PICD-Instruct: A Generative Instruction Learning Framework for Few-Shot Multi-Intent Spoken Language Understanding

Wenbin Hua[1,2,3], **Rui Fan**[1,2,4], **Tingting He**[1,2,3,*], **Ming Dong**[1,2,3]
[1]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning,
[2]National Language Resources Monitor and Research Center for Network Media,
[3]School of Computer Science,
[4]School of Chinese Language and Literature, Central China Normal University, Wuhan, China
{wenbin_hua, fanrui}@mails.ccnu.edu.cn
{tthe, dongming}@ccnu.edu.cn

Abstract

Few-shot multi-intent spoken language understanding (SLU) aims to identify users' multiple intents and key slots using a tiny amount of annotated data. Recent advances in large language models (LLMs) have utilized instruction learning frameworks to model intent-slot interdependencies, typically requiring abundant data for effective training. However, in few-shot scenarios, these frameworks face challenges such as mismatches between the number of generated slots and input lengths, relational confusion in multi-intent scenarios and neglect of task-specific variations in intent counts across utterances. To overcome the challenges, we propose PICD-Instruct, a novel generative framework based on Basic Instructions (BI), Pairwise Interaction Instructions (PII) and Contrastive Distinct Instructions (CDI). Specifically, BI directs LLMs to generate entities along with associated words, thereby mitigating mismatches in quantitative correspondences. PII explicitly captures dual-task interdependencies by guiding LLMs to pair each intent with its related entities. CDI enhances understanding of utterances by guiding LLMs to determine whether two utterances share the same intent count. Experimental results on public datasets indicate that PICD-Instruct achieves state-of-the-art performance[1].

1 Introduction

Spoken Language Understanding (SLU) (Young et al., 2013) is a fundamental component of task-oriented dialogue systems. Among the various aspects of SLU, multi-intent SLU has gained significant attention due to its practical necessity in complex interactive scenarios. This task involves two closely linked subtasks: multi-intent detection and slot filling. Multi-intent detection focuses on identifying the intents embedded within a user utterance, whereas slot filling extracts key semantic information from the utterance. In practical applications, however, obtaining sufficient labeled data for domain-specific SLU models is often time-intensive and costly. These challenges highlight the critical importance of exploring multi-intent SLU in few-shot settings.

Given the bidirectional relationship between intents and slots, recent models leverage multi-task joint frameworks to capture these interdependencies, achieving strong performance with sufficient training data (Goo et al., 2018; Li et al., 2018; Niu et al., 2019; Liu et al., 2019a; Qin et al., 2020, 2021; Song et al., 2022; Chen et al., 2022; Xing and Tsang, 2022a,b; Mei et al., 2023; Song et al., 2024). Meanwhile, large language models (LLMs) show promise in the zero-shot SLU task (Pan et al., 2023; Zhu et al., 2024) but remain largely designed for single-intent scenarios. For instance, Pan et al. (2023) explored prompt-based zero-shot SLU with ChatGPT, but its slot filling lagged far behind fine-tuned models. Similarly, Zhu et al. (2024) proposed a pseudo-labeling framework to enhance task collaboration but faced error propagation issues. To address these limitations, Xing et al. (2024) first introduced instruction learning into generative multi-intent SLU. Their framework leverages instruction learning and contrastive learning to model intent-slot relationships through mutual prediction of ground-truth labels. By distinguishing task-specific semantics across utterances, this approach enhances SLU reasoning. This raises a key question: Can instruction-guided LLMs achieve superior performance in few-shot multi-intent SLU?

Beyond addressing traditional SLU challenges, LLMs introduce new opportunities by enhancing structured and reliable information extraction (Li et al., 2024). SLU plays a crucial role in intelligent agent-driven task completion, where accurate intent detection ensures effective execution of user

*Corresponding author

[1] Our source code is available at https://github.com/heavenCSH/LLM_SLU

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 4942–4956
November 4-9, 2025 ©2025 Association for Computational Linguistics

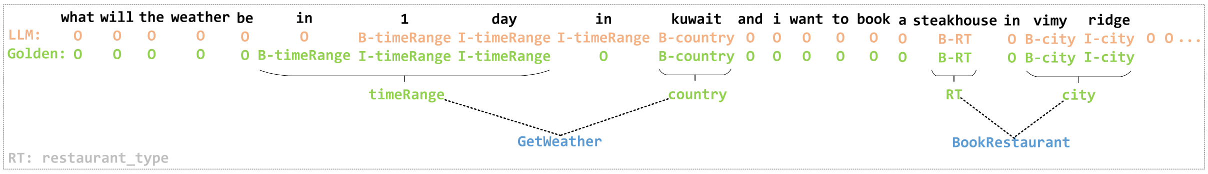

what will the weather be in 1 day in kuwait and i want to book a steakhouse in vimy ridge
LLM: O O O O O O B-timeRange I-timeRange I-timeRange B-country O O O O O O B-RT O B-city I-city O O...
Golden: O O O O O B-timeRange I-timeRange I-timeRange O B-country O O O O O O B-RT O B-city I-city
timeRange
country
RT
city
GetWeather
BookRestaurant
RT: restaurant_type

Figure 1: An example from MixSNIPS dataset. Traditional LLMs-generated slot labels are in orange, while golden slot labels and our proposed entity labels are in green. Intent labels are in blue.

commands (Caren Han et al., 2022). Unlike open-ended generation, SLU requires structured output to maintain schema consistency, which is critical for applications in domains such as voice assistants, customer service automation, and smart device control (Saxon et al., 2021; Irugalbandara, 2024).

However, we discover three core challenges in leveraging LLMs for few-shot multi-intent SLU. Firstly, the uncontrollable nature of LLM-generated outputs poses significant challenges for slot filling, as the number of generated slots often fails to correspond with the input length. This issue is exacerbated in few-shot settings, where limited training data restricts the model's ability to accurately map slots to tokens. As shown in Fig. 1, the example demonstrates the over-generation and mismatch of slot labels. Secondly, existing generative frameworks exhibit a strong dependence on extensive annotated data and fail to effectively capture the semantic dependencies between intents and slots. DC-Instruct (Xing et al., 2024) predicts slot labels based on the provided utterance and intent labels, but it falls short in establishing a one-to-one correspondence between each intent and its associated slots. This leads to confusion in multi-intent scenarios, making it harder for models to learn dual-task interdependencies with limited training data. Thirdly, unlike single-intent scenarios, the number of intents contained in a user's utterance in multi-intent scenarios is often uncertain, making it more challenging for models to accurately identify all intents. Therefore, improving the sensitivity of LLMs to the variations in intent counts across utterances can enhance their understanding of such cases. However, current approaches often overlook this task-specific feature, potentially hindering the models' ability to effectively comprehend utterances with multiple intents.

To overcome these challenges, we propose PICD-Instruct, a novel generative model based on instruction learning. PICD-Instruct employs three types of instructions: Basic Instructions (BI), Pairwise Interaction Instructions (PII) and Contrastive Distinct Instructions (CDI). BI shifts from the traditional approach of assigning a slot label to each word to a formulation based on entity-word pairings, effectively mitigating mismatches between generated slots and input lengths commonly encountered when using LLMs for direct slot generation. Considering that each green entity label in Fig. 1 aligns exactly with its associated words, PII incorporates an auxiliary intent-slot pairing task that explicitly models the bidirectional dependencies between intents and slots. By aligning golden intent labels with corresponding entity labels, PII mitigates relational confusions in multi-intent scenarios. CDI enhances the ability to perceive variations in the number of intents within an utterance by introducing a task that determines whether two utterances contain the same number of intents. By leveraging positive and negative samples alongside the current utterance, CDI trains the model to distinguish between utterances based on intent counts, thereby improving its comprehension capabilities.

We conduct experiments on two few-shot datasets, FewShotMixATIS and FewShotMixSNIPS (Hua et al., 2024). Experimental results show that PICD-Instruct significantly outperforms existing baselines, achieving state-of-the-art (SOTA) performance in the few-shot multi-intent SLU task. Moreover, it demonstrates strong generalization capability, transferring from a single-domain dataset (FewShotMixATIS) to a multi-domain dataset (FewShotMixSNIPS).

In summary, our contributions are three-fold:

(1) We propose PICD-Instruct, a novel generative instruction-learning framework that integrates pairwise interactive instructions and contrastive distinct instructions to overcome challenges in the few-shot multi-intent SLU task.

(2) We advance the explicit modeling of bidirectional dependencies between intents and slots in few-shot settings, reducing relational confusions in multi-intent scenarios through the application of instruction learning.

(3) PICD-Instruct achieves SOTA performance in the few-shot multi-intent SLU task, as evidenced by extensive experiments and analyses.

2 Related Work

Multi-intent SLU Prevailing models (Kim et al., 2017; Gangadharaiah and Narayanaswamy, 2019) often employ joint modeling to simultaneously learn the two tasks in SLU and capture their relations. Gangadharaiah and Narayanaswamy (2019) jointly model multiple intent detection and slot filling via a slot-gate mechanism. To better model the two tasks' interactions, graph neural networks have been widely utilized (Qin et al., 2020, 2021; Xing and Tsang, 2022a,b; Song et al., 2022). The Co-guiding Net (Xing and Tsang, 2022a) pioneers in achieving mutual guidance between the two tasks through a two-stage framework. DC-Instruct (Xing et al., 2024) employs instructions for LLMs to predict one subtask's labels based on the other's golden labels, effectively capturing the relationships between intents and slots. UGEN (Wu et al., 2022) and PromptSLU (Song et al., 2024) performs multi-intent SLU based on the paradigm of prompt learning.

The above approaches primarily focus on scenarios with abundant training data. However, in few-shot settings, capturing the correlations between the two tasks in SLU becomes more challenging, leading to degraded performance for most models (Hua et al., 2024). While UGEN and DC-Instruct have demonstrated performance in low-resource settings, the few-shot training data they utilize does not align well with real-world application scenarios in terms of sample quantity and distribution. To better simulate practical application scenarios, we employ FewShotMixATIS and FewShotMixSNIPS, two datasets specifically tailored for few-shot scenarios, as the data for model training. Different from recent works, we propose a novel generative framework incorporating various instructions to ensure the accuracy of LLM outputs. Our approach explicitly captures dual-task interdependencies by reducing relational confusions and effectively harnesses the variations of intent counts across different utterances, enabling improved performance in the few-shot multi-intent SLU task.

Instruction Learning Recently, the rise of LLMs in the natural language processing (NLP) field has positioned instruction learning as a competitive approach across various NLP tasks (Lou et al., 2024; Safa et al., 2024). This paradigm effectively leverages the advanced conversational abilities of LLMs to perform generative tasks, bridging the gap between pre-training and fine-tuning stages.

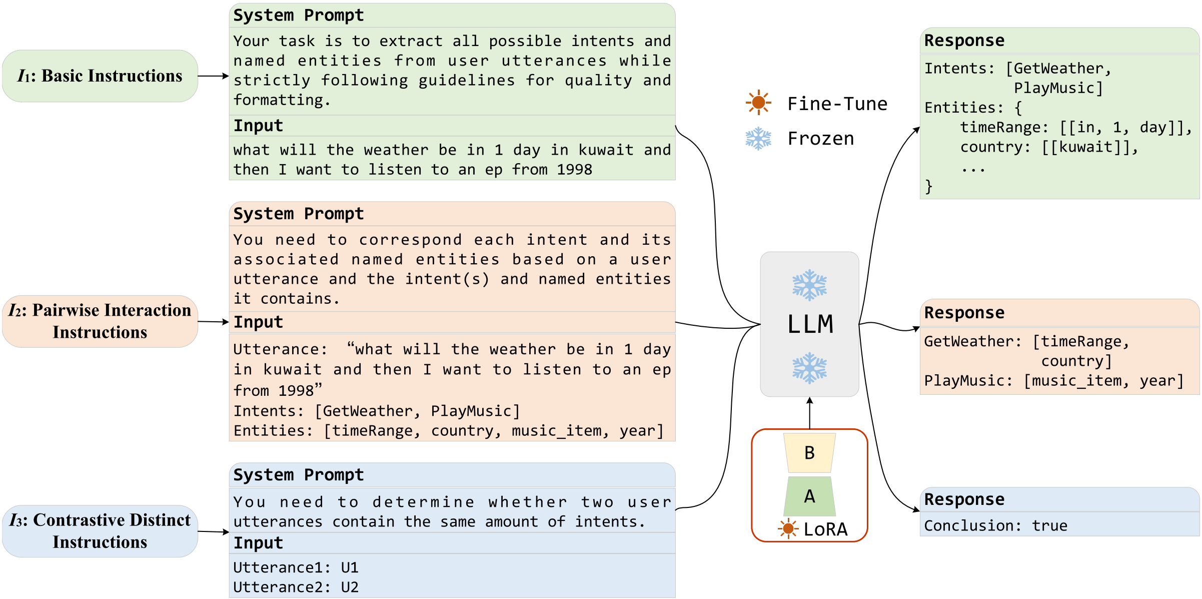
In this work, we investigate instruction learning for few-shot multi-intent SLU and propose a novel model characterized by pairwise interactive instructions and contrastive distinct instructions.

3 Task Definition

As shown in the example in Fig. 1, multi-intent SLU aims to detect all possible intents within an utterance and identify the slot label corresponding to each word. Therefore, multi-intent detection is considered as a multi-label text classification task and slot filling is regarded as a sequence labeling task. The task can be formulated as follows: given an input utterance $X = \{W_1, W_2, \dots, W_n\}$, where n is the length of the utterance. The objective is to predict the correct intents from the candidate intents $I = \{i_1, i_2, \dots, i_m\}$ and identify the slot label for each word W_i from the candidate slot types $S = \{s_1, s_2, \dots, s_k\}$, where m is the number of intent categories, and k is the number of slot types. In the slot filling task, slot labels are typically annotated in the BIO format, where B indicates the beginning of an entity, I denotes the continuation of the entity, and O represents words that do not belong to any entity. As illustrated in Fig. 1, *vimy ridge* is an entity representing a city, and thus it is annotated as $\{$*B-city I-city*$\}$. Due to the nature of the approach proposed in this paper, the task first predicts the entities and their corresponding words within an utterance, and then reassigns the entities with the BIO annotation. For instance, the word corresponding to the *city* entity would be *vimy ridge*.

4 Methodology

In this section, we introduce our proposed PICD-Instruct framework. As depicted in Fig. 2, we formulate our instructions in a question-answer (QA) form. The framework includes three types of instructions, each corresponding to a specific task. This approach mitigates the effects of uncontrollable generation by LLMs and more explicitly models the correlations between the two tasks in SLU, reducing relational confusions. In addition, it enhances the model's ability to understand utterances with multiple intents. The following subsections provide a detailed explanation of our proposed basic instructions (I_1), pairwise interaction instructions (I_2) and contrastive distinct instructions (I_3).


I_1: Basic Instructions
I_2: Pairwise Interaction Instructions
I_3: Contrastive Distinct Instructions
Fine-Tune
Frozen
LLM
B
A
LoRA

```
System Prompt
Your task is to extract all possible intents and
named entities from user utterances while
strictly following guidelines for quality and
formatting.
Input
what will the weather be in 1 day in kuwait and
then I want to listen to an ep from 1998
```

```
System Prompt
You need to correspond each intent and its
associated named entities based on a user
utterance and the intent(s) and named entities
it contains.
Input
Utterance: “what will the weather be in 1 day
in kuwait and then I want to listen to an ep
from 1998”
Intents: [GetWeather, PlayMusic]
Entities: [timeRange, country, music_item, year]
```

```
System Prompt
You need to determine whether two user
utterances contain the same amount of intents.
Input
Utterance1: U1
Utterance2: U2
```

```
Response
Intents: [GetWeather,
          PlayMusic]
Entities: {
    timeRange: [[in, 1, day]],
    country: [[kuwait]],
    ...
}
```

```
Response
GetWeather: [timeRange,
             country]
PlayMusic: [music_item, year]
```

```
Response
Conclusion: true
```

Figure 2: Overview of our framework. Detailed instructions are shown in Appendix A.

4.1 Basic Instructions

The basic instructions (I_1) are designed to guide the model in generating the intents, named entities and their corresponding words expressed in the utterance. The key components of the basic instructions are illustrated as follows:

```
[Persona]: You are an expert in multi-intent
spoken language understanding. Your task is to
extract all possible intents and named entities
from user utterances while strictly following
guidelines for quality and formatting.

[Instructions]: First, identify the intents
in the utterance. The intent options are:
{Intent Label Set}. Next, identify the named
entities and list each entity with its
corresponding words, the entity options are:
{Entity Label Set}.
```

where the persona specifies the model's role and the tasks to be performed, while the instructions detail the step-by-step procedures and requirements. To facilitate result extraction and ensure the controllability of model outputs, the response format for all tasks is standardized in the *JSON* format. It can be formulated as:

$$R = L(SP, I) \tag{1}$$

where SP represents the system prompt, I is the input, L denotes the LLM and R is the response. By converting R into a Python dictionary, we can extract the intents and entities. After obtaining all entities and their corresponding words, inspired by (Wang et al., 2023), we map the words back to their original slot labels using the BIO rule, adhering to the natural left-to-right order of the utterance. This approach allows the LLM to concentrate solely on establishing correspondence between entities and words, disregarding the requirement that the number of final slot labels matches the utterance length. This effectively circumvents the difficulty LLMs face in learning such quantitative correspondences in few-shot scenarios.

4.2 Pairwise Interaction Instructions

To explicitly model dual-task dependencies and reduce relationship confusion, we propose the pairwise interaction instructions (PII). PII is designed to pair each intent with its related entities based on the provided utterance, along with its intent and entity labels. The key components of the PII are as follows:

```
[Persona]: You are an expert in multi-intent
spoken language understanding. You need to
correspond each intent and its associated named
entities based on a user utterance and the
intent(s) and named entities it contains.

[Instructions]: There is a close relationship
between each intent and certain named entities.
You need to pair them separately.
```

As shown in Fig. 2, during training, dual-task dependencies are captured by achieving two kinds of alignments. First, in the input part, both the utterance semantics and the labels for the two subtasks are included, achieving a semantic-label alignment for the tasks. Second, dual-task label alignment is established by pairing intent and entity labels in the generation side. With the straightforward mechanism of separate pairing

Statistic	FewShotMixATIS					FewShotMixSNIPS				
# K-shot	2-shot	4-shot	6-shot	8-shot	10-shot	2-shot	4-shot	6-shot	8-shot	10-shot
# Original training instances	34	66	100	137	172	14	27	40	54	70
# PICD-Instruct training instances	1,717	6,501	14,950	27,948	44,290	287	1,053	2,380	4,347	7,315
# Training slot types	47	53	57	61	65	30	44	50	54	58
# Testing slot types			82					70		
# Testing instances			828					2199		

Table 1: Detail Statistics of FewShotMixATIS and FewShotMixSNIPS.

between each intent and its related entities, the mutual dependencies of the two subtasks can be more easily and directly captured by LLMs with their strong few-shot learning capabilities. In addition, it also subtly reduces relational confusions in multi-intent scenarios.

4.3 Contrastive Distinct Instructions

Unlike single-intent scenarios, the number of intents contained in an utterance in multi-intent scenarios is often uncertain. Previous works overlook variations in intent counts among utterances, a factor that aids in understanding utterances with multiple intents. Inspired by (Xing et al., 2024), we leverage contrastive relationships centered around intent count differences to enhance the comprehension of utterances and further improve SLU performance. As shown in Fig. 3 (a), traditional contrastive learning aims to optimize representations by pulling similar samples closer in the latent space while pushing dissimilar samples away. To adapt this approach to generative models, we propose straightforward yet effective instructions to implement contrastive learning in the instruction learning paradigm, as shown in Fig. 3 (b). We first sample a positive utterance P and a negative utterance N in relation to the current utterance C. Then we construct instructions to ask the LLM whether C and P, or C and N have the same amount of intents. The expected output is a simple binary response:"true" or "false". The key components of the CDI are as follows:

```
[Persona]: You are an expert in multi-intent
spoken language understanding. You need to
determine whether two user utterances contain
the same amount of intents.

[Instructions]: You will be given two user
utterances. Each utterance may contain single
or multiple intents. You need to judge whether
the two utterances contain the same amount of
intents.
```

This approach leverages contrastive relationships to improve the ability of generative LLMs to perceive variations in the number of intents within an utterance in multi-intent scenarios.

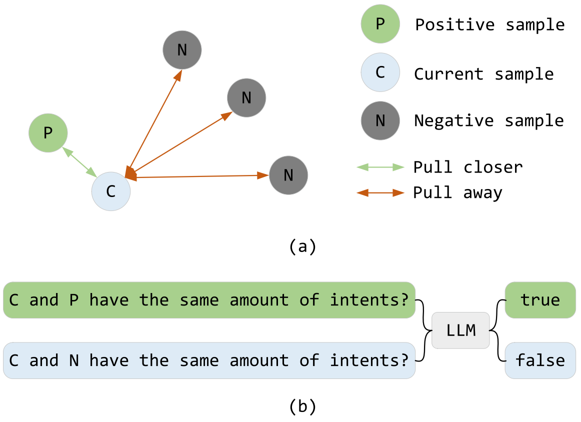


Figure 3: Traditional contrastive learning and our proposed CDI based on instruction learning.

4.4 Training and Inference

Training First, an I_3 is constructed for every two samples. Next, an I_1 and an I_2 are created for each sample. To facilitate efficient annotation, GPT-4o[2] is employed to label I_2. Details of the prompt settings are provided in Appendix B. The shuffled training data is then utilized to train the model in a text-to-text generation form. The training objective is to minimize the negative log-likelihood for each instruction: $\mathcal{L} = -\sum_{n=1}^{N} \log p(y_n \mid y_{<n}, I)$. N is the length of the golden output sequence $y_1, ..., y_N$ and I denotes the current input instruction.

Inference In the inference stage, only I_1 is used to generate predictions for both multiple intent detection and slot filling.

5 Experiments

5.1 Experiment Setup

5.1.1 Dataset

We compare our method with the baselines on two few-shot multi-intent SLU datasets, FewShotMixATIS and FewShotMixSNIPS. They are derived from MixATIS and MixSNIPS datasets (Qin et al., 2020) using the dynamic sampling algorithm proposed by (Wang et al., 2023). As shown in Table 1, each dataset includes five types of few-shot samples, ranging from 2-shot to 10-shot for training. For testing, we use the test sets of original standard

[2] https://chatgpt.com/

datasets (*i.e.*, MixATIS and MixSNIPS). Notably, the test sets contain more slot types than the training sets, better reflecting models' generalization ablility to unseen labels. This setup effectively simulates a realistic application scenario for the task.

To ensure a balanced number of the three instruction types, oversampling (repetitive sampling) is applied to I_1 and I_2 to match the scale of I_3 (C_n^2). The final dataset sizes ranging from 2-shot to 10-shot are presented in the third row of Table 1.

5.1.2 Implementation Details

For PICD-Instruct, we use Qwen2.5-7B[3] as its backbone model. The model employs AdamW (Loshchilov and Hutter, 2017) as the optimizer with an initial learning rate of 3e-5, along with a scheduler that applies linear warm-up for learning rate adjustment. We adopt low-rank adaptation (LoRA) (Hu et al., 2021) to fine-tune the model, with only 55M/28M trainable parameters for FewShotMixATIS/FewShotMixSNIPS. We set the LoRA rank to 128/64 for FewShotMixATIS/FewShotMixSNIPS. The batch size is 16 for both datasets. We conduct experiments based on the llamafactory (Zheng et al., 2024) framework to improve the efficiency of implementation. The Experiments are conducted on two NVIDIA A5000 GPUs. In multi-intent SLU, accuracy (Acc), F1 score and overall accuracy are used as evaluation metrics for multiple intent detection, slot filling and the SLU semantic frame parsing.

5.2 Main Results

We compare our model with BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019b), gpt-3.5-turbo, and several top-performing models. Specifically, AGIF (Qin et al., 2020) presents an adaptive interaction network to achieve fine-grained multiple intent information integration for token-level slot filling. GL-GIN (Qin et al., 2021) introduces a Global-Locally Graph Interaction Network which explores a non-autoregressive model for joint multiple intent detection and slot filling. Wu et al. (2022) proposes a Unified Generative framework (UGEN) based on a prompt-based paradigm and formulates the task as a question-answering problem. BERT-SIF introduces a separate intent-slot interaction framework based on prompt learning to mitigate relational confusions. The results of above baselines are sourced from Hua et al. (2024), who implemented the above models using their official code. To more comprehensively evaluate the effectiveness of our model, we include Uni-MIS (Yin et al., 2024a), ENSI-Qwen2.5 (Yin et al., 2024b) and gpt-4o-mini in the performance comparisons. Specifically, Uni-MIS models multi-intent SLU through a three-view intent-slot interaction fusion mechanism to better capture the interaction information. As an early attempt to apply LLMs to the multi-intent SLU task, ENSI-Qwen2.5 extends Qwen2.5(7B) by introducing the concepts of entity slots and sub-intents to facilitate task completion. For Uni-MIS, results are obtained by executing the official code provided by the authors. For ENSI-Qwen2.5, since the complete code has not yet been released, we reproduce the model's training process to obtain the reported results. The GPT-4o-mini experiment is conducted following the same methodology as in Hua et al. (2024). Due to limitations in prompt length and costs, the gpt-4o-mini experiment is conducted exclusively in the 2-shot setting. As the source code for DC-Instruct is unavailable and key experimental parameters are not fully reported, we are unable to include it in our comparative experiments. Performance comparisons are presented in Tabel 2 and 3, from which we have the following observations:

(1) ***PICD-Instruct achieves new state-of-the-art performance on both datasets.*** On the FewShotMixATIS dataset, PICD-Instruct surpasses BERT-SIF in the 2-shot setting by 39.26%, 2.63%, and 13.16% on intent accuracy, slot F1 and overall accuracy, respectively. On the FewShotMixSNIPS dataset, it outperforms BERT-SIF in the 2-shot setting by 48.84%, 20.21% and 4.86% on intent accuracy, slot F1 and overall accuracy. As the amount of training data increases, the performance of our model and all baselines consistently improves across both datasets. This improvement is attributed to our model's explicit capture of dual-task dependencies via pairwise interaction instructions. The straightforward and effective mechanism significantly reduces training complexity in few-shot scenarios. In addition, our designed contrastive distinct instructions enhance the LLM's capability to differentiate variations in intent counts across utterances, which further improves its understanding in multi-intent scenarios. Furthermore, our method of guiding the LLM to generate entities along with their corresponding words effectively mitigates the mismatch between the number of slots and the utterance length, a challenge that LLMs typically face when learning quantitative

[3] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Model	**FewShotMixATIS**														
	2-shot			4-shot			6-shot			8-shot			10-shot		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
PLM-based Models															
BERT	0	57.38	0	4.47	68.37	2.66	12.44	69.54	6.40	25.36	74.23	10.99	36.11	76.66	17.15
RoBERTa	0	48.90	0	0	56.68	0	6.04	65.17	1.33	6.52	68.27	2.17	16.79	70.96	9.18
AGIF$_{\text{(BERT)}}$	0	38.28	0	0.60	32.73	0	10.75	48.13	3.02	15.10	38.79	3.50	29.83	56.91	8.94
GL-GIN$_{\text{(BERT)}}$	1.21	6.49	0	6.52	21.32	1.57	14.49	32.09	2.90	18.84	33.89	3.26	23.67	49.54	5.56
UGEN$_{\text{(T5)}}$	4.47	54.31	1.33	21.98	68.44	6.52	53.50	72.78	15.94	59.30	74.84	19.57	66.67	76.40	22.71
Uni-MIS$_{\text{(RoBERTa)}}$	10.75	29.91	1.93	40.10	46.68	6.16	67.15	62.02	12.56	70.65	62.16	16.43	70.65	68.86	21.14
BERT-SIF	30.31	62.51	5.80	37.56	65.74	7.97	58.09	68.20	13.53	61.47	**74.90**	21.26	62.56	**77.61**	23.55
LLM-based Models															
gpt-3.5-turbo	30.07	6.85	0.60	-	-	-	-	-	-	-	-	-	-	-	-
gpt-4o-mini	58.21	8.87	2.05	-	-	-	-	-	-	-	-	-	-	-	-
ENSI-Qwen2.5	25.60	41.99	4.47	39.98	46.62	6.52	45.41	52.73	7.13	47.58	54.90	9.42	51.09	56.78	9.66
PICD-Instruct	**69.57**	**65.14**	**18.96**	**70.29**	**69.07**	**21.38**	**72.71**	**72.11**	**24.76**	**78.86**	73.84	**27.54**	**81.64**	74.38	**28.02**

Table 2: Overall results on FewShotMixATIS. I-Acc, S-F1, O-Acc refer to the intent accuracy, slot F1, and overall accuracy (both intents and slots need to be correct), respectively. PLM denotes pre-trained language model.

Model	**FewShotMixSNIPS**														
	2-shot			4-shot			6-shot			8-shot			10-shot		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
PLM-based Models															
BERT	4.46	24.84	0.14	3.91	34.59	0	23.78	38.96	0.73	38.06	49.29	3.00	50.34	57.61	4.91
RoBERTa	0.55	8.87	0	1.36	19.04	0	24.51	33.05	0.50	30.38	33.41	0.68	37.79	37.25	0.68
AGIF$_{\text{(BERT)}}$	1.27	2.74	0	6.23	7.11	0	17.69	9.12	0.09	21.15	10.03	0.05	14.78	12.53	0.68
GL-GIN$_{\text{(BERT)}}$	7.50	0.61	0	14.19	1.48	0	28.06	2.03	0.09	34.20	5.49	0.27	58.21	9.62	0.18
UGEN$_{\text{(T5)}}$	2.64	13.10	0	29.65	33.07	0.23	38.84	40.31	1.96	61.57	46.80	4.37	73.08	58.38	7.78
Uni-MIS$_{\text{(RoBERTa)}}$	33.33	9.39	0.36	45.70	13.24	0.68	49.89	12.53	0.45	67.03	30.43	2.68	68.17	35.81	4.14
BERT-SIF	37.61	26.29	0.64	56.34	38.32	2.18	64.39	43.34	3.23	65.39	50.18	7.14	74.12	**61.75**	11.10
LLM-based Models															
gpt-3.5-turbo	64.48	3.91	0.18	-	-	-	-	-	-	-	-	-	-	-	-
gpt-4o-mini	**86.95**	8.29	0.73	-	-	-	-	-	-	-	-	-	-	-	-
ENSI-Qwen2.5	5.41	6.66	0.18	25.24	15.20	0.77	36.74	22.93	1.36	41.47	28.32	2.05	47.61	29.66	2.50
PICD-Instruct	86.45	**46.50**	**5.50**	**86.77**	**50.18**	**7.32**	**86.99**	**52.26**	**8.64**	**88.18**	**55.10**	**10.55**	**88.09**	58.14	**11.51**

Table 3: Overall results on FewShotMixSNIPS.

correspondences from a limited amount of annotated data. An additional point of interest lies in the use of GPT-4o to assist in annotating pairwise interaction instructions for the sake of efficiency, which may introduce a certain level of annotation noise. Nevertheless, PICD-Instruct consistently and significantly outperforms the baseline models, highlighting the robustness of our approach to potentially noisy annotations.

(2) ***Current LLM-based approaches can hardly handle few-shot multi-intent SLU.*** The performance of ChatGPT is consistent with recent findings (Pan et al., 2023; Qin et al., 2023). While gpt-4o-mini outperforms earlier pre-trained language models in the multiple intent detection task, its performance in slot filling falls significantly behind most of them. We suspect there are two main reasons. First, insufficiently descriptive prompt wording may negatively impact ChatGPT's performance. We believe advanced in-context learning strategies, such as chain-of-thought prompting, could partially enhance ChatGPT's performance, while this is beyond the scope of this paper. Second, multi-intent SLU requires task-specific knowledge, which is more effectively acquired through fine-tuning. This finding underscores the need for vertical domain-specific development, particularly for tasks requiring high levels of domain-specific expertise. ENSI-Qwen2.5 addresses the mismatch between the slot generation length of LLMs and the actual utterance length, as well as improve alignment between sub-intents and clauses, by introducing the concepts of entity slots and sub-intents. However, it falis to capture the relationships between intents and slots and does not effectively model the varying informational richness across different utterances. As a result, its performance on multi-intent SLU remains limited in few-shot settings.

5.3 Ablation Study

In this section, we conduct ablation experiments to explore the effect of each component of our PICD-Instruct model. The results are shown in Table. 4.

Basic Instructions (BI). Retaining only BI (I_1) still yields significant improvements compared to the previous best-performing model, BERT-SIF, especially in slot filling, where it outperforms ChatGPT. This demonstrates that BI effectively guides the LLM to generate entities along with their corresponding words, simplifying the process of slot fill-

Model	**FewShotMixATIS**														
	2-shot			4-shot			6-shot			8-shot			10-shot		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
w/o PII, CDI (I_2, I_3)	67.51	64.43	17.75	68.84	68.07	20.65	71.50	70.65	22.83	78.02	72.75	26.69	77.66	73.54	26.81
w/o PII (I_2)	68.24	64.57	17.87	68.96	68.24	20.77	71.62	70.98	22.95	78.26	72.91	26.81	78.14	73.68	27.05
w/o CDI (I_3)	68.48	64.86	18.24	69.20	68.71	21.01	71.98	71.46	23.67	78.50	73.13	27.05	79.23	73.84	27.17
PICD-Instruct	**69.57**	**65.14**	**18.96**	**70.29**	**69.07**	**21.38**	**72.71**	**72.11**	**24.76**	**78.86**	**73.84**	**27.54**	**81.64**	**74.38**	**28.02**
	FewShotMixSNIPS														
w/o PII, CDI (I_2, I_3)	84.86	45.14	4.50	85.31	48.27	6.18	86.08	51.16	7.64	86.22	54.31	9.23	86.45	56.25	10.56
w/o PII (I_2)	85.08	45.48	4.64	85.54	48.62	6.41	86.36	51.48	7.82	86.45	54.58	9.64	86.68	56.64	10.83
w/o CDI (I_3)	85.54	46.11	4.96	85.95	49.03	6.82	86.90	51.93	8.05	86.81	54.97	10.14	87.04	57.13	11.28
PICD-Instruct	**86.45**	**46.50**	**5.50**	**86.77**	**50.18**	**7.32**	**86.99**	**52.26**	**8.64**	**88.18**	**55.10**	**10.55**	**88.09**	**58.14**	**11.51**

Table 4: Results of ablation experiments.

Model	**FewShotMixATIS**			**FewShotMixSNIPS**		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
Qwen2.5-3B	57.25	57.78	16.55	73.22	36.00	3.32
Qwen2.5-7B	69.57	65.14	18.96	86.45	46.50	5.50
Qwen2.5-14B	**71.74**	**70.04**	**23.67**	**88.45**	**51.12**	**8.23**

Table 5: Results comparison of different model sizes in the 2-shot setting.

ing. Besides, well-crafted instructions fully leverage the few-shot learning capabilities of LLMs, enabling a deeper understanding of the multi-intent SLU task and improving task execution.

Pairwise Interaction Instructions (PII). Adding PII (I_2) results in obvious improvements across all metrics and in all few-shot settings. It indicates that PII effectively and explicitly captures the dual-task correlations, leading to substantial performance enhancements. Moreover, PII helps mitigate relational confusions in multi-intent scenarios. The results further verify the fact that a direct and effective interaction mechanism in the instruction learning paradigm is highly beneficial for few-shot learning.

Contrastive Distinct Instructions (CDI). The aim of CDI is to enhance the LLM's capability to understand variations in intent counts across utterances. The experimental results reveal that including CDI contributes to improvements in all metrics, verifying its necessity. Besides, combining CDI and PII further enhances the model's performance. This synergy arises from their individual contributions: CDI and PII excel at their respective tasks, and their integration establishes a strong interdependence. CDI improves the LLM's initial comprehension of an utterance's intent count, thereby facilitating multiple intent detection. PII explicitly captures dual-task dependencies, reinforcing the relationship between tasks and enhancing slot filling performance. Therefore, removing any one of CDI and PII leads to performance decreases on all of intent accuracy, slot F1 and overall accuracy.

5.4 Effects of Model Size

To further evaluate the impact of model size on performance, we experiment with 3B, 7B and 14B versions of Qwen2.5 on both datasets. Due to space limitation, we only put results in the 2-shot setting in Table 5, detailed results for other settings are provided in Appendix C. This analysis will help determine whether it is necessary to pursue larger model sizes and understand the trade-offs involved.

As shown in Table 5, the experimental results indicate that an increase in Qwen model size leads to improved performance. However, the performance gains in multiple intent detection and slot filling diminish as the model size increases further. For FewShotMixATIS dataset, increasing model parameters from 3B to 7B results in improvements of 12.32% and 7.36% in intent accuracy and slot F1, respectively. However, further increasing parameters from 7B to 14B only yields gains of 2.17% and 4.9% in intent accuracy and slot F1, respectively. A similar trend is observed for the FewShotMixSNIPS dataset, although overall accuracy shows more pronounced improvements when parameters are scaled from 7B to 14B. This suggests that the overall reasoning capability of the LLM improves significantly with increased model size. Consequently, pursuing larger-scale language models may not be essential for achieving substantial performance gains across all metrics in the context of multi-intent SLU. Moreover, we conduct experiments to explore the impact of model type on the performance in few-shot multi-intent SLU. Detailed results are provided in Appendix D and Appendix E.

5.5 Evaluation of GPT-4o Annotation Quality

This section presents a comprehensive analysis of the GPT-4o annotation quality on PII, with detailed prompt settings provided in Appendix B. Specifi-

Input	**Utterance:** rate rajinikanth: the definitive biography one out of 6 stars and then what's the movie schedule for b&b theatres **Intents:** [RateBook, SearchScreeningEvent] **Entities:** [object_name, rating_value, best_rating, rating_unit, object_type, location_name]
Output	**RateBook:** [object_name, rating_value, rating_unit] **SearchScreeningEvent:** [object_type, location_name]

Table 6: Case Study of a GPT-4o Annotation Error.

Model	FewShotMixATIS	FewShotMixSNIPS
GPT-4o	71.51	72.86

Table 7: Results of PII quality labeled by GPT-4o. The score represents the proportion of correctly labeled samples to the total samples.

Model	FewShotMixATIS				
	2-shot	4-shot	6-shot	8-shot	10-shot
w/o CDI(I_3)	95.17	95.41	95.77	96.50	97.83
PICD-Instruct	**96.62**	**97.83**	**97.71**	**98.55**	**98.67**
	FewShotMixSNIPS				
w/o CDI(I_3)	95.68	95.82	95.82	96.82	97.14
PICD-Instruct	**96.04**	**96.32**	**96.50**	**97.31**	**97.68**

Table 8: Ablation results of CDI's impact on intent count. The score represents the proportion of samples where the number of intents predicted by the model for an utterance is equal to the number of true intents.

cally, we manually verify the accuracy of the PII labels generated by GPT-4o for FewShotMixATIS and FewShotMixSNIPS. As shown in Table 7, the annotation accuracy is 71.51% for the FewShotMixATIS dataset and 72.86% for the FewShotMixSNIPS dataset. This indicates that there is still potential for improvement in annotation accuracy. To further investigate the causes of annotation errors, we analyze the most frequent errors made by GPT-4o in specific cases, discovering that it often omits certain entities. For example, as shown in Table 6, GPT-4o fails to annotate the *best_rating* entity, which should have been linked to the *RateBook* intent. Although we explicitly instruct GPT-4o in the prompt to annotate all entities, its adherence to this instruction is imperfect, introducing some noise into the PII annotations. Nevertheless, employing large language models for data annotation remains a valuable direction worth exploring. Despite the noise in the PII annotations, experimental results show that our proposed PICD-Instruct model still significantly outperforms other baseline models, demonstrating its robustness in practical applications.

5.6 Effects of CDI on Intent Count

The ablation experiment has demonstrated the effectiveness of CDI, and this section will further explore the impact of CDI on the number of intents generated by our proposed PICD-Instruct. Specifically, since CDI enhances the LLM's capability to understand variations in intent counts across utterances, we evaluate the proportion of samples where the predicted number of intents matches the true number, further demonstrating its effectiveness. As shown in Table 8, experimental results from both datasets indicate that incorporating CDI enables the model's predicted intent count to better align with the true number of intents within utterances. This improvement is attributed to CDI's significant enhancement of the model's ability to perceive intent counts in utterances through a contrastive learning mechanism, which further enhances intent detection accuracy and overall model performance in the few-shot multi-intent spoken language understanding task.

6 Conclusion

In this paper, we conduct an in-depth investigation of few-shot multi-intent SLU. We propose PICD-Instruct, a framework designed to address the challenges of generative few-shot multi-intent SLU from three key perspectives. Firstly, we propose basic instructions to tackle mismatches between the number of generated slots and input lengths. Secondly, we introduce pairwise interaction instructions to explicitly model dual-task dependencies while minimizing relational confusions in multi-intent scenarios. Thirdly, we present contrastive distinct instructions that leverage contrastive relations in intent counts to enhance understanding. Experimental results demonstrate that our proposed model achieves SOTA performance on FewShotMixATIS and FewShotMixSNIPS, thereby highlighting our model's robust generalization capabilities in a simulated real-world application scenario.

Limitations

This paper presents a comprehensive analysis of generative few-shot multi-intent SLU and introduces the PICD-Instruct model, which is based on the paradigm of instruction learning. In fact, detailed descriptions of intent and slot labels could significantly enhance LLMs' comprehension of multi-intent SLU, as high-quality external knowledge helps mitigate the hallucination issue in LLMs (Wan et al., 2024). In the future, we will explore how to integrate external label knowledge into LLMs to further improve the performance of few-shot multi-intent SLU.

Acknowledgments

This work is supported by Hubei Province Education Science Planning 2023 Special Funding Key Project: Research on Digital Transformation and Innovation of Teaching Models in Higher Education 2023ZA018 and the National Language and Character Research Base.

References

Soyeon Caren Han, Siqu Long, Henry Weld, and Josiah Poon. 2022. Spoken language understanding for conversational ai: Recent advances and future direction. *arXiv e-prints*, pages arXiv–2212.

Lisong Chen, Peilin Zhou, and Yuexian Zou. 2022. Joint multiple intent detection and slot filling via self-distillation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7612–7616. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota. Association for Computational Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Wenbin Hua, Yufan Wang, Rui Fan, Xinhui Tu, and Tingting He. 2024. Unraveling intricacies: A decomposition approach for few-shot multi-intent spoken language understanding. In *2024 IEEE International Conference on Big Data (BigData)*, pages 918–927. IEEE.

Chandra Irugalbandara. 2024. Meaning typed prompting: A technique for efficient, reliable structured output generation. *arXiv preprint arXiv:2410.18146*.

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76:11377–11390.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833.

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024. A simple but effective approach to improve structured language model output for information extraction. *arXiv preprint arXiv:2402.13364*.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019a. CM-net: A novel collaborative memory network for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1051–1060, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, pages 1–10.

Jie Mei, Yufan Wang, Xinhui Tu, Ming Dong, and Tingting He. 2023. Incorporating bert with probability-aware gate for spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:826–834.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. 2023. A preliminary evaluation of chatgpt for zero-shot dialogue understanding. *arXiv preprint arXiv:2304.04256*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1339–1384, Singapore. Association for Computational Linguistics.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 178–188, Online. Association for Computational Linguistics.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Online. Association for Computational Linguistics.

Abdulfattah Safa, Tamta Kapanadze, Arda Uzunoğlu, and Gözde Gül Şahin. 2024. A systematic survey on instructional text: From representation and downstream nlp tasks. *arXiv preprint arXiv:2410.18529*.

Michael Saxon, Samridhi Choudhary, Joseph P McKenna, and Athanasios Mouchtaris. 2021. End-to-end spoken language understanding for generalized voice assistants. *arXiv preprint arXiv:2106.09009*.

Feifan Song, Lianzhe Huang, and Houfeng Wang. 2024. A unified framework for multi-intent spoken language understanding with prompting. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9966–9970. IEEE.

Mengxiao Song, Bowen Yu, Li Quangang, Wang Yubin, Tingwen Liu, and Hongbo Xu. 2022. Enhancing joint multiple intent detection and slot filling with global intent-slot co-occurrence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7967–7977.

Fanqi Wan, Xinting Huang, Leyang Cui, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge verification to nip hallucination in the bud. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2616–2633, Miami, Florida, USA. Association for Computational Linguistics.

Yufan Wang, Jie Mei, Bowei Zou, Rui Fan, Tingting He, and Ai Ti Aw. 2023. Making pre-trained language models better learn few-shot spoken language understanding in more practical scenarios. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13508–13523, Toronto, Canada. Association for Computational Linguistics.

Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. 2022. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7203–7208.

Bowen Xing, Lizi Liao, Minlie Huang, and Ivor Tsang. 2024. Dc-instruct: An effective framework for generative multi-intent spoken language understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14520–14534.

Bowen Xing and Ivor Tsang. 2022a. Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 159–169, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022b. Group is better than individual: Exploiting label topologies and label relations for joint multiple intent detection and slot filling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3964–3975, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shangjian Yin, Peijie Huang, and Yuhong Xu. 2024a. Uni-mis: United multiple intent spoken language understanding via multi-view intent-slot interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19395–19403.

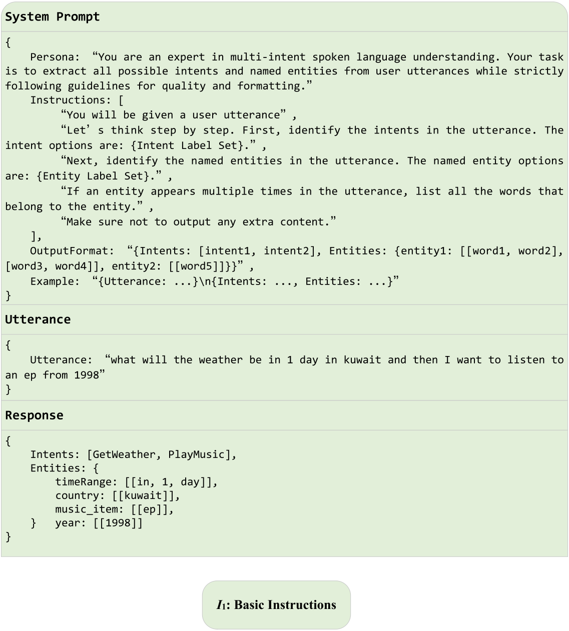
Shangjian Yin, Peijie Huang, Yuhong Xu, Haojing Huang, and Jiatian Chen. 2024b. Do large language model understand multi-intent spoken language? *arXiv preprint arXiv:2403.04481*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Zhihong Zhu, Xuxin Cheng, Hao An, Zhichang Wang, Dongsheng Chen, and Zhiqi Huang. 2024. Zero-shot spoken language understanding via large language models: A preliminary study. In *Proceedings of the*

2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 17877–17883, Torino, Italia. ELRA and ICCL.



System Prompt

```
{
    Persona:  “You are an expert in multi-intent spoken language understanding. Your task
is to extract all possible intents and named entities from user utterances while strictly
following guidelines for quality and formatting.”
    Instructions: [
        “You will be given a user utterance” ,
        “Let’ s think step by step. First, identify the intents in the utterance. The
intent options are: {Intent Label Set}.” ,
        “Next, identify the named entities in the utterance. The named entity options
are: {Entity Label Set}.” ,
        “If an entity appears multiple times in the utterance, list all the words that
belong to the entity.” ,
        “Make sure not to output any extra content.”
    ],
    OutputFormat:  “{Intents: [intent1, intent2], Entities: {entity1: [[word1, word2],
[word3, word4]], entity2: [[word5]]}}” ,
    Example:  “{Utterance: ...}\n{Intents: ..., Entities: ...}”
}
```

Utterance

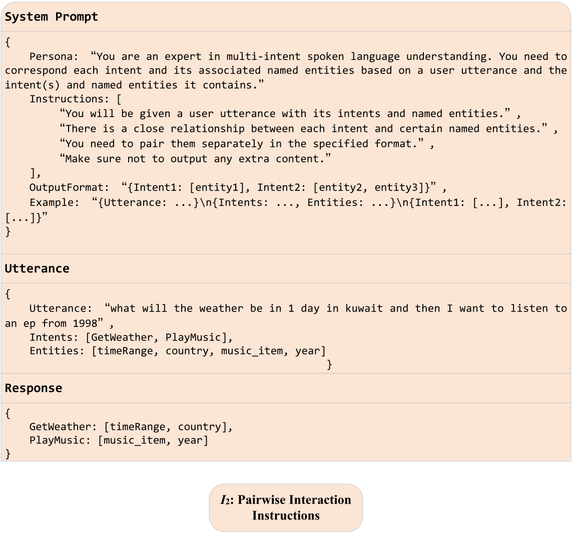
```
{
    Utterance:  “what will the weather be in 1 day in kuwait and then I want to listen to
an ep from 1998”
}
```

Response

```
{
    Intents: [GetWeather, PlayMusic],
    Entities: {
        timeRange: [[in, 1, day]],
        country: [[kuwait]],
        music_item: [[ep]],
    }   year: [[1998]]
}
```

I_1: Basic Instructions

Figure 4: Details of BI (I_1).



System Prompt

```
{
    Persona:  “You are an expert in multi-intent spoken language understanding. You need to
correspond each intent and its associated named entities based on a user utterance and the
intent(s) and named entities it contains.”
    Instructions: [
        “You will be given a user utterance with its intents and named entities.” ,
        “There is a close relationship between each intent and certain named entities.” ,
        “You need to pair them separately in the specified format.” ,
        “Make sure not to output any extra content.”
    ],
    OutputFormat:  “{Intent1: [entity1], Intent2: [entity2, entity3]}” ,
    Example:  “{Utterance: ...}\n{Intents: ..., Entities: ...}\n{Intent1: [...], Intent2:
[...]}”
}
```

Utterance

```
{
    Utterance:  “what will the weather be in 1 day in kuwait and then I want to listen to
an ep from 1998” ,
    Intents: [GetWeather, PlayMusic],
    Entities: [timeRange, country, music_item, year]
                                                    }
```

Response

```
{
    GetWeather: [timeRange, country],
    PlayMusic: [music_item, year]
}
```

I_2: Pairwise Interaction Instructions

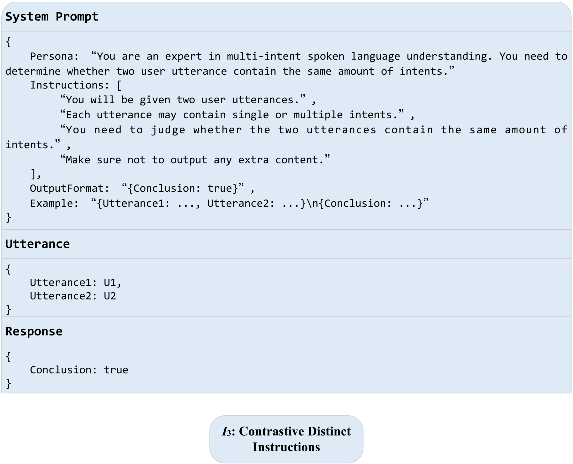
Figure 5: Details of PII (I_2).

A The Detailed Instructions

This section presents the detailed instructions for BI, PII, and CDI, as illustrated in Figs. 4, 5, and 6, respectively.

B The Prompt Used by GPT-4o

To ensure efficient annotation, we employ GPT-4o to label I_2, with the corresponding prompt illustrated in Fig. 7. First, we define GPT-4o's role and provide an example annotation. Next, we introduce a labeling technique designed to improve the quality of the annotations. Finally, we specify the output format.



System Prompt

```
{
    Persona:  “You are an expert in multi-intent spoken language understanding. You need to
determine whether two user utterance contain the same amount of intents.”
    Instructions: [
        “You will be given two user utterances.” ,
        “Each utterance may contain single or multiple intents.” ,
        “You need to judge whether the two utterances contain the same amount of
intents.” ,
        “Make sure not to output any extra content.”
    ],
    OutputFormat:  “{Conclusion: true}” ,
    Example:  “{Utterance1: ..., Utterance2: ...}\n{Conclusion: ...}”
}
```

Utterance

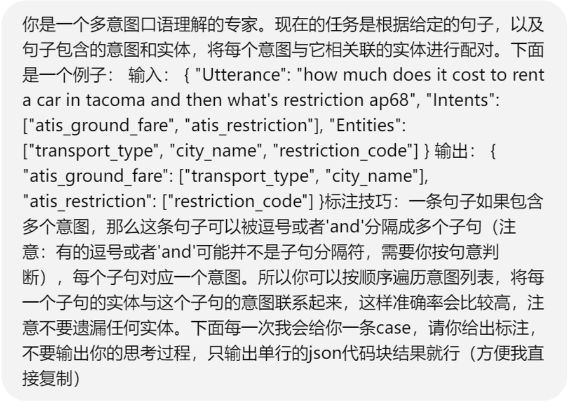
```
{
    Utterance1: U1,
    Utterance2: U2
}
```

Response

```
{
    Conclusion: true
}
```

I_3: Contrastive Distinct Instructions

Figure 6: Details of CDI (I_3).



你是一个多意图口语理解的专家。现在的任务是根据给定的句子，以及句子包含的意图和实体，将每个意图与它相关联的实体进行配对。下面是一个例子： 输入：｛"Utterance": "how much does it cost to rent a car in tacoma and then what's restriction ap68", "Intents": ["atis_ground_fare", "atis_restriction"], "Entities": ["transport_type", "city_name", "restriction_code"] } 输出：｛ "atis_ground_fare": ["transport_type", "city_name"], "atis_restriction": ["restriction_code"] }标注技巧：一条句子如果包含多个意图，那么这条句子可以被逗号或者'and'分隔成多个子句（注意：有的逗号或者'and'可能并不是子句分隔符，需要你按句意判断），每个子句对应一个意图。所以你可以按顺序遍历意图列表，将每一个子句的实体与这个子句的意图联系起来，这样准确率会比较高，注意不要遗漏任何实体。下面每一次我会给你一条case，请你给出标注，不要输出你的思考过程，只输出单行的json代码块结果就行（方便我直接复制）

Figure 7: The prompt used by GPT-4o.

C The Detailed Experimental Results for Model Size

This section presents the detailed experimental results for three parameter sizes across all few-shot settings. As shown in Table 9, performance improves with an increase in model size. Consistent with the findings in Section 5.4, performance gains for most metrics diminish as the model size continues to increase. Therefore, it is crucial to consider both model size and performance together, especially in scenarios with limited computational resources.

D Effects of Model Type

To investigate the effectiveness of different model types, we compare the recently released versions of two mainstream LLMs, LLaMA[4] and Qwen.

As shown in Table 10, the results reveal that Qwen outperforms LLaMA in terms of all metrics in most few-shot settings. A possible explanation for this performance gap lies in their foundational

[4] https://huggingface.co/meta-llama

Model	**FewShotMixATIS**														
	2-shot			4-shot			6-shot			8-shot			10-shot		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
Qwen2.5-3B	57.25	57.78	16.55	64.86	63.06	17.27	63.89	64.95	20.17	69.57	67.16	21.38	72.83	68.26	21.50
Qwen2.5-7B	69.57	65.14	18.96	70.29	69.07	21.38	72.71	72.11	24.76	78.86	73.84	27.54	81.64	74.38	28.02
Qwen2.5-14B	**71.74**	**70.04**	**23.67**	**78.38**	**70.77**	**24.76**	**78.86**	**72.14**	**25.36**	**80.92**	**75.38**	**30.68**	77.17	**76.16**	**29.71**
	FewShotMixSNIPS														
Qwen2.5-3B	73.22	36.00	3.32	74.90	40.71	4.14	79.04	41.51	4.73	81.08	45.21	6.23	82.36	46.53	7.23
Qwen2.5-7B	86.45	46.50	5.50	86.77	50.18	7.32	86.99	52.26	8.64	88.18	55.10	10.55	88.09	58.14	11.51
Qwen2.5-14B	**88.45**	**51.12**	**8.23**	**86.77**	**56.65**	**9.00**	**88.49**	**57.50**	**11.41**	**91.27**	**61.58**	**13.78**	**90.81**	**63.02**	**14.51**

Table 9: Results comparison of different model sizes on FewShotMixATIS and FewShotMixSNIPS.

Model	**FewShotMixATIS**														
	2-shot			4-shot			6-shot			8-shot			10-shot		
	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc	I-Acc	S-F1	O-Acc
w/o PII, CDI	48.03	54.21	8.06	56.28	59.08	9.84	58.21	59.96	11.72	56.64	60.81	11.35	63.41	64.41	15.22
LLaMA3.2-3B	49.52	55.66	9.30	56.64	59.68	11.23	58.21	61.55	12.08	56.76	62.98	15.10	67.63	68.92	18.96
Qwen2.5-3B	**57.25**	**57.78**	**16.55**	**64.86**	**63.06**	**17.27**	**63.89**	**64.95**	**20.17**	**69.57**	**67.16**	**21.38**	**72.83**	68.26	**21.50**
	FewShotMixSNIPS														
w/o PII, CDI	62.26	30.02	0.82	66.58	37.12	2.84	67.76	40.75	3.87	75.22	44.91	5.46	76.81	47.86	6.87
LLaMA3.2-3B	68.62	32.79	2.05	69.40	37.31	3.05	68.49	41.08	4.09	77.67	45.12	6.37	81.95	48.54	7.19
Qwen2.5-3B	**73.22**	**36.00**	**3.32**	**74.90**	**40.71**	**4.14**	**79.04**	**41.51**	**4.73**	**81.08**	**45.21**	6.23	**82.36**	46.53	**7.23**

Table 10: Results comparison of different model types on FewShotMixATIS and FewShotMixSNIPS.

Model	**FewShotMixATIS**				
	2-shot	4-shot	6-shot	8-shot	10-shot
LLaMA3.2-3B	1.33	0.97	1.33	0.36	0.24
Qwen2.5-3B	**0.24**	**0.12**	**0.24**	**0.24**	**0.24**
	FewShotMixSNIPS				
LLaMA3.2-3B	2.36	1.23	0.68	0.36	0.59
Qwen2.5-3B	**0.09**	**0.27**	**0.18**	**0.09**	**0.05**

Table 11: Error rate of *JSON* parsing on FewShotMixATIS and FewShotMixSNIPS.

capabilities. While LLaMA is primarily trained on English corpora, Qwen excels in both Chinese and English, potentially allowing it to learn more diverse language patterns during pre-training, which could benefit multi-intent SLU. Another noteworthy observation is the disparity in their *JSON* output format capabilities. As shown in Table 11, Qwen exhibits superior *JSON* output capabilities compared to LLaMA, likely due to its tailored post-training process for generating structured outputs as ducumented in the official source[5]. Specifically, LLMs frequently generate content such as *"Cutting Knowledge Date: December 2023 Today Date: ..."*, where the ellipsis represents the original input, often resulting in errors during *JSON* parsing. Despite inferior performances of LLaMA, it still outperforms the strong baseline model BERT-SIF, which demonstrates the effectiveness of our proposed instructions in few-shot multi-intent SLU. Notably, removing PII and CDI for LLaMA results in significant performance declines across all metrics. In summary, this analysis underscores the critical importance of model selection, particularly with respect to capabilities relevant to the task at hand.

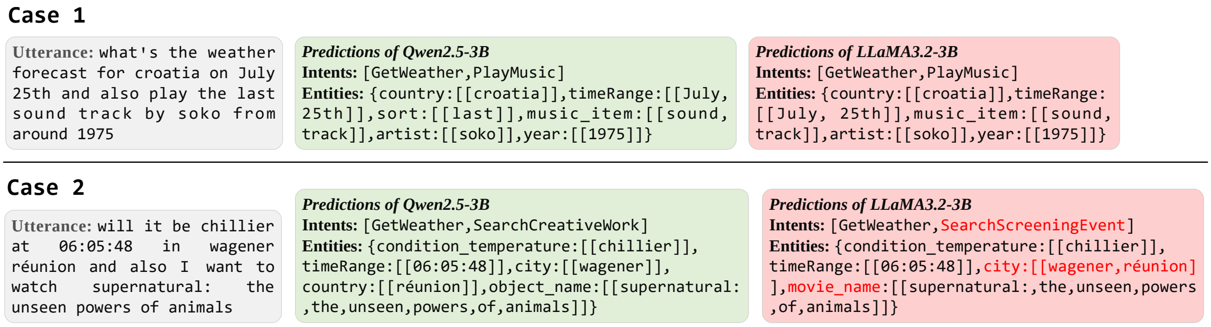
E Case Study of Model Type

This section presents two case studies to further examine the effectiveness of different model types. A detailed illustration is provided in Fig. 8.

In case 1, both Qwen and LLaMA successfully detect all intents; however, LLaMA fails to predict the slot for *"last"*. This indicates that while LLaMA performs well in intent detection, it struggles with modeling fine-grained semantic details, particularly in interpreting the semantically implied word *"last"*. The word *"last"* is highly functional and context-dependent. However, LLaMA may not have effectively learned or modeled its role within specific contexts. This suggests that LLaMA's generalization ability may be somewhat limited, particularly in predicting abstract functional slots associated with non-entity words.

In case 2, LLaMA cannot identify *"SearchCreativeWork"* intent and outputs a wrong intent *"SearchScreeningEvent"*, while Qwen can give the correct prediction. LLaMA's incorrect intent prediction directly results in misclassifying *"supernatural: the unseen powers of animals"* as a *"movie_name"*. Moreover, LLaMA incorrectly identifies *"réunion"* as a city. It suggests that

[5] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct



Case 1

Utterance: `what's the weather forecast for croatia on July 25th and also play the last sound track by soko from around 1975`

Predictions of Qwen2.5-3B
Intents: `[GetWeather,PlayMusic]`
Entities: `{country:[[croatia]],timeRange:[[July,25th]],sort:[[last]],music_item:[[sound,track]],artist:[[soko]],year:[[1975]]}`

Predictions of LLaMA3.2-3B
Intents: `[GetWeather,PlayMusic]`
Entities: `{country:[[croatia]],timeRange:[[July, 25th]],music_item:[[sound,track]],artist:[[soko]],year:[[1975]]}`

Case 2

Utterance: `will it be chillier at 06:05:48 in wagener réunion and also I want to watch supernatural: the unseen powers of animals`

Predictions of Qwen2.5-3B
Intents: `[GetWeather,SearchCreativeWork]`
Entities: `{condition_temperature:[[chillier]],timeRange:[[06:05:48]],city:[[wagener]],country:[[réunion]],object_name:[[supernatural:,the,unseen,powers,of,animals]]}`

Predictions of LLaMA3.2-3B
Intents: `[GetWeather,SearchScreeningEvent]`
Entities: `{condition_temperature:[[chillier]],timeRange:[[06:05:48]],city:[[wagener,réunion]],movie_name:[[supernatural:,the,unseen,powers,of,animals]]}`

Figure 8: Illustrative case studies comparing Qwen2.5-3B and LLaMA3.2-3B predictions.

LLaMA exhibits a shallow understanding of the phrase *"I want to watch"* in the utterance, tending to associate it with movie screening events rather than with abstract content search. In contrast, Qwen accurately interprets *"supernatural: the unseen powers of animals"* as the title of a work, correctly associating it with the content rather than screening-related information, demonstrating a stronger contextual understanding. Furthermore, Qwen demonstrates more accurate entity classification, particularly with respect to geographical locations.