant to commercial SLU applications. To augment our training set, we incorporate synthetic data, following prior work (Arora et al., 2022a).

Semantic Parsing (SP) is a sequence generation task that focuses on converting a spoken utterance into a semantic parse sequence, enabling the voice assistant to execute tasks effectively. The STOP dataset (Tomasello et al., 2022) (STOP Dataset License Agreement9) is the largest semantic parsing dataset. It comprises over 200,000 audio files from more than 800 speakers across 8 different domains.

Speech Command Recognition (SCR) refers to the task of detecting when a keyword has been spoken in an utterance. The Google Speech Commands dataset (Warden, 2018) (CC BY 4.0 license) consists of nearly 100,000 utterances of spoken keywords. The Grabo dataset (Renkens et al., 2014) (License for use in academic research or commercial product) comprises English and Dutch spoken commands given to a robot for navigation. The dataset includes recordings from 11 speakers issuing 36 different commands, each repeated 15 times. The Lithuanian SC dataset (Kolesau and Šešok, 2020) (license for use in academic research) consists of recordings of 28 individuals uttering 20 Lithuanian words using a mobile phone. The Arabic SC database (Ghandoura et al., 2021) (CC BY 4.0 license) consists of 1,600 speech recordings from 40 speakers in Arabic, covering 6 control words and the digits 0 through 9.

Language Identification (LID) aims to recognize the language of a spoken utterance. VoxForge (MacLean, 2018) (GPL license) is an open-source dataset established to collect transcribed multilingual speech using open-source engines. We utilize this dataset to train a language classification system for several European languages, including English, Spanish, French, German, Russian, and Italian. Consistent with previous work (Chang et al., 2023), we use random split with 1200 audio samples in the training, 300 in the validation, and 300 in the test set for each language.

Fake Speech Detection (FSD) aims to detect fake or spoofed speech generated by Text-to-Speech and Voice Conversion systems from real speech in order to prevent adversarial attacks. ASVspoof (Nautsch et al., 2021) (ODC-By license) is a benchmark designed for building fake speech detection systems. In our experiments, we utilize the Logical Access (LA) portion of the ASVspoof 2019 dataset, which includes both bonafide and spoofed speech. The dataset consists of a total of 121,461 utterances with real speech collected from 107 English speakers.

Emotion Recognition (ER) aims to understand the emotion or sentiment conveyed in an utterance. IEMOCAP (Busso et al., 2008)10 is a dataset con-

9https://github.com/facebookresearch/spoken_task_oriented_parsing/blob/main/LICENSE

10License:https://sail.usc.edu/iemocap/Data_Release_Form_IEMOCAP.pdf

2768
<table>
<thead>
<tr>
<th>Tasks Type</th>
<th>Dataset</th>
<th>Language</th>
<th>Output Type</th>
<th>Example Outputs ($Y^*$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCR</td>
<td>Google SC v1</td>
<td>En</td>
<td>speech command class</td>
<td>yes, down, no, …</td>
</tr>
<tr>
<td></td>
<td>Grabo SC</td>
<td>Du</td>
<td>speech command class</td>
<td>turn_rel throttle=&quot;slow&quot; angle=&quot;east&quot;, turn_rel throttle=&quot;slow&quot; angle=&quot;west&quot;, turn_rel throttle=&quot;fast&quot; angle=&quot;south&quot;, …</td>
</tr>
<tr>
<td>Lithuanian SC</td>
<td>Lt</td>
<td>intent class</td>
<td>ačiū, iki, išjunk, …</td>
<td></td>
</tr>
<tr>
<td>Arabic SC</td>
<td>Ar</td>
<td>increase heat washroom, deactivate lights, deactivate lights bedroom, …</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fluent SC</td>
<td>En</td>
<td>increase brightness, set light color, set light brightness, …</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>SNIPS</td>
<td>En</td>
<td>music likeness, music settings, play music, …</td>
<td></td>
</tr>
<tr>
<td>SLURP</td>
<td>En</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LID</td>
<td>Voxforge</td>
<td>En, Es, Fr, De, Ru, It</td>
<td>language class</td>
<td>En, Es, Fr, De, Ru, It</td>
</tr>
<tr>
<td>FSD</td>
<td>ASVspoof</td>
<td>En</td>
<td>synthetic speech class</td>
<td>spoof, bonafide</td>
</tr>
<tr>
<td>ER</td>
<td>IEMOCAP</td>
<td>En</td>
<td>emotion class</td>
<td>angry, neutral, sad, happy, other</td>
</tr>
<tr>
<td>AcC</td>
<td>AccentDB</td>
<td>En</td>
<td>speaker accent class</td>
<td>american, australian, bangla, …</td>
</tr>
<tr>
<td>SD</td>
<td>MUSTARD</td>
<td>En</td>
<td>sarcasm class</td>
<td>sarcasm, not sarcasm</td>
</tr>
<tr>
<td>+ MUSTARD++</td>
<td>En</td>
<td>sarcasm class</td>
<td>sarcasm, not sarcasm</td>
<td></td>
</tr>
<tr>
<td>GID</td>
<td>VoxCeleb1</td>
<td>En</td>
<td>speaker gender class</td>
<td>male, female</td>
</tr>
<tr>
<td>VAD</td>
<td>Google SC v2 + Freesound</td>
<td>En</td>
<td>voice activity class</td>
<td>speech, no speech</td>
</tr>
<tr>
<td>AuC</td>
<td>ESC-50</td>
<td>X</td>
<td>audio class</td>
<td>dog, pouring water, breathing, …</td>
</tr>
<tr>
<td>NER</td>
<td>SLURP</td>
<td>En</td>
<td>sequence of (entity phrase, entity tag) pairs</td>
<td>sl:date FILL tuesday SEP sl:event_name FILL meeting, …</td>
</tr>
<tr>
<td>SP</td>
<td>STOP</td>
<td>En</td>
<td>semantic parse sequence</td>
<td>[in:create_alarm [sl:date_time_recurring for six am every day ]], …</td>
</tr>
</tbody>
</table>

Table 7: Overview of the fine-tuning datasets and task types in the UniverSLU model. Each row indicates a “separate” task. Note that a task refers to a specific combination of task type and dataset. For instance, SNIPS and FSC have the same task type but are different tasks due to differences in their label set $\mathcal{L}$. We also show the example outputs $Y^*$ (seperated by “,”) for each task.

Sisting of approximately 12 hours of speech data with four emotion classes: neutral, happy, sad, and angry. We use Sessions 1-4 as training and Session 5 as a test set.

Accent Classification (AcC) aims to classify speech from speakers with different accents within the same language. Accent DB (Ahamad et al., 2020) (CC BY-NC 4.0 License) dataset contains approximately 20 hours of English speech data, encompassing 4 non-native accents, 1 metropolitan Indian accent, and 4 native accents.

Sarcasm Detection (SD) aims to identify whether a given utterance is sarcastic or not, based on verbal and non-verbal cues. The MUSTARD (Castro et al., 2019) (MIT License) dataset is a multimodal sarcasm detection dataset consisting of 1,991 utterances compiled from popular TV shows and annotated with sarcasm labels. The MUSTARD++ (Ray et al., 2022) (Publicly available) dataset is an enhanced version of MUSTARD dataset. It includes original MUSTARD dataset and features enhanced annotations. Additionally, the dataset has been doubled in size, comprising 2,695 utterances.

Gender Identification (GID) aims to determine the gender of a speaker based on their speech. For gender identification, we utilize the VoxCeleb1 (Hechmi et al., 2021) (Creative Commons Attribution-ShareAlike 4.0 International License) dataset. It contains approximately 148,000 training and 5,000 test utterances. The gender labels are extracted from the speaker metadata.

Voice Activity Detection (VAD) systems aim to determine whether a given audio contains human speech or background noise. We leverage the Google Speech Commands v2 (Warden, 2018)
dataset as speech data and Freesound (Fonseca et al., 2017) (GNU Affero General Public License v3.0) dataset as background noise to train these systems.

Audio Classification (AuC) is a multi-class single-label classification task that aims to correctly classify the audio present in the environment. ESC-50 (Fonseca et al., 2017) (Creative Commons Attribution-NonCommercial license) is a labeled set of 2000 environment recordings with 50 audio classes.

Zero-shot evaluation datasets: To assess the performance of our MTL model in an unseen language, we employ the Arabic Emotion Recognition dataset, KSU_emotions (Meftah et al., 2021) (License: LDC User Agreement for Non-Members). We utilize a subset of this dataset that mirrors the emotions present in the IEMOCAP dataset—namely, neutral, happy, sad, and angry—comprising 2301 utterances. Additionally, we extend our evaluation to a new, unseen task type: Dialogue Act Classification (DAC) which aims to identify the function of a given speech utterance in a dialog. For this task, we utilize the Daily Talks (Lee et al., 2023) (CC-BY-SA 4.0 license) spoken dialogue dataset, annotated with four dialogue act categories—question, inform, directive, and commissive—comprising 4758 utterances, following the setup outlined in (Huang et al., 2023).

Finally, it is important to note that all the datasets are publicly available.

A.3 Experimental Setups

Our models are implemented in PyTorch (Paszke et al., 2019), and experiments are conducted with ESPnet-SLU toolkit (Arora et al., 2022a).

Task-specific baselines: We fine-tune the Whisper medium model with more frequent saving of checkpoints (i.e., after every 1,000 iterations instead of after each epoch) following the approach in (Arora et al., 2023). We train for 100 epochs with a learning rate of 1e-5 and 1500 warmup steps. Early stopping is performed if the validation accuracy saturates. Additionally, we add special tokens to the vocabulary of the Whisper model for SLU labels, such as slot and intent tags. Similar to prior work (Arora et al., 2022a), we use an auxiliary ASR objective to train our SLU models when we have access to transcripts.

MTL models using task specifiers: Similar training setup is followed for training our multitask-learning model with task specifiers $S_{\text{lang}}$, $S_{\text{task_type}}$, $S_{\text{data}}$ (Eq. 2). In addition to SLU tags, we also add $S_{\text{task_type}}$, $S_{\text{data}}$, and “⟨audio⟩” tokens to the Whisper vocabulary, which are used as prompts for the model, as discussed in Sec. 4.1. For UniverSLU-17 Task Specifier we increase the number of iterations per epoch to 3,000 and train for 100 epochs, since training data becomes larger due to the combination.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whisper model</td>
<td>medium</td>
</tr>
<tr>
<td>Dropout Rate</td>
<td>0</td>
</tr>
<tr>
<td>LR schedule</td>
<td>inv. sqrt.</td>
</tr>
<tr>
<td>Max learning rate</td>
<td>[1e-5, 2e-5, 5e-5]</td>
</tr>
<tr>
<td>Warmup steps</td>
<td>[5000, 15000, 25000]</td>
</tr>
<tr>
<td>Number of steps</td>
<td>[25, 50, 100]</td>
</tr>
<tr>
<td>Adam eps</td>
<td>1e-6</td>
</tr>
<tr>
<td>Adam betas</td>
<td>(0.9, 0.99)</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.01</td>
</tr>
<tr>
<td>Beam Size</td>
<td>[1, 5, 20]</td>
</tr>
<tr>
<td>Length Penalty</td>
<td>[0,0.1,0.2]</td>
</tr>
<tr>
<td>Maxlen ratio</td>
<td>[1.0,1.2]</td>
</tr>
</tbody>
</table>

Table 8: Model Training and Inference Search for UniverSLU Models.
Task Metric | Dataset | Language | #Class | SALMONN Natural Phrase | UniverSLU-17 Task Specifier | Natural Phrase
---|---|---|---|---|---|---
 SCR | Acc ↑ | Google SC v1 | En | 12 | 64.6 | 99.1 | 99.2
(Lithuanian SC | Lt | 15 | 18.2 | 98.9 | 98.9
 Arabic SC | Ar | 16 | ✗ | 100.0 | 100.0
 IC | F1 ↑ | SNIPS | En | 6 | 10.9 | 92.3 | 63.5
 LID | Acc ↑ | Voxforge | En, Es, Fr, De, Ru, It | 30.6 | 99.9 | 99.9
 FSD | EER ↓ | ASVspoof | En | 2 | 19.9 | 2.0 | 0.9
 Acc | Acc ↑ | AccentDB | En | 9 | 8.7 | 99.9 | 100.0
 SD | F1 ↑ | MUSTARD | En | 2 | 37.2 | 73.5 | 72.9
 GID | F1 ↑ | VoxCeleb1 | En | 2 | 98.6 | 99.9 | 99.9

Table 9: Performance of SALMONN using publicly available checkpoint on unseen SLU tasks, that were used for fine-tuning UniverSLU models.

Task Specifier, we only add “⟨audio⟩” token to the Whisper vocabulary, which is used to fine-tune on audio classification tasks.

The combined parameter size for all our MTL and task-specific models is approximately 762.3M. We initialize the UniverSLU-17 Task Specifier and UniverSLU-17 Natural Phrase models with weights obtained from the MTL model Univer-14 Task Specifier and UniverSLU-14 Natural Phrase, respectively. We additionally perform upsampling on low-resource datasets: the Arabic speech commands dataset by a factor of 6, the Lithuanian speech commands dataset, ESC-50, SNIPS, and sarcasm detection datasets by a factor of 3, and the emotion recognition dataset by a factor of 2. We chose these upsampling ratios roughly in inverse proportion to the number of utterances in the datasets. In future work, we will further investigate various data sampling approaches and their impact on model performance.

We apply SpecAugment (Park et al., 2019) and use dropout (Srivastava et al., 2014) and label smoothing (Müller et al., 2019) techniques. The models are trained using 4 NVIDIA A40 (40GB) GPUs. All model, training, and inference parameters are selected based on validation performance. Table 8 shows training and inference hyperparameters for our hyperparameter search. Full Details about models, configuration files, and data preparation will be made publicly available as part of the ESPnet-SLU toolkit: https://github.com/espnet/espnet.

A.4 Detailed Comparison with LLMs

In the evaluation process, we utilized the publicly available SALMONN (Tang et al., 2023) 13B checkpoint and applied it to unseen UniverSLU tasks in a zero shot manner, employing instructions similar to those used in UniverSLU as prompts. The results, however, are presented for a subset of tasks due to various challenges encountered during the evaluation process. Difficulties were faced in obtaining results for sequence prediction tasks (Tomasello et al., 2022), primarily because the output did not match the expected label format. GPU memory constraints were encountered for classification tasks (Lugosch et al., 2019; Renkens et al., 2014) with more than 20 labels since the option labels were included in the instruction. Voice activity detection task with very long audios posed a challenge, leading to out of memory error on a 40 GB GPU.

Table 9 shows that SALMONN 13B did not perform well on most tasks. However, it’s crucial to recognize that this comparison with Univer-SLU is not entirely fair, as SALMONN is operating in a zero-shot manner. Notably, SALMONN exhibited even poorer performance than random for accent classification, intent classification (SNIPS) and sarcasm detection tasks. It was also unable to perform arabic speech command recognition, never predicting one of the label options. Furthermore, in zero-shot inference on SNIPS, SALMONN performed worse than UniverSLU-14 Natural Phrase (See Tab. 5). These results strongly support our assertion that existing LLMs struggle with the challenges posed by diverse SLU tasks. The need for
UniverSLU, as demonstrated by its superior performance in handling a wide array of SLU tasks, becomes evident in light of these findings.

A.5 Generating Instructions from ChatGPT

In this work, we utilize ChatGPT to generate natural language task descriptions for each task type. We list our prompts for obtaining task descriptions for each task type in Table 10. We further perform manual inspection to ensure that these task descriptions resemble how a user might describe a task. To illustrate, for the voice activity detection (VAD) task type, we manually selected the following example paraphrases such as "Determine the presence of spoken words in the audio.", "Identify if there is speech in the provided audio.", "Verify the existence of speech in the given audio.". For the VAD task type, we deliberately excluded paraphrases that did not explicitly address the detection of "speech" within the audio. For example, "Appraise if there are audible utterances in the audio" appeared to discuss the possibility of general audio, including background noise like a dog barking, rather than focusing specifically on speech. Similarly, "Assess if there is language spoken in the audio" seemed to lean towards identifying the language being spoken, diverging from the primary objective of VAD. We found these descriptions to deviate somewhat from the essence of the VAD task; hence, we removed them as we believe that the end user would not be using these descriptions to describe that task.

We keep querying the ChatGPT until we have sufficient task descriptions for training and inference of our models. We plan on exploring a more automated and systematic approach that can efficiently scale to select descriptions across a broader range of tasks in future work.
<table>
<thead>
<tr>
<th>Task Type</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCR</td>
<td>Different ways of saying “recognize speech command” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>IC</td>
<td>Different ways of saying “classify intent of spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>LID</td>
<td>Different ways of saying “identifying the language in verbal communication.” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>FSD</td>
<td>Different ways of saying “differentiate speech synthesis and voice conversion speech from real speech” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>ER</td>
<td>Different ways of saying “emotion recognition of spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>AcC</td>
<td>Different ways of saying “identifying the accent in spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>SD</td>
<td>Different ways of saying “detect whether given speech has sarcasm” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>GID</td>
<td>Different ways of saying “detect gender of speaker in given speech” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>VAD</td>
<td>Different ways of saying “detect if there is speech in the given audio” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>AuC</td>
<td>Different ways of saying “classify the environment sound present in audio” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>NER</td>
<td>Different ways of saying “recognise the named entities in spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>SP</td>
<td>Different ways of saying “write the semantic parse of spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
<tr>
<td>DAC</td>
<td>Different ways of saying “classify the dialog act of spoken utterance” in 15 ways in a natural language without changing meaning</td>
</tr>
</tbody>
</table>

Table 10: Prompts for ChatGPT to generate task descriptions for each task type.
<table>
<thead>
<tr>
<th>Task Type</th>
<th>Split</th>
<th>Natural Language Phrase</th>
</tr>
</thead>
</table>
| IC       | Train/Dev | Assessing and categorizing the meaning behind spoken utterances  
Uncovering the intent behind verbal communication  
Sorting and classifying the purpose of spoken statements  
Decoding and categorizing the intention behind spoken language  
Understanding and labeling the intent in spoken expressions  
Organizing and categorizing the meaning of spoken utterances  
Interpreting and classifying the purpose of verbalized intentions  
Analyzing and categorizing the underlying purpose of spoken words  
Grouping and identifying the intention behind spoken utterances  
Differentiating and classifying the intent expressed through speech  
Extracting and labeling the classification of spoken intentions  
Segregating and classifying the purpose of verbalized expressions  
Unraveling and categorizing the intended meaning of spoken statements  
Intent classification of spoken utterance  |
|          | Test   | Classifying the purpose of verbal expression  
Identifying the intent conveyed through speech  
Decipher the objective conveyed through speech  
Determine the purpose of the spoken expression  
Identify the motive embedded in speech  |
| SCR      | Train/Dev | Recognizing speech command  
Decoding oral requests  
Grasping speech guidance  
Discerning spoken triggers  
Acknowledging vocal commands  
Comprehending voiced instructions  
Interpreting verbalized prompts  
Detecting spoken signals  
Capturing voice-activated orders  
Classifying speech-based commands  
Inferring uttered guidance  
Analyzing vocal instructions  
Differentiating voiced prompts  
Sorting vocalized guidance  
Parsing spoken directives  |
|          | Test   | Understanding spoken instructions  
Identifying uttered directives  
Identify spoken instructions  
Understand voiced directives  
Decode vocal orders  |
| ER       | Train/Dev | Emotion recognition of spoken utterance  
Understanding the emotional content of spoken utterance  
Interpreting and categorizing emotions in verbal communication  
Decoding the emotional aspect of spoken expressions  
Discerning and recognizing emotions in spoken language  
Assessing and labeling the emotional state of spoken utterances  
Grasping the emotional intent behind spoken words  
Uncovering and categorizing the emotional response in speech  
Extracting and identifying the emotions conveyed through spoken utterances  
Emotionally analyzing and categorizing spoken expressions  
Capturing and recognizing emotional cues in spoken utterance  
Processing and discerning emotions in verbalized communication  
Differentiating and classifying the emotional tone of spoken utterances  
Unraveling and categorizing the intended emotional expression of spoken statements  
Inferring and organizing the classification of emotions in spoken language  |
|          | Test   | Detecting emotional content in spoken utterance  
Identifying emotions conveyed through speech  
Perceive the emotion conveyed through speech  
Discern the feelings embedded in spoken utterance  
Categorize the emotional content of verbal expression  |

Table 11: Detail of Natural Language Phrases used as Prompts to train and evaluate our MTL model in Table 2