KNN-INSTRUCT: Automatic Instruction Construction with K Nearest Neighbor Deduction

Jianshang Kou, Benfeng Xu, Chiwei Zhu, Zhendong Mao*

University of Science and Technology of China, Hefei, China
{koujs, benfeng, tanz}@mail.ustc.edu.cn zdmao@ustc.edu.cn

Abstract

Supervised fine-tuning (SFT) is a critical procedure for aligning large language models. Despite its efficiency, the construction of SFT data often struggles with issues of quality, diversity, and scalability. Many existing methods, inspired by the SELF-INSTRUCT framework, typically generate synthetic instructions by prompting aligned proprietary models like ChatGPT. However, such process suffers from stale distribution, resulting in instructions that are merely trivial variations of existing ones. In this paper, we introduce a novel bootstrapping approach termed KNN-INSTRUCT, which incorporates KNN deduction to produce meaningful new instructions by effectively summarizing and learning from similar existing ones. We conduct an economical controlled experiment to preliminarily validate its effectiveness. In the further experiment, we construct a high-quality SFT dataset named KNN-INST-12K*. Applying the dataset to Qwen-2-7B, we get a MT-Bench score of 7.64, which outperforms all 7B models on the LMSYS leaderboard, including Starling-LM-7B (7.48), OpenChat-3.5 (7.06) and Zephyr-7B-beta (6.53). Our code and data are available at https://github.com/CrossmodalGroup/KNN-Instruct/.

1 Introduction

Large language models (LLMs) pre-trained on large volumes of unlabeled corpus have demonstrated amazing capability in numerous natural language processing (NLP) tasks (Devlin et al., 2018; Brown et al., 2020). The pre-trained models can be aligned to human intention with the straightforward supervised fine-tuning (SFT) (Wei et al., 2022; Victor et al., 2022). For the instruction-following task, a SFT dataset consists of some conversations $(instruction, response)$. The SFT model can be trained with reinforcement learning from human feedback (RLHF) for further alignment (Ouyang et al., 2022), which is very intricate and expensive, and has been widely applied in strong models such as ChatGPT (OpenAI, 2022).

Recently, Zhou et al. (2024) show that LLMs could be well aligned to human intention simply through SFT, while the scale, quality and diversity of SFT dataset are key to the performance of SFT model (Wang et al., 2022). To obtain a large, high-quality and diverse SFT dataset, previous works (Ouyang et al., 2022; Zhou et al., 2024) usually employ human experts to carefully curate meaningful instructions and corresponding responses, which is very laborious, costly and thus lacks scalability.

With the development of instruction-following LLMs, the past few years have witnessed a lot of works on automatic SFT data construction, which employ capable LLMs like ChatGPT instead of human experts to construct meaningful conversations (Wang et al., 2022; Xu et al., 2023; Taori et al., 2023; Chiang et al., 2023; Zhao et al., 2024). Nevertheless, we notice that current works still more or less fall short of quality, diversity or scalability. For instance, we reproduce SELF-INSTRUCT with the latest GPT-3.5-Turbo API, and our case study in Appendix B.1 shows that a number of synthetic instructions are not so high-quality, which may be detrimental to the performance of SFT model.

In this paper, we propose an automatic SFT data construction method named KNN-INSTRUCT. We analyze the types, pros and cons of several existing methods, and mainly make two improvements over the notable work SELF-INSTRUCT:

1. **On KNN Deduction** Currently many bootstrapping methods use several random samples to "prompt" the ChatGPT to write a new sample (Wang et al., 2022; Taori et al., 2023; Xu et al., 2023). We abandon the practice of random sampling. Instead, we sample a single instruction first, and involve text embedding to deduce its k nearest neighbors, which

* Corresponding author: Zhendong Mao.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10337–10350

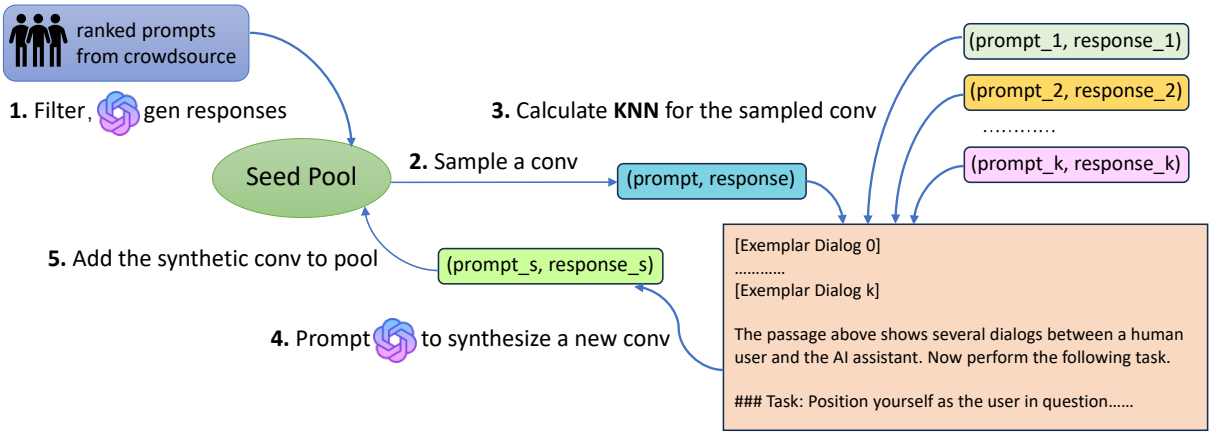

Figure 1: A high-level overview of KNN-INSTRUCT.

encourages the relevance between sampled instructions.

2. **On Seed Dataset** We believe that for bootstrapping methods, a large, high-quality, and diverse seed dataset helps to build a SFT dataset with the same features. According to this intuition, we carefully construct our initial seed dataset from a human-annotated instruction dataset.

To validate the effectiveness of our method, we carry out two experiments. In the preliminary experiment, we construct KNN-INST-12K, a SFT dataset of 12,104 single-turn conversations. We select four competitive methods as baselines, and take necessary measures to ensure fair comparison. We apply these five datasets to two capable pre-trained models, LLaMA-2-7B (Touvron et al., 2023) and Qwen-7B (Bai et al., 2023), respectively. We then evaluate all SFT models with AlpacaEval (Li et al., 2023) and MT-Bench (Zheng et al., 2024a), which are both widely-recognized LLM benchmarks, for five times. The evaluation results show that KNN-INSTRUCT surpasses all baseline methods on both benchmarks and for both backbones. It is noteworthy that we employ GPT-3.5-Turbo for both data construction and LLM evaluation in the preliminary experiment to save expenditure.

To further explore the performance of KNN-INSTRUCT, we conduct our second experiment, where GPT-3.5-Turbo is replaced with the strong and expensive GPT-4-Turbo. We apply the new synthesized dataset, KNN-INST-12K* to the recently released pre-trained model Qwen2-7B, and evaluate it with MT-Bench, where the rigorous GPT-4-Turbo serves as a judge. The result is that our Qwen2-7B$_{\text{KNN-INST-12K*}}$ scores 7.64, surpassing the top three 7B models on the LMSYS Leaderboard [1]: Starling-LM-7B (Zhu et al., 2023), OpenChat-3.5 (Wang et al., 2023), and Zephyr-7B-β (Tunstall et al., 2023). Specifically, it also outperforms GPT-3.5-Turbo and Qwen2-7B-Instruct in the first turn of MT-Bench.

In Section 5, we design a number of ablation studies to gain a deeper insight into KNN-INSTRUCT. We thus demonstrate that our improvements on KNN deduction and seed dataset do help boost SFT performance, and the performance could be further improved by scaling up the dataset or optimize the hyper-parameter K.

Our contributions are summarized as follows:

- We propose KNN-INSTRUCT, a scalable framework that automatically derives high-quality and diverse instructions from existing similar instructions.
- We design a series of controlled experiments and ablation studies to demonstrate the capability and validity of KNN-INSTRUCT.

2 Related Work

Automatic SFT Data Construction In recent years, many works have attempted to take advantage of strong instruction-following LLMs such as ChatGPT instead of human experts to construct high-quality, diverse SFT data, which remarkably reduce the expenditure and have achieved competitive performance (Wang et al., 2022; Kong et al., 2023; Xu et al., 2023; Taori et al., 2023; Chiang et al., 2023; Zhao et al., 2024). To some degree,

[1] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

SFT data construction is just instruction data construct, because with a instruction, we can easily obtain its response by querying ChatGPT. According to different principles, these automatic methods can be roughly divided into three categories:

- **Bootstrapping** This kind of methods uses several high-quality instructions as an initial seed dataset, iteratively sample a few instructions and request ChatGPT to write a meaningful new instruction according to these examples. SELF-INSTRUCT (Wang et al., 2022), Alpaca (Taori et al., 2023) and WizardLM (Xu et al., 2023) are representatives of them. For this line of works, the selection of demo instructions and the design of user prompt are critical for the final performance.

- **User Simulation** These methods implement a user simulator and does not need any seed dataset. For instance, UltraLM (Ding et al., 2023) designs a comprehensive and easily-expandable framework, employing the ChatGPT to play the role of human user to chat with another ChatGPT back and forth to generate massive multi-turn conversations. PlatoLM (Kong et al., 2023) is another example, which trains a Socratic user simulator with the ShareGPT (ShareGPT, 2023) dataset to pose human-like questions.

- **Crowdsourcing** These methods also use crowdsourcing, which is similar to the traditional manual method to some extent. For example, WildChat (Zhao et al., 2024) offers free ChatGPT API and collect 1.0M multi-turn conversations with user consent.

LLM Evaluation In past years, there have been a lot of benchmarks to evaluate LLM's capacity on one or more NLP tasks (Hendrycks et al., 2020; Cobbe et al., 2021; Chen et al., 2021). Nevertheless, these benchmarks are mostly based on close-end questions and simple metrics like BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) or just pattern matching are used to automate the evaluation, thus fail to well measure human preference (Zheng et al., 2024a) on real-world questions. Nowadays with the development of LLMs, some strong instruction-following models, e.g. ChatGPT, are employed as a judge to evaluate a LLM on a set of open-end questions, where MT-Bench (Zheng et al., 2024a) and AlpacaEval (Li et al., 2023) are two of the most frequently adopted benchmarks.

3 KNN-Instruct

In this section, we introduce our proposed KNN-INSTRUCT in detail. Section 3.1 briefly describes the pipeline of KNN-INSTRUCT. Section 3.2, Section 3.3 and Section 3.4 respectively elaborate on the three key improvements we made over SELF-INSTRUCT. Section 3.5 deals with the actual implementation of KNN-INSTRUCT.

3.1 Overview

As can be seen in Figure 1, KNN-INSTRUCT is a bootstrapping method. To get an initial seed dataset, we select some high-quality ones from a dataset of human-annotated instructions, and call ChatGPT to generate corresponding responses for them. After the seed dataset is initialized, we then iteratively sample a conversation from the seed dataset, calculate it's K nearest neighbors, request ChatGPT to write a new instruction and response in a (K+1)-shot setting, and put the synthesized conversation back to the seed dataset.

3.2 KNN Deduction

In each iteration, SELF-INSTRUCT randomly sample 8 instructions to "prompt" the GPT-3 to write a new prompt. Nevertheless, our case study in Appendix B.1 reveals that the randomly sampled instructions are usually irrelevant with each other, which puzzles the ChatGPT and it is prone to talk about general topics like environmental pollution or climate change, instead of raising a specific and practical question.

Intuitively, we believe that instructions with close semantic relations are better demonstrations to "prompt" the ChatGPT to craft a high-quality new instruction. Therefore, we abandon the randomly sampling strategy, but take the KNN deduction shown below, i.e., a sampling strategy by deducing K nearest neighbors:

1. Sample a single instruction from the seed dataset as a core instruction.

2. Select the K nearest neighbors of the core instruction in the embedding space.

The (K+1) sampled instructions and their corresponding responses would be used in subsequent few-shot learning. We take advantage of the notable SimCSE(Gao et al., 2021) to embed the instructions, and adopt cosine similarity for distance measurement.

3.3 Improvement of Efficiency

In SELF-INSTRUCT, if the ROUGE-L similarity between the new instruction and any existing instructions is above or equal to 0.7, the new instruction would not be kept. The purpose of this action is to encourage instruction diversity. In our reproduction of the vanilla SELF-INSTRUCT (GPT-3 is replaced with GPT-3.5-Turbo), we observe that nearly 60% (4k out of 7k) synthesized instructions are discarded for high similarity, which brings more expense and less efficiency.

Nowadays, with strong instruction-following LLMs like ChatGPT, we believe that the diversity of synthetic instructions could be ensured simply through "prompting". According to this intuition, we carefully design a brief prompt template for few-shot learning as follow:

Prompt Used in KNN-INSTRUCT

Position yourself as the user in question, and craft a new, high-quality instruction. Keep the following in mind:
1. **Relevance**: Incorporate your previous analysis, fully utilize these informative prior, ensuring that the new instruction aligns well with this user.
2. **Originality**: The new instruction should be distinguished to existing ones instead of naive imitation or transfer, so try your best in CREATIVITY;
3. **Standalone**: The new instruction should be self-contained and not depend on prior conversations.
4. **Format**: You should simply return a string as the new instruction.

As can be seen, we directly emphasize that the new instruction must be original, standalone and relevant to the K+1 examples. Our prompt is relatively brief and no synthetic instructions would be eliminated, which reduces a lot of expenditure and improves efficiency. More details of our template are available in Appendix A.1.

3.4 Seed Dataset

It is noteworthy that SELF-INSTRUCT uses a dataset of 175 human-written instructions and responses as the initial seed dataset. We take it that this dataset is too small, and a large, diverse, high-quality seed dataset does help to build a large, diverse, high-quality SFT dataset.

Based on this assumption, we take advantage of 10k-prompts-ranked [2] to construct our seed dataset, which is made up of 10,331 human-annotated instructions. Each instruction is labeled with one or more quality scores ranged from 1 to 5. The data sources include both real-world and synthetic conversation datasets, such as ShareGPT (ShareGPT, 2023), Evol-Instruct (Xu et al., 2023), and UltraChat (Ding et al., 2023).

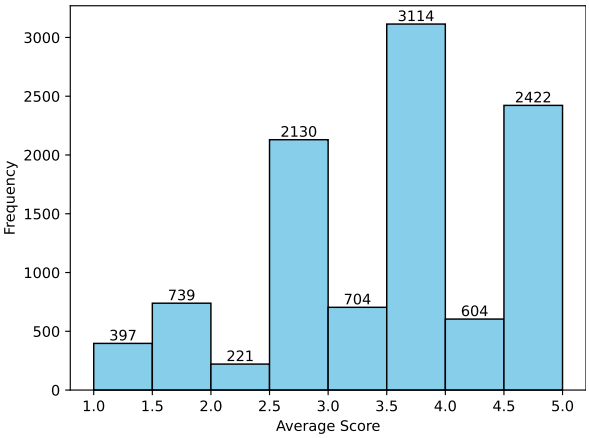

Figure 2: The quality distribution of 10k-prompts-ranked. The last interval is closed while other intervals are half open, half closed.

We calculate the average quality distribution of this dataset as shown in Figure 2. In consideration of quantity and quality, we only use instructions with a average score above 4.0. There are 3,026 instructions that meet this requirement. We request the ChatGPT to generate responses for the selected 3,026 instructions, and therefore get a diverse, high-quality seed dataset of 3,026 single-turn conversations. We name it **Seeds-3k**.

3.5 Implementation

In our practical implementation, we make some trade-offs to improve efficiency. The pseudo-code of KNN-INSTRUCT is shown in Algorithm 1. We calculate the embeddings for all instructions in the seed dataset in advance. We sample the instructions in order. The synthetic conversation will not be added to seed dataset immediately, but after all iterations have completed. More details of implementation are available in Section A.1.

In this paper, we set the hyper-parameter $K = 2$. A full round of KNN-INSTRUCT will double the seed dataset. With a seed dataset of scale 3,026, we run the algorithm twice and thus get a SFT dataset

[2] https://huggingface.co/datasets/DIBT/10k_prompts_ranked

Algorithm 1 KNN-INSTRUCT

Input: Seeds, an array of (instruction, response)

1: Calculate embeddings for all instructions in Seeds
2: **for** instruction **in** Seeds **do**
3: Calculate its K nearest instructions in the embedding space
4: Format the K+1 conversations in a few-shot prompt template
5: Request ChatGPT to write a new instruction according to the examples
6: Request ChatGPT to write a response for the synthesized instruction
7: **end for**
8: Add all synthesized conversations to Seeds

Output: Seeds, with double size

of 12,104 single-turn conversations, KNN-INST-12K. In a single-threaded configuration, it takes about 4 hours to obtain KNN-INST-12K.

We also estimate the computational overhead of KNN-INSTRUCT: To synthesize a sample $(instruction, response)$,

- The average input tokens is $50 + (40 + 60) \times 3 = 350$
- The average output tokens is $40 + 60 = 100$

where 50 is the length of our prompt template (see in Appendix A.1), 40 is the average length of $instruction$, 60 is the average length of $response$.

In this paper, GPT-3.5-Turbo-0125 (released on Jan 25 2024) is employed as the teacher model. If we use the recently released GPT-4o-mini, which is affordable, intelligent and lightweight, the total cost to synthesize KNN-INST-12K is about:

$$9,026 \times (350 \times \$0.15/M + 100 \times \$0.6/M) \approx \$1.02$$

3.6 Instances of KNN-INSTRUCT

We provide an instance of KNN-INSTRUCT as follow. As can be seen, there are three demonstrations as we adopt $K = 2$. The first demonstration queries the difference between Italian and Norwegian coffee culture; the second demonstration focuses on Italian dishes in Norway; the third demonstration talks about coffee brewing methods.

KNN-INSTRUCT Case 1

demo_1: Can you provide me with a comparison between traditional Italian espresso and Norwegian coffee culture, highlighting the unique aspects of each and how they contribute to the overall coffee experience in both countries?
demo_2: What are some traditional Italian dishes that one can enjoy in Norway, and what makes them stand out compared to authentic Italian cuisine?
demo_3: Can you provide a comprehensive analysis of the different types of coffee brewing methods, highlighting their respective strengths and weaknesses?
new_instruction: What are the key cultural differences between Japanese tea ceremonies and English afternoon tea traditions, and how do these rituals reflect and contribute to the respective societies' values and social norms?

Based on the three demonstrations, ChatGPT is inspired to produce a new instruction, which deals with the Japanese and English tea culture, and their impacts on the societies. We think this instruction is of high quality, and is relevant to all three demonstrations. More instances of KNN-INSTRUCT are shown in Appendix B.2.

4 Experiments

To justify the validity of KNN-INSTRUCT, we conduct two experiments: a preliminary experiment in Section 4.1, and a further one in Section 4.2.

4.1 Preliminary Experiment

This experiment aims to preliminarily verify the effectiveness of KNN-INSTRUCT at a low cost. To achieve this, we select several representative SFT data construction methods as baselines, use them to create 12k scale SFT datasets, conduct SFT on one or more pre-trained models, evaluate these SFT models with widely-accepted LLM benchmarks and make comparison.

Baselines We select four competitive baselines belonging to three different categories: Alpaca (Taori et al., 2023), Evol-Instruct (Xu et al., 2023), ShareGPT (ShareGPT, 2023) and UltraChat (Ding et al., 2023). Notably, we have taken necessary measures to ensure fair comparison:

- For bootstrapping methods Alpaca and Evol-Instruct, their seed datasets and ChatGPT APIs are different from ours. Therefore, we reproduce them with our Seeds-3k and the latest

Model	Alignment	MT-Bench	AlpacaEval(%)
Qwen-7B$_{\texttt{Alpaca-12k}}$	SFT	6.93±0.03	70.08±0.61
Qwen-7B$_{\texttt{ShareGPT-12k}}$	SFT	7.15±0.05	74.00±0.35
Qwen-7B$_{\texttt{UltraChat-12k}}$	SFT	7.20±0.06	71.64±0.44
Qwen-7B$_{\texttt{Evol-Inst-12k}}$	SFT	7.20±0.05	74.98±0.48
Qwen-7B$_{\text{KNN-INST-12K}}$	SFT	**7.38**±0.03	**75.86**±0.09
Qwen-7B-Chat	SFT+RLHF	7.33±0.01	74.78±0.57
LLaMA-2-7B$_{\texttt{Alpaca-12k}}$	SFT	6.25±0.04	53.67±0.39
LLaMA-2-7B$_{\texttt{ShareGPT-12k}}$	SFT	5.66±0.08	55.68±0.79
LLaMA-2-7B$_{\texttt{UltraChat-12k}}$	SFT	6.15±0.02	50.26±0.51
LLaMA-2-7B$_{\texttt{Evol-Inst-12k}}$	SFT	6.39±0.06	59.07±0.51
LLaMA-2-7B$_{\text{KNN-INST-12K}}$	SFT	**6.45**±0.06	**59.22**±0.60
LLaMA-2-7B-Chat	SFT+RLHF	7.23±0.04	88.50±0.45

Table 1: Performance of KNN-INSTRUCT and baseline models on MT-Bench and AlpacaEval. We evaluate each model for 5 times and report the mean and standard deviation. The best results achieved by SFT-only models are bolded, while the global best results are underlined. NOTE: In this table, GPT-3.5-Turbo-0125 serves as a judge.

GPT-3.5-Turbo. The two synthetic baseline datasets would be referred to as `Alpaca-12k` and `Evol-Inst-12k`.

- Vicuna is a crowdsourcing method, while UltraChat is based on user simulation. We cannot not reproduce them, so we directly randomly sample 12k conversations from their publicly released data. The two sampled baseline datasets are named after `ShareGPT-12k` and `UltraChat-12k`.

Backbones We conduct SFT experiments on LLaMA-2-7B (Touvron et al., 2023) and Qwen-7B (Bai et al., 2023), which are both high-performance 7B pre-trained models.

Benchmarks Given that traditional LLM benchmarks like MMLU (Hendrycks et al., 2020) fail to be consistent with human users (Zheng et al., 2024a), we select two benchmarks based on open-end questions, which employ the rigorous GPT-4 as a judge and have received wide recognition:

- **AlpacaEval** (Taori et al., 2023) consists of 805 single-turn questions. The GPT-4 serves as a judge to compare the LLM and the baseline model text-davinci-003 on their responses. The average win rate will be reported as evaluation result.

- **MT-Bench** (Zheng et al., 2024a) consists of 80 high-quality 2-turn questions across 8 categories. For each question, the GPT-4 is used to give a score from 1 to 10. The average score will be reported as evaluation result.

In consideration of evaluation expenditure, we substitute GPT-3.5-Turbo (released on Jan 25, 2024) for GPT-4. All evaluations are repeated 5 times to ensure stability, and their mean and standard deviation will also be reported.

Training Details We conduct full-parameter SFT on four NVIDIA A800 80G GPUs for three epochs. We adopt an initial learning rate of 1×10^{-5}, a maximum sequence length of 2048 tokens, and a total train batch size of 128. The training takes about 1.5 hours. More training details are available at Appendix A.1.

Results Table 1 reveals that for both LLaMA-2-7B and Qwen-7B, KNN-INSTRUCT surpasses all baselines on both MT-Bench and AlpacaEval. In addition, we incorporate two strong models trained by RLHF, Qwen-7B-Chat and LLaMA-2-7B-Chat in the table. We can see that for Qwen-7B, KNN-INSTRUCT has an edge over Qwen-7B-Chat on both benchmarks. These empirical results preliminarily prove the effectiveness of KNN-INSTRUCT.

Data Analysis We also explore several statistics of KNN-INST-12K and the four baseline datasets: the vocabulary size, the average turns, the average length of instructions and turns, and the lexical diversity. We use the QwenTokenizer (Bai et al., 2023) for tokenization, and the MTLD (McCarthy and Jarvis, 2010) to compute lexical diversity. We

present the results in Table 2, where our KNN-INST-12K scores 4th, 3th, 3th, 2nd, 2nd on the five metrics, respectively.

4.2 Further Experiment

We employ the intelligent but expensive GPT-4-Turbo (released on Apr 9 2024) to construct a high-quality dataset, KNN-INST-12K*, and apply it to Qwen2-7B with all hyper-parameters consistent with before.

Baselines We select five competitive baseline models: 1) The state-of-the-art 7B model Starling-LM-7B (Zhu et al., 2023) 2) OpenChat-3.5 (Wang et al., 2023) 3) Zephyr-7B-β (Tunstall et al., 2023) 4) Qwen2-7B-Instruct 5) GPT-3.5-Turbo-0125, where 1) 2) 3) are the top three 7B models in LMSYS Leaderboard.

Benchmark We use MT-Bench to do evaluation, where GPT-4-Turbo-2024-04-09 serves as a judge.

Results As can be seen from Table 3, or model scores 7.64 on MT-Bench (8.23 in the first turn, 7.05 in the second turn), which outperforms all 7B models. More impressively, our model surpasses Qwen2-7B-Instruct (8.20 in the first turn, 7.31 in the second turn) and GPT-3.5-Turbo-0125 (7.96 in the first turn, 7.75 in the second turn) in the first turn. However, our model struggles to maintain the advantage in the first turn, lagging 0.12 and 0.22 in the total score, respectively. A possible explanation is that KNN-INSTRUCT only produces single-turn conversations.

5 Ablation Study

We conduct a few ablation studies: on selection of hyper-parameter K, on the procedure of KNN deduction, on different seeds, on the scalability of our method, and on the embedding similarity filter, which helps to gain a deeper understanding of KNN-INSTRUCT. In this section, we use GPT-3.5-Turbo-0125 as a judge for MT-Bench and AlpacaEval, and LLaMA-Factory (Zheng et al., 2024b) to evaluate LLMs on MMLU.

5.1 On Selection of K

Intuitively, a larger hyper-parameter K brings more demo samples, encourages the quality of synthetic data, and meanwhile adds the costs of ChatGPT API (with more input tokens). In this paper, we select $K = 2$ (i.e., 3-shot) for simplicity and economic reasons.

In this section, we employ a strong open-source LLM, Qwen2-1.5B-Instruct as the teacher model to help explore the impact of K on model performance. We adopt $K = 2, 3, 4, 5, 6$ to construct five 12k-scale SFT datasets. We conduct SFT with them on Qwen2-1.5B, and evaluate the fine-tuned models with MT-Bench.

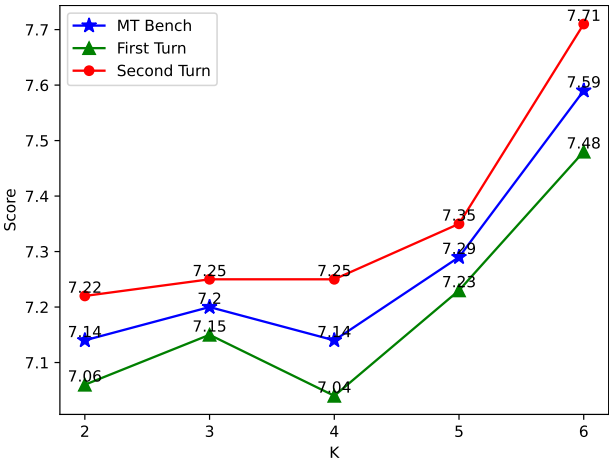

Figure 3: Model performance with different K.

Results As shown in Figure 3, there is an upward trend of model performance with the K increases. We have not observed a downward trend, and larger K still remains to be explored.

5.2 On KNN Deduction

KNN deduction is a key improvement that we make to the vanilla SELF-INSTRUCT. To verify its significance, we remove this procedure, and use three random samples to do few-shot learning instead. The corresponding method is called RAND-INSTRUCT. We construct a dataset RAND-INST-12K in the same setting with KNN-INST-12K. Applying the two datasets to Qwen-7B and LLaMA-2-7B, we make evaluation with MMLU, MT-Bench and AlpacaEval. For MMLU, we use the notable project LLaMA-Factory (Zheng et al., 2024b).

Results In Table 4, KNN-INSTRUCT outperforms RAND-INSTRUCT in five of six situations, which strongly demonstrates the significance of KNN deduction.

5.3 On Different Seeds

In previous section, we assume that for bootstrapping methods, a large, diverse, high-quality seed dataset helps to build a dataset with the same characteristics. Here, we select two different datasets to explore the impact of seed dataset:

Dataset	Vocab. Size	Avg Turns	Avg. Inst. Len.	Avg. Turn Len.	Lex. Diversity
Alpaca-12k	49312	1.0	79.10	240.58	67.81
Evol-Inst-12k	58907	1.0	**105.41**	**427.86**	82.58
ShareGPT-12k	**106339**	**3.42**	65.17	316.84	70.42
UltraChat-12k	62528	3.16	76.66	366.05	**94.81**
KNN-INST-12K	54648	1.0	78.79	421.57	86.58

Table 2: Several statistic of KNN-INST-12K and the four baseline datasets. For each colomn, the highest value is bolded, while the second highest value is underlined.

Model	Alignment	First Turn	Second Turn	Average
Starling-LM-7B	SFT+RLAIF	7.69	7.26	7.48
OpenChat-3.5	SFT	7.41	6.72	7.07
Zephyr-7B-β	SFT+DPO	7.03	6.04	6.53
Qwen2-7B$_{\text{KNN-INST-12K}}$*	SFT	**8.23**	7.05	7.64
Qwen2-7B-Instruct	SFT+RLHF	8.20	7.31	7.76
GPT-3.5-Turbo-0125	/	7.96	**7.75**	**7.86**

Table 3: Performance of KNN-INSTRUCT and five strong baselines on MT-Bench. We additionally present detailed scores for the first and second turns of MT-Bench. For each colomn, the highest value is bolded, while the second highest value is underlined. NOTE: In this table, GPT-4-Turbo-2024-04-09 serves as a judge.

- **Manual-175** is the seed dataset of SELF-INSTRUCT. It consists of 175 carefully curated conversations by human experts. The instances of Manual-175 contains three keys: `instruction,input,output`, while our instances contain two keys: `instruction,response`. To bridge this gap, we concatenate the `instruction` and `input` of vanilla Manual-175 with a `"\n"`.

- **ShareGPT-en** is filtered from ShareGPT, a realistic human-AI conversation dataset shared by users. We only select single-turn conversations from the publicly available ShareGPT dataset. To ensure quality, we design some rules to eliminate low-quality ones of the publicly available ShareGPT dataset. We also discard conversations whose instructions contain non-English characters. The filtered ShareGPT-en dataset consists of 13,460 single-turn conversations.

We replace our Seeds-3k with Manual-175, ShareGPT-en, reproduce KNN-INSTRUCT, and thus obtain two synthetic datasets: KNN-INST-MA-12K, KNN-INST-SG-12K, respectively. We apply them to Qwen-7B, evaluate the fine-tuned models with MT-Bench, and make comparison with Qwen-7B$_{\text{KNN-INST-12K}}$.

Results Table 5 reveals that despite Manual-175's high-quality, Qwen-7B$_{\text{KNN-INST-MA-12K}}$ does lag a lot compared with other models. Qwen-7B$_{\text{KNN-INST-SG-12K}}$ ranks second, we think it's the scale and diversity of KNN-INST-SG-12K that contributes to its good performance. Finally, our Seeds-3k consists of both real-world and synthetic data, and incorporates human annotation to ensure quality, so Qwen-7B$_{\text{KNN-INST-12K}}$ ranks first. This comparison demonstrates that, for bootstrapping methods, a large, diverse and high-quality seed dataset will significantly contributes to the high performance of SFT model.

5.4 On Scalability

Our previous experiments in Section 4.1 limit the scale of SFT dataset to 12k. In this section, we employ Qwen2-1.5B-Instruct to successively derive 6 datasets of 6k, 12k, 18k, 24k, 30k, 36k scale.We apply these datasets to Qwen2-1.5B and conduct evaluation on the SFT models with MT-Bench.

Results Figure 4 shows that with a larger SFT dataset, the model performance increases at first and then decreases. This empirical result reveals that for teacher model Qwen2-1.5B-Instruct, initial dataset Seeds-3k and $K = 2$, the optimal data scale for KNN-INSTRUCT is 30k, which is about ten times the original scale.

Model	MT-Bench	AlpacaEval(%)	MMLU
Qwen-7B$_{\text{KNN-INST-12K}}$	**7.38**	**75.86**	**56.61**
Qwen-7B$_{\text{RAND-INST-12K}}$	7.10	74.47	54.63
LLaMA-2-7B$_{\text{KNN-INST-12K}}$	**6.45**	**59.22**	43.46
LLaMA-2-7B$_{\text{RAND-INST-12K}}$	6.13	55.20	**44.36**

Table 4: Impact of KNN deduction. For each pre-trained model, the higher score on a benchmark is bolded.

Model	Seed	Seed Size	MT-Bench
Qwen-7B$_{\text{KNN-INST-MA-12K}}$	Manual-175	175	6.72
Qwen-7B$_{\text{KNN-INST-SG-12K}}$	ShareGPT-en	**13,460**	7.21
Qwen-7B$_{\text{KNN-INST-12K}}$	Seeds-3k	3,026	**7.38**

Table 5: Model performance with different seed datasets.

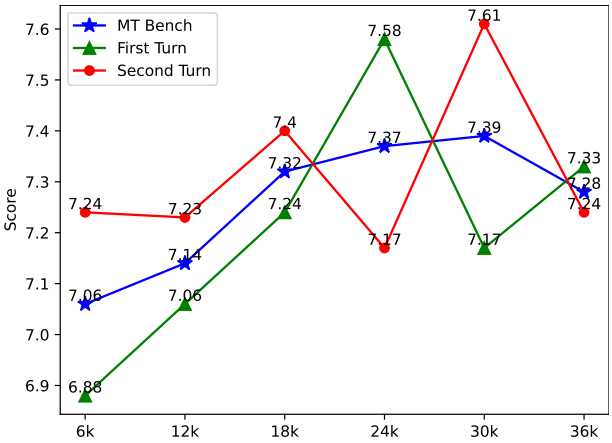

Figure 4: Model performance with different data scale.

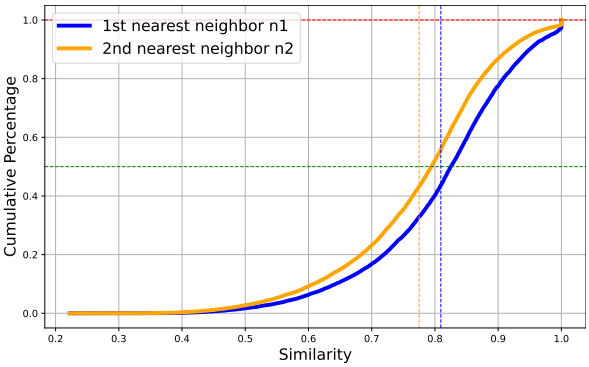

Figure 5: The similarity distribution of (n, n1), (n, n2).

5.5 On Embedding Similarity Filter

We explore our method on the embedding similarity. During the process of KNN-INSTRUCT, for each selected sample n, we collect its similarity with its 2 nearest neighbors n_1, n_2 and plot the distribution. As shown in Figure 5, the median of (n, n_1) is over 0.8, while that of (n, n_2) is about 0.77, both are very high. To control this, we add a similarity filter during iteration: `If the similarity` (n, n_1) `surpasses 0.7, continue.`

Method	MT-Bench	MMLU
KNN-INSTRUCT	7.14	55.23
KNN-INSTRUCT (+FILTER)	**7.42**	**55.37**

Table 6: Model performance with / without filter.

With Qwen2-1.5B-Instruct, we obtain two 12k datasets by KNN-INSTRUCT and KNN-INSTRUCT (+FILTER), apply them to Qwen2-1.5B, and use MT-Bench, MMLU to make evaluation.

Results As shown in Figure 6, the model performance increases with a similarity filter. The threshold 0.7 is arbitrarily set and more explorations are need on the selection of threshold.

6 Conclusion

In this paper, we introduce an automatic instruction construction framework named KNN-INSTRUCT. We make improvements on the sampling strategy and seed dataset over current bootstrapping methods. Moreover, we remove the redundant instruction elimination procedure to improve efficiency. We conduct a lot of controlled experiments to demonstrate the significance of KNN-INSTRUCT, where our strongest model Qwen2-7B$_{\text{KNN-INST-12K*}}$ surpasses all current 7B models on MT-Bench. We hope this work would contribute to the research of LLM alignment.

Limitations

In this section, we discuss some limitations and potential research directions of this work.

Multi-turn Conversations As shown in Appendix B.2, our KNN-INSTRUCT construct only single-turn conversations. The second experiment has revealed that our model does well in the first-turn question, but fail to perform as well as GPT-3.5-Turbo-0125 in the second-turn. In the future, we would like to upgrade KNN-INSTRUCT to construct multi-turn conversations, which might help strengthen the multi-turn conversational capability of SFT-only LLMs.

Larger Scale Dataset In this work, our data size is limited to 12k for the sake of saving time and cost, which is smaller than many previous works like Alpaca (52k), Vicuna (70k), and WizardLM (1.5M). We also explore the scalability of KNN-INSTRUCT in Section 5.4, which is around $10\times$ the original scale. Nowadays, with stronger and cheaper LLMs like GPT-4o-mini, we believe the scalability of KNN-INSTRUCT could be further improved.

Acknowledgments

We would like to thank all the reviewers for their insightful suggestions to improve this paper. This research is supported by Artificial Intelligence-National Science and Technology Major Project 2023ZD0121200 and National Natural Science Foundation of China under Grant 62222212.

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*.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Chuyi Kong, Yaxin Fan, Xiang Wan, Feng Jiang, and Benyou Wang. 2023. Platolm: Teaching llms via a socratic questioning user simulator. *Preprint*, arXiv:2308.11534.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Philip M McCarthy and Scott Jarvis. 2010. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

OpenAI. 2022. Chatgpt. Accessed on March 28, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th annual meeting of the Association for Computational Linguistics, pages 311–318.

ShareGPT. 2023. ShareGPT: Share your wildest ChatGPT conversations with one click. — sharegpt.com. https://sharegpt.com/. [Accessed 03-04-2024].

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. https://github.com/tatsu-lab/stanford_alpaca.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Sanh Victor, Webson Albert, Raffel Colin, Bach Stephen, Sutawika Lintang, Alyafeai Zaid, Chaffin Antoine, Stiegler Arnaud, Raja Arun, Dey Manan, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024a. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024b. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

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.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

A Implementation Details

In this section, we report details for our method KNN-INSTRUCT, and the SFT procedure.

A.1 KNN-INSTRUCT

We employ `sup-simcse-roberta-large` (Gao et al., 2021) for tokenization and embedding, and (1 - cosine similarity) for distance measurement. Our seed dataset is organized in the ShareGPT format (multi-turn conversation) format. A full round of KNN-INSTRUCT doubles the seed dataset.

With the Seeds-3k, we run KNN-INSTRUCT twice and get KNN-INST-12K. The system prompt and few-shot prompt template of KNN-INSTRUCT are shown as follows:

System Prompt

You are a helpful assistant designed to interpret and analyze user queries that have actually been proposed to AI assistant ChatGPT. Based on your analysis, you are capable of further crafting new queries.

Few-Shot Prompt Template

[Exemplar Dialog 1]
[Human User]: {Q1} [AI Assistant] {A1}
[Exemplar Dialog 1 Ends]
[Exemplar Dialog 2]
[Human User]: {Q2} [AI Assistant] {A2}
[Exemplar Dialog 2 Ends]
[Exemplar Dialog 3]
[Human User]: {Q3} [AI Assistant] {A3}
[Exemplar Dialog 3 Ends]
The passage above showcases several dialogs between a human user and the AI assistant, ChatGPT. As can be seen, real-world user queries are usually meaningful, diversified and grounded, reflecting practical needs, critical thinking or intriguing ideas. Now perform the following task.
Task: Position yourself as the user in question, and craft a new, high-quality instruction. Keep the following in mind:
1. **Relevance**: Incorporate your previous analysis, fully utilize these informative prior, ensuring that the new instruction aligns well with this user.
2. **Originality**: The new instruction should be distinguished to existing ones instead of naive imitation or transfer, so try your best in CREATIVITY;
3. **Standalone**: The new instruction should be self-contained and not depend on prior conversations.
4. **Format**: You should simply return a string as the new instruction.

A.2 Supervised Fine-Tuning

We conduct full-parameter SFT with open source repositories:

- For LLaMA-2-7B, we use FastChat (Zheng et al., 2024a), which is the repo for Vicuna and MT-Bench;
- For Qwen-7B, we use its official repo (Bai et al., 2023);
- For Qwen2-7B, we use the notable repository LLaMA-Factory.

Our operating system, CPU, GPU, Pytorch version and CUDA version are listed in Table 7. Table 8 shows the hyper-parameters we set in SFT. Parameters not listed would be set to the default value of its corresponding repo.

B Case Study

This section present instances for SELF-INSTRUCT and our KNN-INSTRUCT.

B.1 Instances of SELF-INSTRUCT

We reproduce SELF-INSTRUCT (3-shot) with GPT-3.5-Turbo API, and observe some low-quality examples as follows:

SELF-INSTRUCT Case 1

demo_1: A one-hour color video in YUV format has a frame resolution of 1920x1080 with a 4:2:2 color subsampling format, 8 bits for each component, and a frame rate of 30 frames/s. Determine the storage requirement for the video in Gbytes (3 sig. fig.).
demo_2: How has the city of Cape Town changed and evolved over the years, particularly in relation to urban development and socioeconomic factors?
demo_3: I'm a medical student and I want to practice my medical spanish. Can you pretend to be a patient and, in Spanish, we can pretend to do a patient encounter (I'll

OS	CPU	GPU	Pytorch	CUDA(Runtime)
Ubuntu 20.04.6 LTS	Intel Xeon Platinum 8358P@2.60GHz	8 * NVIDIA A800-SXM4-80G	2.3.0+cu121	12.1.66

Table 7: Experimental environment.

Hype-Param	Value	Hype-Param	Value
Num of Epochs	3	Per Device Batch Size	2
Gradient Accu. Steps	16	Learning Rate	1e-5
Max Seq Length	2048	LR Scheduler Type	cosine
BF16	True	Weight Decay	0.1
Warmup Ratio	0.01	Gradient Checkpointing	True

Table 8: SFT Hyper-parameters.

be the doctor and we'll speak spanish).
new_prompt: Can you provide an overview of the impact of climate change on coral reefs, particularly focusing on changes in ocean temperature and acidity levels?

SELF-INSTRUCT Case 2

demo_1: Can you take up a more elaborate culinary challenge? Create a three-course meal that highlights the unique flavors of your region and is composed of locally-sourced ingredients that are in season. However, in addition to using each ingredient only once throughout the meal, you must also ensure that at least one course is vegan or gluten-free. Push your culinary skills to the limit and showcase your adaptability and creativity in designing a delicious menu!
demo_2: How has the music of Bob Marley influenced reggae in Jamaica and beyond?
demo_3: Extract the dates from the following paragraph: The 20th (twentieth) century began on January 1, 1901 (MCMI), and ended on December 31, 2000 (MM).[1] The 20th century was dominated by significant events that defined the modern era: sixth mass extinction, Spanish flu pandemic, World War I and World War II, nuclear weapons, nuclear power and space exploration, nationalism and decolonization, the Cold War and post-Cold War conflicts, and technological advances. These reshaped the political and social structure of the globe.
new_prompt: Can you provide a detailed explanation of the impact of climate change on the biodiversity of coral reefs worldwide, including specific examples of how rising sea temperatures and ocean acidification have affected coral ecosystems?

B.2 Instances of KNN-INSTRUCT

KNN-INSTRUCT Case 2

demo_1: What is Gaussian noise?
demo_2: What are the various applications of Gaussian noise in image processing beyond denoising tasks?
demo_3: What are the key differences and similarities between the Gaussian elimination method and the method of substitution when solving a system of linear equations?
new_prompt: What are the advantages and limitations of using Gaussian noise as a steganographic method for hiding secret information within digital images?

KNN-INSTRUCT Case 3

demo_1: Create a weird conversation with old people about some popular topic where they are using modern colloquialisms in the wrong way.
demo_2: an you provide some humorous examples of misinterpreted idioms or sayings that could confuse non-native English

speakers?
demo_3: What are some common phrases or expressions that people often use incorrectly because they are based on literal interpretations rather than their intended figurative meanings?
new_prompt: What are some clever ways to incorporate modern slang and technology references into conversations with older adults without confusing them or making it seem forced?