Unleashing the Power of Large Language Models in Zero-shot Relation Extraction via Self-Prompting

Siyi Liu[1†] **Yang Li**[2] **Jiang Li**[2] **Shan Yang**[3] **Yunshi Lan**[4]
[1]EPFL, Lausanne, Switzerland
[2]MYBank, Ant Group, Beijing, China
[3]Beihang University, Beijing, China
[4]East China Normal University, Beijing, China
ssui.liu1022@gmail.com ly200170@alibaba-inc.com lj311207@mybank.cn
shanyang@buaa.edu.cn yslan@dase.ecnu.cn

Abstract

Recent research in zero-shot Relation Extraction (RE) has focused on using Large Language Models (LLMs) due to their impressive zero-shot capabilities. However, current methods often perform suboptimally, mainly due to a lack of detailed, context-specific prompts needed for understanding various sentences and relations. To address this, we introduce the Self-Prompting framework, a novel method designed to fully harness the embedded RE knowledge within LLMs. Specifically, our framework employs a three-stage diversity approach to prompt LLMs, generating multiple synthetic samples that encapsulate specific relations from scratch. These generated samples act as in-context learning samples, offering explicit and context-specific guidance to efficiently prompt LLMs for RE. Experimental evaluations on benchmark datasets show our approach outperforms existing LLM-based zero-shot RE methods. Additionally, our experiments confirm the effectiveness of our generation pipeline in producing high-quality synthetic data that enhances performance.

1 Introduction

Recent advances in Large Language Models (LLMs) have significantly progressed Natural Language Processing (NLP). Leveraging LLMs' potential in zero-shot learning, there is growing interest in applying their capabilities to zero-shot Relation Extraction (RE) (Han et al., 2018; Chen and Li, 2021), which identifies relationships between entities in text without extensive data annotation. Specifically, current methods convert the RE task into a Question Answering (QA) task by reformulating sentences as questions and candidate relations as options (Zhang et al., 2023b). Further advancements integrate a self-consistency approach (Wang et al., 2022b) within QA to reduce uncertainty through majority voting (Li et al., 2023a).

However, current methods frequently demonstrate suboptimal performance, mainly because of insufficient guidance for RE. The intricate demands of RE necessitate more detailed and context-specific prompts to effectively comprehend the diverse and complex nature of sentences and relations (Bassignana and Plank, 2022; Zhao et al., 2023b).

Inspired by recent studies on **Self-Prompting** (Li et al., 2022; Wan et al., 2023a,b)—that is, *employing the outputs generated by LLMs themselves as prompts*—our research introduces a novel prompting paradigm for RE. This paradigm leverages LLMs' inherent capabilities to create synthetic RE data tailored to specific relations. When using LLMs for RE from specific sentences, these synthetic samples, enriched with essential relational knowledge, serve as effective in-context demonstrations.

To be specific, for each distinct relation, we initially prompt LLMs to generate a corresponding sample comprising a sentence and its related relation triple. However, directly prompting LLMs to generate samples may result in a lack of **diversity** and **coverage** (Chung et al., 2023; Yu et al., 2024), which are crucial for in-context learning (Levy et al., 2022; Li and Qiu, 2023). Consequently, to guarantee the quality and comprehensive coverage of these synthetic samples, we implement a three-stage diversification strategy: **1. Relation Synonyms**: Utilizing LLMs, we generate synonyms for each relation, broadening semantic understanding and data variability. **2. Entity Filtering**: We filter out generated samples containing high-frequency entities to prevent repetitions, thereby ensuring the uniqueness of each data point. **3. Sentence Rephrase**: By rephrasing generated

[†]The work is performed when Siyi Liu is an intern in MYBank, Ant Group.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13147–13161
November 12-16, 2024

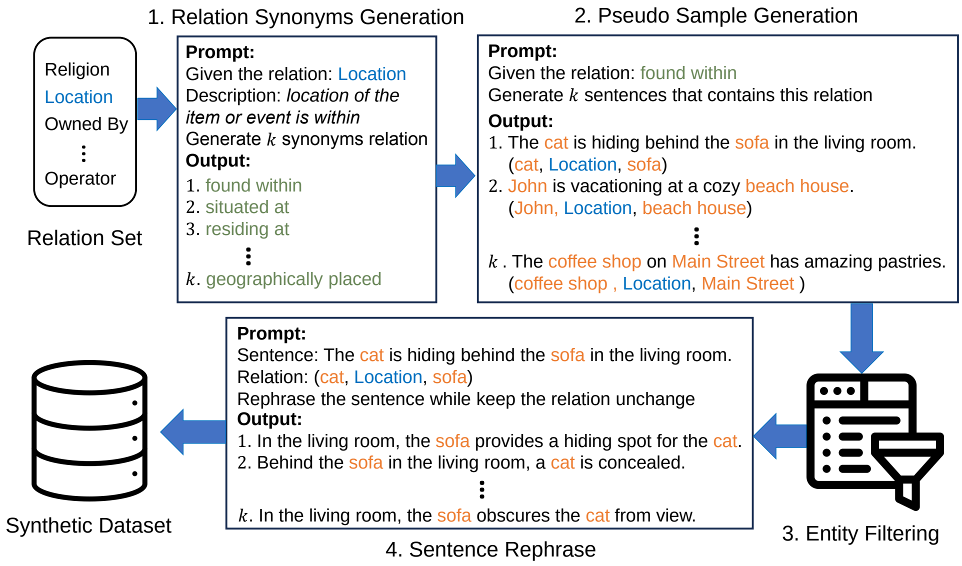

Figure 1: Depiction of the three-stage synthetic sample generation pipeline, where blue indicates candidate relations, green signifies synonym relations, and orange highlights entities within sentences.

sentences, we introduce structural variation and enhance the linguistic complexity of our dataset. The integration of these diversification methods results in a robust and varied set of synthetic data for RE. During inference, we select salient examples from this synthetic dataset as in-context demonstrations for each test sample, concatenating them with the test question to form the final input sequence for the LLM to generate the final answer.

To verify our method's effectiveness, we evaluated it across multiple zero-shot RE datasets. Compared to previous prompting strategies for LLM-based zero-shot RE SoTA, our method significantly outperforms them. Furthermore, extensive experiments have shown that our three-stage diversification strategy substantially enhances the diversity and coverage of in-context samples, thereby boosting model performance.

2 Methodology

2.1 Relation Synonyms

Our methodology's initial phase generates relation synonyms to broaden relation synonym coverage. This strategy recognizes that a dataset's relation often represents a broad concept, covering various synonymous or semantically related terms. As detailed in Figure 1, Step 1, we utilize LLMs to generate k synonyms for each targeted relation. To ensure the generated synonyms align with the relation's meaning, we provide the description of the relation to the LLMs. We then integrate the original relation with these synonyms to form a comprehensive semantic group. This process ensures the group encompasses the original relation alongside its synonyms, enhancing the relation's contextual comprehension.

2.2 Synthetic Sample Generation with Entity Filtering

After establishing semantic groups for each relation, we then prompt LLMs to create synthetic samples (as shown in Step 2 of Figure 1). However, these directly generated samples often lack sufficient entity coverage, reflecting the real world's complexity and variability in sentence structures. Such reliance on LLMs may result in a skewed distribution of entities, favoring those frequently found in pretraining and Supervised Fine-Tuning (SFT) data (Li et al., 2023b; Xu et al., 2023). This issue is not unique to our approach but has also been observed in other LLM-based domain-specific data generation efforts (e.g., Li et al. (2023b); Xu et al. (2023)).

To ensure comprehensive entity coverage, we implement a filtration mechanism for generated samples. This method discards samples with entities appearing more than n times, preventing over-representation. Conversely, samples with less frequent entities are retained, and their occurrence counts are updated. This strategy mitigates bias towards prevalent entities, promoting a diverse and balanced entity representation in our synthetic sample collection.

2.3 Sentence Rephrase

In our Self-Prompting framework, semantic coverage is crucial for sample diversity. The placement of subject and object entities in sentences can vary widely, and relations may be expressed implicitly or explicitly. Thus, incorporating diverse linguistic forms in synthetic data is essential.

To address this, we use LLMs to rephrase each sentence in the synthetic samples, creating r variants with similar meanings (as shown in Figure 1, Step 4). These rephrased versions differ in structure but maintain the original relation, whether explicit or implicit. This method enhances the range of linguistic expressions in our dataset while ensuring consistent portrayal of the relationship across different semantic representations.

2.4 Self-Prompting Inference

In the inference phase for a given test sentence, we retrieve d semantically similar samples as in-context demonstrations. This involves encoding the test sentence with the sentence embedding model and selecting the most similar examples from our sample set using cosine similarity.

To organize the retrieved samples effectively, we implement a ranking strategy based on similarity scores (Liu et al., 2022a), arranging samples from the lowest to the highest score. This method positions the most relevant sample nearest to the test sentence, optimizing the impact of contextually appropriate samples on the LLM's inference process.

2.5 Addressing the Error Propagation Problem

Error propagation is a critical concern in complex pipelines like ours, where early inaccuracies can accumulate and adversely impact downstream tasks. For instance, if incorrect or imprecise synonyms are generated during the Relation Synonyms Generation step, these errors may cascade through subsequent stages, resulting in further inaccuracies in relation extraction and other tasks that rely on these synonyms. To mitigate this risk, we incorporate relation descriptions (as detailed in Section 2.1 and Table 14 in the appendix). This enables the language model to better grasp the context of the relations, thereby enhancing the accuracy of synonym generation, as demonstrated in Table 4.

3 Experimental Setup

3.1 Datasets

We evaluate our methods on four RE datasets: (1) **FewRel** (Han et al., 2018), (2) **Wiki-ZSL** (Sorokin and Gurevych, 2017), (3) **TACRED.** (Zhang et al., 2017), (4) **SemEval** (Hendrickx et al., 2009). Further details about the dataset preprocessing and data statistics are in Appendix B.

3.2 Implementation Details

In our study, we employed ChatGPT with the API version `gpt-3.5-turbo-0301`, in line with previous research (Zhang et al., 2023b; Li et al., 2023a). The text embedding model utilized was `text-embedding-ada-002`, accessed via the OpenAI API. For more details about hyperparameter setting on API usage and synthetic sample generation, please refer to Appendix C.

3.3 Baselines

Zero-shot Baselines: For the FewRel and WikiZSL datasets, our baseline models include R-BERT (Wu and He, 2019), ESIM (Chen et al., 2017), CIM (Rocktaschel et al., 2016), ZS-BERT (Chen and Li, 2021), RE-Prompt (Chia et al., 2022) and RE-Matching (Zhao et al., 2023a). For RE-Prompt, the NoGen variant represents outcomes without data generation. Regarding TACRED and SemEval, our baseline comparisons involve NLI (Sainz et al., 2021) and SuRE (Lu et al., 2022). Here, the underlying base models are DeBERTa-XLarge (He et al., 2020) for NLI and PEGASUS-Large (Zhang et al., 2020) for SuRE.

LLMs Baselines: In evaluating prompt-based LLM baselines, we selected SumAsk (Li et al., 2023a) and QA4RE (Zhang et al., 2023b) for comparison. We also present the performance using a vanilla prompt strategy (denoted as **Vanilla**). This approach involves directly prompting LLMs to deduce the relation within a sentence, absent any in-context demonstrations ($d = 0$).

4 Results and Analysis

4.1 Main Results

Our evaluation of zero-shot prompting in LLMs, conducted on the FewRel and Wiki-ZSL datasets (as detailed in Tables 1 and 2), shows competitive performance against existing zero-shot RE methods. Notably, our Self-Prompting technique significantly enhances ChatGPT's performance over

Type	Method	$m = 5$			$m = 10$			$m = 15$		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Zero-shot	R-BERT	39.22	43.27	41.15	26.18	29.69	27.82	17.31	18.82	18.03
	ESIM	48.58	47.74	48.16	44.12	45.46	44.78	27.31	29.62	28.42
	CIM	49.63	48.81	49.22	46.54	47.90	45.57	29.17	30.58	29.86
	ZS-BERT	71.54	72.39	71.96	60.51	60.98	60.74	34.12	34.38	34.25
	RE-Prompt (NoGen)	51.78	46.76	48.93	54.87	36.52	43.80	54.45	29.43	37.45
	RE-Prompt	70.66	**83.75**	76.63	68.51	**74.76**	71.50	63.69	**67.93**	65.74
	RE-Matching	**78.19**	78.41	**78.30**	74.39	73.54	73.96	67.31	67.33	67.32
LLMs	Vanilla	74.45	59.25	65.98	61.15	57.68	59.36	57.82	61.27	59.01
	SumAsk	75.64	70.96	73.32	62.31	61.08	61.69	43.55	40.27	41.85
	Self-Prompting	78.13	77.01	77.57	**75.21**	74.43	**74.81**	**69.95**	67.50	**68.70**

Table 1: Main results on Wiki-ZSL. We mark the best results in **bold**, the second-best underlined. The results of the baselines are retrieved from Li et al. (2023a) and Zhao et al. (2023a).

Type	Method	$m = 5$			$m = 10$			$m = 15$		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Zero-shot	R-BERT	42.19	48.61	45.17	25.52	33.02	28.20	16.95	19.37	18.08
	ESIM	56.27	58.44	57.33	42.89	44.17	43.52	29.15	31.59	30.32
	CIM	58.05	61.92	59.92	47.39	49.11	48.23	31.83	33.06	32.43
	ZS-BERT	76.96	78.86	77.90	56.92	57.59	57.25	35.54	38.19	36.82
	RE-Prompt (NoGen)	72.36	58.61	64.57	66.47	48.28	55.61	66.49	40.05	49.38
	RE-Prompt	90.15	88.50	89.30	80.33	79.62	79.96	74.33	72.51	73.40
	RE-Matching	**92.82**	**92.34**	**92.58**	**83.21**	82.64	**82.93**	73.80	73.52	73.66
LLMs	Vanilla	91.70	88.87	90.26	72.64	76.12	74.34	65.46	65.50	65.48
	SumAsk	78.27	72.55	75.30	64.77	60.94	62.80	44.76	41.13	42.87
	Self-Prompting	90.47	89.83	90.19	81.15	**83.02**	82.07	**75.54**	**78.01**	**76.76**

Table 2: Main results on FewRel. We mark the best results in **bold**, the second-best underlined. The results of the baselines are retrieved from Li et al. (2023a) and Zhao et al. (2023a).

Vanilla prompting, outperforming the RE-Prompt method in most scenarios and markedly surpassing the SumAsk prompt strategy.

As the number of unseen relations (m) increases, predicting the correct relation becomes more challenging due to a broader range of choices. However, the benefits of Self-Prompting become more apparent, while Vanilla and SumAsk approaches show significant performance declines. This is likely because in-context demonstrations effectively narrow down potential relations. As a result, Self-Prompting better guides LLMs in inferring correct relations and demonstrates greater resilience with increasing relations.

Further validation is demonstrated through the application of our method on the TACRED and SemEval datasets. As shown in Table 3, our Self-Prompting technique achieved the highest F1 score on the SemEval dataset and the second-highest on TACRED, outperforming other zero-shot methods and significantly exceeding the performance of the QA4RE prompt strategy. This highlights the effectiveness of our approach, particularly given QA4RE's established performance.

Datasets	TACRED			SemEval		
	Prec.	Rec.	F1	Prec.	Rec.	F1
$\text{NLI}_{\text{DeBERTa}}$	42.9	**76.9**	55.1	22.0	25.7	23.7
$\text{SuRE}_{\text{PEGASUS}}$	13.8	51.7	21.8	0.0	0.0	0.0
Vanilla	32.1	74.8	44.9	18.2	20.8	19.4
SumAsk	**62.2**	53.8	**57.7**	-	-	-
QA4RE	32.8	68.0	44.2	29.9	35.2	32.3
Self-Prompting	56.8	57.5	57.1	**55.3**	**50.9**	**52.7**

Table 3: Main results on TACRED and SemEval. We mark the best results in **bold**, the second-best underlined. The results of the baselines are retrieved from (Zhang et al., 2023b)

4.2 Ablation Study on Different Diversity Strategies

In our ablation study, we systematically examine the impact of different components of our synthetic data generation method on the FewRel and Wiki-ZSL datasets. The absence of each component is denoted by a specific condition in our experiments: **w/o Rephrasing** (omission of sentence rephrasing), **w/o Synonyms** (exclusion of relation synonyms generation), **w/o Entity Filtering** (absence of entity frequency filtering), **w/o All** (direct generation without any enhancements), **Vanilla** (zero-shot learning without any generated samples, serving

Strategy	Wiki-ZSL			FewRel		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Vanilla	70.1	72.4	71.2	81.8	78.2	80.1
w/o Synonyms	73.0	73.3	73.2	82.1	80.3	81.3
w/o Entity Filtering	74.9	75.3	75.1	82.7	80.8	81.7
w/o Rephrase	77.0	75.7	76.3	84.5	81.9	83.2
w/o All	65.7	67.9	66.8	78.5	81.4	80.0
Complete	**82.4**	**77.7**	**80.0**	**85.4**	**83.3**	**84.3**

Table 4: Performance comparison of different strategies on FewRel and Wiki-ZSL datasets ($m = 10$).

as a baseline), and **Complete** (all diversification strategies are included).

As we can see in Table 4, the findings emphasize the importance of each component. Removing sentence rephrasing slightly decreases Precision and F1 scores. Excluding relation synonym generation results in a more substantial drop across all metrics, highlighting the importance of synonyms for capturing semantic breadth. Omitting entity frequency filtering significantly impacts Recall, indicating that entity variety is crucial for comprehensive relation extraction.

Moreover, directly prompting LLMs to generate samples and using them for inference impairs the model's performance, as evidenced by the w/o All condition, which underperforms compared to the Vanilla baseline. This suggests that unrefined sample generation can adversely affect the quality of RE. In contrast, our method (Complete), which incorporates all techniques, consistently outperforms the other conditions. It notably secures the highest Precision, Recall, and F1 scores across both datasets, confirming our comprehensive approach's effectiveness.

5 Conclusion

In this study, we introduced the Self-Prompting framework to optimize zero-shot RE in LLMs. Our three-stage diversification strategy generates synthetic samples, enhancing LLMs' accuracy and efficiency in RE. Experimental results on benchmark datasets demonstrate our method's effectiveness, surpassing existing LLM-based zero-shot RE techniques. Further experiments confirm that our strategy successfully addresses the challenges of diversity and coverage in synthetic sample generation, thereby improving model performance.

Limitations

While our Self-Prompting method demonstrates promising outcomes in zero-shot RE, it also presents certain limitations. Firstly, the selection of appropriate in-context demonstrations from synthetic datasets requires further exploration, as improper samples may introduce noise, adversely affecting LLM performance in zero-shot RE. Additionally, the performance of our Self-Prompting method on domain-specific data remains uncertain, given that domain-specific data generation poses an ongoing challenge. We acknowledge these issues and leave them for future work to address.

Ethics Statement

This work employs text generated by Large Language Models (LLMs), which may inadvertently produce content with ethical or safety concerns. However, given that ChatGPT, the LLM utilized in our experiments, is rigorously designed to minimize the generation of untrustworthy and harmful information, and considering the specific context of zero-shot relation extraction, we contend that the ethical considerations related to this research are limited. Consequently, a detailed discussion of these issues is deemed unnecessary.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Elisa Bassignana and Barbara Plank. 2022. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 67–83, Dublin, Ireland. Association for Computational Linguistics.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the*

55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 575–593, Toronto, Canada. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 94–99, Boulder, Colorado. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Itay Levy, Ben Bogin, and Jonathan Berant. 2022. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023a. Revisiting large language models as zero-shot relation extractors. *arXiv preprint arXiv:2310.05028*.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for open-domain qa. *arXiv preprint arXiv:2212.08635*.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023b. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022b. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.

Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. 2022. Summarization as indirect supervision for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6575–6594, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Graham Neubig and Zhiwei He. 2023. Zeno GPT Machine Translation Report.

Tim Rocktaschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023a. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Martin Eisenschlos, Sercan O Arik, and Tomas Pfister. 2023b. Universal self-adaptive prompting. *arXiv preprint arXiv:2305.14926*.

Wenya Wang, Vivek Srikumar, Hanna Hajishirzi, and Noah A Smith. 2022a. Elaboration-generating commonsense question answering at scale. *arXiv preprint arXiv:2209.01232*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

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*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Ran Xu, Hejie Cui, Yue Yu, Xuan Kan, Wenqi Shi, Yuchen Zhuang, Wei Jin, Joyce Ho, and Carl Yang. 2023. Knowledge-infused prompting: Assessing and advancing clinical text data generation with large language models.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Zhaojian Yu, Xin Zhang, Ning Shang, Yangyu Huang, Can Xu, Yishujie Zhao, Wenxiang Hu, and Qiufeng Yin. 2024. Wavecoder: Widespread and versatile enhanced instruction tuning with refined data generation.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023a. Contrastive learning of sentence embeddings from scratch. *arXiv preprint arXiv:2305.15077*.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023b. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Jun Zhao, Wenyu Zhan, Wayne Xin Zhao, Qi Zhang, Tao Gui, Zhongyu Wei, Junzhe Wang, Minlong Peng, and Mingming Sun. 2023a. Re-matching: A fine-grained semantic matching method for zero-shot relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6680–6691.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2023b. A comprehensive survey on deep learning for relation extraction: Recent advances and new frontiers.

A Related Works

A.1 Zero-shot Relation Extraction

Zero-shot RE has recently become a crucial focus in advancing predictive model capabilities. Levy et al. (2017) pioneered zero-shot RE, developing models capable of identifying novel relations beyond predefined types. Furthering this field, Sainz et al. (2021) explored the use of smaller Language Models (LMs) fine-tuned on Natural Language Inference (NLI) datasets. Their approach employs an entity-filled relation template matching the test sentence, utilizing inference for relation prediction. Chen and Li (2021) incorporate text descriptions of both seen and unseen relations. It employs nearest neighbor search for predicting unseen relations, using embeddings of these relations and new sentences. Lu et al. (2022) framed RE as a summarization task, applying generative models to concisely express the relationships between target entities. However, a persistent challenge with existing zero-shot methods is their heavy reliance on extensive labeled data. Our research focuses on conducting zero-shot RE without any labeled data.

A.2 LLMs for Zero-shot Relation Extraction

In the exploration of Zero-shot RE using LLMs, most existing research has concentrated on designing effective prompts to enhance LLMs' extraction performance. For instance, ChatIE (Wei et al., 2023) employs ChatGPT for zero-shot RE, utilizing a two-stage prompting strategy to refine the LLMs' search scope. QA4RE (Zhang et al., 2023b) adopts a multiple-choice question-answering format, representing relations through manually crafted templates and assigning LLMs the task of predicting a single character. In a different approach, SumAsk (Li et al., 2023a) deconstructs the LLMs' reasoning into three distinct stages, thereby aiding them in understanding and interpreting the relationships between subjects and objects. This method is further enriched by the use of self-consistency (Wang et al., 2022b) to reduce response uncertainty. However, these methods do not fully harness the LLMs' inherent RE capabilities, primarily because of insufficient context-specific prompting. Our work aims to explore the LLMs' RE potential by utilizing Self-Prompting, which focuses on generating context-specific prompts from synthetic samples.

A.3 Synthetic Data Generation via LLMs

Recent research has been focused on leveraging the content generated by LLMs to enhance the training of smaller models in various domains. For instance, Ye et al. (2022) applied this technique in classification tasks, Wang et al. (2022a) in commonsense question-answering, Zhang et al. (2023a) in contrastive learning, and Chia et al. (2022) in RE. Additionally, another strand of research directly utilizes the outputs from LLMs. Some studies have employed LLMs to generate relevant contexts or background documents as supplementary inputs for QA tasks (Yu et al., 2022; Liu et al., 2022b; Li et al., 2022). Others have focused on eliciting detailed reasoning steps, termed chain-of-thought, particularly for solving arithmetic problems (Wei et al., 2022; Wan et al., 2023a,b). In this work, we capitalize on synthetic RE samples generated by LLMs to bolster their capabilities in RE, exploring a novel approach to enhance the effectiveness of these models in this specific task.

B Datasets information

The statistics of the datasets are shown in Table 5 and Table 6. Following previous works (Zhang et al., 2023b; Li et al., 2023a), for the FewRel and Wiki-ZSL datasets, we randomly selected 5 relations for validation and selected a varying number of unseen relations (m) for testing, where m could be 5, 10, or 15. To ascertain the robustness of our results, this classification process was repeated five times, and we report the average macro-F1 scores from these iterations. For TACRED and SemEval, we conduct experiments using only the test samples and present the micro-averaged F1 scores. All relations are included except for *none-of-the-above*.

To effectively manage OpenAI API usage and associated costs, we randomly selected 1,000 samples from the test set of each dataset. We ensured that these samples proportionally represented each relation class.

Dataset	# samples	# entities	# relations
FewRel	56,000	72,954	80
Wiki-ZSL	94,383	77,623	113

Table 5: Statistics of FewRel and Wiki-ZSL

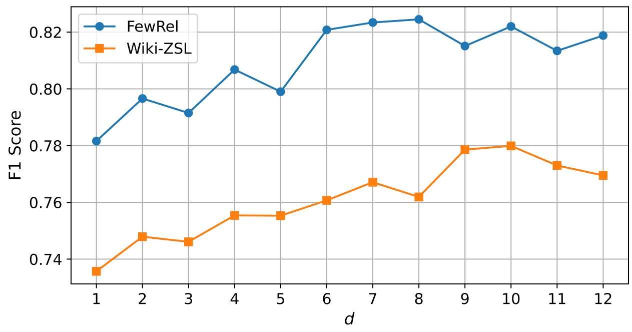

Figure 2: Average F1 when using different numbers of demonstrations in Self-Prompting.

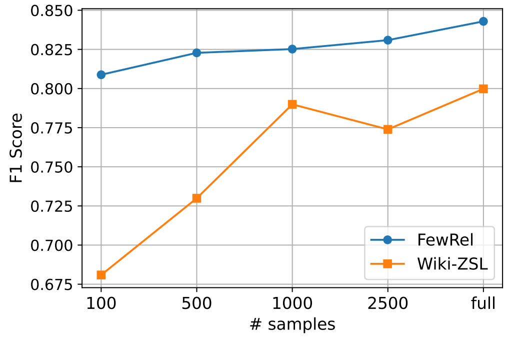

Figure 3: Average F1 when using different sizes of synthetic samples in Self-Prompting.

Dataset	# train	# dev	# test	# relations
TACRED	68,124	22,631	15,509	42
SemEval	6,507	1,493	2,717	9

Table 6: Statistics of TACRED and SemEval

C Hyperparameter Settings

During the synthetic sample generation phase, the temperature setting was adjusted to 1.2 to enhance sample diversity. Conversely, for inference, we set the temperature to 0, ensuring reproducibility, with other hyperparameters maintained at default settings.

For generating relation synonyms, we produced 10 synonyms per relation ($k = 10$). In the synthetic sample generation and filtering process, the LLMs were prompted to generate 10 samples at a time, excluding those with entities occurring more than three times ($n = 3$). The generation process ceased either upon reaching 200 samples or when no new samples contained unique entities after three iterations for each relation. Each sample underwent sentence rephrasing to generate three variants ($r = 3$). A detailed cost analysis is provided in Appendix H. Regarding the selection of demonstration samples at inference, we fixed d at 10. Following Kojima et al. (2022), our approach only retains the first part of the model's output that conforms to the specified answer format.

D Influence of Demonstration Quantity

To identify the optimal number of in-context samples d, we analyzed how varying the number of examples in the input affects performance, as illustrated in Figure 2. These experiments, aimed at assessing cost-effectiveness, were limited to a single subset of relations with $m = 10$. Analyzing F1 scores across two datasets revealed a pattern of performance improvement as the number of examples increased from 1 to 12. Yet, we found that utilizing more than 10 examples did not offer substantial benefits and, notably for Wiki-ZSL, resulted in diminished performance. Therefore, balancing performance efficiency with cost considerations, we determined that 10 demonstrations ($d = 10$) were optimal for our experiments.

E Influence of Generated Data Size

Evaluating the impact of synthetic sample size on experimental outcomes, our comprehensive analysis, shown in Figure 3, focuses on a relation subset with $m = 10$, exploring synthetic sample sizes from 100 to approximately 6,000.

The analysis reveals a clear trend: an increase in synthetic sample size generally boosts the F1 score across both FewRel and Wiki-ZSL datasets. Specifically, the FewRel dataset shows a steady increase in performance, reaching its peak with the full dataset utilized. In contrast, the Wiki-ZSL dataset experiences a marked improvement in F1 scores from 100 to 1,000 samples, after which the gains taper off, with scores stabilizing at 2,500 samples and beyond. This indicates that while enlarging the synthetic sample pool enhances model performance, a saturation point exists beyond which no significant benefits are observed.

F Data Generation Quality Analysis

We employed GPT-4 to determine the presence of specified relations within various datasets to evaluate the quality of generated samples. We randomly selected 10 relations from each dataset, generating 10 samples for each, thereby creating a set of 100 samples per dataset. This analysis encompassed three datasets: the original real data, our

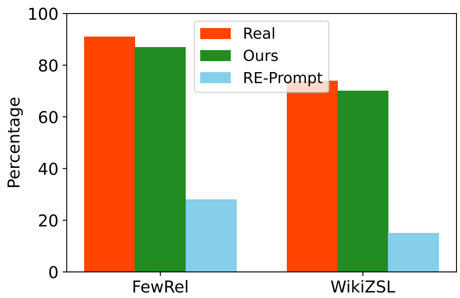

Figure 4: Percentage of correct samples in FewRel and Wiki-ZSL

generated data, and data generated using the RE-Prompt method. GPT-4 was tasked with verifying the specified relations in these samples. A sample was deemed correct if the head and tail entities exhibited the relation as labeled.

Figure 4 shows that our generated samples more accurately encapsulate the targeted relations compared to those generated by the RE-Prompt method. This close alignment with real data benchmarks demonstrates the effectiveness of our generation methodology, validating our samples' utility for in-context learning in RE tasks.

G Comparing among Different Demonstration Data

To further compare the quality of synthetic data from our method against RE-Prompt, we utilized RE-Prompt's synthetic data as demonstration samples in our inference framework. We documented the experimental outcomes on the FewRel and Wiki-ZSL datasets, with $m = 10$, in Table 7. These outcomes uniformly demonstrate that our method surpasses RE-Prompt in all instances, highlighting the superior data quality generated by our approach. This advantage is attained without task-specific fine-tuning, showcasing our data generation process's ability to produce high-quality synthetic samples for RE tasks effectively.

Datasets	FewRel			Wiki-ZSL		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Vanilla	82.51	78.32	80.36	68.50	72.23	70.31
RE-Prompt	83.73	81.30	82.50	73.33	72.14	72.73
Self-Prompting	**85.47**	**83.13**	**84.28**	**83.64**	**76.54**	**79.93**

Table 7: Performance on FewRel and Wiki-ZSL datasets using varied synthetic demonstrations with $m = 10$ unseen relations

H Cost of Synthetic Data Generation

For synthetic data generation, we employed `gpt-3.5-turbo`, an economical choice at \$0.001 per 1K tokens for prompts and \$0.002 per 1K tokens for completions[1]. The synthesis involves three phases: generating relation synonyms, creating samples, and rephrasing sentences. The costs for each relation's data generation are itemized in Table 9, totaling approximately \$0.264 for around 600 samples per relation. Considering the Wiki-ZSL dataset includes up to 113 relations, the full data generation cost is estimated at \$30. This is cost-effective compared to manual annotation expenses, such as in machine translation tasks, which can reach around \$0.1 per word (Neubig and He, 2023). Thus, using `gpt-3.5-turbo` for synthetic data generation in RE tasks is validated as an economically viable method.

I General Effectiveness with LLMs of Different Sizes

To examine the impact of LLM size, we also employed the Qwen (Bai et al., 2023) series LLMs (1.8B, 7B, 14B) as alternative base models for evaluating our Self-Prompting methods. Our research explored Self-Prompting's efficacy across LLMs of various sizes, with the findings detailed in the accompanying table. This analysis covered models ranging from Qwen-1.8B to ChatGPT, applying both Vanilla and Self-Prompting methods to different sets of unseen relations ($m = 5, 10, 15$) in the FewRel dataset.

The Qwen series models (1.8B, 7B, and 14B parameters) demonstrated clear enhancements using Self-Prompting compared to the Vanilla approach. For the smallest model, Qwen-1.8B, Self-Prompting achieved a 14.57% average increase in F1 scores, highlighting its significant benefit for smaller-scale models. With larger models, the average improvement lessened but remained impactful: 10.07% for Qwen-7B and 6.63% for Qwen-14B.

J Case Study

Generation: Tables 10 and 11 showcase examples of the generation process for the *location* and *operator* relations, respectively.

Inference: Table 12 presents a successful instance of Self-Prompting, while Table 13 illustrates a failure. The success case demonstrates how synthetic

[1] https://openai.com/pricing

Type	Method	$m = 5$			$m = 10$			$m = 15$			Avg. Improv.
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	
Qwen-1.8B	Vanilla	51.23	47.47	49.28	22.81	27.36	24.89	20.75	24.42	22.49	14.57%
	Self-Prompting	59.30	59.28	59.29	47.31	46.80	47.05	33.66	34.43	34.04	
Qwen-7B	Vanilla	64.85	62.60	63.69	37.80	40.24	38.98	27.71	30.05	28.82	10.07%
	Self-Prompting	64.09	65.49	64.78	54.85	55.85	55.35	41.97	41.20	41.58	
Qwen-14B	Vanilla	66.13	65.20	65.66	53.03	52.31	52.67	47.73	45.60	46.64	6.63%
	Self-Prompting	75.00	69.86	72.33	63.17	60.05	61.67	51.70	50.03	50.85	
ChatGPT	Vanilla	91.70	88.87	90.26	72.64	76.12	74.34	65.46	65.50	65.48	5.24%
	Self-Prompting	88.47	88.92	88.70	80.27	82.08	81.17	74.82	77.05	75.92	

Table 8: Performance of our method for LLMs with different size

Stage	# Prompt	# Completion	# Total	Cost ($)
Relation Synonyms	0.132	0.077	0.209	0.00029
Sample Generation	38.18	23.14	61.33	0.08447
Sentence Rephrase	112.58	33.55	146.12	0.17967
Total	150.89	56.77	207.66	0.26443

Table 9: Average number of token usage (k) and cost ($) for a single relation samples generation

in-context samples, when closely related to the test sample, can offer a nuanced guide, aiding the model in distinguishing between *location* and *located on terrain feature*. Conversely, in the failure case, Self-Prompting did not yield an accurate prediction due to the in-context samples being less relevant, thereby introducing noise during inference.

K Prompts for LLMs

We listed each stage's prompts used in the synthetic data generation process in Table 14.

Stage	Examples
Relation Synonyms	**Relation**: Location **Description**: location of the item, physical object, or event is within. **Synonyms**: [situated at, found within, positioned in, nestled amongst, geographically placed, lying in, set within, residing at, located near, anchored in]
Sample Generation	**Relation**: Location 1. The grocery store in my neighborhood has a wide variety of organic produce. 2. The rainforest, filled with exotic wildlife, is set within the Amazon River basin. 3. The Louvre Museum, one of the world's largest art museums, sits within the city of Paris.
Rephrase Sentence	**Relation**: Location **Sentence**: The historic Colosseum is set within the heart of Rome, surrounded by ancient ruins and archaeological sites. **Rephrased Sentence:** 1. At the core of Rome, the Colosseum stands amidst ancient ruins and archaeological wonders. 2. Surrounded by relics of the past, the Colosseum exists at the center of Rome, a city with a rich history. 3. Rome's heart holds the majestic Colosseum, encircled by remnants of the ancient era.

Table 10: Case of sample generation for relation **Location**

Stage	Examples
Relation Synonyms	**Relation**: Operator **Description**: person, profession, or organization that operates the equipment, facility, or service. **Synonyms**: [controller, manager, handler, technician, operator, administrator, machinist, supervisor, system operator, service provider]
Sample Generation	**Relation**: Operator 1. The doctor, who works at the hospital, is responsible for overseeing the medical equipment. 2. The IT technician is in charge of maintaining and operating the computer server. 3. The internet connection provided by the telecommunications company has been unreliable lately.
Rephrase Sentence	**Relation**: Operator **Sentence**: The train station is operated by the city transportation authority. **Rephrased Sentence:** 1. The train station falls under the jurisdiction of the city transportation authority. 2. The city transportation authority oversees the operations of the train station. 3. The city transportation authority is in charge of managing the train station.

Table 11: Case of sample generation for relation **Operator**

Stage	Examples
Background Prompts	**Relation**: You are a helpful information extractor that can conduct relation extraction task. In detail, you final goal is to extract the relation between two entities in a sentence. The relation candidate is a list of relations that you can choose from: ['religion', 'location', 'competition class', 'operating system', 'owned by', 'contains administrative territorial entity', 'field of work', 'spouse', 'located on terrain feature', 'distributed by']
Synthetic In-Context Prompts	**Sentence**: The ski resort town, nestled against the natural feature of snow-capped mountains, is a popular destination for winter sports enthusiasts. Given the Sentence, the relation between town and snow-capped mountains is: located on terrain feature **Sentence**: The village, with its enchanting vineyards and stunning vistas, finds itself nestled in the picturesque valley. Given the Sentence, the relation between village and valley is: location **Sentence**: The beautiful vineyard, with rolling hills as its backdrop, is situated near the quaint village and nearby tourist destinations. Given the Sentence, the relation between vineyard and village is: location **Sentence**: Perched on the hill, the building provides a stunning vista of the valley beneath. Given the Sentence, the relation between building and hill is: located on terrain feature **Sentence**: Renowned for its geysers and hot springs, Yellowstone National Park is situated in the western United States. Given the Sentence, the relation between Yellowstone National Park and western United States is: located on terrain feature
Test Sample Prompt	**Sentence**: It is located west of, and adjacent to Bridalveil Fall, on the south side of the Merced River in Yosemite Valley. Given the Sentence, the relation between Bridalveil Fall and Yosemite Valley is:
Output	**Ground truth**: located on terrain feature **Vanilla**: location ✗ **Self-Prompting**: located on terrain feature ✓

Table 12: Case of successful test sample inference

Stage	Examples
Background Prompts	**Relation**: You are a helpful information extractor that can conduct relation extraction task. In detail, you final goal is to extract the relation between two entities in a sentence. The relation candidate is a list of relations that you can choose from: ['religion', 'location', 'competition class', 'operating system', 'owned by', 'contains administrative territorial entity', 'field of work', 'spouse', 'located on terrain feature', 'distributed by']
Synthetic In-Context Prompts	**Sentence**: An operating system known as macOS powers the Mac computers, which are produced by Apple Inc. Given the Sentence, the relation between computers and Mac is: operating system **Sentence**: Linux, a widely used open-source operating system, is favored by programmers and developers. Given the Sentence, the relation between Linux and open-source is: operating system **Sentence**: The Unix operating system, known for its stability and security, is widely used in enterprise computer systems. Given the Sentence, the relation between Unix operating system and computer systems is: operating system **Sentence**: Windows, commonly known as Microsoft Windows, is a group of several proprietary graphical operating system families. Given the Sentence, the relation between Windows and Microsoft is: operating system **Sentence**: The construction and distribution of the iconic Lego sets are handled by The Lego Group, a Danish toy production company. Given the Sentence, the relation between Lego sets and The Lego Group is: distributed by
Test Sample Prompt	**Sentence**: Sentence: His muscle algorithms for face animation were widely used in the computer film industry, most notably by Pixar, which first used the technique in their animation short Tin Toy. Given the Sentence, the relation between Tin Toy and Pixar is:
Output	**Ground truth**: distributed by **Vanilla**: distributed by ✓ **Self-Prompting**: field of work ✗

Table 13: Case of failed test sample inference

Stage	Prompts
Relation Synonyms	For a giving relation type: {*relation*}, your objective is to create {*k*} synonyms about this relation. The description of this relation is: {*description*} Ensure that your generated examples adhere to the following guidelines: 1. The synonyms should explicitly or implicitly align with the relation {*relation*}. 2. Ensure the diversity among different synonyms. 3. The synonyms could be a single word or phrase. Please format your output in Python list-style: [synonyms1, synonyms2, ..., synonyms{*k*}]
Sample Generation	Imagine you are a sophisticated language model functioning as a textual data generator for a relation extraction task. Your objective is to create {*k*} synthetic sentences, each containing a specific type of relationship denoted as: {*relation*} The description of this relation is: {*description*}. These sentences must be informative and clearly demonstrate the intended relation, either explicitly or implicitly. Please format your output as follows: Sentence: [Your generated sentence here]. Relation: [(entity1, {*relation*}, entity2), (entity3, {*relation*}, entity4), ...]. Where the relation list could contain one to three relation tuples. Ensure that your generated examples adhere to the following guidelines: 1. The relation should be the same as the previously defined relation. 2. Head and tail entities must appear in the original sentence. 3. Separate the head and tail into several triples that have the same relation. 4. Generate sentences with varying lengths and complexities, including simple, compound, and complex sentences. 5. Ensure a broad and realistic variety in the types of head and tail entities to reflect real-world contexts.
Rephrase Sentence	As a text paraphrasing agent, your task is to paraphrase a given sentence to generate {*k*} new versions. The original sentence includes one or more relationships. Rewrite the sentence to subtly imply the relationships that were originally stated explicitly, while also enhancing the semantic depth and diversifying the grammatical structure. Input format: Sentence: The sentence to be paraphrased. Relation: A list of relation tuples in the format (head, relation, tail). Output Format: Provide {*k*} paraphrased sentences, where the relation list could contain one to three relation tuples. Ensure that your generated examples adhere to the following guidelines: 1. Preservation of Entities: Ensure that the head and tail entities from the original sentence are present in each paraphrased version. 2. Variety and Realism: Aim for a wide range of sentence structures and contexts in your paraphrases, reflecting realistic and diverse scenarios. 3. In the generated relation list for each paraphrased sentence, the relation MUST remain consistent with the relation: {*relation*}, while minor modifications to the entities are permissible.
Inference	Your goal is to extract the relation between two entities in a sentence. The relation candidate is a list of relations that you can choose from: {*relation list*} {*demonstrations*} Sentence: {*extract sentence*} Given the Sentence, the relation between {*head*} and {*tail*} is:

Table 14: Prompts used for synthetic data generation and test sample inference