DC-Instruct: An Effective Framework for Generative Multi-intent Spoken Language Understanding

Bowen Xing[1]*, **Lizi Liao**[2], **Minlie Huang**[3] **and Ivor W. Tsang**[45]

[1]Beijing Key Laboratory of Knowledge Engineering for Materials Science, School of Computer and Communication Engineering, University of Science and Technology Beijing

[2]Singapore Management University

[3]The Conversational Artificial Intelligence (CoAI) Group, Tsinghua University

[4]CFAR and IHPC, Agency for Science, Technology and Research, Singapore

[5]College of Computing and Data Science, Nanyang Technological University

Abstract

In the realm of multi-intent spoken language understanding, recent advancements have leveraged the potential of prompt learning frameworks. However, critical gaps exist in these frameworks: the lack of explicit modeling of dual-task dependencies and the oversight of task-specific semantic differences among utterances. To address these shortcomings, we propose DC-Instruct, a novel generative framework based on Dual-task Inter-dependent Instructions (DII) and Supervised Contrastive Instructions (SCI). Specifically, DII guides large language models (LLMs) to generate labels for one task based on the other task's labels, thereby explicitly capturing dual-task inter-dependencies. Moreover, SCI leverages utterance semantics differences by guiding LLMs to determine whether a pair of utterances share the same or similar labels. This can improve LLMs on extracting and discriminating task-specific semantics, thus enhancing their SLU reasoning abilities. Extensive experiments on public benchmark datasets show that DC-Instruct markedly outperforms current generative models and state-of-the-art methods, demonstrating its effectiveness in enhancing dialogue language understanding and reasoning.

1 Introduction

In dialogue systems, spoken language understanding (SLU) (Young et al., 2013) is a fundamental component for comprehensively understanding users' queries. In recent developments, multi-intent SLU (Kim et al., 2017) has garnered significant attention for its various and practical application scenarios. It typically includes two subtasks: multiple intent detection and slot filling. Multiple Intent detection aims to identify the intents expressed in the utterance, while slot filling extracts specific pieces of semantics information from the utterance. Some examples are shown in Fig. 1.

Figure 1: Some samples from MixATIS dataset. Intent labels are in blue, and slot labels are in green.

Since there exist inherent inter-dependencies between intents and slots, recent models widely adopt the multi-task framework based on a shared semantics encoder and model the dual-task interactions through some specialized components (Gangadharaiah and Narayanaswamy, 2019; Goo et al., 2018; Liu et al., 2019; Qin et al., 2020, 2021; Xing and Tsang, 2022b,a). Among them, Co-guiding Net Xing and Tsang (2022a) achieves the mutual guidance between the two tasks via heterogeneous graph attention networks. These models show potential, but their specialized components limit their generalization ability. To this end, the prompt learning paradigm is integrated, and Wu et al. (2022) propose a unified generative framework (UGEN), which includes five kinds of templates in the question-answer formulation.

Nonetheless, we discover that up-to-date multi-intent SLU methods still suffer from two issues. Firstly, current prompt instructions fail to effectively model the inter-dependencies between multiple intent detection and slot filling. The five instructions (I_1-I_5) in UGEN tackle the two tasks separately: I_1 targets multiple intent detection, while I_2-I_5 focus on slot filling. We propose that explicitly modeling the dual-task inter-dependencies

*bwxing714@gmail.com

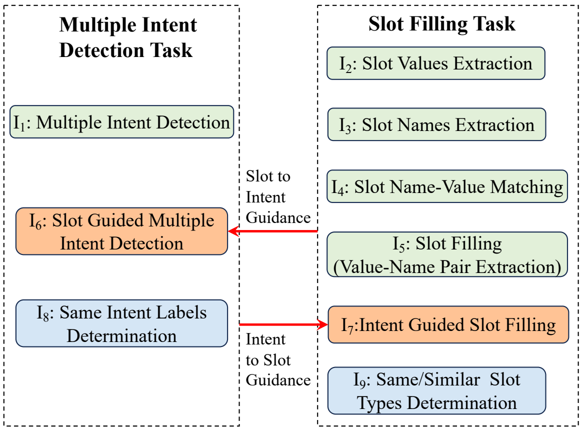

Figure 2: Overall illustration of our DC-Instruct. I_1-I_5 are basic instructions. I_6 and I_7 are dual-task inter-dependent instructions. I_8 and I_9 are supervised contrastive instructions.

within the prompt learning framework could significantly enhance the reasoning ability of large language models (LLMs). Secondly, there is an oversight of the semantic variations among utterances. In Fig. 1, there exist specific semantics differences among the three utterances, which can be reflected by their labels. Utterance A and B have the same ***intent*** labels, while Utterance C has different ***intent*** labels with them. Utterance B and C have similar ***slot*** labels, while Utterance A has quite different ***slot*** labels with them. We argue that these task-specific contrastive relations can benefit LLMs on understanding task-specific semantics while existing methods ignore them.

To resolve the above challenges, in this work, we introduce a novel generative model termed DC-Instruct. We propose Dual-task Inter-dependent Instructions (DII) and Supervised Contrastive Instructions (SCI) to adeptly model dual-task inter-dependencies and exploit task-specific semantic differences within the prompt learning framework. DII introduces two auxiliary tasks: slot-guided multiple intent detection and intent-guided slot filling, integrating inter-dependent instructions by embedding one task's golden labels into the instructional context of the other. This enables LLMs to conditionally generate task-specific labels, effectively capturing dual-task dependencies and alignments. To address the challenge of exploiting utterance contrastive relations, SCI introduces an auxiliary task for determining whether a pair of utterances share the same intents or similar slot types. SCI guides LLMs to discern True/False outcomes regarding the task-specific semantics of both the anchor utterance and its corresponding positive/negative examples. In this way, SCI can enhance LLMs' ability to distinguish and align task-specific semantics for improving SLU reasoning.

Taking the public benchmark datasets as test beds, we conduct extensive experiments based on various LLMs scaling from 220M to 13B. The experimental results show that our models can achieve consistent and significant improvements over state-of-the-art models. The ablation study and experiments in different low-resource settings further verify our method's advantages.

Our major contributions are three-fold:
(1) We propose DC-Instruct, a novel generative model based on dual-task inter-dependent instructions and supervised contrastive instructions.
(2) We make the first attempt to explicitly capture dual-task dependencies and exploit utterance contrastive relations in the prompt learning paradigm.
(3) Experimental results demonstrate the superiority of our method, which can achieve new state-of-the-art performances.

2 Related Works

Multi-intent SLU A group of models (Zhang and Wang, 2016; Goo et al., 2018; Li et al., 2018; E et al., 2019; Liu et al., 2019; Qin et al., 2019; Chen et al., 2019; Zhang et al., 2019; Wu et al., 2020) have been proposed to jointly tackle the two tasks in SLU and model their interactions. However, these models can only handle single-task scenarios, while there are usually multi-intent utterances in real-world scenarios. To this end, (Kim et al., 2017) propose a multi-intent SLU model, and (Gangadharaiah and Narayanaswamy, 2019) jointly model the tasks of multiple intent detection and slot filling via a slot-gate mechanism. To effectively model the dual-task interactions, graph neural networks have been widely utilized (Qin et al., 2020, 2021; Xing and Tsang, 2022a,b; Song et al., 2022). Co-guiding Net (Xing and Tsang, 2022a) makes the first attempt to model the mutual guidances between multiple intent detection and slot filling in the heterogeneous semantics-label graphs. Rela-Net (Xing and Tsang, 2022b) and LCLR (Zhu et al., 2023) propose to leverage the dual-task correlations in the decoding process. More recently, prompt learning has been investigated for multi-intent SLU. UGEN (Wu et al., 2022) performs multi-intent SLU in a unified generative framework using five kinds of question-answer-formed instructions.

Different from the above works, we propose a novel generative method that includes various instructions to explicitly model the dual-task dependencies and effectively leverage the contrastive relations among utterances.

Prompt Learning Recently, prompt learning has been attracting increasing attention since it achieves promising performance in various NLP tasks (Liu et al., 2023a,b; Wu et al., 2022; Chan et al., 2023; Shen et al., 2023). This paradigm can unify the pre-training and fine-tuning stages into the text-to-text generation tasks. For example, Shen et al. (2023) propose a a dual-slot multi-prompt template to unify entity locating and entity typing. Chan et al. (2023) propose a path prediction method based on prompt learning to incorporate the hierarchy structure.

In this work, we investigate prompt learning for multi-intent SLU and propose a novel model distinguished by dual-task inter-dependent instructions and supervised contrastive instructions.

3 Preliminary

Multi-intent SLU aims to predict all the intents expressed in the utterance and the slot label corresponding to each word. Traditional framework regards multiple intent detection as a multi-label sentence classification task and regards slot filling as a sequence labeling task. UGEN (Wu et al., 2022) makes the first attempt to explore generative multi-intent SLU based on the prompt learning paradigm. In generative multi-intent SLU, the output of the multiple intent detection task is a sequence of intents expressed in the utterance. The output of the slot filling task is a sequence of slot value-name pairs. A slot value is a word or phrase expressing specific semantics and the slot name is the slot type or label corresponding to the slot value. Considering the first example in Fig. 1, there are two slot value-name pairs: [`cheapest`, `cost relative`] and [`general mitchell international`, `airport name`].

4 Method

In this section, we introduce our proposed DC-Instruct framework, as shown in Fig. 3. Following (Wu et al., 2022), we formulate our instructions in the question-answer (QA) form. There are total five types of instructions in UGEN to tackle the two tasks separately, while they cannot capture the dual-task dependencies nor contrastive relations. Our framework also includes these five instructions, and they are referred to as basic instructions. We first briefly introduce the basic instructions ($I_1, ..., I_5$) and then depict our proposed dual-task inter-dependent instructions (I_6, I_7) and supervised contrastive instructions (I_8, I_9).

4.1 Basic Instructions

The first basic instruction (I_1) is to guide the model to predict the intents expressed in the utterance. The other four basic instructions (I_2 I_5) are for slot filling. I_2 aims to guide LLMs to extract the slot values in the utterance. I_3 guides LLMs to assign slot names to the provided slot values. I_4 is a slot value-name matching task associating the correct slot name with the specific slot value. I_5 is the slot value-name pair extraction task, which guides the model to generate the sequence of all slot value-name pairs. In the inference process, only I_1 and I_5 are used to generate multi-intent SLU predictions.

4.2 Dual-task Inter-dependent Instructions

To explicitly model the dual-task dependencies in the prompt learning paradigm, we propose the dual-task inter-dependent instructions, whose formulation is shown in Fig. 4. In the training stage, dual-task dependencies are captured by achieving three kinds of alignments. First, since the instruction guides the LLM to predict task A's labels, the semantics-label alignment between the utterance context and task A's labels can be achieved. Second, the dual-task label alignment between task B's labels in the prompt template and task A's labels in the generation side is modeled. Third, in the prompt template, both the utterance semantics and task B's labels are provided, thus the dual-task semantics-label alignment between them and task A's labels is achieved.

4.2.1 Slot-guided Multiple Intent Detection

In this instruction (I_6), all slot types included in the utterance are provided in the instruction to guide the multiple intent detection task. Considering the first example in Fig. 1, its instruction of slot-guided multiple intent detection is:

> `Utterance:` show me the cheapest fare ... then where is general mithchell international located. `This utterance includes these slot types:` cost relative , airport name. `What are the intents of the utterance according to options? Options: {Intent Label Set}.`

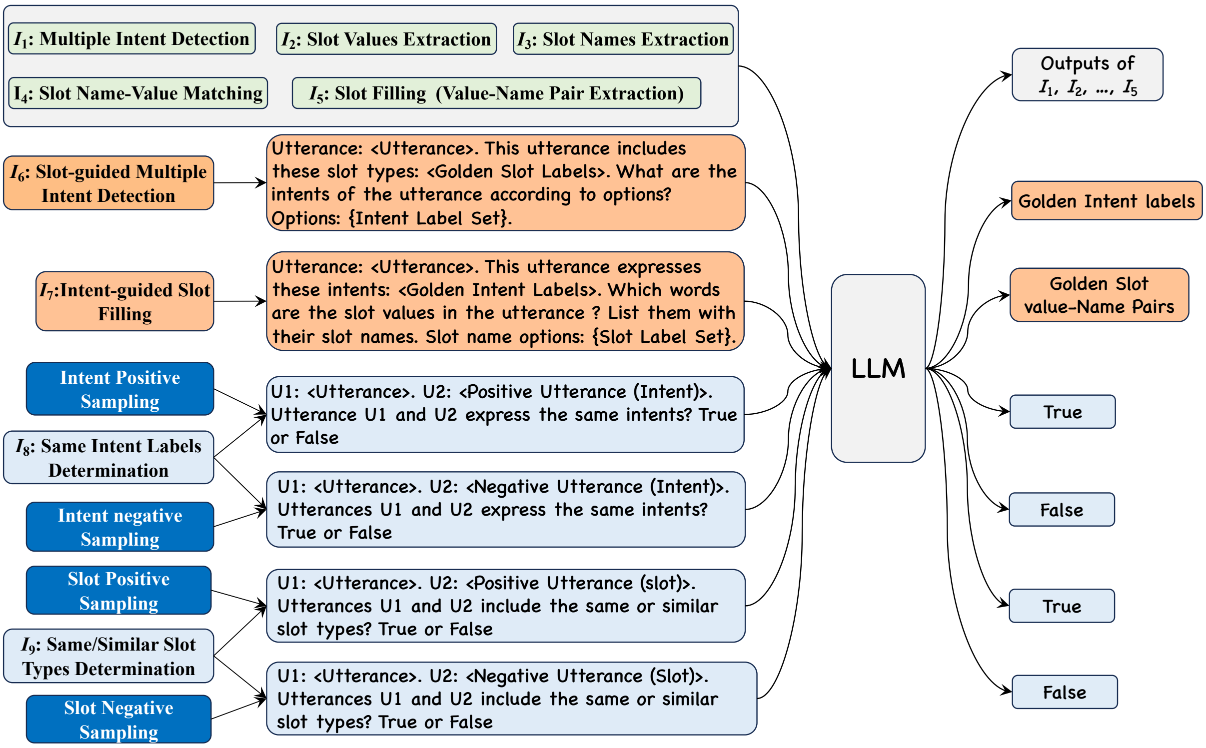

Figure 3: Illustration of our framework. Due to space limitation, we omit the details of I_1-I_5. We show some examples of detailed instructions in Appendix (Table 5 and Table 6).

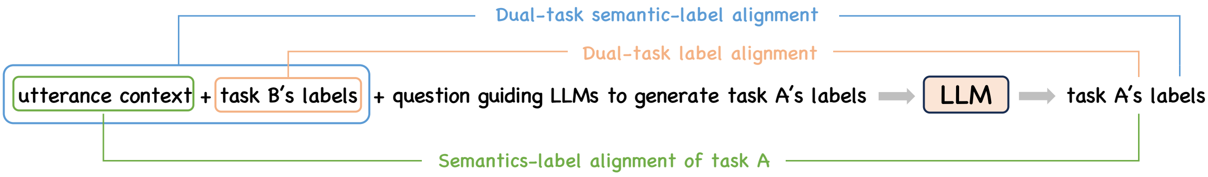

Figure 4: Illustration of our proposed dual-task inter-dependent instructions. The left side of the LLM is the input instruction, and the right side of the LLM is the generated sequence of label(s). Task A denotes the current task and task B denotes the other task.

where the utterance context is in blue and the slot types are in green.

The golden output is: *cheapest, city*.

4.2.2 Intent-guided Slot Filling

In this instruction (I_7), the golden intent labels are provided in the instruction to guide the slot filling task. Considering the first example in Fig. 1, its instruction of intent-guided slot filling is:

```
Utterance: show me the cheapest fare ... then
where is general mithchell international located. This
utterance expresses these intents: cheap-
est , city. Which words are the slot values
in the utterance? List them with
their slot names. Slot name options:
{Slot Label Set}.
```

where the utterance context is in blue and the intent labels are in pink.

And the golden generation is: *cheapest* `is` *a cost relative* ; *general mitchell international* `is` *a airport name*. The underlined *words* at the left side of '`is a`' are the slot values, and the ones on the right side are the corresponding slot names.

4.3 Supervised Contrastive Instructions

Previous works ignore the semantics differences among the samples, which are reflected in the different labels. This kind of contrastive relations can be leveraged to perform supervised contrastive learning (SCL), enhancing the semantics understanding ability and further improving reasoning. As shown in Fig. 5 (a), traditional SCL leverages the supervision signal from the contrastive labels to pull together the representations corresponding to the same label while pushing apart the representations corresponding to different labels. However, our generative model is based on the prompt learning paradigm, which cannot operate on the

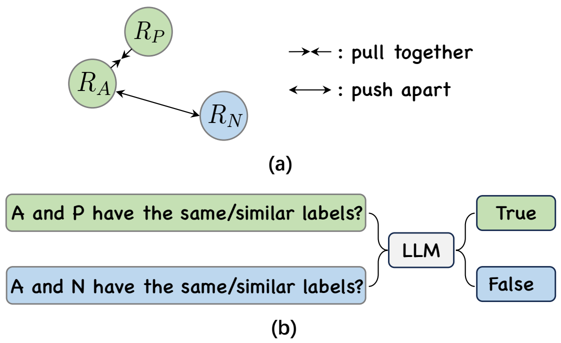

Figure 5: Comparsion of traditional SCL based on representation learning and our proposed SCI based on prompt learning. A, P and N denote the anchor, positive sample and negative sample, respectively. R_A, R_P and R_N denote the representation of A, P and N.

representations. To this end, we propose a set of simple while effective instructions to achieve SCL in the prompt learning paradigm, as shown in Fig. 5 (b). We first sample a negative utterance N and a positive utterance P regarding the anchor utterance A. Then we construct the instructions using natural language to ask the LLM whether A and P or A and N have the same/similar intent/slot labels. The corresponding golden output is "True" or "False". By this means, we can leverage the contrastive relations to improve generative LLMs on task-specific semantics understanding and reasoning.

4.3.1 Same Intent Labels Determination

To integrate intent SCL, we design a set of instructions (I_8) performing *same intent labels determination*. They teach the LLM to align the intent semantics of utterances expressing the same intent labels and discriminate the intent semantics of utterances expressing different intent labels. For an anchor utterance, the intent positive samples are the ones having the same intent labels as the anchor, and the other ones are the intent negative samples. Considering the first utterance in Fig. 1 as the anchor, the second utterance is its intent positive sample, and the third utterance is its negative sample. In this case, I_8 can be constructed as:

`U1:` show me the cheapest fare ... then where is general mithchell international located . `U2:` show me the cheapest fare ... from boston to dallas earlier than 1017 in the morning . `Utterance U1 and U2 express the same` **`intents`**`? True or False`

`U1:` show me the cheapest fare ... then where is general mithchell international located. `U2:` repeating leaving denver to san francisco before 10 am ... flight number from toronto to st. petersburg. `Utterance U1 and U2 express the same` **`intents`**`? True or False`

where the anchor utterance is in blue, the intent positive sample is in green and the negative sample is in red. The word '**`intents`**' can guide the LLM to extract the high-level intent semantics of the samples for determination. The corresponding golden outputs are "True" and "False", respectively.

4.3.2 Same/Similar Slot Types Determination

Similarly, we design *same/similar slot types determination* for slot SCL. This set of instructions (I_9) aims to teach the LLM to align the slot semantics of utterances that include the same/similar slot labels and discriminate the slot semantics of utterances that include different slot labels. The positive or negative slot samples are defined based on the slot type set similarities, which are calculated by: $S(a,b) = \frac{\mathrm{len}\left(\mathrm{overlap}(L_a^s, L_b^s)\right)}{\max\left(\mathrm{len}(L_a^s), \mathrm{len}(L_b^s)\right)}$. L_a^s is the set of all slot types included in the anchor and $\mathrm{overlap}(L_a^s, L_b^s)$ denotes the set of overlap slot types of the anchor and sample b. If $S(a,b) \geq 1-\mu$, sample b is regarded as a slot positive sample; if $S(a,b) \leq \mu$, sample b is regarded as a slot negative sample. μ is the threshold[1].

In Fig. 1, considering the second utterance as the anchor, the first utterance is its slot negative sample, and the third utterance is its slot positive sample. In this case, we can construct I_9 as:

`U1:` show me the cheapest fare ... from boston to dallas earlier than 1017 in the morning. `U2:` show me the cheapest fare ... then where is general mithchell international located. `Utterance U1 and U2 express the same or similar` **`slot types`**`? True or False`

`U1:` show me the cheapest fare ... from boston to dallas earlier than 1017 in the morning . `U2:` repeating leaving denver to san francisco before 10 am ... flight number from toronto to st. petersburg. `Utterance U1 and U2 express the same or similar` **`slot types`**`? True or False`

where the anchor utterance is in blue, the slot positive sample is in green and the slot negative sample is in red. The phrase '**`slot types`**' can guide the LLM to extract the high-level slot semantics of the samples for determination. The golden outputs are "True" and "False", respectively.

4.4 Training and Inference

Training We first construct all instructions of $I_1 \sim I_9$. Then we randomly select α ratio of the instructions of I_6, I_7, I_8 and I_9 and merge them with all instructions of $I_1 \sim I_5$, forming

[1] In this work we use $\mu = 1/3$. We also try 1/4 and 1/5, while no significant performance gap is observed.

	Models	MixATIS			MixSNIPS		
		Overall(Acc)	Slot (F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)
Classification-based Models	Attention BiRNN (Liu and Lane, 2016)	39.1	86.4	74.6	59.5	89.4	95.4
	Slot-Gated (Goo et al., 2018)	35.5	87.7	63.9	55.4	87.9	94.6
	Bi-Model (Wang et al., 2018)	34.4	83.9	70.3	63.4	90.7	95.6
	SF-ID (E et al., 2019)	34.9	87.4	66.2	59.9	90.6	95.0
	Stack-Propagation (Qin et al., 2019)	40.1	87.8	72.1	72.9	94.2	96.0
	JMID-SF (Gangadharaiah and Narayanaswamy, 2019)	36.1	84.6	73.4	62.9	90.6	95.1
	AGIF (Qin et al., 2020)	40.8	86.7	74.4	74.2	94.2	95.1
	GL-GIN (Qin et al., 2021)	43.5	88.3	76.3	75.4	94.9	95.6
	GISC (Song et al., 2022)	48.2	88.5	75.0	75.9	95.0	95.5
	Co-guiding Net (Xing and Tsang, 2022a)	51.3	89.8	79.1	77.5	95.1	97.7
	ReLa-Net (Xing and Tsang, 2022b)	52.2	90.1	78.5	76.1	94.7	97.6
	Co-guiding Net+LCLR (Zhu et al., 2023)	52.0	90.2	79.4	78.1	95.5	**98.1**
	DARER^2 (Xing and Tsang, 2023)	49.0	89.2	77.3	76.3	94.9	96.7
	GL-GIN* (RoBERTa-base, full fine-tuning, TP=125M+)	50.1	86.9	80.8	82.6	96.4	97.3
	Go-guiding* (RoBERTa-base, full fine-tuning, TP=125M+)	54.3	88.4	83.2	83.9	**97.6**	**98.1**
	DARER^{2*} (RoBERTa-base, full fine-tuning, TP=125M+)	53.8	88.2	83.1	83.5	97.3	97.9
Generative Models	UGEN* (T5-base, full fine-tuning, TP=220M) (Wu et al., 2022)	55.4	89.1	83.1	79.1	94.8	96.8
	UGEN* (T5-large, full fine-tuning, TP=770M)	57.9	89.6	84.2	81.0	95.8	97.2
	UGEN* (LLama2-7B, LoRA fine-tuning, TP=17M)	62.4	90.1	94.2	82.0	96.4	96.1
	UGEN* (LLama2-13B, LoRA fine-tuning, TP=26M)	64.3	90.3	96.1	84.0	96.7	96.3
	ChatGPT (gpt-3.5-turbo 175B, https://chat.openai.com/)	1.9	34.5	22.1	1.4	30.0	67.5
	DC-Instruct† (T5-base, full fine-tuning, TP=220M)	58.1	90.4	84.4	81.2	95.7	97.6
	DC-Instruct† (T5-large, full fine-tuning, TP=770M)	60.5	90.7	84.9	83.9	96.4	97.8
	DC-Instruct† (LLama2-7B, LoRA fine-tuning, TP=17M)	65.0	90.7	94.6	84.0	96.7	96.3
	DC-Instruct† (LLama2-13B, LoRA fine-tuning, TP=26M)	**66.7**	**91.7**	**96.9**	**84.6**	96.9	97.1

Table 1: Results comparison. * denotes we implement the model using the official code. † denotes DC-Instruct models significantly outperform UGEN counterparts (p < 0.01 under t-test). TP denotes the trainable parameter size.

the training data. We use the shuffled training data to train the model in the text-to-text generation form. The training objective is to minimize the negative log-likelihood for each instruction: $\mathcal{L} = -\sum_{n=1}^{N} \log p\left(y_n \mid y_{<n}, I\right)$. 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 and I_5 are used to generate the predictions for multiple intent detection and slot filling, respectively.

5 Experiments

5.1 Main Results

Due to space limitation, we put experiment settings in Appendix A. The performance comparison of our model and baselines are shown in Table 1, from which we have the following observations:

(1) ***Our model achieves new state-of-the-art performance on all tasks and datasets.*** Specifically, on MixATIS dataset, DC-Instruct (T5-base) overpasses UGEN (T5-base) by 2.7%, 1.3%, and 1.3% on overall accuracy, slot F1, and intent accuracy, respectively; on MixSNIPS dataset, it overpasses UGEN by 2.1%, 0.9% and 0.8% on overall accuracy, slot F1, and intent accuracy. This is because our model explicitly captures dual-task dependencies via dual-task inter-dependent instructions, and our designed supervised contrastive instructions further enhance the LLM's ability on task-specific semantics understanding. Besides, T5-based models perform worse than RoBERTa-based models on MixSNIPS dataset. We suspect the reason is that MixSNIPS has much more training samples while much fewer labels, which makes it easier for classification-based models to precisely choose the correct label index from the limited label space.

(2) ***Based on up-to-date larger generative LLMs (e.g., LLama2), our DC-Instruct model can still achieve significant improvements***. The reason is that the advantages of our approach are orthogonal to the ability of LLMs. Our method can teach LLMs to capture dual-task dependencies and extract task-specific semantics, which can hardly be learned in the pre-training process.

(3) ***ChatGPT can hardly handle multi-intent SLU,*** *consistent with the recent observations (Pan et al., 2023; Qin et al., 2023).* We suspect the reason is that this task requires task-specific knowledge, which is better captured in the fine-tuning process. Besides, the schema of intent and slot labels is complex. We believe advanced in-context-learning strategies like chain-of-thought can improve ChatGPT to some extent, while this is not our focus in this paper. Since ChatGPT cannot obtain promising results on multi-intent SLU, prompt tuning is necessary for LLMs. We give the error analysis in Sec. 5.5 and some error cases in Appendix.

MixATIS	1-shot (TSN=571)			5-shot (TSN=2707)			5% (TSN=658)			10% (TSN=1316)		
	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)
Co-guiding Net (RoBERTa)	36.5	81.2	73.6	51.9	86.9	79.6	44.3	82.5	78.0	46.7	86.5	76.7
DARER^2 (RoBERTa)	36.1	80.8	73.7	51.6	84.7	79.2	43.1	82.1	78.2	46.5	86.9	75.8
UGEN (T5)	42.8	85.3	78.6	53.1	88.5	81.8	47.0	85.5	80.9	50.4	87.3	81.4
DC-Instruct (T5)	45.9	86.9	80.6	55.0	89.7	82.9	49.6	86.6	82.1	52.4	88.5	82.7

MixSNIPS	5-shot (TSN=416)			10-shot (TSN=708)			5% (TSN=1988)			10% (TSN=3977)		
	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)
Co-guiding Net (RoBERTa)	42.1	82.5	89.5	54.1	87.8	91.4	69.9	92.7	94.9	76.8	94.7	96.7
DARER^2 (RoBERTa)	42.2	82.4	89.6	53.9	87.8	91.2	69.5	92.3	94.8	75.6	94.3	96.5
UGEN (T5)	43.3	85.0	92.5	52.0	88.0	93.5	68.7	92.4	96.5	72.8	93.9	96.7
DC-Instruct (T5)	46.7	86.2	93.0	54.8	89.0	94.5	70.6	93.5	96.9	74.4	94.5	97.4

Table 2: Experiment results on different low-resource settings. TSN denotes the number of training samples.

Models	MixATIS			MixSNIPS		
	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)
DC-Instruct	58.1	90.4	84.4	81.2	95.7	97.6
w/o DII (I_6, I_7)	56.9	89.6	83.6	80.3	95.3	97.2
w/o SgMID (I_6)	57.6	90.3	83.8	80.7	95.6	97.3
w/o IgSF (I_7)	57.8	89.9	84.3	80.8	95.4	97.5
w/o SCI (I_8, I_9)	57.3	89.8	83.8	80.7	95.3	97.2
w/o Intent-SCI (I_8)	57.6	90.3	84.0	80.9	95.6	97.3
w/o Slot-SCI (I_9)	57.7	90.0	84.3	81.0	95.2	97.5

Table 3: Results of ablation experiments.

5.2 Ablation Study

We conduct ablation experiments to study the effect of each component of our DC-Instruct model and the results are shown in Table 3.

Dual-task Inter-dependent Instructions (DII). When removing DII (I_6, I_7), obvious drops can be witnessed on all metrics, especially overall accuracy. This proves that DII can effectively and explicitly model the dual-task dependencies, which significantly improves the performance. We can also find that removing any one of the slot-guided multiple intent detection instruction (SgMID, I_6) and intent-guided slot filling instruction (IgSF, I_7) not only causes the own task's performance drops but also leads to drops on overall accuracy and the other task's performance. This can further verify the fact that DII can effectively align the two tasks and make them deeply coupled.

Supervised Contrastive Instructions (SCI). The aim of SCI is to enhance the LLM's ability on task-specific semantics understanding. We can find that removing SCI leads to significant decreases in all metrics, verifying its necessity. Besides, removing Intent-SCI harms multiple intent detection and causes performance decreases in slot filling and sentence-level semantics parsing simultaneously. Similarly, removing Slot-SCI leads to performance decreases not only in slot F1 but also in intent accuracy and overall accuracy. This can be attributed to two facts. First, Intent-SCI and Slot-SCI can effectively improve the performances on their own tasks. Second, our proposed DII makes the two tasks deeply coupled and interrelated with each other's performances. Therefore, removing any one of Intent-SCI and Slot-SCI leads to performance decreases on all of overall accuracy, slot F1 and intent accuracy.

5.3 Experiments on Low-resource Setting

In real-world scenarios, obtaining a large number of golden-labeled SLU samples is usually expensive and difficult. Therefore, we conducted experiments on 1/5/10-shot and 5%/10%-ratio settings to simulate the low-resource setting and study the quick adaptation ability of our model. The implementation details are shown in Appendix B and the experiment results are shown in Table 2. From the results, we can observe that:

(1) UGEN and our DC-Instruct model outperform other baselines by a large margin on MixATIS dataset. This is because the prompt learning paradigm has a strong ability for generalization and it unifies the decoding process of the two tasks, which is beneficial for capturing dual-task dependencies. Our model can further achieve significant and consistent improvement over UGEN under all low-resource settings on all metrics. This can be attributed to the fact that our proposed dual-task inter-dependent instructions and supervised contrastive instructions can effectively distill more beneficial dual-task correlative knowledge and task semantics knowledge from the limited training data.

(2) On MixSNIPS dataset, as the training sample number increases, the performance gap between

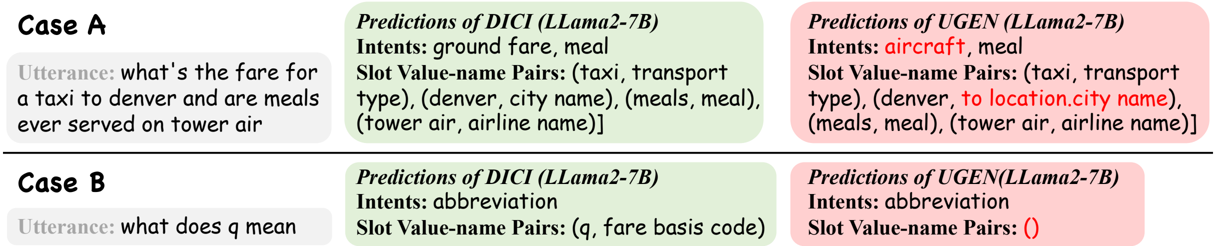

Figure 6: Illustration of two cases with predictions from DC-Instruct (LLama2-7B) and UGEN (LLama2-7B).

RoBERTa-based models and T5-based models decreases, and finally, RoBERTa-based models outperform T5-based models. We suspect the reason is that MixSNIPS has much fewer labels, which makes it easy for classification-based models to predict the correct label index.

MixATIS	MID			SF	
	Fewer	Eq. No.	More	×Value	×Name
ChatGPT	4.1	24.2	49.0	91.4	92.9
UGEN(LLama2-7B)	0.6	5.2	0.0	37.8	37.8
DC-Instruct(LLaama2-7B)	0.2	5.1	0.0	34.1	34.1
MixSNIPS	**MID**			**SF**	
	Fewer	Eq. No.	More	×Value	×Name
ChatGPT	11.4	20.9	8.8	97.4	98.1
UGEN(LLama2-7B)	0.2	3.7	0.0	17.1	17.1
DC-Instruct(LLaama2-7B)	0.1	3.6	0.0	14.9	14.9

Table 4: Results of error analysis.

5.4 Case Study

To demonstrate the superiority of our DC-Instruct model over the state-of-the-art generative model UGEN, we present two cases in Fig. 6.

In case A, UGEN cannot identify '`ground fare`' intent and outputs a wrong intent '`aircraft`', while our model can give the correct prediction. This is because our proposed SgMID instruction (I_6) can guide the LLM to comprehensively consider utterance semantics of '`fare for a taxi`' and the slot semantics of '`transport type`' for intent prediction. Besides, our proposed Intent-SCI (I_8) can enhance the LLM to extract and discriminate the intent-related semantics. UGEN also makes a mistake on the slot name of '`denver`'. In MixATIS dataset, '`to location.city name`' only relates to the flight destination. Our DC-Instruct model can correctly predict because the subtle semantics difference between '`to location.city name`' and '`city name`' can be captured by our proposed Slot-SCI.

In case B, although UGEN can correctly predict intent '`abbreviation`', it cannot extract the slot value-name pair `(q, fare basis code)`. Thanks to our proposed IgSF instruction (I_7), DC-Instruct can correctly extract the slot with the awareness that there exists at least one abbreviation in the utterance. Besides, our proposed Slot-SCI can help identify the correct slot name by enhancing the LLM to extract and discriminate the slot-related semantics.

5.5 Error Analysis

We count and categorize errors made by ChatGPT, UGEN (LLama2-7B), and our DC-Instruct (LLama2-7B). The results are listed in Table 4. Due to space limitation, we give the definitions of different kinds of errors in Appendix F.

ChatGPT tends to assign redundant wrong intents on the MixATIS dataset. We suspect the reason is that the MixATIS dataset has more intent labels whose semantics is hard to discriminate for ChatGPT. Besides, ChatGPT can hardly predict all correct slots for an utterance. It usually makes mistakes on the span of the slot value and cannot discriminate the semantics of slot names. Designing advanced in-context-learning methods tailored for the above errors may improve ChatGPT on multi-intent SLU. We present some error cases of ChatGPT in Appendix (Table 7).

Compared with UGEN, DC-Instruct makes fewer errors on both MID and SF tasks. Especially, DC-Instruct can correctly predict all slot value-name pairs for more utterances than UGEN.

6 Conclusion

In this paper, we propose DC-Instruct, addressing the challenges in generative multi-intent SLU from two perspectives. Firstly, we propose dual-task inter-dependent instructions to explicitly model the dual-task dependencies. Secondly, we propose supervised contrastive instructions, which exploit the utterance contrastive relations in the prompt learning paradigm. Extensive evaluations on benchmarks demonstrate the superiority of our method, which can achieve promising improvements over various LLMs scaling from 220M to 13B.

Limitations

Despite the promising results of DC-Instruct for multi-intent SLU, we suppose that DC-Instruct has two limitations: (1) **new intents and slots detection**. Currently, the application of our model is limited to identifying known intents and slots. In real-world scenarios, detecting new intents and slots is an important and challenging task. In the future, we can investigate to enhance our model on detecting unknown intents and slots. (2) **new intent and slot label generation**. Except for new intents and slots detection, directly generating their labels based on the utterance semantics is more useful while harder. We suppose this is a promising research direction and we put it as our future work.

Acknowledgements

This work was supported by National Key Research and Development Project of China (No. 2022YFC3502303). This research is also supported by the Ministry of Education, Singapore, under its AcRF Tier 2 Funding (Proposal ID: T2EP20123-0052). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore. Ivor W. Tsang was supported by A*STAR Centre for Frontier AI Research.

References

Chunkit Chan, Xin Liu, Jiayang Cheng, Zihan Li, Yangqiu Song, Ginny Wong, and Simon See. 2023. DiscoPrompt: Path prediction prompt tuning for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 35–57, Toronto, Canada. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

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.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Byeongchang Kim, Seonghan Ryu, and Gary Geunbae Lee. 2017. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76(9):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, Brussels, Belgium. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Proc. Interspeech 2016*, pages 685–689.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. 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.

Yiyi Liu, Ruqing Zhang, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. 2023b. Prompt tuning with contradictory intentions for sarcasm recognition. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–339, Dubrovnik, Croatia. Association for Computational Linguistics.

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

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, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection 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 2078–2087, Hong Kong, China. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. 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 309–314, New Orleans, Louisiana. Association for Computational Linguistics.

Di Wu, Liang Ding, Fan Lu, and Jian Xie. 2020. SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1932–1937, Online. 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, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Bowen Xing and Ivor W. Tsang. 2023. Relational temporal graph reasoning for dual-task dialogue language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13170–13184.

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.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267, Florence, Italy. Association for Computational Linguistics.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2993–2999. IJCAI/AAAI Press.

Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Dongsheng Chen, and Yuexian Zou. 2023. Towards unified spoken language understanding decoding via label-aware compact linguistics representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12523–12531, Toronto, Canada. Association for Computational Linguistics.

A Experiment Settings

Datasets Following previous works, we evaluate our model on MixATIS (Hemphill et al., 1990; Qin et al., 2020) and (Coucke et al., 2018; Qin et al., 2020). MixATIS includes 13162/756/828 utterances for training/validation/testing. MixSNIPS includes 39776/2198/2199 utterances for training/-validation/testing.

Evaluation Metrics In multi-intent SLU, accuracy (Acc), F1 score and overall accuracy are used as the metrics for multiple intent detection, slot filling, and sentence-level semantic frame parsing, respectively. Overall accuracy denotes the ratio of sentences with all intents and slots correctly predicted. Implementation Details

Implementation Details For experiments based on T5-base and T5-large (Raffel et al., 2020), we use Adam optimizer with a learning rate of $3e^{-5}$. The batch size is 16/40 for MixATIS/MixSNIPS dataset. The number of gradient accumulation step is 16/10 for MixATIS/MixSNIPS dataset. Experiments are conducted on a single NVIDIA A40 GPU. For experiments based on LLama2-7B and LLama13B (Touvron et al., 2023), we use low-rank adaptation (LoRA) (Hu et al., 2022) to finetune them with only 17M and 26M trainable parameters, respectively. AdamW optimizer is used with a learning rate of $3e^{-4}$. The batch size is 128/256 for MixATIS/MixSNIPS dataset. Experiments are

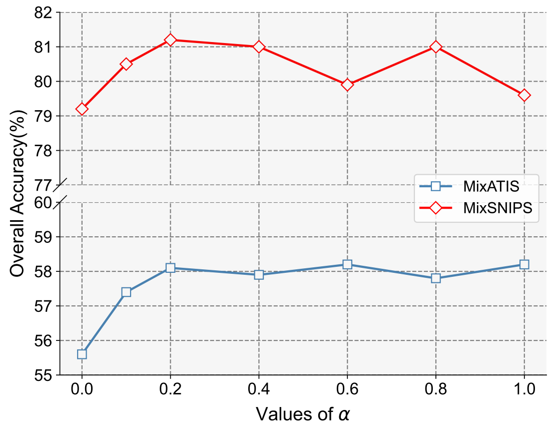

Figure 7: Experiment results on overall accuracy corresponding to different values of α.

conducted on two NVIDIA A40 GPUs. The α ratio of the instruction of $I_6 \sim I_9$ is set as 0.2.

B Implementation Details of Low-resource Experiments

We prepare the training samples of the k-shot setting by collecting samples from the original training set until each intent and slot label appears at least k times. As for the 5%/10%-ratio setting, we randomly select 5%/10%-ratio samples from the original training set.

C Effect of the Value of α

The value of α in our work is set as 0.2. We conducted experiments and tuned this ratio in the range of [0.0, 0.1 0.2, 0.4, 0.6, 0.8, 1.0]. To study its effect, we plot the experiment results corresponding to different values of α in Fig. 7. We can observe that as α increases from 0.0 to 0.2, the performance improves consistently, while a larger ratio (>0.2) did not lead to significant improvement in performance but computational cost. Therefore, we finally chose the value of 0.2 for α.

D Discussion of Different Inference Manners

In the inference stage, we tried two different manners. The first one adopts I_1 and I_5 to separately conduct inference for each task, which is the current one. The other one leverages the predictions of I_1 and I_5 and then uses them to inform I_6 and I_7 to let multiple intent detection and slot filling guide each other with their predicted labels. From the experimental results, we found that using one task's generated labels to inform the other task's generation led to comparable performances with the currently adopted inference setting, which separately conducts inference for each task. We suspect the reason is that although one task's labels can provide beneficial knowledge to guide the other task's label generation, there may exist error propagations, which may cause one task's incorrectly generated labels to mislead the label generation of the other task. Besides, the second manner leads to exposure bias because in the training stage, I_6 and I_7 include all correct labels of another task, while in the inference stage, I_6 and I_7 may include incorrectly predicted labels.

As for the first manner, in the training stage, our designed instructions can force the LLM to learn to capture the dual-task inter-dependencies and enhance the LLM's ability on extracting task-specific semantics. The trained LLM can benefit from the learned capabilities and comprehensively generate one task's labels with the consideration of the dual-task dependencies, while this is a soft manner without causing error propagation and exposure bias.

E Generalization of our Method

The two main contributions of our method – (1) dual-task inter-dependent instructions and (2) supervised contrastive instructions – can be generalized to other tasks and datasets. The dual-task inter-dependent instructions can be formatted as [sample + task A question + task B's labels] ->[generate]->[task A's labels] and in the same way, [sample + task B question + task A's labels] ->[generate]->[task B's labels]. By this means, the inter-dependencies between the tasks can be explicitly modeled in the prompt learning paradigm. As for our proposed supervised contrastive instructions, it can be formatted as [sample pair + whether the two samples have the same xx labels?]->[generate]->[True or False]. By this means, our proposed supervised contrastive instructions can be used in all scenarios where the train samples have golden labels.

F Definitions of different kinds of Errors in Sec. 5.5

Multiple Intent Detection (MID):
(1) 'fewer': the ratio of the incorrectly inferred test samples whose predicted intents are fewer than the golden intents.
(2) 'Eq. No.': the ratio of the incorrectly inferred test samples whose predicted intents number is equal to the golden intents number.
(3) 'More': the ratio of the incorrectly inferred test samples whose predicted intents are more than the golden intents.
Slot Filling (SF):
(1) '$\times$ Value': the ratio of the incorrectly inferred test samples that have at least one error in the predicted slot values.
(2) '$\times$ name': the ratio of the incorrectly inferred test samples that have at least one error in the predicted slot names.

	Instruction	Golden Output
I_1	utterance: define airline ua, names of airports and also show me city served both by nationair and canadian airlines international. question: what are the intents of the utterance according to options? options: <intent label set>	abbreviation, airport, city
I_2	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. question: which words are the slot values in the utterance?	ua, nationair, canadian airlines international
I_3	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. question: list those slot values' related slot names in the utterance: ua,nationair,canadian airlines international options: <slot label set>	ua is one airline code, nationair is one airline name, canadian airlines international is one airline name
$I_4(1)$	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. the related slot name for show me city served both is the time relative?	False
$I_4(2)$	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. the related slot name for canadian airlines international is the airline name?	True
I_5	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. question: which words are the slot values in the utterance? List them with their slot names. options: <slot label set>	ua is one airline code, nationair is one airline name, canadian airlines international is one airline name
I_6	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. This utterance includes these slot types: airline code,airline name,airline name. question: what are the intents of the utterance according to options? options: <intent label set>	abbreviation, airport, city
I_7	utterance: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. This utterance expresses these intent: abbreviation,airport,city. question: which words are the slot values in the utterance? List them with their slot names. options: <slot label set>	ua is one airline code, nationair is one airline name, canadian airlines international is one airline name
$I_8(1)$	U1: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. U2: what is the yn code, houston airports and then what are the cities that american airlines serves. utterances U1 and U2 express the same intents?	True
$I_8(2)$	U1: define airline ua, names of airports and also show me city served both by nationair and canadian airlines international. U2: which companies fly between boston and oakland and what types of meals are available. utterances U1 and U2 express the same intents?	False
$I_9(1)$	U1: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. U2: what does ea mean and show me the cities served by nationair. Utterances U1 and U2 include the same or similar slot types?	True
$I_9(2)$	U1: define airline ua , names of airports and also show me city served both by nationair and canadian airlines international. U2: what does the fare code yn mean and then how many fares are there one way from tacoma to montreal. Utterances U1 and U2 include the same or similar slot types?	False

Table 5: Detailed illustration of I_1~I_9 of utterance "define airline ua , names of airports and also show me city served both by nationair and canadian airlines international.", which is from the MixATIS dataset.

	Instruction	Golden Output
I_1	utterance: play isham jones and swine not deserves four points. question: what are the intents of the utterance according to options? options: <intent label set>	play music, rate book
I_2	utterance: play isham jones and swine not deserves four points. question: which words arethe slot values in the utterance?	isham jones,swine not, four,points
I_3	utterance: play isham jones and swine not deserves four points. question: list those slot values' related slot names in the utterance: ua,nationair,canadian airlines international. options: <slot label set>	isham jones is one artist, swine not is one object name, four is one rating value, points is one rating unit
$I_4(1)$	utterance: play isham jones and swine not deserves four points. the related slot name for deserves four points is the object part of series type?	False
$I_4(2)$	utterance: play isham jones and swine not deserves four points. the related slot name for four is the rating value?	True
I_5	utterance: play isham jones and swine not deserves four points. question: which words are the slot values in the utterance? List them with their slot names. options: <slot label set>	isham jones is one artist, swine not is one object name, four is one rating value, points is one rating unit
I_6	utterance: play isham jones and swine not deserves four points. This utterance includes these slot types: artist,object name,rating value,rating unit. question: what are the intents of the utterance according to options? options: <intent label set>	abbreviation, airport, city
I_7	utterance: play isham jones and swine not deserves four points. This utterance expresses these intent: play music,rate book. question: which words are the slot values in the utterance? List them with their slot names. options: <slot label set>	isham jones is one artist, swine not is one object name, four is one rating value, points is one rating unit
$I_8(1)$	U1: play isham jones and swine not deserves four points. U2: play me songs from agreable and rate this novel 5 stars. utterances U1 and U2 express the same intents?	True
$I_8(2)$	U1: play isham jones and swine not deserves four points. U2: add shelby lynne to my playlist this is luis fonsi and then show me the landing at low tide painting. utterances U1 and U2 express the same intents?	False
$I_9(1)$	U1: play isham jones and swine not deserves four points. U2: play michael angelo batio and then rate lords of the rim zero stars. Utterances U1 and U2 include the same or similar slot types?	True
$I_9(2)$	U1: play isham jones and swine not deserves four points. U2: play me some music from 199. utterances U1 and U2 include the same or similar slot types?	False

Table 6: Detailed illustration of I_1~I_9 of utterance "play isham jones and swine not deserves four points", which is from the MixSNIPS dataset.

	Utterances	Golden Intents	Predicted Intents	Gold Slot Value-name Pairs	Predicted Slot Value-name Pairs
1	list california airports , list la and how many canadian airlines international flights use aircraft 320	airport, city, quantity	airport, city, airline, aircraft	(california, state name), (la, city name), (canadian airlines international, airline name), (320, aircraft code)	(list california airports, airport name), (list la, city name), (canadian airlines, airline name), (aircraft 320, aircraft code)
2	i need a ticket from nashville to seattle and then flight numbers from chicago to seattle on continental	airfare, flight no	airline, flight no	(nashville, from location.city name), (seattle, to location.city name), (chicago, from location.city name), (seattle, to location.city name), (continental, airline name)	(i need a ticket, transport type), (nashville, from location.city name), (seattle, to location.city name), (chicago, stop location.city name), null (continental, airline name)
3	what cities does northwest fly to	city	city	(northwest, airline name)	(northwest, airline name), (cities, city name)

	Utterances	Golden Intents	Predicted Intents	Gold Slot Value-name Pairs	Predicted Slot Value-name Pairs
1	add the song to the soundscapes for gaming playlist and then play signe anderson chant music that is newest	add to playlist, play music	add to playlist, play music	(song, music item), (soundscapes for gaming, playlist), (signe anderson, artist), (chant, music item), (newest, sort)	(add the song, music item), (soundscapes for gaming playlist, playlist), (signe anderson chant music, artist), null, (newest, sort)
2	i want to eat close to bowlegs seven years from now and then i want to play the video game espn major league soccer	book restaurant, search creative work	book restaurant null	(close, spatial relation), (bowlegs, city), (seven years from now, time range), (video game, object type), (espn major league soccer, object name)	null (bowlegs, location name), (seven years from now, time range), (video game, object type), (espn major league soccer, object name)
3	i want to hear any tune from the twenties and then what time is holiday heart showing at the movie house	play music, search screening event	play music, search screening event	(tune, music item), (twenties, year), (holiday heart, movie name), (movie house, object location type)	null, (twenties, year), (holiday heart, movie name), (movie house, facility), (time, time range)

Table 7: Some error cases of ChatGPT. Errors are in red and ‘null’ denotes the corresponding slot is not extracted.