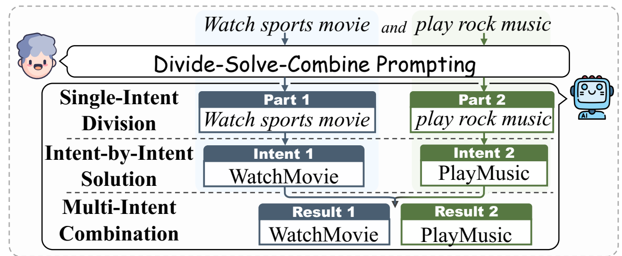
Divide-Solve-Combine: An Interpretable and Accurate Prompting Framework for Zero-shot Multi-Intent Detection

Libo Qin[1,2*], Qiguang Chen[3*], Jingxuan Zhou[1], Jin Wang[4], Hao Fei[5], Wanxiang Che[3], Min Li[1]

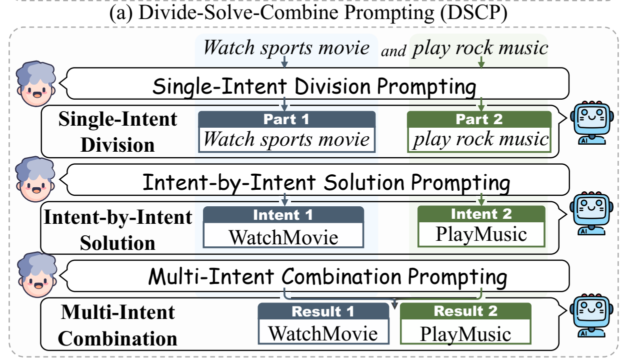
[1]School of Computer Science and Engineering, Central South University, China
[2]Key Laboratory of Data Intelligence and Advanced Computing in Provincial Universities, Soochow University, China
[3]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[4]Yunnan University, China
[5]National University of Singapore, Singapore
lbqin@csu.edu.cn, qgchen@ir.hit.edu.cn

Abstract

Zero-shot multi-intent detection is capable of capturing multiple intents within a single utterance without any training data, which gains increasing attention. Building on the success of large language models (LLM), dominant approaches in the literature explore prompting techniques to enable zero-shot multi-intent detection. While significant advancements have been witnessed, the existing prompting approaches still face two major issues: *lacking explicit reasoning* and *lacking interpretability*. Therefore, in this paper, we introduce a Divide-Solve-Combine Prompting (`DSCP`) to address the above issues. Specifically, `DSCP` explicitly decomposes multi-intent detection into three components including (1) *single-intent division prompting* is utilized to decompose an input query into distinct sub-sentences, each containing a single intent; (2) *intent-by-intent solution prompting* is applied to solve each sub-sentence recurrently; and (3) *multi-intent combination prompting* is employed for combining each sub-sentence result to obtain the final multi-intent result. By decomposition, `DSCP` allows the model to track the explicit reasoning process and improve the interpretability. In addition, we propose an interactive divide-solve-combine prompting (`Inter-DSCP`) to naturally capture the interaction capabilities of large language models. Experimental results on two standard multi-intent benchmarks (i.e., MIXATIS and MIXSNIPS) reveal that both `DSCP` and `Inter-DSCP` obtain substantial improvements over baselines, achieving superior performance and higher interpretability.



(a) Divide-Solve-Combine Prompting (DSCP)



(b) Interactive Divide-Solve-Combine Prompting (Inter-DSCP)

Figure 1: The example of Divide-Solve-Combine Prompting `DSCP` (a) and Interactive Divide-Solve-Combine Prompting `Inter-DSCP` (b).

1 Introduction

Intent detection plays a pivotal role in task-oriented dialog systems, which can be used to extract the intents of user queries for accurate system response generation (Tur and De Mori 2011; Qin et al. 2021c). In real-world scenarios, users often express multiple intents within a single utterance. Consequently, dominant approaches in the literature shift their focus from single intent detection to multi-intent detection (Qin et al. 2020; Zhu et al. 2023b).

With the revolution of the pre-trained models, remarkable success has been witnessed in the multi-intent detection. Specifically, recent multi-intent research can be broadly grouped into two main categories. The first category leverages slot filling for enhancing multi-intent detection through their mutual interaction (Gangadharaiah and Narayanaswamy 2019; Qin et al. 2021b; Song et al. 2022; Xing and Tsang 2022; Cheng, Yang, and Jia 2023). The second category seeks to improve multi-intent detection through intent semantic space optimization to enhance the performance (Wu, Su, and Juang 2021; Hou et al. 2021; Song, Huang, and Wang 2022; Vulić et al. 2022; Zhu et al. 2023a).

While remarkable success has been achieved, the current multi-intent detection approaches still follow the traditional training-test paradigm that requires a large amount of train-

[*]Equal Contribution

ing data. This reliance on abundant data presents challenges for real-world applications, where sufficient data may not be readily available. Recently, the Large Language Models (LLMs) have been shown surprising performance in zero-shot settings that does not require model updates, which can greatly alleviate the difficulty of annotated data. Inspired by this, Pan et al. (2023) introduce a vanilla multi-intent detection prompting method (VMID) for zero-shot multi-intent detection, which prompts LLM to generate multi-intent results without any parameter tuning. Despite its simplicity, this method still faces two major drawbacks: (1) ***Lacking explicit reasoning process***: VMID yields direct results without revealing any explicit reasoning process, limiting the performance (Wei et al. 2022); (2) ***Lacking interpretability***: The whole zero-shot prompting prediction process in VMID is a black box for users, making it challenging to establish a clear connection between sentence spans and their corresponding intents (Jiang et al. 2023b).

Motivated by this, in this paper, we introduce a Divide-Solve-Combine Prompting (`DSCP`) strategy to address the above issues. Specifically, as illustrated in Figure 1(a), `DSCP` consists of three components: (1) *single-intent division prompting*, (2) *intent-by-intent solution prompting*, and (3) *multi-intent combination prompting*. Concretely, given a user query, *single-intent division prompting* is first used to divide the utterance into spans where each span contains a single intent. Then, *intent-by-intent solution prompting* is applied to extract the intents from those spans. Furthermore, after obtaining all independent intent results, *multi-intent combination prompting* is required to combine all intents for the final multi-intent result. By explicitly emulating the multi-intent thought process, `DSCP` has the following advantages: (1) With the help of *single-intent division prompting* and *intent-by-intent solution prompting*, `DSCP` is able to explicitly solve zero-shot multi-intent SLU step by step, achieving the explicit reasoning process; (2) By decomposing the zero-shot multi-intent SLU into three sub-component solvers, `DSCP` enables us to analyze which span corresponds to each intent, helping us understand the reasons behind the results and thereby enhancing interpretability. In addition, we further propose an interactive divide-solve-combine prompting (`Inter-DSCP`) to naturally elicit the interaction capabilities of large language models (see Figure 1 (b)).

We conduct experiments on the two widely-recognized benchmarks, MIXATIS and MIXSNIPS. The experimental results demonstrate that `DSCP` achieves superior performance. Furthermore, the `Inter-DSCP` can bring further improvement by successfully capturing the interaction ability in LLM.

Our contributions can be summarized as follows:

- We introduce the Divide-Solve-Combine Prompting (`DSCP`) framework for zero-shot multi-intent detection, which can elicit the explicit reasoning process of multi-intent detection in LLM and improve interpretability.
- We further introduce an interactive divide-solve-combine prompting (`Inter-DSCP`), which is capable of naturally modeling the interaction ability of LLM to enhance performance.
- Experimental results on MIXATIS and MIXSNIPS demonstrate the effectiveness of `DSCP` and `Inter-DSCP` by achieving superior performance. Besides, extensive analysis reveals the superiority of our approaches.

To facilitate the reproducibility of our work, our code will be available at https://github.com/LightChen233/DSCP.

2 Divide-Solve-Combine Prompting

Unlike the vanilla multi-intent detection prompting, as shown in Figure 2, *divide-solve-combine prompting* explicitly decomposes the process into three components to solve it step by step, containing (1) *Single-Intent Division Prompting* (§2.1); (2) *Intent-by-Intent Solution Prompting* (§2.2) and *Multi-intent Combination Prompting* (§2.3).

2.1 Single-Intent Division Prompting

Single-Intent Division Prompting (SIDP) requires the model to explicitly divide a sentence into multiple spans corresponding to different intents. Formally, *SIDP* is shown as:

> [Task Instruction $\mathcal{T}$]: Assuming you are a professional multi-intent annotator, you need to label ...
> [Label Constraint $\mathcal{L}^C$]: You need to select the intent of the sentence from the following intent list ...
> [Single-Intent Division $\mathcal{D}$] : Firstly, you need to divide the sentence into multiple parts that contain different intents;

Each part of the prompt is introduced as follows:

(1) Task Instruction $\mathcal{T}$ describes the task requirements and definitions of multi-intent detection, aiming to clearly specify the task that the model needs to handle.

(2) Label Constraint $\mathcal{L}^C$ contains all predefined label set $\mathcal{L}$ from the multi-intent detection task.

(3) Single-Intent Division $\mathcal{D}$ is provided to require models to split given input into a series of single-intent spans.

In summary, the division process can be expressed as:

$$\hat{\mathcal{Y}}^D = \underset{\mathcal{S}}{\text{argmax}} \sum p(s_1, \ldots, s_n | \mathcal{T}, \mathcal{L}^C, \mathcal{D}, \mathcal{X}), \quad (1)$$

where $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ denotes one of the possible span split divided from input $\mathcal{X}$, and $\hat{\mathcal{Y}}^D = \{\hat{y}^D_1, \hat{y}^D_2, \ldots, \hat{y}^D_n\}$ denotes the predicted split single-intent span set for next step.

2.2 Intent-by-Intent Solution Prompting

After obtaining the split single-intent span, we further introduce a *Intent-by-Intent Solution Prompting (IISP)* to detect intent on each span. Specifically, *IISP* can be defined as:

> [Intent-by-Intent Solution Prompting $\mathcal{S}$] : Secondly, you need to consider what intents each part contains;

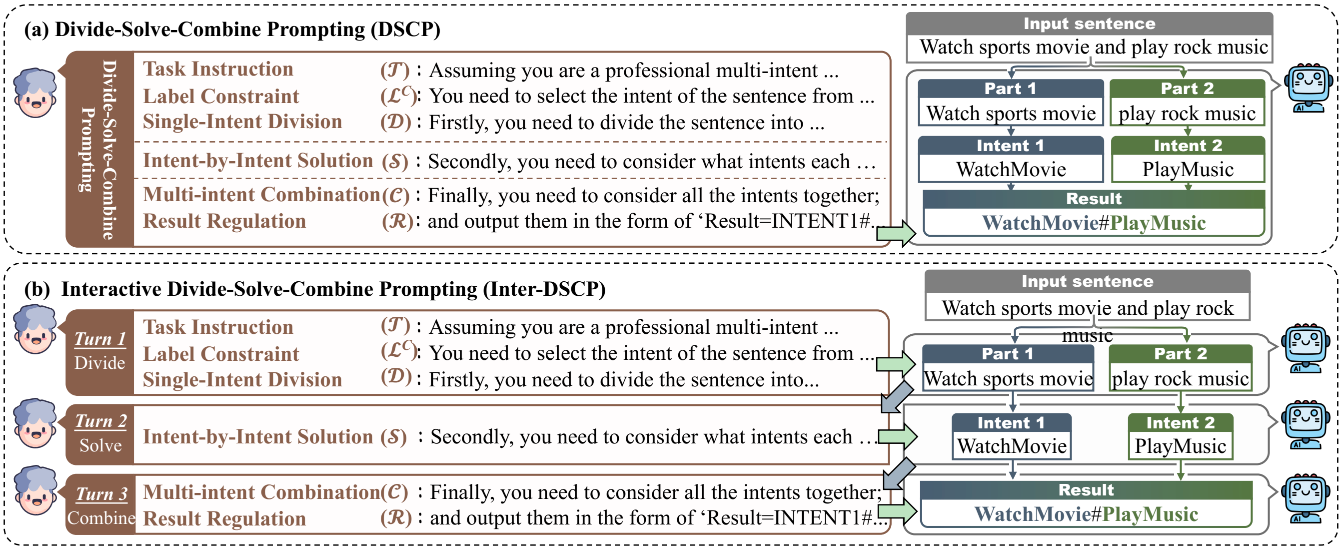


Figure 2: The main framework of `DSCP` (a) and `Inter-DSCP` (b). The left shows the specific prompting format, while the right illustrates the intuitive reasoning process.

where Intent-by-Intent Solution Prompting $\mathcal{S}$ requires models to predict the intents separately, described as:

$$\hat{\mathcal{Y}}^S = \underset{\mathcal{Y}^S}{\operatorname{argmax}}\, p(y_1^S, \ldots, y_n^S | \mathcal{H}^D, \hat{\mathcal{Y}}^D, \mathcal{S}), \tag{2}$$

where $\mathcal{Y}^S$ denotes one of the possible label lists for the corresponding span list, and $y_i^S \in \mathcal{Y}^S$ are all selected from the label set $\mathcal{L}$. $\mathcal{H}^D$ denotes the generation history which contains $\mathcal{T}, \mathcal{L}^C, \mathcal{D}, \mathcal{X}$ in the *SIDP* stage.

2.3 Multi-intent Combination Prompting

Furthermore, we provide *Multi-Intent Combination Prompting (MICP)* to combine all intents, which is defined as:

> Multi-intent Combination Prompting $\mathcal{C}$: Finally, you need to consider all the intents together.
> Result Regulation $\mathcal{R}$: and output them in the form of 'Result=INTENT1#INTENT2...

Similarly, the prompt is introduced in detail as follows:

(1) Multi-intent Combination Prompting $\mathcal{C}$ requires the model to combine all intent results to obtain the final multi-intent result.

(2) Result Regulation $\mathcal{R}$ contains all label set $\mathcal{L}$ from the intent detection task.

Formally, the *MICP* is determined as:

$$\hat{\mathcal{I}} = \underset{\mathcal{I}_j \in \mathcal{I} \wedge \mathcal{I}_j \subseteq \mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{I} | \mathcal{R}, \mathcal{C}, \hat{\mathcal{Y}}^S, \mathcal{H}^S), \tag{3}$$

where $\hat{\mathcal{I}}$ denotes the predicted multi-intent label results, and $\mathcal{H}^S$ denotes the generation history which contains $\mathcal{H}^D$, $\hat{\mathcal{Y}}^D$, $\mathcal{S}$ in *IISP* stage.

2.4 Interactive Divide-Solve-Combine Prompting

With the emergence of LLM, their strong interactive capabilities have gathered considerable attention (Zheng et al. 2023). Inspired by this, as shown in Figure 2 (b), we introduce Interactive Divide-Solve-Combine Prompting (`Inter-DSCP`) to leverage the interactive potential of LLMs. Unlike `DSCP` that directly obtain results in a single dialogue turn, `Inter-DSCP` leverages three turns of dialogue to stimulate the interactive capabilities of LLMs.

Formally, in `Inter-DSCP`, the overall process can be formulated as follows:

Single-Intent Division Prompting *In the first dialog interactive turn,* the initial response can be mathematically represented as follows:

$$\hat{\mathcal{Y}}^D_{\texttt{Inter}} = \underset{\mathcal{S}^p}{\operatorname{argmax}} \sum p(s_1^p, \ldots, s_n^p | \texttt{User}(\textit{SIDP})), \tag{4}$$

where $\texttt{User}(\cdot)$ signifies that the model receives the prompt in the "user" role.

Intent-by-Intent Solution Prompting *In the second dialog interactive turn,* the process of Intent-by-Intent Solution Prompting (*IISP*) is mathematically formulated as:

$$\hat{\mathcal{Y}}^S_{\texttt{Inter}} = \underset{\mathcal{Y}^S}{\operatorname{argmax}}\, p(\mathcal{Y}^S | \tilde{\mathcal{H}}^D, \texttt{User}(\textit{IISP})), \tag{5}$$

where $\mathcal{Y}^S = \{y_1^S, \cdots, y_n^S\}$. $\tilde{\mathcal{H}}^D = \{\texttt{User}(D), \texttt{Ass}(\hat{\mathcal{Y}}^D_{\texttt{Inter}})\}$ represents the generation history, and $\texttt{Ass}(\cdot)$ indicates the transformation of the model's result into the "assistant" role.

Multi-intent Combination Prompting *In the third dialog interactive turn,* the prediction of intent in this framework can be defined as:

$$\hat{\mathcal{I}} = \underset{\mathcal{I}_j \in \mathcal{I} \wedge \mathcal{I}_j \in \mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{I} | \tilde{\mathcal{H}}^S, \texttt{User}(\textit{MICP})), \tag{6}$$

Model	MixSNIPS			MixATIS		
	Intent Acc.(%)	Macro F1(%)	Micro F1(%)	Intent Acc.(%)	Macro F1(%)	Micro F1(%)
Mistral-7B (Jiang et al. 2023a)						
`VMID` (Pan et al. 2023)	32.87	66.35	67.48	5.97	39.46	37.88
`Zero-CoT` (Kojima et al. 2022)	32.98	69.61	69.79	8.19	43.59	42.40
`Least-to-Most` (Zhou et al. 2022)	29.00	61.91	62.65	6.67	32.44	32.97
`Plan-and-Solve` (Wang et al. 2023)	28.99	61.78	62.27	7.93	34.25	34.78
`DSCP`	36.65	69.44	**71.28**	11.21	42.20	41.81
`Inter-DSCP`	**41.52**	**69.62**	70.71	**14.72**	**46.96**	**46.60**
GPT-3.5 (OpenAI 2022)						
`VMID` (Pan et al. 2023)	63.12	87.87	87.80	21.94	67.40	66.01
`Zero-CoT` (Kojima et al. 2022)	63.21	87.79	87.38	26.11	68.77	66.11
`Least-to-Most` (Zhou et al. 2022)	66.48	89.90	89.47	25.69	68.99	65.45
`Plan-and-Solve` (Wang et al. 2023)	69.12	90.35	90.04	27.08	65.32	62.59
`DSCP`	72.90	91.14	90.79	29.17	73.75	70.05
`Inter-DSCP`	**74.90**	**91.50**	**91.30**	**41.11**	**75.04**	**75.41**
PaLM-2 (Anil et al. 2023)						
`VMID` (Pan et al. 2023)	79.82	92.32	92.96	26.25	65.57	66.15
`Zero-CoT` (Kojima et al. 2022)	79.95	93.07	92.88	26.39	65.94	65.74
`Least-to-Most` (Zhou et al. 2022)	76.31	92.08	91.95	26.94	69.73	66.25
`Plan-and-Solve` (Wang et al. 2023)	82.04	93.98	93.94	26.56	65.95	63.33
`DSCP`	83.95	94.18	94.16	28.19	65.99	67.24
`Inter-DSCP`	**86.81**	**94.27**	**94.43**	**31.53**	**66.75**	**67.39**
GPT-4 (OpenAI 2023)						
`VMID` (Pan et al. 2023)	88.45	95.46	95.47	32.64	76.74	74.02
`Zero-CoT` (Kojima et al. 2022)	87.77	95.46	95.47	40.97	78.62	77.40
`Least-to-Most` (Zhou et al. 2022)	87.27	94.60	94.71	48.33	82.26	80.67
`Plan-and-Solve` (Wang et al. 2023)	84.45	92.81	92.90	36.67	79.44	76.68
`DSCP`	89.68	95.69	95.80	50.69	**84.57**	**81.97**
`Inter-DSCP`	**92.00**	**96.59**	**96.66**	**52.22**	83.99	81.86

Table 1: Main Results for Mistral-7B, GPT-3.5, PaLM-2, and GPT-4 on MixSNIPS and MixSNIPS test sets. Intent Acc. denotes the intent accuracy. All metrics are calculated based on OpenSLU framework (Qin et al. 2023b). The improvements over all baselines are statistically significant with $p < 0.05$ under t-test

where $\tilde{\mathcal{H}}^S = \{\texttt{User}(\textit{SIDP}), \texttt{Ass}(\hat{\mathcal{Y}}^D_{\texttt{Inter}}), \texttt{User}(\textit{IISP}), \texttt{Ass}(\hat{\mathcal{Y}}^S_{\texttt{Inter}})\}$ captures the history of interactive prompts in a dialog format.

3 Experiments and Analysis

3.1 Implementation Settings

Following Qin et al. (2020); Cheng et al. (2024); Zhu et al. (2024b), we evaluate `DSCP` and `Inter-DSCP` on two widely used multi-intent benchmark: MixATIS and MixSNIPS (Qin et al. 2020). We follow Pan et al. (2023) to use similar regular expressions to extract multi-intent results to calculate the relevant metrics. The top-p parameter in all processes is selected from $\{0.95, 1\}$. The temperatures for *Single-Intent Division Prompting*, *Intent-by-Intent Solution Prompting* and *Multi-Intent Combination Prompting* are selected from $[0, 2]$.

3.2 Backbones and Baselines

We evaluate `DSCP` and `Inter-DSCP` on some representative LLMs, including: Mistral-7B (Jiang et al. 2023a); PaLM-2 (Anil et al. 2023); GPT-3.5 (OpenAI 2022) and GPT-4 (OpenAI 2023) backbone. In addition, we adapt the following prompting baselines, including:

- *Vanilla Multi-Intent Detection Prompting* (`VMID`) (Pan et al. 2023) directly requires LLMs to output the corresponding multiple intents through vanilla prompting;
- *Zero-shot Chain-of-Thought Prompting* (`Zero-CoT`) (Kojima et al. 2022) adds "Let's think step-by-step!" to stimulate the LLMs' thinking chain ability;
- *Least-to-Most Prompting* (`Least-to-Most`) (Zhou et al. 2022) generates a series of sub-question and then solves them one-by-one;
- *Plan-and-Solve Prompting* (`Plan-and- Solve`) (Wang et al. 2023) automatically makes a solution plan and then solves them.

3.3 Main Results

Following Qin et al. (2023b), we use Intent Accuracy, Micro F1, and Macro F1 for evaluating multi-intent detection performance. The main results are illustrated in Table 1. From the results, we have the following observations:
GPT-4 achieves the best results compared to other back-

Model	Intent Acc.(%)	Macro F1(%)	Micro F1(%)
	MIXSNIPS		
w/o *SIDP*	70.44	90.63	90.22
w/o *IISP*	67.62	85.02	84.54
w/o *MICP*	66.07	87.61	87.26
`DSCP`	72.90	91.14	90.79
	MIXATIS		
w/o *SIDP*	13.61	66.29	63.39
w/o *IISP*	24.72	72.42	70.00
w/o *MICP*	12.64	68.25	63.65
`DSCP`	29.17	73.75	70.05

Table 2: Ablation experiments on MIXSNIPS and MIXATIS based on GPT-3.5 backbone.

bones. Among all backbones, GPT-4 has the best performance on the MIXATIS and MIXSNIPS benchmarks. Specifically, compared to other backbones, `DSCP` can be improved by at least 5.73% on Intent Acc, which shows that larger models and better training lead to better performance.
`DSCP` achieves superior performance. `DSCP` surpasses all previous baselines on all backbones and achieves superior performance, while traditional CoT strategies fail in these benchmarks. Specifically, `DSCP` outperforms the `Plan-and-Solve` method by at least 1.91% and 1.63% on MIXSNIPS and MIXATIS, respectively, which shows the effectiveness of `DSCP`.
`Inter-DSCP` brings further performance improvements. `Inter-DSCP` can further significantly improve performance. As illustrated in Table 1, `Inter-DSCP` shows superiority over `DSCP` across all backbones (with at least 1.53% improvements on Intent Acc.). We attribute it to the fact that utilizing the interaction ability of LLM can effectively boost zero-shot multi-intent detection.

3.4 Analysis

In this section, we conduct thorough analyses to better understand our approach by answering the following questions: (1) *Does SIDP help utterance understanding?* (2) *Can IISP attain better intent detection performance?* (3) *What effects can MICP bring?* (4) *How `DSCP` performs on multi-intent detection?* (5) *Is `DSCP` a robust method across different prompting?* (6) *Can `DSCP` be improved by few-shot demonstrations?* (7) *Is `DSCP` interpretable and user-friendly for humans?* (8) *Why `DSCP` works?*

Answer 1: *SIDP* can boost the understanding of the user utterance. We investigate the effectiveness of *single-intent division prompting (SIDP)* by removing the *SIDP* from `DSCP` and preserving the other prompting. As shown in Table 2 (w/o *SIDP*), we find that the performance on Intent Acc. drops significantly about 2.5% on MIXSNIPS. This is because missing single-intent division operations make it hard to effectively understand complex multiple intents in the utterance, which limits the performance.

Answer 2: *IISP* matters for intent detection on the divided span. This section explores the effectiveness of *IISP* by removing *IISP* and keeping other prompting unchanged. As shown in Table 2 (w/o *IISP*), we find the Intent Acc., Macro F1 , and Micro F1 exhibit significant decreases of over 5%. We attribute it to the fact that *IISP* can address simple single-intent detection more easily than directly performing multi-intent detection.

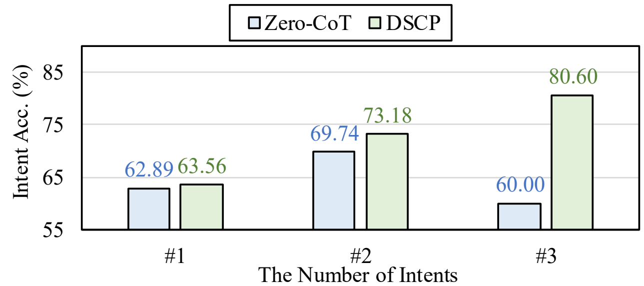


Figure 3: Performance differences under different number of intents on `DSCP` and `Zero-CoT` based on GPT-3.5.

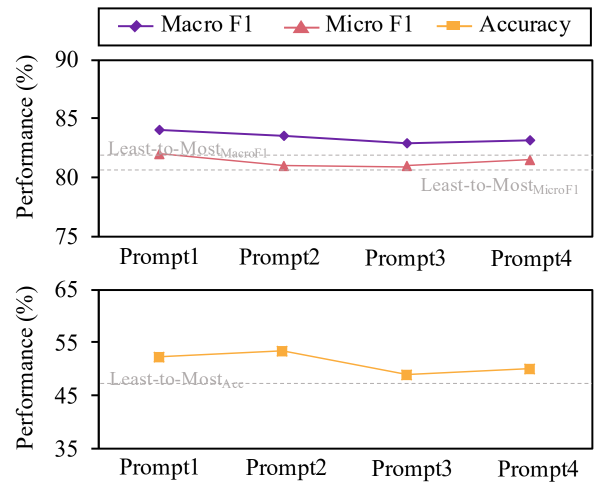
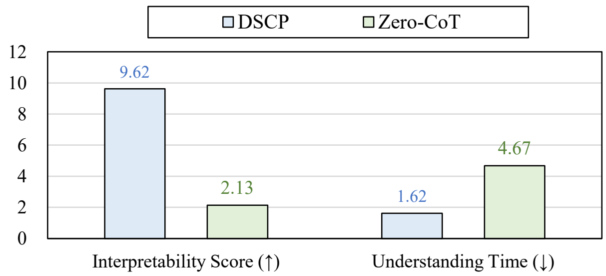


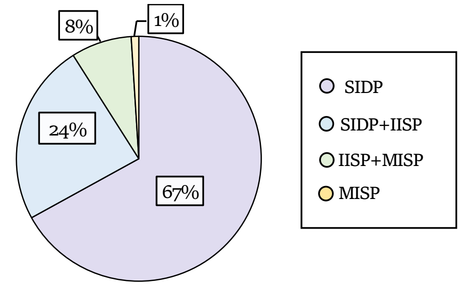
Figure 4: The robust analysis across different prompts based on GPT-4 backbone.

Answer 3: *MICP* can eliminate redundancy and unreasonable intents. We verify the effectiveness of *multi-intent combination prompting (MICP)* by removing the procedure of *MICP*. We still preserve "Result Regulation" prompting ($\mathcal{R}$) for better result extraction. As illustrated in Table 2 (w/o *MICP*), when removing *MICP*, we observe that it drops by 6.83% on Intent Acc, 3.53% on Macro F1 and 3.53% on Micro F1 for MIXATIS. In addition, in our in-depth exploration, we observe that *MICP* not only collects and regulates the output, but also eliminates redundancy and unreasonable intents, which enhances the performance.

Answer 4: `DSCP` can performs better on more intents. We further explore whether `DSCP` can perform better in multi-intent scenarios. Specifically, we conduct a grouping based on the number of intents in the multi-intent datasets on MIXSNIPS. From the Figure 3, we observe that as the

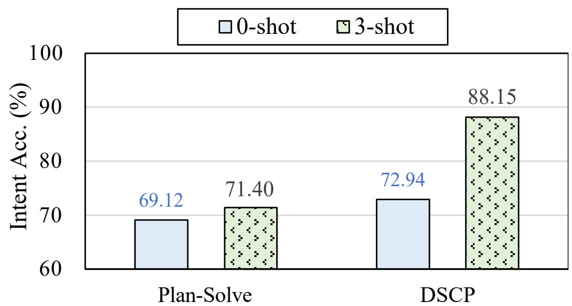


(a) The artificial interpretability scores of different model outputs and the average time spent by humans in understanding these outputs.

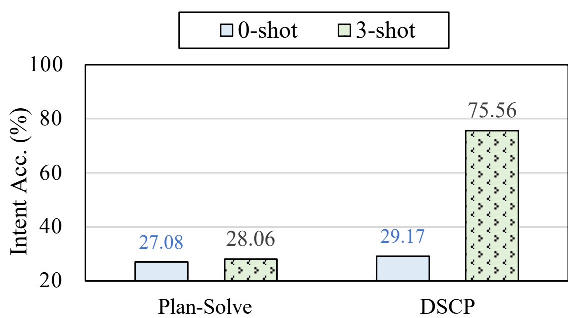


(b) The explanatory analysis of why the DSCP surpasses `Zero-CoT`. We manually attribute the performance enhancement to three phases of `DSCP`.

Figure 5: The interpretability analysis across different prompts.



(a) Few-shot Performance on MixATIS



(b) Few-shot Performance on MixSNIPS

Figure 6: Evaluation of GPT-3.5's few-shot learning capability. The 3-shot examples are randomly chosen from the development dataset.

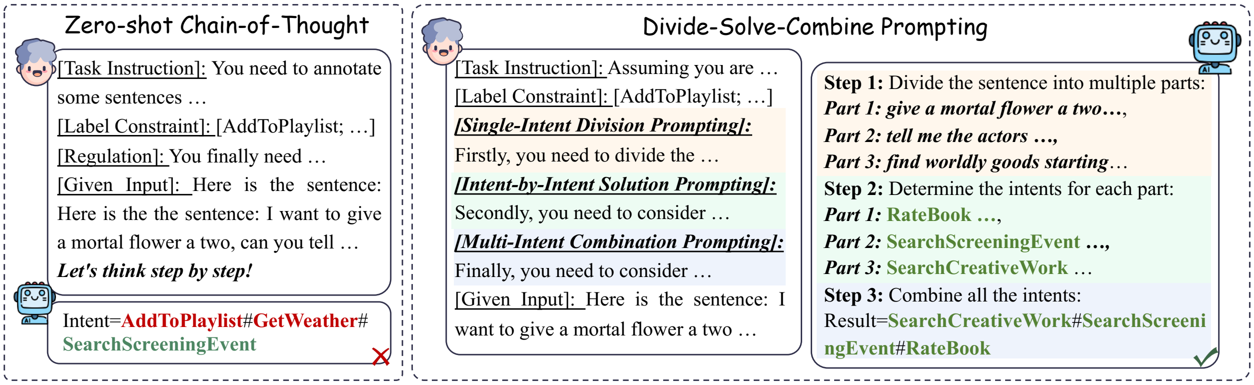
number of intents increased, the performance of DSCP improved. More importantly, the performance gap between DSCP and `Zero-CoT` also increased as the number of intents increased, which demonstrates that our model can attain better performance on complex multi-intent scenarios.

Answer 5: `DSCP` is robust across different prompting. To analyze the robustness of `DSCP`, we employ four synonymous prompts that convey the same meaning but use different expressions. Specifically, we leverage GPT-4 to generate three guiding prompts that are synonymous with the three prompt contents we have proposed. Figure 4 presents the performance of four distinct `DSCP` prompts. Across all metrics, the average performance of `DSCP` surpasses that of `Least-to-Most` prompting, which further verifies the robustness of `DSCP`.

Answer 6: `DSCP` can be well improved by few-shot demonstrations. To further analyze the model's performance in a few-shot setting, we randomly select three samples from the validation set, and manually modify them to relevant natural language format. As shown in Figure 6, `DSCP` can significantly improve the effectiveness of GPT-3.5 by few-shot learning, even exceeding the effect of GPT-4. Moreover, compared with the `Plan-and-Solve` model, the performance of the model is significantly improved, which further shows the effectiveness of `DSCP`.

Answer 7: `DSCP` is an interpretable and user-friendly framework To evaluate the interpretability and user-friendliness of `DSCP`, one must consider both the transparency of its underlying processes and the ease with which users can interact with it. To this end, we conduct a manual evaluation of the generated outputs, requiring evaluators to assign interpretability scores and record the time taken to understand the outputs and annotate the scores. As shown in Figure 5 (a), our framework achieves an interpretability score of 9.62 and shorter time spent in output understanding, significantly better than `Zero-CoT`. This is because the "Let's think step-by-step!" approach fails to induce step-by-step reasoning in multi-intent tasks, often directly providing answers, which greatly suppresses the model's interpretability. Furthermore, we require annotators to figure out which prompting strategies in `DSCP` lead to the enhanced predictions. As illustrated in Figure 5 (b), we can observe that all output can be explained by the relevant prompting strategies. And 67% of the examples resulted from the introduction of SIDP, which introduces more fine-grained interpretability.

Answer 8: Qualitative Analysis We conduct qualitative analysis to better understand DSCP by providing a case study comparing the outputs generated by the `Zero-CoT` approach and our `DSCP` approach. As illustrated in Figure 7, we observe that `Zero-CoT` incorrectly predicts "SearchCreativeWork#RateBook" as "AddToPlaylist#GetWeather". This is because `Zero-CoT`



(a) The reasoning result for Zero-CoT. (b) The reasoning result for DSCP.

Figure 7: Case study of `DSCP` and `Zero-CoT` on GPT-3.5.

makes it difficult to understand complex multi-intent detection. In contrast, `DSCP` can correctly predict the multiple intents. Unlike `Zero-CoT`, `DSCP` explicitly requires LLM to decompose the utterance first, then solve it intent-by-intent, and finally combine all intents, which not only reduces the difficulty of the task, but also improves the interpretability.

4 Related Work

4.1 Prompt Learning

Recent advancements in Large Language Models (LLMs) have significantly enhanced their performance in various NLP tasks (Chowdhery et al. 2022; OpenAI 2023). A key development is the improved chain-of-thought (CoT) reasoning capabilities (Wei et al. 2022; Kojima et al. 2022; Zhou et al. 2022; Zheng et al. 2023; Qin et al. 2023a; Chen et al. 2024a,b; Feng et al. 2024; Chu et al. 2023). Specifically, Wei et al. (2022) first highlight LLMs' remarkable ability for multi-step chain-of-thought reasoning. Based upon this, Kojima et al. (2022) first introduce a novel and efficient trigger phrase, "Let's think step by step!", to activate zero-shot multi-step reasoning. Building on this, Zhou et al. (2022) propose the Least-to-Most to divide the hard request into simple requests and solve them one by one. Wang et al. (2023) develop the Plan-and-Solve method, which separates the process into planning and executing sub-tasks. Yao et al. (2023) and Hu et al. (2024) propose a tree-format thought path for better reasoning performance. Furthermore, Yang et al. (2024) introduce the concept of a meta-buffer to store a series of high-level cognitive templates for better reasoning, which are distilled from the problem-solving processes. Qin et al. (2023a) and Chen et al. (2024b) extend the vanilla CoT into cross-lingual and multi-step cross-modal scenarios.

In this work, we investigate the zero-shot multi-intent scenario. To this end, we introduce the `DSCP` to explicitly decouple the zero-shot multi-intent detection process and `Inter-DSCP` to further utilize the interactivity of LLMs.

4.2 Intent Detection

Intent detection is a fundamental component of task-oriented dialogue systems, as it enables the extraction of the user's intent during interactions (Tur and De Mori 2011). Recent advancements in deep neural networks have led to diverse methodologies for single-intent detection (Goo et al. 2018; Qin et al. 2019, 2021a). However, these methods predominantly focus on single-intent scenarios, which limits their applicability in real-world environments. To overcome this, research has pivoted towards multi-intent SLU (Qin et al. 2020; Cheng et al. 2023; Qin et al. 2024). Gangadharaiah and Narayanaswamy (2019); Qin et al. (2020) pioneer joint modeling approaches for this domain and introduce MIXATIS and MIXSNIPS benchmarks for the multi-intent SLU community. Based on this, Qin et al. (2021b) propose a non-autoregressive framework for faster speed. Additionally, Xing and Tsang (2022) introduce a heterogeneous semantics-label graph for multi-intent SLU. Song et al. (2022) and Pham, Tran, and Nguyen (2023) utilize the label knowledge between intents and slots. Recently, Pan et al. (2023); He and Garner (2023) and Zhu et al. (2024a) explore a vanilla prompt framework to solve zero-shot intent detection, which does not require any training data.

In contrast to their approaches, we introduce a divide-solve-combine prompting (`DSCP`) framework and an interactive version (`Inter-DSCP`), achieving to track the explicit reasoning process and improve the interpretability.

5 Conclusion

In this paper, we introduce a Divide-Solve-Combine Prompting (`DSCP`) for zero-shot multi-intent detection to enhance explicit reasoning and interpretability. Specifically, `DSCP` contains *single-intent division prompting*, *intent-by-intent solution prompting*, and *multi-intent combination prompting* to first split an input utterance into multiple sub-sentences, solve the intents one-by-one, and finally combine the final multi-intents, respectively. Additionally, we further propose an Interactive Divide-Solve-Combine Prompting (`Inter-DSCP`) to capture the LLM interaction capabilities. Experiments on MIXATIS and MIXSNIPS show that `DSCP` and `Inter-DSCP` obtain promising performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62306342, 62236004, 62441603 and 62476073. This work was sponsored by the Excellent Young Scientists Fund in Hunan Province (2024JJ4070), the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3024 and CCF-Zhipu Large Model Innovation Fund (NO.CCF-Zhipu202406). This work was also supported by Key Laboratory of Data Intelligence and Advanced Computing in Provincial Universities, Soochow University. We are grateful for resources from the High Performance Computing Center of Central South University. Libo Qin is the corresponding author.

References

Anil, R.; Dai, A. M.; Firat, O.; et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Chen, Q.; Qin, L.; WANG, J.; Zhou, J.; and Che, W. 2024a. Unlocking the Capabilities of Thought: A Reasoning Boundary Framework to Quantify and Optimize Chain-of-Thought. In *Proc. of NeurIPS*.

Chen, Q.; Qin, L.; Zhang, J.; Chen, Z.; Xu, X.; and Che, W. 2024b. M^3CoT: A Novel Benchmark for Multi-Domain Multi-step Multi-modal Chain-of-Thought. In *Proc. of ACL*.

Cheng, L.; Yang, W.; and Jia, W. 2023. A scope sensitive and result attentive model for multi-intent spoken language understanding. In *Proc. of AAAI*.

Cheng, X.; Zhu, Z.; Cao, B.; et al. 2023. MRRL: Modifying the Reference via Reinforcement Learning for Non-Autoregressive Joint Multiple Intent Detection and Slot Filling. In *Proc. of EMNLP Findings*.

Cheng, X.; Zhu, Z.; Li, H.; Li, Y.; Zhuang, X.; and Zou, Y. 2024. Towards multi-intent spoken language understanding via hierarchical attention and optimal transport. In *Proc. of the AAAI*.

Chowdhery, A.; Narang, S.; Devlin, J.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Chu, Z.; Chen, J.; Chen, Q.; Yu, W.; He, T.; Wang, H.; Peng, W.; Liu, M.; Qin, B.; and Liu, T. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.

Feng, G.; Zhang, B.; Gu, Y.; Ye, H.; He, D.; and Wang, L. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36.

Gangadharaiah, R.; and Narayanaswamy, B. 2019. Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog. In *Proc. of NAACL*.

Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. Slot-Gated Modeling for Joint Slot Filling and Intent Prediction. In *Proc. of NAACL*.

He, M.; and Garner, P. N. 2023. Can ChatGPT Detect Intent? Evaluating Large Language Models for Spoken Language Understanding. *arXiv preprint arXiv:2305.13512*.

Hou, Y.; Lai, Y.; Wu, Y.; et al. 2021. Few-shot learning for multi-label intent detection. In *Proc. of AAAI*.

Hu, M.; Mu, Y.; Yu, X.; Ding, M.; Wu, S.; Shao, W.; Chen, Q.; Wang, B.; Qiao, Y.; and Luo, P. 2024. Tree-Planner: Efficient Close-loop Task Planning with Large Language Models. In *Proc. of ICLR*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; et al. 2023a. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jiang, S.; Zhu, S.; Cao, R.; et al. 2023b. SPM: A Split-Parsing Method for Joint Multi-Intent Detection and Slot Filling. In *Proc. of ACL*.

Kojima, T.; Gu, S. S.; Reid, M.; et al. 2022. Large language models are zero-shot reasoners. *Proc. of NeurIPS*.

OpenAI. 2022. Introducing ChatGPT.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Pan, W.; Chen, Q.; Xu, X.; et al. 2023. A preliminary evaluation of chatgpt for zero-shot dialogue understanding. *arXiv preprint arXiv:2304.04256*.

Pham, T.; Tran, C.; and Nguyen, D. Q. 2023. MISCA: A Joint Model for Multiple Intent Detection and Slot Filling with Intent-Slot Co-Attention. In *Proc. of EMNLP Findings*.

Qin, L.; Che, W.; Li, Y.; et al. 2019. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. In *Proc. of EMNLP*.

Qin, L.; Chen, Q.; Wei, F.; Huang, S.; and Che, W. 2023a. Cross-lingual Prompting: Improving Zero-shot Chain-of-Thought Reasoning across Languages. In *Proc. of EMNLP*.

Qin, L.; Chen, Q.; Xu, X.; Feng, Y.; and Che, W. 2023b. OpenSLU: A Unified, Modularized, and Extensible Toolkit for Spoken Language Understanding. *arXiv preprint arXiv:2305.10231*.

Qin, L.; Chen, Q.; Zhou, J.; Li, Q.; Lu, C.; and Che, W. 2024. Decoupling Breaks Data Barriers: A Decoupled Pre-training Framework for Multi-intent Spoken Language Understanding. In *Proc. of IJCAI*.

Qin, L.; Liu, T.; Che, W.; et al. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *Proc. of ICASSP*.

Qin, L.; Wei, F.; Xie, T.; et al. 2021b. GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling. In *Proc. of ACL*.

Qin, L.; Xie, T.; Che, W.; and Liu, T. 2021c. A Survey on Spoken Language Understanding: Recent Advances and New Frontiers. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4577–4584. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Qin, L.; Xu, X.; Che, W.; et al. 2020. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. In *Proc. of EMNLP Findings*.

Song, F.; Huang, L.; and Wang, H. 2022. A Unified Framework for Multi-intent Spoken Language Understanding with prompting. arXiv:2210.03337.

Song, M.; Yu, B.; Quangang, L.; Yubin, W.; Liu, T.; and Xu, H. 2022. Enhancing Joint Multiple Intent Detection and Slot Filling with Global Intent-Slot Co-occurrence. In *Proc. of EMNLP*.

Tur, G.; and De Mori, R. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Vulić, I.; Casanueva, I.; Spithourakis, G.; et al. 2022. Multi-Label Intent Detection via Contrastive Task Specialization of Sentence Encoders. In *Proc. of EMNLP*.

Wang, L.; Xu, W.; Lan, Y.; et al. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proc. of ACL*.

Wei, J.; Wang, X.; Schuurmans, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Proc. of NeurIPS*.

Wu, T.-W.; Su, R.; and Juang, B. 2021. A Label-Aware BERT Attention Network for Zero-Shot Multi-Intent Detection in Spoken Language Understanding. In *Proc. of EMNLP*.

Xing, B.; and Tsang, I. 2022. Co-guiding Net: Achieving Mutual Guidances between Multiple Intent Detection and Slot Filling via Heterogeneous Semantics-Label Graphs. In *Proc. of EMNLP*.

Yang, L.; Yu, Z.; Zhang, T.; et al. 2024. Buffer of Thoughts: Thought-Augmented Reasoning with Large Language Models. In *Proc. of NeurIPS*.

Yao, S.; Yu, D.; Zhao, J.; et al. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Zheng, C.; Liu, Z.; Xie, E.; Li, Z.; and Li, Y. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q. V.; et al. 2022. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *Proc. of ICLR*.

Zhu, Z.; Cheng, X.; An, H.; et al. 2024a. Zero-Shot Spoken Language Understanding via Large Language Models: A Preliminary Study. In *Proc. of COLING*.

Zhu, Z.; Cheng, X.; Huang, Z.; et al. 2023a. Towards Unified Spoken Language Understanding Decoding via Label-aware Compact Linguistics Representations. In *Proc. of ACL Findings*.

Zhu, Z.; Cheng, X.; Li, Y.; Li, H.; and Zou, Y. 2024b. Aligner^2: Enhancing Joint Multiple Intent Detection and Slot Filling via Adjustive and Forced Cross-Task Alignment. In *Proc. of the AAAI*.

Zhu, Z.; Xu, W.; Cheng, X.; et al. 2023b. A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In *Proc. of ICASSP*.