Sample Design Engineering: An Empirical Study on Designing Better Fine-Tuning Samples for Information Extraction with LLMs

Biyang Guo[1†], **He Wang**[1†], **Wenyilin Xiao**[1†]
Hong Chen[2†], **Zhuxin Lee**[3], **Songqiao Han**[4,1*], **Hailiang Huang**[1,5*]
[1]AI Lab, SIME, Shanghai University of Finance and Economics
[2]Ant Group, [3]Guangdong Yunxi Technology
[4]Key Laboratory of Interdisciplinary Research of Computation and Economics, Ministry of Education, China
[5]Shanghai University of Finance and Economics-Ant Group Joint Laboratory of Frontier Financial Intelligence

Abstract

Large language models (LLMs) have achieved significant leadership in many NLP tasks, but aligning structured output with generative models in information extraction (IE) tasks remains a challenge. Prompt Engineering (PE) is renowned for improving IE performance through prompt modifications. However, the realm of the sample design for downstream fine-tuning, crucial for task-specific LLM adaptation, is largely unexplored. This paper introduces **Sample Design Engineering** (SDE), a methodical approach to enhancing LLMs' post-tuning performance on IE tasks by refining input, output, and reasoning designs. Through extensive ID and OOD experiments across six LLMs, we first assess the impact of various design options on IE performance, revealing several intriguing patterns. Based on these insights, we then propose an integrated SDE strategy and validate its consistent superiority over heuristic sample designs on three complex IE tasks with four additional LLMs, demonstrating the generality of our method. Additionally, analyses of LLMs' inherent prompt/output perplexity, zero-shot, and ICL abilities illustrate that good PE strategies may not always translate to good SDE strategies. Code is available at https://github.com/beyondguo/LLM-Tuning.

1 Introduction

Information extraction (IE) aims to extract structured information from unstructured text, which is highly valuable in a wide range of industrial scenarios. The emergence of Large Language Models (LLMs) such as GPT-3 (Brown et al., 2020), LLaMA (Touvron et al., 2023a) has broadened the capabilities of language models to tackle various complex IE tasks with a single model. Nonetheless, a fundamental challenge arises from the discrepancy between the unstructured nature of the

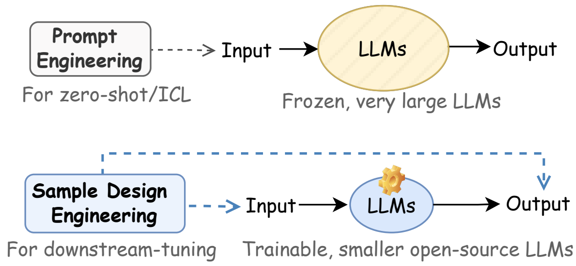

Figure 1: A simplified comparison between PE and our proposed SDE.

LLMs' generative paradigm and the requirement for structured output. In this background, ***Prompt Engineering*** (**PE**) has become a key area in leveraging cutting-edge LLMs to address this challenge (Wan et al., 2023; Wang et al., 2023a; Xie et al., 2023; Pang et al., 2023).

However, the efficacy of PE relies on the size of LLMs. In industrial applications, the high costs of deploying large models and data privacy risks drive many companies to seek the customization of smaller, open-source models tailored to their specific needs by downstream fine-tuning. Inspired by PE, we believe that the design of samples is also vital in downstream fine-tuning scenarios. This paper, therefore, aims to design effective fine-tuning samples for IE tasks, which we term ***Sample Design Engineering*** (**SDE**). Different sample designs may make it easier or harder for the LLMs to learn, especially given the complexity and scarcity of training samples for downstream tasks. Figure 1 is a simplified demonstration of PE and SDE.

We begin by identifying a range of SDE options and conduct experiments on a typical IE task – multi-aspect sentiment analysis (MASA) to explore the impact of each option. Some enlightening insights can be revealed such as the position of task instructions and the use of placeholders for unmentioned targets, which demonstrate the significant impact of various SDE options on LLMs'

[†]Equal Contribution
[*]Corresponding authors, emails:
han.songqiao@shufe.edu.cn, hlhuang@shufe.edu.cn

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 573–594
November 12-16, 2024 ©2024 Association for Computational Linguistics

fine-tuning performance. Leveraging these findings, we propose an integrated strategy **ES-SDE** (Empirically Strong - SDE), which outperforms weaker SDE combinations and heuristic designs from other studies on several complex IE tasks, showcasing its robustness and effectiveness on different models and training settings. Furthermore, our exploratory analysis of perplexity, zero-shot, and in-context learning (ICL) furthers our understanding of the relationship between PE and SDE. Our analysis indicate that a well-crafted PE strategy may not necessarily translate to a successful SDE strategy, prompting further investigation into the mechanisms of SDE to optimize LLMs for downstream applications. These discoveries underscore the potential for refining SDE mechanisms to augment LLMs' fine-tuning. The main contributions of our research are as follows:

- We propose Sample Design Engineering, a new data-centric perspective for enhancing the performance of Large Language Models in downstream tasks. we emphasize the importance of sample design during the fine-tuning of LLMs, whereas much of the existing research has focused primarily on prompt design.
- We provide a comprehensive summary and systematic evaluation of various sample design strategies, many of which have either been overlooked in previous research or only explored in a fragmented manner.
- Through extensive experiments involving ten models and three task types, we demonstrate the necessity and effectiveness of this novel Sample Design Engineering perspective.

2 Related Work

2.1 Prompt Engineering (PE) for Information Extraction

With the rapid advancement of LLMs, several studies have explored the zero-shot and few-shot capabilities of large models on typical IE tasks (Wei et al., 2023; Li et al., 2023a; Han et al., 2023), revealing notable performance gaps compared to traditional supervised SoTA models. To bridge the gap between IE tasks and text generation models, previous studies have proposed various prompt strategies to improve prompt quality. These strategies include carefully designed prompt templates or generation methods (Xie et al., 2023; Pang et al., 2023; Xu et al., 2023; Xie et al., 2024), sample retrieval techniques to provide better few-shot examples (Wan et al., 2023; Wang et al., 2023a), and code-based methods (Wang et al., 2023c; Li et al., 2023b) to enhance the model's adaptation to structured tasks.

However, most research focus on very large models (Sahoo et al., 2024). These most advanced and effective LLMs are either black-box models that are only accessible via APIs, or extremely large models with large resource requirements. Consequently, many practitioners turn to smaller but open-source LLMs, especially 10B around models.

2.2 Fine-tuning LLMs

According to the different purposes, we can divide LLMs' fine-tuning into two types: *instruction-tuning* (IT) and *downstream-tuning* (DT)[1]. IT trains LLMs to comprehend and follow human instructions across diverse NLP tasks (Longpre et al., 2023; Taori et al., 2023). DT customizes LLMs for complex industrial tasks, requiring high output stability for easier parsing and downstream application. To intrinsically enhance the LLMs' comprehension of IE tasks, some IT-based methods have been proposed and have shown some success (Wang et al., 2022; Zhang et al., 2023b; Sainz et al., 2024; Wang et al., 2023b). However, above works merely adopt a vanilla format of fine-tuning data and do not further explore the organization of structured data. Our study centers in DT scenarios, highlighting sample design challenges, but the insights may also benefit IT sample design, a topic for future exploration.

In addition, parameter-efficient fine-tuning (PEFT) methods, such as prefix-tuning(Li and Liang, 2021), prompt-tuning(Lester et al., 2021), p-tuning(Liu et al., 2023), and LoRA(Hu et al., 2021) provide cost-effective alternatives that retain FFT's effectiveness, gaining popularity in industrial applications. In this research, we use the widely-used LoRA as the default fine-tuning technique. However, we believe results from our study are also applicable to other PEFT methods.

3 Sample Design Engineering

3.1 Typical SDE Options

We categorize sample design options into *input*, *output*, and *reasoning*. We take the Multi-Aspect Sentiment Analysis (MASA) task as an example to clarify each option. MASA requires analyzing

[1] It is also known as task tuning (TT) in some literature, like (Weber et al., 2023).

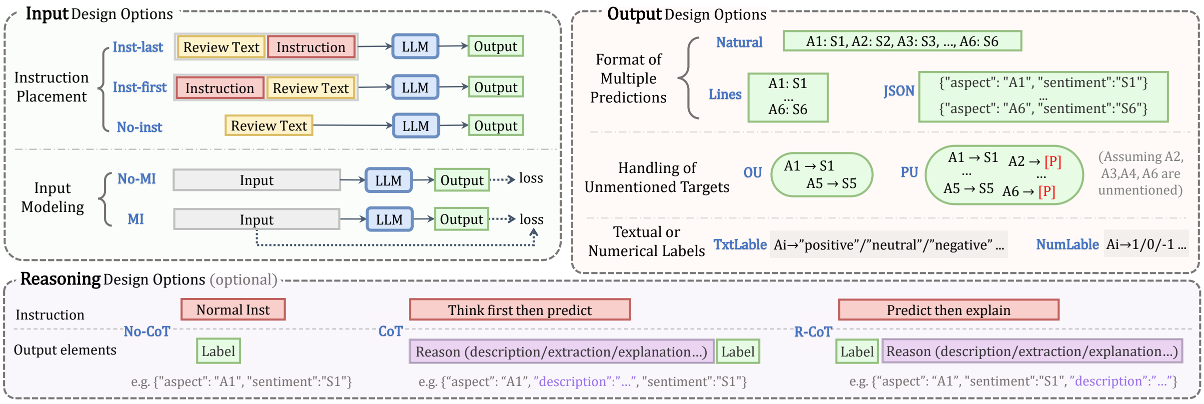

Figure 2: Typical SDE options to be considered when designing downstream-tuning samples, taking the MASA task as an example. Ai means aspect i, Si means its sentiment label, `[P]` refers to placeholder tokens.

review texts to assign sentiments to predefined aspects, while some aspects may be unmentioned, a specific example can be found in A.2. Figure 2 is an overview of different SDE options.

Input Design Options:

(1) **Instruction Placement**: Put the instruction before / after the task text ***(Inst-first / Inst-last)***, or with no instruction ***(No-inst)*** as used in many previous tasks (Lewis et al., 2019; Guo et al., 2022; Zhang et al., 2023a).

(2) **Input Modeling**: Compare ***No-MI*** that excludes input from loss calculation, akin to LLaMA2's SFT process (Touvron et al., 2023b)) against ***MI*** (modeling input in back-propagation).

Output Design Options:

(1) **Multiple Predictions Formatting:** Set the output formatting from less to more structured, ***Natural*** (free-form text), ***Lines*** (each aspect on a new line), and ***JSON*** (JSON-lines for precision and explicitness).

(2) **Unmentioned Targets:** Each text may only contain content related to a part of predefined targets. For those unmentioned targets, omit them, termed ***OU*** (Omit Unmentioned), or place placeholders such as "None", "", or others for them, termed ***PU*** (Placeholders for Unmentioned).

(3) **Textual or numerical labels:** Use the default textual labels (***TxtLabel***) or numbers (***NumLabel***) to represent outcomes.

Reasoning Design Options:

Chain-of-Thought (CoT) (Wei et al., 2022) has shown promise in improving LLM's reasoning in zero-shot, ICL, and IT(Kim et al., 2023), but requires more study in DT. We introduce the ***CoT*** option to "think before predict". Conversely, the ***R-CoT*** (Reverse-CoT) enabling "predict then explain" to explore CoT's mechanics further. Note that Implementing CoT-like samples incurs additional annotation costs due to the `description` fields, making it task-dependent.

3.2 Integrated SDE Strategy

A final sample design is a combination of the above options, which we call an **integrated SDE strategy**. This paper initially explores the impact of each option through extensive experimentation, then proposes an evidence-based integrated SDE strategy.

4 Experiments I: Evaluating The Impact of Each SDE Option

4.1 Settings

• **Tasks and Datasets.** For the Chinese online review MASA scenario, the data is provided and annotated by our collaborating company, which encounters a real-world business need. The data annotations come from two domains of aspect: **D1**, **D2**. We conduct experiments with both in-domain (ID) and out-of-domain (OOD) scenarios, testing model on domains that appear or not appear in training set, respectively. The models need to give a sentiment label from {*positive*, *neutral*, *negative*} for each aspect, while some aspects may not occur in the review. Based on the two domains, we construct 2 ID tasks (**D1⇒D1**, **D2⇒D2**) , and 2 OOD tasks (**D1⇒D2**, **D2⇒D1**). More details refer to A.2. Specific design examples can be found in A.3.

• **Models.** We utilize the following widely used open-source LLMs of 7B size : (1) *chinese-llama / alpaca-2-7b* (Cui et al., 2023) (note as **c-llama2-**

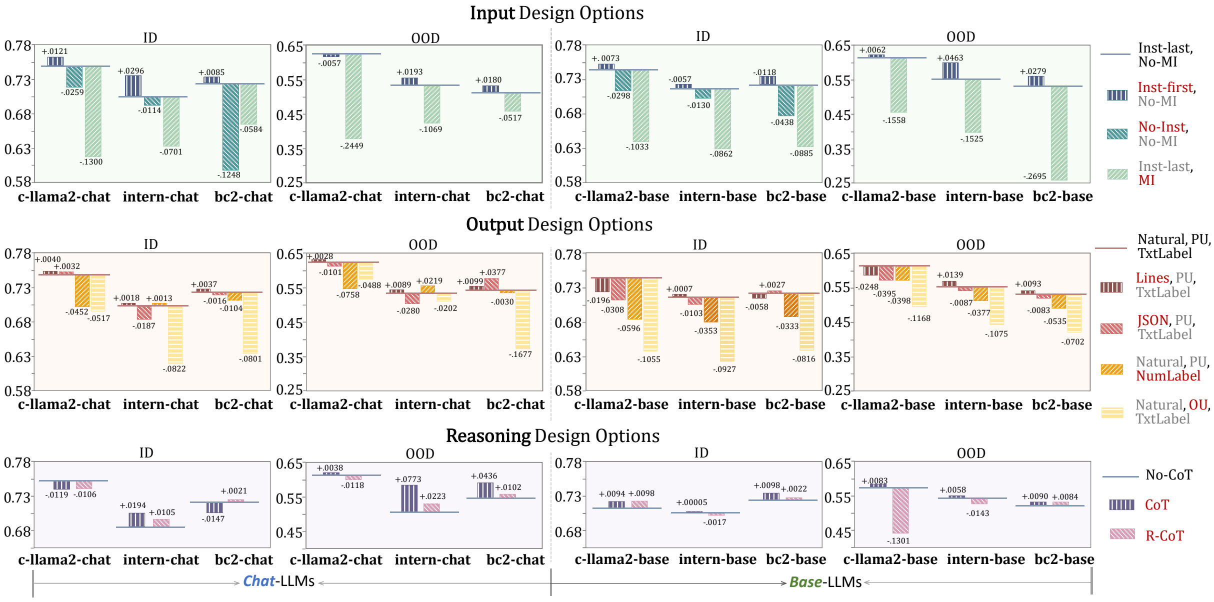

Figure 3: Sentiment analysis performances (κ) of different SDE options. Results of ID are the average of D1⇒D1 and D2⇒D2, same for OOD. The lines depict the performance of default options (baseline) in each group, and the bars depict each method's relative improvement or degradation compared to the baseline, with each method differing from the baseline in only one option (colored in red).

base / chat); (2) *internlm-7b-base / chat* (Team, 2023) (**intern-base / chat**); (3) *baichuan2-7b-base / chat* (Yang et al., 2023) (**bc2-base / chat**). We use LoRA as the default efficient fine-tuning technique. Hyperparameters and other training details can be found in Appendix A.2.

• **Evaluation Metrics.** We evaluate from two perspectives: (1) **Sentiment analysis performance**. We use the weighted Kappa score κ (Cohen, 1968) for this measurement considering the imbalance of different aspects and the ordinal nature of sentiment labels. (2) **Format adherence**, to assess the generation stability. Maintaining format adherence is vital for the subsequent utilization of LLM outputs. We track this with the format-parsing error rate. More details of metrics can be seen in Appendix A.1.

4.2 Experimental Results on Each Option

4.2.1 Sentiment Analysis Performance

We first assess the sentiment analysis performances of LLMs using different sample design options. The comparative results of ID and OOD tasks on 3 Chat-LLMs and 3 Base-LLMs are plotted in Figure 3 (full results see Table 3 to Table 8 in Appendix A.4). Some shared and intriguing patterns are revealed from the results.

Conclusions for Input Options:

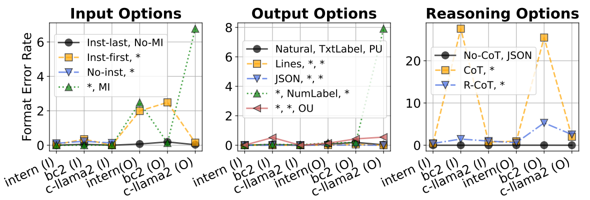

Figure 4: Format adherence performance, measured by parsing error rates (%). '*' means same option as above. I means ID, and O means OOD.

(1) **Instructions enhance DT.** *No-Inst* damages performance in ID tasks and OOD generalization ability. This underlines the importance of including instructions to enhance LLMs' comprehension and adaptability.

(2) **Better to place instruction first.** *Inst-first* outperforms *Inst-last* across both ID and OOD tasks for different LLMs. This demonstrates the significance of instruction placement for LLMs' tuning process. We hypothesize that this may partly be explained by the attention mechanism, see Appendix A.6.

(3) **Modeling input detracts from performance.** *MI* results in worse outcomes across various models and tasks. suggesting a cautious approach in determining which parts of the task to model.

Conclusions for Output Options:

(1) ***Lines* format is reliable for multiple pre-**

dictions. *Lines*, positioned between *Natural* and *JSON*, demonstrates stable and high performance across various models and tasks. It offers structured information while retains natural language readability, making it versatile for different LLMs.

(2) **Format preferences of Base/Chat models.** Base models show consistent responses across formats, while Chat models vary, implying differences in their SFT or RLHF data's structure. Moreover, Base models favor natural styles and are more affected by *NumLabel*, but Chat models are more accommodating to sophisticated or less natural formats, also benefit from the SFT and RLHF process.

(3) **Textual over numeric labels.** Numeric labels worsens performance, possibly due to lacking the descriptive depth and context clues that textual labels provide, which is crucial for LLMs.

(4) **Omitting the unmentioned targets may not be a good choice.** *OU*(Omit Unmentioned) may simplify outputs by omitting unmentioned aspects, but leads to inconsistency of aspects. This variability compels the models to adjust dynamically, increasing task complexity. *PU* (Placeholders for Unmentioned) keeps consistent by adding placeholders, perhaps making it easier for LLMs to learn. Additional analysis shows that the aspects with a higher degree of unmentioning suffer greater underperformance with *OU* compared to *PU*, see Appendix A.7.

Conclusions for Reasoning Options:

(1) **Subtle impact of CoT on ID, while significant on OOD tasks.** CoT design marginally affects ID tasks but markedly improves OOD performance. This contrast highlights CoT's role in enhancing model reasoning and adaptability in unfamiliar contexts, underpinning its value for generalization.

(2) **"Think before predict" beats "predict then explain".** The performance of *R-CoT*, which places the reasoning step after predicting, does not match that of *CoT*. However, *R-CoT* can still outperform *No-CoT* in many cases, suggesting that a single reasoning component is also beneficial.

4.2.2 Format Adherence Performance

Figure 4 presents the results of the format adherence performances for Chat-LLMs, from which we find the following conclusions:

(1) *Inst-first* improves sentiment analysis performance but reduces format stability, especially in OOD tasks, indicating that leading with instructions might increase format errors with unfamiliar content.

(2) Structured design options lead to better format adherence abilities: *JSON* > *Lines* > *Natural*. *JSON* format demonstrates strong adherence to the correct structure, highlighting a balance between output complexity and precision.

(3) *MI*, *NumLabel* and *CoT* can be quite unstable, which should be taken seriously in applications where stability is vital.

(4) Though improving the understanding or reasoning , CoT design puts LLMs at a higher risk of parsing failure for customized downstream tasks, underlining a trade-off for this option.

Considering LLMs' format adherence alongside the understanding abilities is crucial for specialized downstream applications, suggesting a need for a balanced approach in industrial scenarios.

5 Experiments II: A Robust Integrated SDE Strategy

Based on the experimental evidence from the previous section, we propose an **empirically strong SDE strategy** (termed as **ES-SDE**) using the well-performing options: a combination of *Inst-first*, *No-MI* input designs and *Lines*, *PU*(Placeholders for Unmentioned), *TxtLabel* output designs. We don't use the *CoT* design because of its high annotation cost and relatively unstable output.

In this section, we conduct comprehensive experiments to validate its effectiveness across different downstream tasks, as well as the robustness against perturbations in instructions or generation.

5.1 Settings

• **Tasks and datasets.** To evaluate the effectiveness of ES-SDE, we conduct experiments on three typical and challenging IE tasks:

GENIA (Ohta et al., 2002), a nested named entity recognition (Nested-NER) dataset in the molecular biology domain, where ChatGPT-3.5 only achieves an F1 score of 50.89% using 5-shot CoT reasoning (Han et al., 2023).

MAVEN (Wang et al., 2020), a general domain event detection (ED) dataset. Han et al. (2023) demonstrate that the performance of ChatGPT in ED tasks falls below expectations. We use the top-10 event types in our experiments.

Review11, our self-collected Chinese MASA

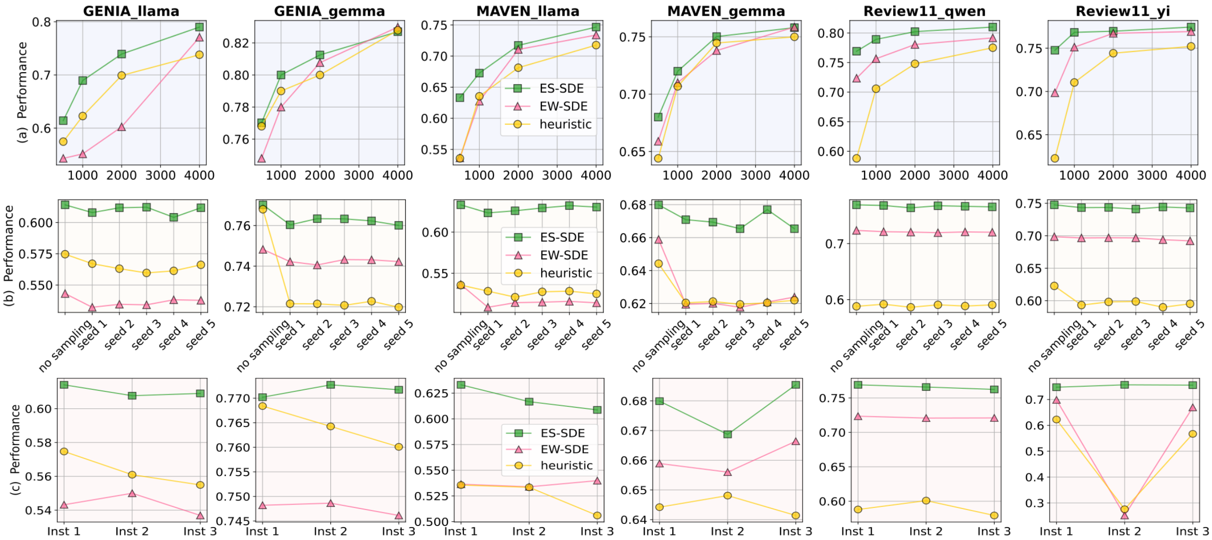

Figure 5: Comparison of different sample design strategies. (a) Performance of different sample design strategies with increasing training sizes: 500, 1000, 2000 and 4000. (b) Robustness on decoding sampling randomness, training size = 500. (c) Robustness on instruction content variation, training size = 500.

dataset that involves 11 aspects, more complicated than the MASA tasks in Section 4.

• **Baselines.** As a comparison to **ES-SDE**, we also propose an **empirically weak SDE strategy** (**EW-SDE**), combining the less effective options *Inst-last*, *Natural*, and *OU*(Omit Unmentioned) options, while keeping other options the same with ES-SDE. Note that ES-SDE and EW-SDE are both evidence-based strategies according to the previous empirical results, therefore, we also set up a **heuristic**-based baseline, referring to the prompt designs from the study of Han et al. (2023), which are similar to a combination of *Inst-first* and *OU* options, with a "lines-of-list" output format. Examples of these strategies see Appendix 11.

• ***Models.*** For a more generalized evaluation, we utilize four new LLMs. Considering the task language, the *llama2-7b-chat* (Touvron et al., 2023b) and *gemma2-9b-chat* (Team, 2024) are used for GENIA and MAVEN, and *qwen1.5-4b-chat* (Bai et al., 2023) and *yi1.5-6b-chat* (Young et al., 2024) are used for Review11. The training details are the same as Section 4.

5.2 Results

Figure 5 reports the comparison between different sample design strategies , from different perspectives . Soft-match F1 scores (Han et al., 2023) are reported for GENIA and MAVEN, and κ reported for Review11. More detailed results see Appendix A.5. Several key conclusions can be observed:

(1) ***ES-SDE maintains advantages across tasks and training sizes.*** Figure 5-(a) demonstrates that **ES-SDE** keeps its advantage as the training size increases, indicating the high quality of ES-SDE samples. Although the performance differences between designs are narrowed with large training size, ES-SDE achieves similar results with fewer training samples, facilitating fine-tuning with limited resources.

(2) ***Stable on decoding randomness.*** By default, the model employs a greedy decoding strategy (no sampling). Figure 5-(b) shows the results when activating decoding sampling with varying random seeds. **ES-SDE** maintains exceptional stability across different seeds compared with SW-SDE and heuristic strategies.

(3) ***Robust to instruction variation.*** We can use diverse expressions for the same instruction, so we validate how different strategies react to varied instruction phrasing (examples in Appendix 12). As shown in Figure 5-(c), ES-SDE keeps its edge in different variations, showing its robustness to instruction content.

Overall, **ES-SDE** represents a reliable and potent approach for the DT of LLMs, illustrating that—through a careful SDE process, LLMs can achieve much higher performances in downstream tasks. This method could also extend to other tasks requiring structured output. For example, analyzing financial reports with LLMs, which involves multi-dimensional understanding and forecasting,

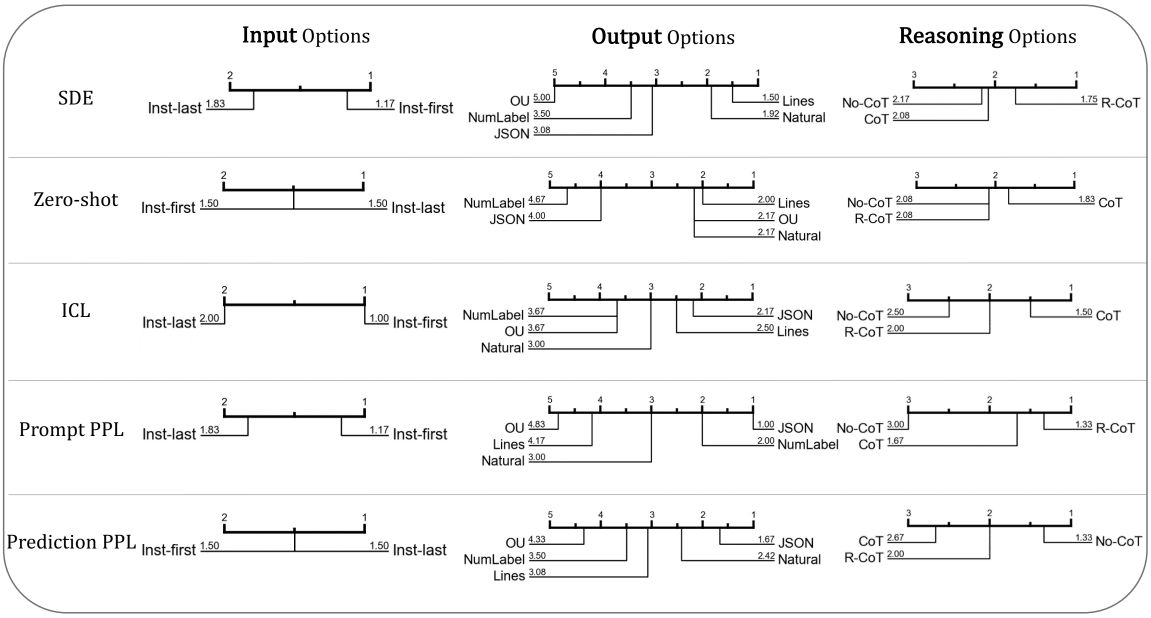

Figure 6: Average rankings of the DT performances of SDE options and zero-shot/ICL/PPL rankings of their corresponding prompts. Results based on the MASA ID tasks across 6 LLMs.

is not a typical IE task but is similar to our sample design considerations. Decisions like whether to use JSON or lines format for multi-dimensional predictions, or whether to use placeholders for missing dimensions, closely relate to our findings. We believe our conclusions are relevant and can be applied to analogous tasks beyond the scope of traditional IE. Note that ES-SDE may not be the best strategy for all cases. A detailed investigation into SDE across a broader spectrum of tasks and models could yield even more effective strategies.

6 Can PE guide SDE?

Effective PE can reveal a LLM's strengths and preferences. We explore if PE can guide SDE by crafting zero-shot and ICL prompts according to different SDE options. Figure 6 reports the average rankings of SDE options and their corresponding prompts in the MASA ID tasks, with detailed results in Appendix A.8.

For both PE and SDE evaluations, *Inst-first* and *CoT* works well. However, there are also many inconsistent patterns between PE and SDE, such as the performance of *OU*, and the comparison between *Natural* and *Lines*. Gonen et al. (2023) showed that the lower perplexity (PPL) generally leads to better prompt designs. Inspired by this, we conduct PPL analysis on the ICL prompts/predictions. There are also some discrepancies between the PPL scores and the performance in PE and SDE. For instance, *OU* has poor PPL scores, but performs well in zero-shot scenarios, and *JSON* shows weaker performance in SDE compared to *Lines*, despite its better PPL score.

These findings highlight a complex landscape where **prompt design patterns do not always align with SDE effectiveness**, underscoring the nuanced relationship between PE and SDE.

7 Conclusion

In this study, we introduce SDE as an effective method to enhance the downstream-tuning performances of LLMs on IE tasks. Through comprehensive ID and OOD experiments involving six LLMs, we demonstrate the effects of various sample design strategies, uncovering some interesting patterns that are consistent across different LLMs. Building on these findings, we develop the ES-SDE approach, which integrates the most effective options. Our experiments on three new tasks with four additional LLMs consistently show ES-SDE's superiority over baseline methods. Further analysis of the relationship between PE and SDE suggests that effective prompt designs do not necessarily translate to successful sample designs. This observation opens up avenues for more detailed investigations into the mechanisms of SDE in future research.

Limitations

This research follows a two-step experimental approach. In the first step, we investigate the impact of each SDE option, the results are then used as evidence for the second step—proposing an empirically strong SDE combination strategy. As an empirical study, this research is subject to certain limitations:

1. While we demonstrate that the experimental findings from the first phase are extendable to different downstream tasks, the applicability to other untested scenarios remains uncertain. For instance, although the *Lines* output design outperforms the *JSON* format in our current experiments, it is unclear if this advantage persists in more complex tasks with intricate structures. Future research will address these more challenging contexts;

2. With the rapid pace of advancements in LLMs, new and more sophisticated models are being introduced frequently. The models we used in our study were among the best open-source options available at the start of our research but have since been surpassed by newer releases. Although we assessed a total of 10 LLMs, including both base and chat variants, there remains a possibility that our findings may not be universally applicable to other models;

3. Combining different SDE options poses significant challenges, particularly without prior validation experiments such as those described in Section 4. The challenges are twofold. Firstly, unlike typical hyperparameters like learning rate or network layers, choosing different SDE options alters the training data itself, rendering traditional hyperparameter-tuning techniques such as Bayesian Optimization (Snoek et al., 2012) less practical. Secondly, evaluating LLMs on downstream tasks is both resource-intensive and costly, due to the need for customized task metrics, parsing rules, and high model inference costs. Therefore, developing a more efficient framework for SDE studies is a critical objective for future research.

Acknowlegements

We thank reviewers for their insightful feedback and comments. We would like to express sincere gratitude to Associate Professor Yun Chen of SHUFE for her valuable guidance and support during this research. We thank the financial support from the National Natural Science Foundation of China (No. 72271151, No. 72172085, No. 72342009), and FlagInfo-SHUFE Joint Laboratory.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Arie Ben-David. 2008. Comparison of classification accuracy using cohen's weighted kappa. *Expert Systems with Applications*, 34(2):825–832.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

J Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213–220.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Biyang Guo, Yeyun Gong, Yelong Shen, Songqiao Han, Hailiang Huang, Nan Duan, and Weizhu Chen. 2022. Genius: Sketch-based language model pre-training via extreme and selective masking for text generation and augmentation. *arXiv preprint arXiv:2211.10330*.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Seungone Kim, Se Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12685–12708.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023b. CodeIE: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the human language technology conference*, pages 73–77. Citeseer.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Gemma Team. 2024. Gemma.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. Deepstruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023b. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. Maven: A massive general domain event detection dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671.

Xingyao Wang, Sha Li, and Heng Ji. 2023c. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663. Association for Computational Linguistics.

Lucas Weber, Elsa M. Bruni Bruni, and Dieuwke Hupkes. 2023. Mind the instructions: a holistic evaluation of consistency and interactions in prompt-based learning. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 294–313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593, Mexico City, Mexico. Association for Computational Linguistics.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 190–200.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Ayfer Ezgi Yilmaz and Haydar Demirhan. 2023. Weighted kappa measures for ordinal multi-class classification performance. *Applied Soft Computing*, 134:110020.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023a. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023b. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Metrics for MASA

Weighted Kappa. Considering the imbalance of different aspects and the ordinal nature of labels, weighted agreement measures are proved to be more effective than traditional metrics (Ben-David, 2008; Galar et al., 2011; Grandini et al., 2020). Thus we adopt Weighted Kappa (Cohen, 1968; Yilmaz and Demirhan, 2023) as the measure of classification effect, which is an extension of Cohen's Kappa (Cohen, 1960). Weighted Kappa κ is defined as $\kappa = \frac{P_o - P_e}{1 - P_e}$, which measures a model's performance by considering how much better it performs than random guessing. Here, $P_o = \sum_{i,j=1}^{R} w_{ij} p_{ij}$ and $P_e = \sum_{i,j=1}^{R} w_{ij} p_{i.} p_{.j}$. The probabilities $p_{ij}, p_{i.}, p_{.j}$ are values or accumulated values from the classification confusion matrix. The weighting factor, w_{ij}, enables a nuanced assessment of different error degrees. For example, classifying "positive" as "negative" is more detrimental than classifying "positive" as "neutral," hence a higher penalty should be imposed on the former. Based on the feedback from enterprises in practical applications, we define the weight matrix without loss of generality as Table 1.

	Pre-Pos	Pre-Neu	Pre-Neg	Pre-Unm
Label-Pos	1	1/2	0	1/2
Label-Neu	2/3	1	2/3	2/3
Label-Neg	0	1/2	1	1/2
Label-Unm	1/2	2/3	1/2	1

Table 1: Weight matrix for calculating weighted Kappa.

Format adherence. Format adherence not only ensures that outputs from the model can be reliably parsed and utilized in practical applications, but also reflects the model's ability to understand the context and the nuances of different instructions. We set up parsers according to the prescribed formats of different designs, then we calculate the ratio of predictions that cannot be successfully parsed with our output parser. Considering the inherently uncertainty nature of generative language models, we relaxed the format such as the expression of aspects and sentiments. Meanwhile, in order to compare the content correctness between designs more fairly, for some cases such as common punctuation errors, we will correct it into the required format when calculating the Kappa. If a certain aspect can still not be parsed correctly, this aspect is treated as "unmentioned". Figure 10 shows a variety of representative format error types and how they are processed by the parsers we design.

A.2 Datasets and Training Settings

The data annotations come from two domains of aspects: **D1** about food, beverage, price, hygiene, staff attitude, and parking convenience and **D2** about traffic convenience, queuing, serving speed, decoration, and noise. Figure 7 is an example of the MASA task on **D1**.

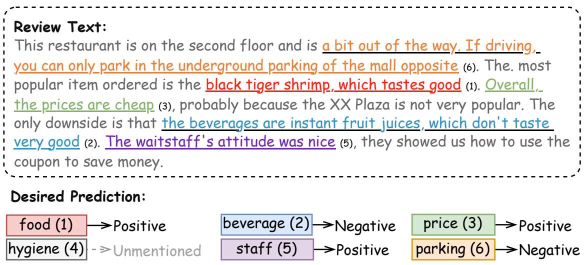

Figure 7: An example for the MASA task.

Considering the high cost of annotation in industries and the fact that fine-tuning LLMs requires less annotated data (Zhou et al., 2024), we train the model with 500 and 1,000 samples, respectively. We use a large test set containing around 8,000 samples to make results more stable and convincing. Table 2 shows the label distribution of each aspect for two domains **D1** and **D2**, where we can see the distributions are highly unbalanced.

The training setup was as follows: learning rate set to 1e-4, batch size of 4, LoRA rank of 8 LoRA alpha of 32, LoRA dropout of 0.1. In the generation phase, the hyperparameter 'max new tokens' is set to 200 for input design options and output design options, while for reasoning design options, it is set to 400. For the same model, the other generation parameters of different designs are kept consistent.

A.3 Sample Design Examples

Figure 9 shows a detailed example of our sample designs on MASA tasks.

A.4 Detailed Evaluations of Each SDE Option

The detailed results of in-domain (ID) and out-of-domain (OOD) evaluations on the MASA task of different SDE options across six LLMs are shown in Table 3 to Table 8, including both the sentiment analysis performances (κ) and the format adherence performances (format error rate). An averaged results of training size 500 and 1000 of ID and OOD scenarios are visualized in Figure 3.

A.5 Detailed Results on GENIA, MAVEN and Review11

Table 9 shows the comparison of different sample design strategies on three downstream tasks—GENIA (Nested NER), MAVEN (Event Detection), and Review11 (MASA). Hard and soft-matching F1 scores are reported for GENIA and MAVEN, while kappa κ and accuracy are reported for Review11. From the results, we can see that ES-SDE maintains its advantage over other methods, across different tasks and training sizes.

Table 10 illustrates the performances of different sample design strategies on three downstream tasks across different instruction variations.

A.6 Additional Analysis on *Inst-last* and *Inst-first*

The experimental results showing that *Inst-first* consistently outperforms *Inst-last* across various tasks and models are thought-provoking, leading us to conduct a more in-depth analysis. We extract the attention weights related to some task-related fields in the instruction, and sum up these task-related

		TrainSet (size=500)				TrainSet (size=1000)				TestSet			
		Pos	Neu	Neg	Unm	Pos	Neu	Neg	Unm	Pos	Neu	Neg	Unm
D1	F	65.20	15.00	18.80	1.00	66.60	13.70	18.30	1.40	66.01	12.23	20.12	1.64
	B	22.20	4.20	8.20	65.40	23.50	3.60	7.20	65.70	21.50	3.15	6.29	69.07
	P	33.40	13.00	15.60	38.00	35.60	10.70	15.80	37.90	36.64	10.24	13.97	39.15
	H	14.80	1.20	6.00	78.00	17.10	1.00	5.50	76.40	16.12	0.82	5.58	77.48
	SA	48.80	3.60	14.00	33.60	47.90	4.10	13.60	34.40	42.73	3.46	13.87	39.94
	PC	4.40	0.60	1.40	93.60	4.80	0.30	1.90	93.00	3.93	0.34	1.56	94.18
D2	TC	52.40	13.20	7.60	26.80	53.10	13.20	8.10	25.60	48.56	12.84	7.03	31.57
	Q	18.80	8.20	11.20	61.80	17.90	10.10	11.00	61.00	14.67	10.00	10.44	64.89
	SS	16.80	3.60	8.20	71.40	15.70	3.80	8.90	71.60	14.86	3.15	8.58	73.41
	D	46.00	8.20	4.20	41.60	48.50	8.10	4.30	39.10	43.10	7.68	5.28	43.93
	N	1.00	1.40	2.80	94.80	1.40	1.30	3.40	93.90	2.10	1.08	3.36	93.46

Table 2: Label distribution(%) in various aspects of train set and test set. **D1** contains annotations for 6 aspects—food (F), beverage (B), price (P), hygiene (H), staff attitude (SA), and parking convenience (PC); **D2** contains annotations for 5 different aspects—traffic convenience (TC), queuing (Q), serving speed (SS), decoration (D), and noise (N). We use 'Pos', 'Neu', 'Neg', 'Unm' to represent Positive, Neutral, Negative and Unmentioned labels, respectively.

attention weights for each token. Figure 8 shows the comparison of the attention weights for a certain customer review. As we can see, **tokens that are closer to the instruction usually get higher task-related attention weights**. Intuitively, when people write reviews, they generally present their core opinions at the beginning. This leads to the possibility that if the instructions are placed at the front, those core parts may receive greater task-related attention weights. This may partly explain why *Inst-first* usually leads to a higher sentiment analysis performance.

A.7 Additional Analysis on *OU* and *PU*

In previous experiments, we found that *OU* performs much worse than *PU*. This intriguing result motivates us to a further analysis. Specifically, we calculate and compare the kappa scores of *OU* and *PU* for each aspect, to analyze the relationship between label distributions and the effect of *OU*.

From the result in Table 11, we can observe that when training the model with 500 samples, for aspects with a higher number of unmentioned, the *OU* method showed a significant gap compared to the *PU* format. When the training set increased to 1000 samples, this gap noticeably narrowed. This suggests that for the *OU* method, aspects with more unmentioned, implying less frequent occurrence in answers, are harder for the model to learn, so requiring more data. From another perspective, it also indicates that even if a certain aspect is not covered in the text, mentioning this aspect in the answers can enhance the model's understanding of it.

A.8 Can PE Guide SDE? Detailed Results

Evaluating the performances of sample designs involves fine-tuning models on downstream tasks, which can be time-consuming. Therefore, we also pondered whether it might be possible to design better samples without training models first. We tried to understand the inherent capabilities and potential of the model by experimenting with different prompt designs in both the zero-shot and in-context learning scenarios.

A.8.1 Zero-shot and In-context Learning Analysis

Zero-shot and In-context learning ability can directly reveal LLMs' familiarity with the given task. In the zero-shot approach, we use the input (which contains the instruction on output format) from each SDE option as the prompt for the original frozen LLMs prediction. For the ICL approach, we add two fixed examples from the training set before each test instance. Considering the inference time cost caused by the increase in sample length, we limit our prediction and analysis to 500 samples. All other experimental setups remain aligned with those described in Experiments I.

Zero-shot Study. All six 7B LLMs used in Section 4 exhibit poor zero-shot MASA ability, failing to follow the instructions to generate proper output in most cases, as shown in Table 13, making it hard to analysis its relationship with SDE results. Variations in format preferences across different models are observed, which we conjecture is strongly related to the datasets employed for instruction

model: **c-llama2-chat**		**Weighted Kappa κ**				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8091	0.6882	0.5243	0.7217	0	0	2	2
	Inst-first, _	0.8136	0.7079	0.5124	0.7223	0	0	9	15
	No-inst, _	0.7757	0.6626	\	\	20	1	\	\
	_, MI	0.6187	0.6187	0.4806	0.2756	1	0	0	1079
Output	Natural, TxtLabel, PU	0.8091	0.6882	0.5243	0.7217	0	0	2	2
	Lines, _, _	0.8083	0.6969	0.5068	0.7447	0	0	0	0
	JSON, _, _	0.8086	0.6952	0.4905	0.7354	0	0	0	0
	_, NumLabel, _	0.7697	0.6373	0.4221	0.6723	3	1	0	1260
	_, _, OU	0.7934	0.6005	0.5282	0.6203	0	0	87	0
Reasoning	No-CoT	0.8086	0.6952	0.4905	0.7354	0	0	0	0
	CoT	0.7928	0.6873	0.5249	0.7085	56	65	36	282
	R-CoT	0.8074	0.6752	0.4726	0.7297	93	65	141	263
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8256	0.7110	0.5518	0.7312	0	0	0	3
	Inst-first, _	0.8236	0.7090	0.5483	0.7264	0	0	5	1
	No-inst, _	0.8003	0.6920	\	\	6	4	\	\
	_, MI	0.8113	0.6700	0.5095	0.5182	0	0	0	728
Output	Natural, TxtLabel, PU	0.8256	0.7110	0.5518	0.7312	0	0	0	3
	Lines, _, _	0.8259	0.7118	0.5560	0.7452	0	0	0	0
	JSON, _, _	0.8249	0.7094	0.5488	0.7432	0	0	0	0
	_, NumLabel, _	0.7624	0.6604	0.4210	0.6840	2	2	0	765
	_, _, OU	0.8172	0.7125	0.5511	0.6746	0	0	493	1
Reasoning	No-CoT	0.8249	0.7094	0.5488	0.7432	0	0	0	0
	CoT	0.8111	0.7111	0.5354	0.7311	59	24	30	253
	R-CoT	0.8214	0.7137	0.5085	0.7532	51	25	75	115

Table 3: MASA evaluations of each SDE option for model **c-llama2-chat**. The first method in each group is the group baseline. "_" means keeping the same option with the group baseline.

model: **c-llama2-base**		**Weighted Kappa κ**				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8067	0.6801	0.5246	0.7000	0	0	6	98
	Inst-first, _	0.8092	0.6921	0.5575	0.6794	0	0	34	3
	No-inst, _	0.7762	0.6511	\	\	0	1	\	\
	_, MI	0.7778	0.5024	0.4946	0.4184	2	0	118	0
Output	Natural, TxtLabel, PU	0.8067	0.6801	0.5246	0.7000	0	0	6	98
	Lines, _, _	0.8066	0.6410	0.5128	0.6622	0	0	19	0
	JSON, _, _	0.8010	0.6242	0.5170	0.6287	0	0	0	0
	_, NumLabel, _	0.7728	0.5949	0.5155	0.6296	14	1	26	356
	_, _, OU	0.7746	0.5012	0.4199	0.5711	0	3	300	7
Reasoning	No-CoT	0.8010	0.6242	0.5170	0.6287	0	0	0	0
	CoT	0.7789	0.6652	0.4649	0.6974	83	82	33	226
	R-CoT	0.8019	0.6428	0.4657	0.4199	88	11	87	1823
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8237	0.7011	0.6010	0.7197	0	0	3	177
	Inst-first, _	0.8231	0.7068	0.6069	0.6956	0	2	16	28
	No-inst, _	0.7957	0.6882	\	\	2	2	\	\
	_, MI	0.8048	0.6174	0.5306	0.6390	0	3	139	6
Output	Natural, TxtLabel, PU	0.8237	0.7011	0.6010	0.7197	0	0	3	177
	Lines, _, _	0.8205	0.6947	0.5900	0.6963	0	0	10	0
	JSON, _, _	0.8212	0.6857	0.5649	0.6875	0	0	0	0
	_, NumLabel, _	0.7619	0.6536	0.4804	0.6709	1	2	0	584
	_, _, OU	0.8179	0.6774	0.5034	0.6277	0	5	64	29
Reasoning	No-CoT	0.8212	0.6857	0.5649	0.6875	0	0	0	0
	CoT	0.8026	0.6979	0.5519	0.7159	70	31	16	125
	R-CoT	0.8195	0.7034	0.5368	0.6454	46	14	24	666

Table 4: MASA evaluations of each SDE option for model **c-llama2-base**. Definition of "_" see Table 3.

model: **intern-chat**		**Weighted Kappa** κ				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.7774	0.6278	0.3947	0.6707	0	0	0	11
	Inst-first, _	0.8035	0.6609	0.3949	0.7090	4	2	13	304
	T2L	0.7862	0.5963	\	\	10	7	\	\
	_, MI	0.7463	0.5178	0.3153	0.5363	0	0	0	395
Output	Natural, TxtLabel, PU	0.7774	0.6278	0.3947	0.6707	0	0	0	11
	Lines, _, _	0.7827	0.6261	0.4032	0.6799	0	1	1	1
	JSON, _, _	0.7713	0.5966	0.3965	0.6129	0	0	0	2
	_, NumLabel, _	0.7765	0.6261	0.4165	0.6926	0	0	3	23
	_, _, OU	0.7520	0.4888	0.4029	0.6221	0	1	16	7
Reasoning	No-CoT	0.7713	0.5966	0.3965	0.6129	0	0	0	2
	CoT	0.7666	0.6401	0.4843	0.6797	43	19	30	121
	R-CoT	0.7764	0.6124	0.3892	0.6648	44	23	23	72
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8049	0.6793	0.4330	0.6982	0	0	0	0
	Inst-first, _	0.8173	0.7125	0.4640	0.7343	0	1	6	259
	No-inst, _	0.8139	0.6811	\	\	8	5	\	\
	_, MI	0.7819	0.6256	0.3332	0.6520	1	0	8	29
Output	Natural, TxtLabel, PU	0.8049	0.6793	0.4330	0.6982	0	0	0	0
	Lines, _, _	0.8060	0.6797	0.4498	0.7038	0	1	0	1
	JSON, _, _	0.8021	0.6649	0.4661	0.6647	0	0	0	0
	_, NumLabel, _	0.8081	0.6764	0.4393	0.7286	0	0	3	3
	_, _, OU	0.8008	0.6369	0.4374	0.6694	0	0	33	1
Reasoning	No-CoT	0.8021	0.6649	0.4661	0.6647	0	0	0	0
	CoT	0.7981	0.6966	0.5190	0.7098	36	7	10	132
	R-CoT	0.8043	0.6709	0.3994	0.7195	50	4	19	42

Table 5: MASA evaluations of each SDE option for model **intern-chat**. Definition of "_" see Table 3.

model: **intern-base**		**Weighted Kappa** κ				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.7849	0.6465	0.4898	0.6129	0	1	1	0
	Inst-first, _	0.7955	0.6472	0.4947	0.7006	3	8	18	221
	No-inst, _	0.7936	0.6119	\	\	11	6	\	\
	_, MI	0.7562	0.5029	0.3305	0.4672	0	1	232	447
Output	Natural, TxtLabel, PU	0.7849	0.6465	0.4898	0.6129	0	1	1	0
	Lines, _, _	0.7873	0.6455	0.4939	0.6365	0	2	4	0
	JSON, _, _	0.7859	0.6250	0.4727	0.6127	0	0	3	82
	_, NumLabel, _	0.7605	0.6003	0.3861	0.6412	14	3	10	102
	_, _, OU	0.7275	0.5185	0.3943	0.4935	0	4	48	6
Reasoning	No-CoT	0.7859	0.6250	0.4727	0.6127	0	0	3	82
	CoT	0.7621	0.6489	0.4581	0.6388	77	12	2347	50
	R-CoT	0.7734	0.6342	0.3752	0.6816	141	49	1496	206
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8112	0.6874	0.5216	0.7065	1	0	0	0
	Inst-first, _	0.8167	0.6965	0.5195	0.7544	0	0	5	46
	No-inst, _	0.8191	0.6963	\	\	5	8	\	\
	_, MI	0.7937	0.6238	0.2780	0.6492	0	2	383	45
Output	Natural, TxtLabel, PU	0.8112	0.6874	0.5216	0.7065	1	0	0	0
	Lines, _, _	0.8113	0.6919	0.5060	0.7126	0	0	3	0
	JSON, _, _	0.8076	0.6781	0.5195	0.6817	0	0	3	1
	_, NumLabel, _	0.8084	0.6776	0.4426	0.7139	3	1	31	20
	_, _, OU	0.8006	0.6330	0.4587	0.6098	0	1	30	3
Reasoning	No-CoT	0.8076	0.6781	0.5195	0.6817	0	0	3	1
	CoT	0.7956	0.6874	0.5196	0.6903	34	12	405	56
	R-CoT	0.8069	0.6725	0.4890	0.7185	46	11	220	125

Table 6: MASA evaluations of each SDE option for model **intern-base**. Definition of "_" see Table 3.

model: **bc2-chat**		**Weighted Kappa** κ				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.7904	0.6544	0.4067	0.6170	8	0	21	10
	Inst-first, _	0.7958	0.6660	0.3858	0.6739	19	36	12	385
	No-inst, _	0.7176	0.4776	\	\	23	13	\	\
	_, MI	0.7645	0.5636	0.3713	0.5490	0	0	5	16
Output	Natural, TxtLabel, PU	0.7904	0.6544	0.4067	0.6170	8	0	21	10
	Lines, _, _	0.7869	0.6653	0.4091	0.6344	0	0	9	1
	JSON, _, _	0.7927	0.6489	0.4714	0.6196	0	0	1	0
	_, NumLabel, _	0.7839	0.6401	0.3671	0.6506	5	4	12	17
	_, _, OU	0.7016	0.5670	0.3599	0.3285	2	81	50	19
Reasoning	No-CoT	0.7927	0.6489	0.4714	0.6196	0	0	1	0
	CoT	0.7722	0.6400	0.5006	0.6776	3641	757	739	3323
	R-CoT	0.7922	0.6535	0.4534	0.6579	107	126	280	563
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8113	0.7060	0.4709	0.6365	0	4	13	18
	Inst-first, _	0.8142	0.7095	0.4733	0.6787	31	12	21	136
	No-inst, _	0.7466	0.6172	\	\	6	6	\	\
	_, MI	0.7935	0.6514	0.3951	0.5885	0	0	7	3
Output	Natural, TxtLabel, PU	0.8113	0.7060	0.4709	0.6365	0	4	13	18
	Lines, _, _	0.8103	0.7057	0.4691	0.6387	0	0	3	0
	JSON, _, _	0.8118	0.7064	0.5237	0.6323	0	0	1	0
	_, NumLabel, _	0.8121	0.6962	0.4042	0.6697	10	17	4	15
	_, _, OU	0.8061	0.6467	0.4843	0.5155	1	25	44	4
Reasoning	No-CoT	0.8118	0.7064	0.5237	0.6323	0	0	1	0
	CoT	0.7995	0.7026	0.4992	0.6975	2273	193	560	2043
	R-CoT	0.8087	0.6961	0.5022	0.6772	57	48	85	167

Table 7: MASA evaluations of each SDE option for model **bc2-chat**. Definition of "_" see Table 3.

model: **bc2-base**		**Weighted Kappa** κ				**# Wrong format** (7969 test samples in total)			
train_size=500		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8017	0.6412	0.4441	0.6146	0	0	75	0
	Inst-first, _	0.8016	0.6649	0.4488	0.6657	0	6	27	4
	No-inst, _	0.7533	0.6020	\	\	2	3	\	\
	_, MI	0.7660	0.4999	0.3220	0.1978	0	0	1	164
Output	Natural, TxtLabel, PU	0.8017	0.6412	0.4441	0.6146	0	0	75	0
	Lines, _, _	0.7996	0.6317	0.4583	0.6191	0	0	2	0
	JSON, _, _	0.8008	0.6476	0.4316	0.6104	0	0	0	0
	_, NumLabel, _	0.7969	0.5794	0.4312	0.5206	7	45	469	47
	_, _, OU	0.7595	0.5202	0.4240	0.4944	0	0	116	2
Reasoning	No-CoT	0.8008	0.6476	0.4316	0.6104	0	0	0	0
	CoT	0.7865	0.6814	0.3854	0.6745	63	17	43	483
	R-CoT	0.7980	0.6548	0.4240	0.6349	32	44	39	32
train_size=1000		D1→D1	D2→D2	D1→D2	D2→D1	D1→D1	D2→D2	D1→D2	D2→D1
Input	Inst-last, No-MI	0.8143	0.6981	0.4747	0.6767	0	0	26	4
	Inst-first, _	0.8155	0.7157	0.5061	0.6974	0	3	26	4
	No-inst, _	0.7543	0.6391	\	\	0	3	\	\
	_, MI	0.8010	0.6489	0.4164	0.5250	0	0	1	431
Output	Natural, TxtLabel, PU	0.8143	0.6981	0.4747	0.6767	0	0	26	4
	Lines, _, _	0.8103	0.7003	0.4732	0.6713	0	0	6	1
	JSON, _, _	0.8120	0.7039	0.4785	0.6819	0	0	0	0
	_, NumLabel, _	0.8119	0.6812	0.4575	0.6467	1	5	292	8
	_, _, OU	0.7894	0.6484	0.4031	0.6235	0	1	31	0
Reasoning	No-CoT	0.8120	0.7039	0.4785	0.6819	0	0	0	0
	CoT	0.8045	0.7063	0.5319	0.6965	21	12	25	494
	R-CoT	0.8160	0.7021	0.4604	0.6949	15	14	24	115

Table 8: MASA evaluations of each SDE option for model **bc2-base**. Definition of "_" see Table 3.

LLM training size	Strategies	**GENIA** (Nested-NER) llama2-7b-chat F1-hard	 llama2-7b-chat F1-soft	 gemma2-9b-it F1-hard	 gemma2-9b-it F1-soft	**MAVEN** (ED) llama2-7b-chat F1-hard	 llama2-7b-chat F1-soft	 gemma2-9b-it F1-hard	 gemma2-9b-it F1-soft	**Review11** (MASA) Qwen-4b-chat κ	 Qwen-4b-chat Acc	 Yi1.5-6b-chat κ	 Yi1.5-6b-chat Acc
500	heuristic	0.5123	0.5747	0.7128	0.7684	0.5197	0.5356	0.6269	0.6442	0.5880	0.7586	0.6227	0.7811
	EW-SDE	0.4833	0.5432	0.6869	0.7482	0.4922	0.5364	0.5394	0.6589	0.7235	0.8327	0.6985	0.8172
	ES-SDE	**0.5407**	**0.6141**	**0.7127**	**0.7702**	**0.5846**	**0.6331**	**0.6662**	**0.6799**	**0.7691**	**0.8626**	**0.7476**	**0.8475**
1, 000	heuristic	0.5654	0.6228	0.7430	0.7955	0.6237	0.6354	0.6987	0.7068	0.7058	0.8262	0.7104	0.8254
	EW-SDE	0.4879	0.5517	0.7259	0.7805	0.6109	0.6275	0.5789	0.7116	0.7565	0.8502	0.7512	0.8471
	ES-SDE	**0.6159**	**0.6895**	**0.7407**	**0.7977**	**0.6432**	**0.6726**	**0.7066**	**0.7167**	**0.7892**	**0.8716**	**0.7683**	**0.8575**
2, 000	heuristic	0.6476	0.6990	**0.7617**	0.8101	0.6722	0.6813	0.7335	0.7446	0.7479	0.8483	0.7442	0.8461
	EW-SDE	0.5435	0.6025	0.7571	0.8077	0.6966	0.7106	0.6144	0.7381	0.7805	0.8649	0.7672	0.8580
	ES-SDE	**0.6807**	**0.7393**	0.7593	**0.8125**	**0.7033**	**0.7172**	**0.7392**	**0.7502**	**0.8023**	**0.8785**	**0.7696**	**0.8589**
4, 000	heuristic	0.6873	0.7383	**0.7804**	0.8279	0.7118	0.7176	0.7418	0.7503	0.7751	0.8644	0.7521	0.8494
	EW-SDE	0.7111	0.7709	0.7781	**0.8299**	0.7265	0.7338	0.6367	**0.7585**	0.7917	0.8715	0.7692	0.8570
	ES-SDE	**0.7273**	**0.7849**	0.7758	0.8265	**0.7295**	**0.7466**	**0.7461**	0.7577	**0.805**	**0.8814**	**0.7744**	**0.8618**

Table 9: Comparison of different sample design strategies on three downstream tasks. In most cases, ES-SDE has advantages over other designs on different tasks and training scales.

LLM Instruction Variation	Strategies	**GENIA** (Nested-NER) llama2-7b-chat F1-hard	 llama2-7b-chat F1-soft	 gemma2-9b-it F1-hard	 gemma2-9b-it F1-soft	**MAVEN** (ED) llama2-7b-chat F1-hard	 llama2-7b-chat F1-soft	 gemma2-9b-it F1-hard	 gemma2-9b-it F1-soft	**Review11** (MASA) Qwen-4b-chat κ	 Qwen-4b-chat Acc	 Yi1.5-6b-chat κ	 Yi1.5-6b-chat Acc
inst-1	heuristic	0.5123	0.5747	0.7128	0.7684	0.5197	0.5356	0.6269	0.6442	0.5880	0.7586	0.6227	0.7811
	EW-SDE	0.4833	0.5432	0.6869	0.7482	0.4922	0.5364	0.5394	0.6589	0.7235	0.8327	0.6985	0.8172
	ES-SDE	**0.5407**	**0.6141**	**0.7127**	**0.7702**	**0.5846**	**0.6331**	**0.6662**	**0.6799**	**0.7691**	**0.8626**	**0.7476**	**0.8475**
inst-2	heuristic	0.4981	0.5610	0.7096	0.7643	0.5134	0.5334	0.6347	0.6481	0.6009	0.7685	0.2756	0.3803
	EW-SDE	0.4859	0.5500	0.6915	0.7486	0.4956	0.5339	0.5252	0.6560	0.7208	0.8344	0.2515	0.4437
	ES-SDE	**0.5348**	**0.6077**	**0.7170**	**0.7727**	**0.5636**	**0.6167**	**0.6578**	**0.6687**	**0.7659**	**0.8615**	**0.7568**	**0.8560**
inst-3	heuristic	0.4873	0.5549	0.7054	0.7601	0.4940	0.5060	0.6306	0.6414	0.5793	0.7533	0.5671	0.7116
	EW-SDE	0.4764	0.5369	0.6863	0.7461	0.4925	0.5399	0.5416	0.6664	0.7210	0.8365	0.6696	0.807
	ES-SDE	**0.5353**	**0.6090**	**0.7147**	**0.7717**	**0.5530**	**0.6087**	**0.6748**	**0.6854**	**0.7624**	**0.8601**	**0.7556**	**0.8581**

Table 10: Performances of different sample design strategies on three downstream tasks across different instruction variations.

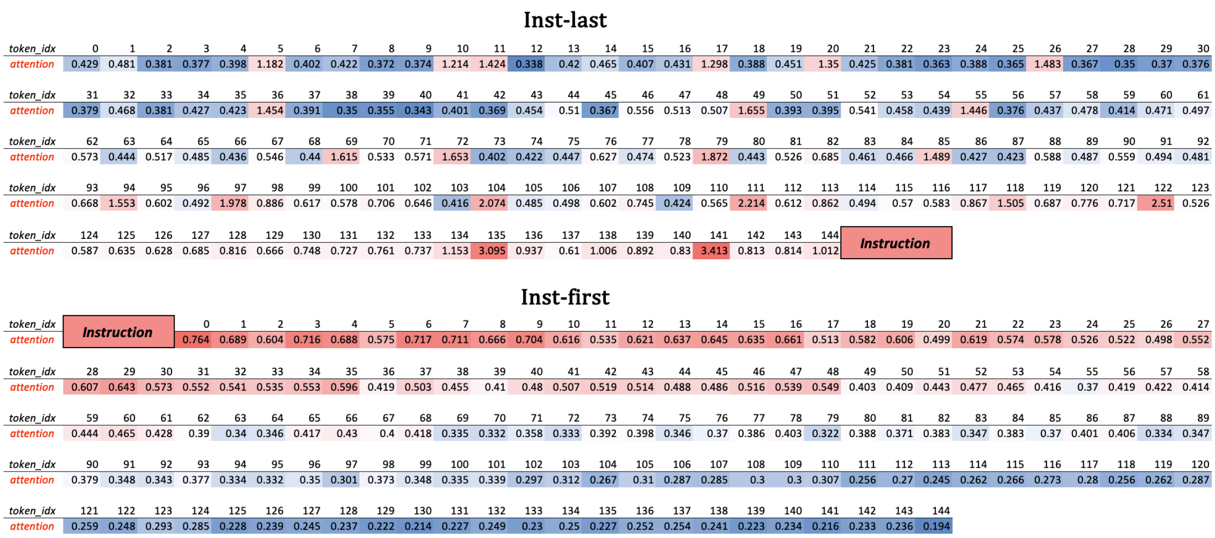

Figure 8: Comparison of task-related attention scores using *Inst-last* and *Inst-first*.

Aspect	Trainsize=500			Trainsize=1000		
	(%)Num_ Unmen	$\Delta\kappa$ Avg_Chat	Avg_Base	(%)Num_ Unmen	$\Delta\kappa$ Avg_Chat	Avg_Base
D1 F	1.00	-.0004	.0007	1.40	-.0026	-.0011
SA	33.60	-.0687	-.0555	34.40	-.0062	-.0212
P	38.00	-.0469	-.0495	37.90	-.0068	-.0255
B	65.40	-.0410	-.0291	65.70	-.0117	-.0079
H	78.00	-.0920	-.1367	76.40	-.0033	-.0207
PC	93.60	-.2338	-.2590	93.00	-.0181	-.0305
D2 TC	26.80	-.0891	-.1341	25.60	-.0497	-.0492
D	41.60	-.1106	-.2475	39.10	-.0280	-.0500
Q	61.80	-.0329	-.0588	61.00	-.0361	-.0149
SS	71.40	-.2537	-.2575	71.60	-.0574	-.0896
N	94.80	-.3347	-.3954	93.90	-.0494	-.1405

Table 11: Number of 'Unmentioned' labels and average $\Delta\kappa$ (κ_{OU}-κ_{PU}) for different aspects.

fine-tuning in each model. Some patterns are also contradictory between zero-shot and SDE. For example, the *OU* SDE option consistently harms DT performances, however, its prompts result in notably fewer format errors in zero-shot inference, for certain LLMs. Therefore, zero-shot performances can hardly tell good or bad SDE options.

In-context Learning Study. ICL can effectively improve LLMs' instruction-following abilities resulting in far fewer formatting errors than zero-shot. Therefore we report the average sentiment analysis performances of each model on two domains in Table 14. The results suggest that *Inst-first* and *CoT* enhance the performance of most models, which provides valuable insights for format selection during the fine-tuning process. For output designs, *JSON* and *OU* options outperform the other approaches for some models, differing from the SDE results.

A.8.2 Perplexity Analysis

Perplexity measures the uncertainty of the model in generating a given text sequence (Chen et al., 1998), with lower perplexity values indicating more confident predictions by the model. In calculations, we estimate perplexity using the common practice of taking the logarithm of the model's loss.

In our task, we compare the PPL scores of the ICL prompts corresponding to each different SDE option, as well as the conditional PPL of the models' ICL predictions. For predictions, we concatenate the prompt and the prediction together as a sequence, then consider the prompt as its context.

The perplexity results for different designs are shown in Table 12. For input designs, the PPL score of *Inst-first* option is lower than that of *Inst-last* in general, which is consistent with the conclusion that *Inst-first* performs better in ICL and SDE experiments. For output designs, the *OU* option gets the highest score, which is inconsistent with its performance on the ICL, but is consistent with its being the worst option in the SDE experiment. Surprisingly, the *JSON* format achieved the significantly lowest ppl score, but it was on par with the *Lines* format in ICL and even worse than *Lines* in SDE. The most interesting result appears in the reasoning designs. The *CoT* and *R-CoT* options have low PPL scores on prompts but have high scores on predictions conversely. Such contradictions make it difficult to analyze the results of ICL or SDE through PPL scores.

The analysis above also highlights the indispensability of our SDE experiments, cause we cannot predetermine the final effectiveness of different designs through preliminary analysis alone.

Perplexity:Prompts		c-llama2-chat	c-llama2-base	intern-chat	intern-base	bc2-chat	bc2-base
Input	Inst-last, No-MI	47.662	111.063	18.422	19.036	59.046	42.030
	Inst-first, _	46.357	110.065	19.561	18.632	54.795	39.003
Output	Natural, TxtLabel, PU	47.662	111.063	18.422	19.036	59.046	42.030
	Lines, _, _	47.918	191.274	18.561	19.219	60.498	42.638
	JSON, _, _	29.008	78.848	14.675	13.260	38.547	25.405
	_, NumLabel, _	41.690	92.717	17.664	16.348	51.963	35.185
	_, _, OU	55.345	129.055	20.862	21.450	69.022	49.426
Reasoning	No-CoT	29.008	78.848	14.675	13.260	38.547	25.405
	CoT	18.263	41.312	10.812	9.379	23.406	15.267
	R-CoT	18.210	42.648	10.789	9.354	22.671	15.333
Perplexity:Predictions		c-llama2-chat	c-llama2-base	intern-chat	intern-base	bc2-chat	bc2-base
Input	Inst-last, No-MI	1.052	1.109	1.051	1.394	1.061	1.127
	Inst-first, _	1.088	1.284	1.046	1.360	1.066	1.113
Output	Natural, TxtLabel, PU	1.052	1.109	1.051	1.394	1.061	1.127
	Lines, _, _	1.052	1.137	1.058	1.386	1.222	1.136
	JSON, _, _	1.038	1.074	1.045	1.407	1.019	1.042
	_, NumLabel, _	1.096	1.142	1.078	1.403	1.088	1.102
	_, _, OU	1.183	1.368	1.089	1.279	1.353	1.823
Reasoning	No-CoT	1.038	1.074	1.045	1.407	1.019	1.042
	CoT	1.234	1.475	1.084	1.186	1.090	1.129
	R-CoT	1.239	1.293	1.069	1.185	1.063	1.090

Table 12: The PPL scores on the ICL prompts and predictions corresponding to each SDE options on the MASA ID tasks.

		c-llama2-chat		**Intern-chat**		**bc2-chat**		**c-llama2-base**		**Intern-base**		**bc2-base**	
		D1	D2	D1	D2	D1	D2	D1	D2	D1	D2	D1	D2
Input	Ins-last	74.24	31.67	85.82	11.75	40.67	22.12	88.92	36.60	94.89	81.60	100	98.18
	Ins-first	70.05	44.82	98.76	99.61	59.56	24.18	88.62	27.49	89.79	75.59	99.66	96.26
Output	Natural, TxtLabel, PU	74.24	31.67	85.82	11.75	40.67	22.12	88.92	36.60	94.89	81.60	100	98.18
	Lines, _, _	1.18	1.31	99.94	97.06	4.17	1.57	72.51	12.10	99.57	99.79	99.99	99.94
	JSON, _, _	5.94	16.49	100	100	96.15	73.53	99.94	100	100	100	100	100
	_, Numerical, _	99.87	92.21	99.99	100	100	100	100	100	100	100	100	100
	_, _, OU	45.75	18.31	70.21	31.38	44.15	50.93	72.79	87.99	76.80	56.87	99.74	95.33
Reasoning	No-CoT	5.94	16.49	100	100	96.15	73.53	99.94	100	100	100	100	100
	CoT	35.25	34.25	100	100	58.66	53.29	100	100	100	100	99.99	99.99
	R-CoT	33.84	75.87	100	100	80.71	77.12	98.24	90.58	100	100	100	100

Table 13: Format error rate(%) in zero-shot scenario

test_size=500		**c-llama2-chat**	**c-llama2-base**	**intern-chat**	**intern-base**	**bc2-chat**	**bc2-base**
Input	Inst-last	0.3834	0.2835	0.1856	0.1212	0.4402	0.4187
	Inst-first	0.4832	0.2959	0.2038	0.2044	0.5091	0.4345
Output	Natural, TxtLabel, PU	0.3834	0.2835	0.1856	0.1212	0.4402	0.4187
	Lines, _, _	0.4220	0.2921	0.2436	0.1846	0.3971	0.4077
	JSON, _, _	0.3773	0.2132	0.3390	0.2954	0.4614	0.3683
	_, NumLabel, _	0.1522	0.1666	0.2470	0.2603	0.2406	0.1960
	_, _, OU	0.3612	0.3168	0.2461	0.1443	0.1948	0.1924
Reasoning	No-CoT	0.3773	0.2132	0.3390	0.2954	0.4614	0.3683
	CoT	0.3383	0.2174	0.3636	0.3167	0.4810	0.4466
	R-CoT	0.3638	0.2445	0.3522	0.2633	0.4668	0.4075

Table 14: The average weighted Kappa κ on the MASA ID tasks in in-context learning scenario

Inst-last,No-MI / Natural, TxtLabel, PU **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。请根据评论对这些方面进行情感分析，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的情感："**方面1：情感类别，方面2：情感类别，...**"\n输出： **O: 方面1：情感类别，方面2：情感类别，...**	<review>\n---\n *Read the above comment and observe the following aspects:* [aspect]. *Based on the comment, please conduct sentiment analysis on these aspects with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**"\n Output: **Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**
Inst-first, _ **I:** 阅读下面这段评论，观察以下这些方面：[aspect]。请根据评论对这些方面进行情感分析，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的情感："**方面1：情感类别，方面2：情感类别，...**"\n---\n评论：<review>\n输出： **O: 方面1：情感类别，方面2：情感类别，...**	*Read the comment below and observe the following aspects:* [aspect]. *Based on the comment, please conduct sentiment analysis on these aspects with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**" \n---\n *Review:* <review>\n Output: **Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**
No-Inst, _ **I:** <review>\n输出： **O: 方面1：情感类别，方面2：情感类别，...**	<review>\n Output: **Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**
Lines, _, _ **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。请根据评论对这些方面进行情感分析，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的情感："**方面1：情感类别\n方面2：情感类别\n...**"\n输出： **O: 方面1：情感类别，** **方面2：情感类别，** **...**	<review>\n---\n *Read the above comment and observe the following aspects:* [aspect]. *Based on the comment, please conduct sentiment analysis on these aspects with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**Aspect 1: Sentiment category\n Aspect 2: Sentiment category\n ...**"\n Output: **Aspect 1: Sentiment category,** **Aspect 2: Sentiment category,** ...
JSON, _, _ / No-CoT **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。请根据评论对这些方面进行情感分析，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的情感："**{"方面":方面1, "情感":情感类别}\n{"方面":方面2, "情感":情感类别}\n...**"\n输出： **O: {"方面": ..., "情感": ...}** **{"方面": ..., "情感": ...}** **...**	<review>\n---\n *Read the above comment and observe the following aspects:* [aspect]. *Based on the comment, please conduct sentiment analysis on these aspects with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**{"Aspect ": Aspect 1, "Sentiment": Sentiment category}\n{"Aspect": Aspect 2, "Sentiment": Sentiment category}\n ...**"\n Output: **{"Aspect 1": ..., "Sentiment category": ...}** **{"Aspect 2": ..., "Sentiment category": ...}** ...
_, NumLabel, _ **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。请根据评论对这些方面进行情感分析，具体有四类情感：正面(1)、负面(-1)、中性(0)、未提及(-2)。请用以下格式给出所有方面的情感："**方面1：情感类别，方面2：情感类别，...**"\n输出： **O: 方面1：0，方面2：1，...**	<review>\n---\n *Read the above comment and observe the following aspects:*[aspect]. *Based on the review, please make a sentiment analysis on these aspects with four specific categories:* positive(1), negative(0), neutral(-1), and unmentioned(-2). *Please provide the sentiment for all aspects in the following format:* "**Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**"\n Output: **Aspect 1: 0, Aspect 2: 1, ...**
_, _, OU **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。***请对评论中提及的方面进行情感分析***，具体有三类情感：正面、负面、中性。请用以下格式给出提及的方面的情感："**方面1：情感类别，方面2：情感类别，...**"，**未提及的方面不用给出。**\n输出： **O: 方面1：情感类别，方面2：情感类别，...**	<review>\n---\n *Read the above review and observe the following aspects:*[aspect]. *Please make a sentiment analysis of the aspects mentioned in the review with three specific categories:* positive, negative, and neutral. *Please provide the sentiment of the mentioned aspects in the following format:* "**Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**", ***and the aspects not mentioned need not be given.***\n Output: **Aspect 1: Sentiment category, Aspect 2: Sentiment category, ...**
CoT **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。***请提取或总结原文中对这些方面的描述，并进行情感分析***，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的结果："**{"方面":方面1, "描述":描述, "情感":情感类别}\n{"方面":方面2, "描述":描述, "情感":情感类别}\n...**"\n输出： **O: {"方面":..., "描述":..., "情感":...}** **{"方面":..., "描述":..., "情感":...}** **...**	<review>\n---\n *Read the above review and observe the following aspects:*[aspect]. *Please extract or summarize the descriptions of these aspects in the original text and make a sentiment analysis with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**{"Aspect ": Aspect 1, "Description ": Description, "Sentiment": Sentiment category}\n{"Aspect": Aspect 2, "Description ": Description, "Sentiment": Sentiment category}\n ...**"\n Output: **{"Aspect 1": ..., "Description ": ..., "Sentiment category": ...}** **{"Aspect 2": ..., "Description ": ..., "Sentiment category": ...}** ...
R-CoT **I:** <review>\n---\n阅读上面这段评论，观察以下这些方面：[aspect]。***请提取或总结原文中对这些方面的描述，并进行情感分析***，具体有四类情感：正面、负面、中性、未提及。请用以下格式给出所有方面的结果："**{"方面":方面1, "情感":情感类别, "描述":描述}\n{"方面":方面2, "情感":情感类别, "描述":描述}\n...**"\n输出： **O: {"方面":..., "情感":..., "描述":...}** **{"方面":..., "情感":..., "描述":...}** **...**	<review>\n---\n *Read the above review and observe the following aspects:* [aspect]. *Please extract or summarize the descriptions of these aspects in the original text and make a sentiment analysis with four specific categories:* positive, negative, neutral, and unmentioned. *Please provide the sentiment for all aspects in the following format:* "**{"Aspect ": Aspect 1, "Sentiment": Sentiment category, "Description ": Description}\n{"Aspect": Aspect 2, "Sentiment": Sentiment category, "Description ": Description,}\n ...**"\n Output: **{"Aspect 1": ..., "Sentiment category": ..., "Description ": ...}** **{"Aspect 2": ..., "Sentiment category": ..., "Description ": ...}** ...

Figure 9: Examples of different sample designs on the MASA task.

Error Type	Output	Processed Output	Count as Format Error
Aspect Expression	交通**情况**：未提及，排队等候情况：负面，点菜上菜速度：负面，装修情况：正面，嘈杂情况：未提及。 traffic **situation**: positive, queuing: negative, serving speed: negative, decoration: unmentioned, noise: unmentioned.	{"交通**便利程度**": "正面", "排队等候情况": 负面", "点菜上菜速度": "负面", "装修情况": "未提及", "嘈杂情况": "未提及"} {"traffic convenience": "positive", "queuing": "negative", "serving speed": "negative", "decoration": "unmentioned", "noise": "unmentioned"}	NO
Extra Aspect	食品评价：负面\n饮品评价：未提及\n价格水平：负面\n卫生情况：未提及\n服务人员态度：负面\n停车方便程度：未提及\n**空调：负面** food: negative\n beverage: unmentioned\n price: negative \n hygiene: unmentioned\n staff attitude: negative\n parking convenience: unmentioned\n **air conditioner: negative**	{"食品评价": "负面", "饮品评价": "未提及", "价格水平": "负面", "卫生情况: 未提及", "服务人员态度": "负面", "停车方便程度": "未提及"} {"food": "unmentioned", "beverage": "unmentioned", "price": "negative", "hygiene: unmentioned", "staff attitude": "negative", "parking convenience": "unmentioned"}	NO
Lack of Aspect	食品评价：负面，饮品评价：未提及，价格水平：负面，服务人员态度：未提及，停车方便程度：未提及。 food: unmentioned, beverage: unmentioned, price: negative, staff attitude: negative, parking convenience: unmentioned.	{"食品评价": "未提及", "饮品评价": "未提及", "价格水平": "负面", "**卫生情况: 未提及**", "服务人员态度": "负面", "停车方便程度": "未提及"} {"food": "unmentioned", "beverage": "unmentioned", "price": "negative", "**hygiene: unmentioned**", "staff attitude": "negative", "parking convenience": "unmentioned"}	NO
Redundant Describe	食品评价：正面，饮品评价：负面，价格水平：中性，卫生情况：未提及，服务人员态度：正面 **（虽然服务水平不算专业但看上去挺善良的爱笑的小姑娘）**，停车方便程度：未提及。 food: positive, beverage: negative, price: neural, staff attitude: unmentioned **(The service level is not professional but seems to be a kind little girl who loves to laugh)**, parking convenience: unmentioned.	{"食品评价": "正面", "饮品评价": "负面", "价格水平": "中性", "卫生情况: 未提及", "服务人员态度": "正面", "停车方便程度": "未提及"} {"food": "positive", "beverage": "negative", "price": "neural", "hygiene: unmentioned", "staff attitude": "positive", "parking convenience": "unmentioned"}	YES
Unformatted	交通非常便利，离地铁青年路站很近，点菜上菜速度较快，装修情况良好，嘈杂情况一般。 The transportation is very convenient, it is very close to the subway Qingnian Road station, the ordering speed is fast, the decoration is in good condition, and the noisy situation is general.	{"交通便利程度": "未提及", "排队等候情况": "未提及", "点菜上菜速度": "未提及", "装修情况": "未提及", "嘈杂情况": "未提及"} {"traffic convenience": "unmentioned", "queuing": "unmentioned", "serving speed": "unmentioned", "decoration": "unmentioned", "noise": "unmentioned"}	YES
Extra Sentiment	食品评价：负面，饮品评价：未提及，价格水平：负面，卫生情况：未提及，服务人员态度：负面，**未提及**，停车方便程度：未提及。 food: unmentioned, beverage: unmentioned, price: negative, hygiene: unmentioned, staff attitude: negative, **unmentioned**, parking convenience: unmentioned.	{"食品评价": "负面", "饮品评价": "未提及", "价格水平": "负面", "卫生情况": "未提及", "服务人员态度": "负面", "停车方便程度": "未提及"} {"food": "negative", "beverage": "unmentioned", "price": "negative", "hygiene: unmentioned", "staff attitude": "negative", "parking convenience": "unmentioned"}	YES
Punctuation Error	交通便利**,** 正面，排队等候情况**,** 未提及，点菜上菜速度**,** 正面，装修情况**,** 中性，嘈杂情况**,** 负面。 traffic convenience**,** positive, queuing**,** unmentioned, serving speed**,** positive, decoration**,** neural, noise**,** negative.	{"交通便利程度": "未提及", "排队等候情况": "未提及", "点菜上菜速度": "未提及", "装修情况": "未提及", "嘈杂情况": "未提及"} {"traffic convenience": "unmentioned", "queuing": "unmentioned", "serving speed" : "unmentioned", "decoration": unmentioned", "noise": "unmentioned"}	YES
Numbers out of range	食品评价：**8**，饮品评价：-2，价格水平：-2，卫生情况：-2，服务人员态度：**10**，停车方便程度：-2。 food: **8**, beverage: -2, price: -2, hygiene: -2, staff attitude: **10**, parking convenience: -2.	{"饮品评价": "未提及", "价格水平": "未提及", "卫生情况": "未提及", "停车方便程度": "未提及", "食品评价": "未提及", "服务人员态度": "未提及"} {"beverage": "unmentioned", "price": "unmentioned", "hygiene: unmentioned", "parking convenience": "unmentioned", "food": "unmentioned", "staff attitude": "negative"}	YES
Repeat Segment	{"方面": "食品评价", "描述": "蔓越莓泡在水里种子也都挤漏出来", "情感": "负面"}\n{"方面": "饮品评价", "描述": "蔓越莓晶钻果茶，蔓越莓泡在水里种子也都挤漏出来，蔓越莓泡在水里种子也都挤漏出来，蔓越莓泡在水里种子也都挤漏出来", "情感": "负面"}\n{"方面": "价格水平", "描述": **"蔓越莓晶钻果茶，蔓越莓晶钻果茶，蔓越莓晶钻果茶，......蔓越莓晶钻** {"aspect": "food", "describe": "Cranberries soak in water and the seeds squeeze out", "sentiment": "negative"}\n{"aspect": "beverage", "describe": "Cranberry tea, cranberry soaked in water and the seeds will leak out, cranberry soaked in water and the seeds will leak out, cranberry soaked in water and the seeds will leak out", "sentiment": "negative"}\n{"aspect": "price", "describe": "Cranberry tea, cranberry tea, cranberry tea,...cranberry tea	{"食品评价": "负面", "饮品评价": "负面", "价格水平": "未提及", "卫生情况": "未提及", "服务人员态度": "未提及", "停车方便程度": "未提及"} {"food": "negative", "beverage": "negative", "price": "unmentioned", "hygiene: unmentioned", "staff attitude": "unmentioned", "parking convenience": "unmentioned"}	YES

Figure 10: Examples of format error types and how they are processed on the MASA task.

Tasks	Strategies	Prompts	Output_Formats
GENIA (Nested-NER)	heuristic	[INST]Read the given sentence carefully, identify all named entities of type "DNA", "RNA", "protein", "cell_type" or "cell_line". Answer in the format **["entity_type", "entity_name"]**. If no entity exists, then just answer "[]". Given sentence: <sentence> [/INST]	["DNA", "xxx"] ["protein", "xxx"] ["protein", "xxx"] ...
	EW-SDE	[INST]Given sentence: <sentence> Read the given sentence carefully, identify all named entities of type "DNA", "RNA", "protein", "cell_type" or "cell_line". For each entity type, answer in the format like **"'entity_type': 'entity_name_1', 'entity_name_2'..."**, then concat answer for each type with ';'. Only output entity types that contain entities.[/INST]	'DNA': 'xxx', 'xxx', ... ; 'protein': 'xxx', 'xxx'; 'cell_type': 'xxx'
	ES-SDE	[INST]Read the given sentence carefully, identify all named entities of type "DNA", "RNA", "protein", "cell_type" or "cell_line". For each entity type, answer in a line in the format like **"'entity_type': 'entity_name_1', 'entity_name_2'..." (when no entities exist, answer "'entity_type': ''")**.Given sentence: <sentence> [/INST]	'DNA': 'xxx', 'xxx', ... 'RNA': '' 'protein': 'xxx', 'xxx' 'cell_type': 'xxx' 'cell_line': ''
MAVEN (ED)	heuristic	We define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Given a sentence, please detect the type of events it contains and extract the trigger word from it. Please generate the result in the following format: **"["event_type", "trigger_word"]\n..."**If no event exists, just answer[]. The sentence is: <sentence> Output: \n"	["Motion", "xxx"] ["Conquering", "xxx"] ["Conquering", "xxx"]
	EW-SDE	Given a sentence: <sentence> \n---\nWe define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Please detect the type of events the given sentence contains and extract the trigger word from it. Please generate the result in the following format: **"event_type1: trigger_word1, trigger_word2, ...; event_type2: trigger_word1, trigger_word2, ...; ..."** Output:\n	Motion: xxx; Conquering: xxx, xxx
	ES-SDE	We define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Given a sentence, please detect all the type of events in the predefined set from it. For the types this sentence contains, please extract the trigger words from it, **and for the types it does not contain, return the trigger words as NONE**. Please generate the result in the following format: **"event_type1: trigger_word1, trigger_word2, ...\nevent_type2: trigger_word1, trigger_word2, ...\n..."** The sentence is: <sentence> Output: \n	Catastrophe: NONE Attack: NONE Hostile_encounter: NONE Causation: NONE Process_start: NONE Competition: NONE Motion: xxx ...
Review11 (MASA)	heuristic	Read the comment below and observe the following aspects: [aspect]. Based on the comment, please conduct sentiment analysis on these aspects with three specific categories: positive, negative, and neutral. Please provide the sentiment of the mentioned aspects in the following format: **"["Aspect 1", "Sentiment category"]\n["Aspect 2", "Sentiment category"]\n ..."**, and the aspects not mentioned need not be given.\n---\n Review: <review>\n Output:	["Aspect 1", "xxx"] ["Aspect 3", "xxx"] ...
	EW-SDE	<review>\n---\n Read the above review and observe the following aspects: [aspect]. Please make a sentiment analysis of the aspects mentioned in the review with three specific categories: positive, negative, and neutral. Please provide the sentiment of the mentioned aspects in the following format: **"Aspect 1: Sentiment category, Aspect 2: Sentiment category, ..."**, and the aspects not mentioned need not be given.\n Output:	Aspect 1: xxx, Aspect 3: xxx, ...
	ES-SDE	Read the comment below and observe the following aspects: [aspect]. Based on the comment, please conduct sentiment analysis on these aspects with four specific categories: positive, negative, neutral, and **unmentioned**. Please provide the sentiment for all aspects in the following format: **"Aspect 1: Sentiment category\nAspect 2: Sentiment category\n..."**\n---\n Review: <review>\n Output:	Aspect 1: xxx Aspect 2: unmentioned Aspect 3: xxx ...

Figure 11: Examples of different sample designs on GENIA, MAVEN and Review11.

heuristic

Original Instruction:
We define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Given a sentence, please detect the type of events it contains and extract the trigger word from it. Please generate the result in the following format: "["event_type", "trigger_word"]\n..."If no event exists, just answer[]. The sentence is: <sentence> Output: \n"

Instruction Variation 1:
We have the following event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. For a sentence, please detect the type of events it contains and extract the trigger word from it. We define the format of the result as: "["event_type", "trigger_word"]\n..."If no event exists, just answer[]. Here is the sentence: <sentence> Output: \n

Instruction Variation 2:
In our event detection task, we specify a set of event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Your goal is to analyze a given sentence and identify the types of events included in the sentence from the predefined set. Extract the trigger words related to each included event types from the sentence. Format the output as shown: "["event_type", "trigger_word"]\n...". If no event exists, just answer[]. Here is the sentence: <sentence> Output: \n

---

EW-SDE

Original Instruction:
Given a sentence: <sentence> \n---\nWe define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Please detect the type of events the given sentence contains and extract the trigger word from it. Please generate the result in the following format: "event_type1: trigger_word1, trigger_word2, ...; event_type2: trigger_word1, trigger_word2, ...; ..." Output:\n

Instruction Variation 1:
For a sentence: <sentence>\n---\nWe have the following event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Please detect the type of events the given sentence contains and extract the trigger word from it. We define the format of the result as: "event_type1: trigger_word1, trigger_word2, ...; event_type2: trigger_word1, trigger_word2, ...; ..." Output: \n

Instruction Variation 2:
Here is a sentence: <sentence>\n---\nIn our event detection task, we specify a set of event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Your goal is to analyze the given sentence and identify the types of events included in the sentence from the predefined set. Extract the trigger words related to each included event types from the sentence. Format the output as shown: "event_type1: trigger_word1, trigger_word2, ...; event_type2: trigger_word1, trigger_word2, ...; ..." Output: \n

---

ES-SDE

Original Instruction:
We define the event types set: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Given a sentence, please detect all the type of events in the predefined set from it. For the types this sentence contains, please extract the trigger words from it, and for the types it does not contain, return the trigger words as NONE. Please generate the result in the following format: "event_type1: trigger_word1, trigger_word2, ...\nevent_type2: trigger_word1, trigger_word2, ...\n..." The sentence is: <sentence> Output: \n

Instruction Variation 1:
We have the following event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. For a sentence, please detect all the type of events in the predefined set from it. For the types this sentence contains, please extract the trigger words from it, and for the types it does not contain, return the trigger words as NONE. We define the format of the result as: "event_type1: trigger_word1, trigger_word2, ...\nevent_type2: trigger_word1, trigger_word2, ...\n..."Here is the sentence: <sentence> Output: \n

Instruction Variation 2:
In our event detection task, we specify a set of event types: Catastrophe, Attack, Hostile_encounter, Causation, Process_start, Competition, Motion, Social_event, Killing, Conquering. Your goal is to analyze a given sentence and identify each event types from the predefined set. Extract the trigger words related to each event type from the sentence. If the sentence does not contain certain event types, please indicate NONE for those types. Format the output as shown: "event_type1: trigger_word1, trigger_word2, ...\nevent_type2: trigger_word1, trigger_word2, ...\n...". Here is the sentence: <sentence> Output: \n

Figure 12: Variations of Instructions on different strategies (taking MAVEN as an example).