Towards Fully-Automated Dataset Construction

Anonymous ACL submission

Abstract

Application of advanced large language models to data annotation and synthesis automatized the process of dataset construction, yet participation of human experts is still inevitable. This paper proposed an approach to *fully-automated* dataset construction. With only the minimal information, high-quality datasets can be constructed fully automatically for various tasks. Utilizing constructed datasets for both supervised finetuning and few-shot learning improved performance constantly. Furthermore, the first mathematical formalization of the process of dataset construction is presented, providing the theoretical foundation of the proposed method.

1 Introduction

In the field of natural language processing (NLP), the importance of data cannot be overstated. High-quality data is prevalent across various tasks, such as syntactic parsing (Xue et al., 2005; Nivre et al., 2020), machine translation (Nakazawa et al., 2016; Kocmi et al., 2024), summarization (Hermann et al., 2015; Koupaee and Wang, 2018), promoting the research significantly.

Construction of datasets is expensive, due to the dependence on professional knowledge of human experts in many domains, and hard work of countless human annotators. Thanks to the progress of large language models (LLMs), it becomes possible to construct datasets in an *automated* manner (Tan et al., 2024). With the help of advanced LLMs like GPT-4 (Achiam et al., 2023) or Llama-2 (Touvron et al., 2023), high-quality synthetic datasets have been constructed in many areas, such as recommendation system (Acharya et al., 2023; Shen et al., 2024), hallucination detection (Liu et al., 2023b; Manakul et al., 2023), information extraction (Jeronymo et al., 2023; Bonn et al., 2024).

In spite of the utilization of LLMs, human experts are still needed for providing *a priori* domain knowledge (Huang et al., 2024a) or assuring *a posteriori* quality (Huang et al., 2024b). We want to ask the following question. **Can datasets be constructed in a *fully-automated* manner?**

We propose the first step towards fully-automated dataset construction. Given the minimal information about NLP task and data format (ref. §3.1), datasets can be constructed in a fully-automated manner (ref. §3.2). On various tasks with completely different characteristics (ref. §4), high-quality datasets are constructed automatically (ref. §5), and the utilization of constructed datasets improved performances on different tasks, with different learning strategies.

The proposed method has a solid mathematical foundation (ref. §2). Furthermore, low-cost and high-flexibility distinguish the propose method from other studies on dataset construction (ref. §7), indicating the particular usefulness of our method on scenarios such as emerging business or niche market, where human experts are scarce.

Our contributions are summarized as follows.

- We first formalized dataset construction, provided the mathematical proof on the possibility of fully-automated dataset construction.
- Based on the mathematical theory, we proposed a novel paradigm of fully-automated dataset construction with minimal information from human.
- Experiments verified the versatility of the proposed method. High-quality datasets can be constructed and contribute to learning for different tasks using different learning strategies.

2 Formalization of Dataset Construction

We first illustrate the whole picture of formalization (ref. §2.1), then delve into details: introducing necessary terminologies: dataset (ref. §2.2), signed word sequence (ref. §2.3), and dialogue (ref. §2.4), then proving their equivalence (ref. §2.5), which

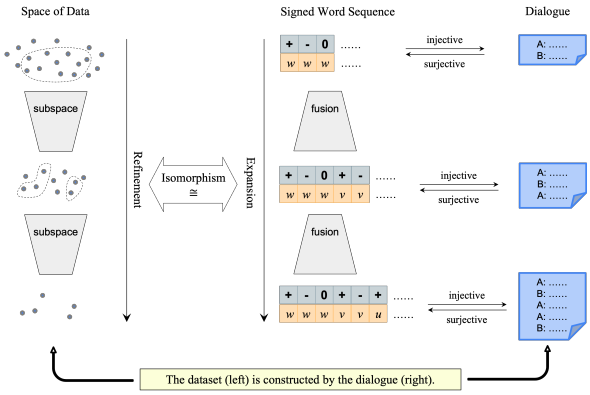

Figure 1: Overview of the formalization of dataset construction.

is the mathematical foundation of dataset construction (ref. §2.6)[1].

2.1 Overview

Overview of the formalization of dataset construction is illustrated in Figure 1. Intuitively, construction of a dataset (in the context of NLP) can be regarded as the process of selecting appropriate samples from the space of all possible word sequences. This selecting process can be divided into several steps. At each step, a subspace is found.

We define the concept of signed word sequence, which bridges the concepts of data and dialogues. We prove that finding subspaces of data space (i.e., refinement) is isomorphic to fusion operation of signed word sequences (i.e. expansion).

However, no signs are existed in real-world corpora. Fortunately, the mapping from dialogues to signed word sequences is surjective, making it possible to approximate signed word sequences using dialogues. Therefore, we can construct datasets by generating appropriate dialogues, which is achieved by the orchestration of advanced LLMs.

2.2 Dataset

In the context of NLP[2], data can take various forms: sentences, passages, dialogues, etc., and can be attached with different types of labels: category labels, sequence labels, etc. Despite the apparent variety of forms, they can be defined universally.

Definition 1 (Data)**.** *Data* $\mathcal{D}$ is a set of sequences $s = (t_1, t_2, \cdots) \in \mathcal{D}$, where $t_i (i \geq 1)$ is called a *token*, which can be a word, subword, label, etc.

[1]Refer Appendix A for more information, including detailed explanation, more examples, and proofs of all theorems.

[2]Throughout this paper, our discussion is focusing on text. Modals like images or audios are out of scope.

Example 1. For sentiment analysis, t_1 is the sentiment label, and $t_2, t_3, \cdots$ are words.

Example 2. For constituent parsing, following the format of Penn Treebank (Marcus et al., 1993), t_i can be a word, a syntax label, or a bracket.

We can prove the following theorem.

Theorem 1. *Any data* $\mathcal{D}$ *is topological space, and is Hausdorff.*

This allows us to give the definition of *dataset*.

Definition 2 (Dataset)**.** *Dataset* $\mathfrak{D}$ is a finite subspace of data $\mathcal{D}$.

We can prove the following property of data.

Theorem 2. $(\mathcal{D}, \cap)$ *is a monoid, where* $\cap$ *is the intersection of subspaces.*

Last, we define concepts related to *refinement*.

Definition 3 (Refinement)**.** *Refinement* of dataset $\mathfrak{D}$ is a tower $\mathfrak{D}_n \subseteq \mathfrak{D}_{n-1} \subseteq \cdots \subseteq \mathfrak{D}_1 \subseteq \mathfrak{D}$.

Definition 4 (Refinement mapping)**.** Given a refinement $\mathfrak{D}_n \subseteq \mathfrak{D}_{n-1} \subseteq \cdots \subseteq \mathfrak{D}_1$, a *refinement mapping* is a surjective mapping $f_i : \mathfrak{D}_i \to \mathfrak{D}_{i+1}$. We call f_i a *real refinement* when $\mathfrak{D}_i \subset \mathfrak{D}_{i+1}$, and we call f_i a *fake refinement* when $\mathfrak{D}_i = \mathfrak{D}_{i+1}$.

Theorem 3. *For any dataset* $\mathfrak{D}$*, there is a data* $\mathcal{D}$*, and a refinement, such that* $\mathfrak{D} \subseteq \cdots \subseteq \mathcal{D}$.

2.3 Signed Word Sequence

Most of the common objects in NLP like sentences, passages, documents can be regarded as word sequences. We extend this concept.

Definition 5 (Signed word sequence)**.** A *signed word sequence* is a sequence $W = (\mathrm{sgn}(w_1) \cdot w_1, \mathrm{sgn}(w_2) \cdot w_2, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$, where $w_i (i \in [1, n])$ is a word, and $\mathrm{sgn}(w_i) \in \{+1, -1, 0\}$. Particularly, if $\mathrm{sgn}(w_i) = 0$ for all $i \in [1, n]$, we call W an *empty sequence*, denoted by $\mathfrak{o}$.

We can define the *fusion* operation on two signed word sequences.

Definition 6 (Fusion)**.** Given two signed word sequences $W = (\mathrm{sgn}(w_1) \cdot w_1, \mathrm{sgn}(w_2) \cdot w_2, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$ and $V = (\mathrm{sgn}(v_1) \cdot v_1, \mathrm{sgn}(v_2) \cdot v_2, \cdots, \mathrm{sgn}(v_n) \cdot v_m)$, if for any $i \in [1, \max(m, n)]$, one of the following equations holds true:

$$\mathrm{sgn}(w_i) = 0,$$
$$\mathrm{sgn}(v_i) = 0,$$
$$w_i = v_i,$$

then we define the *fusion* of W and V as

$$W \star V = (u_1, u_2, \cdots, u_{\max(m,n)}),$$

where

$$u_i = \begin{cases} 0 & \text{if } \mathrm{sgn}(w_i) + \mathrm{sgn}(v_i) = 0, \\ \mathrm{sgn}(w_i) \cdot w_i & \text{if } \mathrm{sgn}(v_i) = 0, \\ \mathrm{sgn}(v_i) \cdot v_i & \text{otherwise.} \end{cases}$$

Theorem 4. *$(W, \star)$ is an abelian group, where W is a signed word sequence.*

The *subsequence* of a signed word sequence is defined as follows.

Definition 7 (Subsequence)**.** The *subsequence* of signed word sequence $W = (\mathrm{sgn}(w_1) \cdot w_1, \mathrm{sgn}(w_2) \cdot w_2, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$ is

$$W' = (\mathrm{sgn}(w'_1) \cdot w_1, \mathrm{sgn}(w'_2) \cdot w_2, \cdots, \mathrm{sgn}(w'_n) \cdot w_n),$$

where $\mathrm{sgn}(w'_i) \in \{0, \mathrm{sgn}(w_i)\}$, denoted by $W' \triangleleft W$ or $W \triangleright W'$.

Theorem 5. *The relation of subsequence is a partial order.*

Last, we define the concept of *expansion*.

Definition 8 (Expansion)**.** *Expansion* of signed word sequence W is a tower $W \triangleleft W_1 \triangleleft W_2 \triangleleft \cdots \triangleleft W_n$.

2.4 Dialogue

In real NLP tasks, we do not encounter signed word sequences. However, we can use *dialogues* to approximate signed word sequences.

Definition 9 (Dialogue)**.** A *dialogue* $\mathcal{L} = (u_1, u_2, \cdots, u_n)$ is a sequence of *utterances*. An utterance is a sequence $u_i = (\mathfrak{s}_i, u_i^1, \cdots, u_i^{n_u})$, where $\mathfrak{s}_i$ is a special token named *speaker*, and u_i^j are words.

Theorem 6. *The fusion of any numbers of signed word sequences can be approximated by a dialogue.*

2.5 Galois Equivalence

Now we discuss the connections of the concepts defined above.

Theorem 7. *$(\mathcal{D}, \cap)$ and $(W, \star)$ are isomorphic.*

This demonstrates the equivalence of dataset and signed word sequence. Furthermore, we have the following theorem[3].

[3]The name of this theorem originates from its similarity to the fundamental theorem of Galois theory.

Theorem 8 (Galois Theorem on Dataset)**.** *For any dataset $\mathfrak{D}$ and signed word sequence W, there is a bijection between the refinement of $\mathfrak{D}$ and the expansion of W.*

In particular, we have the following corollary.

Corollary 1. *For any dataset $\mathfrak{D}$, there is a bijection between its refinement and an expansion of $\mathfrak{o}$.*

2.6 Dataset Construction

We are ready to give a formal definition of *construction* of a dataset.

Definition 10 (Construction)**.** *Construction* of dataset $\mathfrak{D}$ is the refinement of data $\mathcal{D}$ such that $\mathfrak{D} \subseteq \cdots \subseteq \mathcal{D}$.

Theorem 3 ensures the existence of the construction, Theorem 8 ensures the possibility of construction a dataset by expansion of signed word sequence, and Corollary 1 says that the signed word sequence can be the empty sequence $\mathfrak{o}$. Let the expansion of $\mathfrak{o}$ be $\mathfrak{o} \triangleleft \cdots \triangleleft W$. According to Theorem 6, W can be approximate by a dialog $\mathcal{L}$. To summarize, we have the following corollary.

Corollary 2. *Any kind of dataset can be constructed by dialogues.*

Based on the type of refinement mappings, we can categorize utterances into two groups.

- Utterance for operation, corresponding to real refinement mappings.
- Utterance for guidance, corresponding to fake refinement mappings.

In the context of LLM-based dialogue, word sequences for operation can be generated by either the LLM itself or external tools, indicating the importance of tool-augmented LLMs (Schick et al., 2024; Wang et al., 2024b; Qin et al., 2024).

3 Methodology

3.1 Minimal Information from Human

The ideal scenario would be to construct dataset with zero information from human, which is theoretically impossible. According to Definition 2, to construct a dataset

$$\mathfrak{D} = \{s \in \mathcal{D} | s \text{ satisfies some conditions}\} \subseteq \mathcal{D}$$

we need to (and only need to) provide the following information.

- Information about $\mathcal{D}$, i.e., data format.
- Conditions on s, i.e., task description.

This is the minimal information from human.

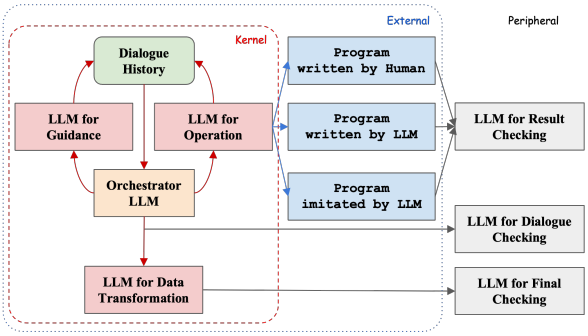

Figure 2: Framework of fully-automated dataset construction.

3.2 Fully-Automated Dataset Construction

Figure 2 shows the proposed framework of fully-automated dataset construction. The minimal information from human is used to initialize all LLMs. The framework is composed of three layers: kernel layer, external layer, peripheral layer.

3.2.1 Kernel Layer

The kernel layer consists of four LLMs and one internal state (i.e., dialogue history).

Dialogue history. Initially, the dialogue history is empty (i.e., a signed word sequence $W = (0, 0, \cdots)$, corresponding the the case $\mathfrak{D} = \mathcal{D}$). Utterances in the dialogue are signed word sequences, and the dialogue history is the fusion of all utterances. The dialog history with n-turns corresponds a refinement $\mathfrak{D}_n \subseteq \cdots \subseteq \mathfrak{D}_1 \subseteq \mathcal{D}$.

LLM for guidance. This LLM is used to generate word sequences corresponding to *fake* refinement mappings. Generally, these word sequences guides the dialogue flow to the correct direction.

LLM for operation. This LLM is used to generate word sequences corresponding to *real* refinement mappings. Intuitively, these word sequences can be regarded as instructions on operating the dataset. Following the instructions, LLM discard examples in the dataset constructed, either by itself or programs in the external layer.

Orchestrator LLM. The discrimination between *real* and *fake* refinement mappings is conducted by the orchestrator LLM. Based on the dialogue history (i.e., an expansion of signed word sequence $W_1 \triangleleft \cdots \triangleleft W_n$), the orchestrator LLM choose the appropriate LLM to generate word sequences W_{n+1}. The following two points need to be noted.

- There are two choices: LLM for guidance, LLM for operation. Invoking external programs is beyond the scope.
- It is possible that W_i and W_{i+1} are generated by the same LLM.

LLM for data transformation. Although a dialogue approximate the fusion of signed word sequences (Theorem 6) which corresponds to a dataset construction (Theorem 8), the constructed dataset is invisible. We resort to LLM to retrieve the constructed dataset based on the dialogue.

3.2.2 External Layer

Given the minimal information from human, the dataset can be constructed by the kernel layer. In *theory*, the kernel layer is sufficient. In *practice*, it is common that additional information is provided and external programs are available. The external layer is designed for invoking external programs, which fall into one of three categories.

Program written by human. These programs condensed rich knowledge of human experts. Invoking them ensures the quality of dataset.

Program written by LLM. When additional information on external programs is provided, we ask LLMs to generate high-quality codes (Liu et al., 2023a; Jiang et al., 2024).

Program imitated by LLM. An alternative way to utilize additional information is to imitate programs. Instead of generating real codes, LLMs are asked to predict the output given the inputs.

3.2.3 Peripheral Layer

To further ensure the quality of constructed datasets, we add the peripheral layer for checking intermediate and final results, consisting of three LLMs, to which LLM-as-a-Judge methods (Bai et al., 2023; Gu et al., 2024) are adopted.

LLM for result checking. Programs may generate abnormal outputs, especially when programs are written or imitated by LLMs, resulting in pathological utterances in the dialogue. Running results of programs are checked by LLM.

LLM for dialogue checking. The dialogue that approximate dataset construction may contain erroneous utterances, such as mistakes on invocation of external programs. Quality of the dialogue is checked by LLM.

LLM for final checking. Dataset construction is complete after transforming the dialogue to appropriate format. We use an LLM for final check, ensuring high-quality in multiple dimensions.

4 Experimental Setup

Throughout this paper, unless otherwise noted, we use GPT-4o (version 2024-08-26) as the LLM. Efforts on prompt engineering and task design are introduced below.

4.1 Prompt Engineering

Performance of LLM is affected by prompts significantly (Sahoo et al., 2024), indicating the necessity of prompt engineering.[4]

4.1.1 Principles of Prompt Engineering

Prompt engineering of the proposed framework is a trade-off between two principles below.

Principle 1: Carefully designing prompts to achieve task-independent high-quality dataset construction. To make sure that datasets can be constructed using the minimal information from human independent of tasks, we design prompts carefully, fully extracting the potential of LLMs while eliminating all task-specific information.

Principle 2: Deliberately sloppy engineering prompts to avoid overfitting. The proposed method is the first step towards fully-automated dataset construction. We do *not* expect to construct perfect dataset using current state-of-the-art LLMs on tasks tackled in this paper. Excessive prompt engineering may pose a danger of overfitting, which should be avoided.

4.1.2 Prompts for LLMs in Peripheral Layer

Intermediate/Final results are checked in multiple dimensions using LLM-as-a-Judge methods.

LLM for result checking. This LLM is aimed at deciding the correctness of program outputs, following previous research (Shinn et al., 2023), we ask the LLM to answer the following Yes/No question: *was the program exectued normally?*

LLM for dialogue checking. This LLM check the quality of dialogue in five dimensions.

- Awareness of program invoking. The timing of program invoking is appropriate.
- Choice of program. Among given programs, the correct program has been chosen.
- Arguments of program. The input arguments of the chosen program are correct.
- Guidance utterance. The guidance is useful for dataset construction.
- Operation utterance. The operation on dataset matches the given task.

LLM for final checking. This LLM check the quality of constructed dataset in two dimensions.

- Agreement to requirements of the task.
- Gap with real-world scenarios.

Following previous research (Zhu et al., 2023; Li et al., 2024), for each dimension, the LLM gives a discrete score ranging from 1 (worst) to 10 (best). Generated dataset will be adopted only when scores in all dimensions are larger than or equal to 7.

4.2 Tasks

To verify the versatility, we construct dataset on three tasks with different characteristics.

4.2.1 English-Chinese Machine Translation

English-Chinese machine translation (MT) is a traditional task with long history (King and Chang, 1963), and has achieved high performance with LLMs (Zhu et al., 2024). This task is aimed at demonstrating the effectiveness of the proposed method on traditional NLP tasks.

Dataset format. The dataset consists of sentence pairs $s = (e_1, \cdots, e_m, \texttt{[SEP]}, c_1, \cdots, c_n)$, where $e_1, \cdots, e_m$ are English words, and $c_1, \cdots, c_n$ are Chinese words.

Information from human. [5] Besides the minimal information, we provide a brief description on Google Cloud Translation API[6], as the program written by human.

4.2.2 Chatbot for Hotel Reservation

Chatbot for hotel reservation (Putri et al., 2019; Hayashibe, 2020) is more focused on business scenarios, where factors related to real world have to be taken into consideration. This task is aimed at demonstrating the effectiveness on complex business tasks.

[4] Prompts of all LLMs are listed in appendix C.

[5] Refer Appendix D for the information from human of all three tasks.

[6] https://cloud.google.com/translate

Task	Data	Quality
English-Chinese Machine Translation	The sun sets in a rosier hue, casting long shadows over the city. 夕阳西下，霞光渐浓，在城市上空投下长长的影子。 Hey, how's it going? Long time no see! 嘿，最近怎么样？好久不见了！ Laughter is the shortest distance between two people. 笑声是两人之间最短的距离。 The quantum theory revolutionized our understanding of atomic and subatomic processes. 量子理论彻底改变了我们对原子和亚原子过程的理解。 Breaking news: The economy has shown signs of recovery after a challenging year. 突发新闻：经历了充满挑战的一年之后，经济已出现复苏迹象。 Can you send me the files by email when you get a chance? 你方便的时候能给我发邮件吗? The latest research in renewable energy shows promising advancements. 最新的可再生能源研究显示了令人鼓舞的进展。 Democracy relies on the active participation of its citizens. 民主依赖于公民的积极参与。	Agreement: 10 Reality: 9
Chatbot for Hotel Reservation	Customer: Hi, I'm planning a trip to New York from 11/10 to 11/15. Can you recommend any hotels? Chatbot: Sure! How many people will be staying, and what type of room would you prefer? Customer: It'll be me and my partner. We would like a non-smoking double room. Chatbot: Great! Do you have a budget or price range in mind for your stay? Customer: We'd like to keep it under $200 per night. Chatbot: Understood. Would you need any specific amenities, such as wifi, breakfast included, or a gym? Customer: Wifi and breakfast are essential. Chatbot: I've found several hotels that fit your criteria. Would you like information on cancellation policies or customer reviews? Customer: Yes, could you tell me about the cancellation policies? Chatbot: Certainly. Most hotels allow free cancellation up to 48 hours before check-in. Any other preferences or questions? Customer: No, that's all for now. Thank you!	Agreement: 10 Reality: 9
Mathematical Problem Solver	Problem: A chemist has a 30% saline solution and a 70% saline solution. How many liters of each should be mixed to obtain 10 liters of a 50% saline solution? Equations: $x + y = 10, 0.3x + 0.7y = 5$ Answer: $x = 5$ liters, $y = 5$ liters Problem: A car travels 150 miles in the same amount of time a bicycle travels 30 miles. If the speed of the car is 20 mph faster than twice the speed of the bicycle, what are the speeds of the car and the bicycle? Equations: $t = 150/v_c, t = 30/v_b, v_c = 2 * v_b + 20$ Answer: $v_b = 6.67$ mph, $v_c = 33.3$ mph Problem: Sarah invests $10,000 in two parts: one part at 5% simple interest and the other at 8%. At the end of the year, the total interest earned is $700. How much did she invest at each rate? Equations: $x + y = 10000, 0.05x + 0.08y = 700$ Answer: $x = \$3,333, y = \$6,667$ Problem: The perimeter of a rectangle is 60 meters. If the length is 5 meters more than twice the width, what are the dimensions of the rectangle? Equations: $2l + 2w = 60, l = 2w + 5$ Answer: $l = 21.67$ meters, $w = 8.33$ meters	Agreement: 10 Reality: 9

Table 1: Examples of constructed dataset. Quality scores are given by the LLM for final checking. For each task, all examples are constructed within one dialog.

Dataset format. The dataset consists of dialogues, whose format is defined at Definition 9.

Information from human. The task description briefly introduces the scenario of the chatbot. Besides the minimal information, we also provided the interfaces of four external programs that are necessary for hotel reservation, all of which are imitated by LLMs.

4.2.3 Mathematical Problem Solver

Despite the success of LLMs on mathematics (Trinh et al., 2024; Azerbayev et al., 2024), there are still many issues that need to be addressed (Tong et al., 2024). We focus on solving mathematical problems using systems of equations. This task is aimed at demonstrating the effectiveness on deterministic reasoning tasks.

Dataset format. The dataset consists of 3-tuples of question, equations, and answer, i.e., $s = (p_1, \cdots, p_l, \texttt{[SEP]}, e_1, \cdots, e_m, \texttt{[SEP]}, a_1, \cdots, a_n)$, where $p_1, \cdots, p_l$ are words in the problem, $e_1, \cdots, e_m$ are equations, and $a_1, \cdots, a_n$ are words in the answer.

Information from human. Domain of problems is given in the task description. We also provided the interfaces of three external programs. One is written (i.e., code generation) by LLM, and the other two are imitated by LLMs.

5 Quality Assessment by Human

The quality of constructed datasets of three tasks was assessed by human, both qualitatively and quantitatively.

5.1 Qualitative Assessment

Table 1 shows examples of constructed dataset[7]. High-quality datasets are constructed for all three tasks. English passages are fluent and natural, the mathematical calculations are correct. Quality scores given by the LLM reflect the characteristics of datasets. This demonstrated the versatility of the proposed method.

5.2 Quantitative Assessment

We ask human to quantitatively assess the quality of constructed dataset. Specifically, for each task, we randomly sampled 100 examples from the constructed dataset. We hire human workers and asked them to evaluate the quality of each dataset from three aspects.

[7]Refer Appendix E for more examples of the constructed dataset for all three tasks.

Category	Metrics	Translation	Chatbot	Math
Dialogue	Awareness	9.55	9.06	9.72
	Choice	10.0	9.21	9.06
	Argument	10.0	9.95	9.98
	Guidance	9.70	9.32	9.77
	Operation	9.69	9.21	9.85
Data	Agreement	9.96	9.95	9.81
	Reality	9.86	9.03	9.95
Overall	Dialog	100%	97%	100%
	Data	98%	90%	97%

Table 2: Quantitative assessment by human. Scores in the dialogue and data categories are the average values of all examples.

- Quality of dialogues, which is evaluated from 5 dimensions identical to the LLM for dialogue checking.
- Quality of final data, which is evaluated from 2 dimensions identical to the LLM for final checking.
- Overall quality, which is the percentage of examples that obtained high scores (≥ 7) on all dimensions given by human workers.

Table 2 shows quantitative assessment results. For all three tasks, the average scores of all dimensions are larger than 9, indicating the high qualities of these datasets. The overall quality is larger than 95%, so the constructed dataset can be safely utilized for learning models.

6 Utilization of Constructed Datasets

Two series of experiments of dataset utilization are conducted: supervised finetuning and few-shot learning, demonstrating the usefulness of constructed datasets.

6.1 Supervised Finetuning

Experiments of supervised finetuning were conducted on the task of mathematical problem solver. The number of epochs was 2, and the learning rate was 3×10^{-5}. Note that the objective of supervised finetuning experiments is to verify the usefulness of constructed datasets, rather than to achieve state-of-the-art performance. Therefore, search of optimal hyperparameters was not conducted.

Using the proposed method, we constructed a dataset consisted of $1,000$ tuples of problems, equations and answers. Half (or all) of this dataset was used to finetune LLMs. Performance was evaluated on GSM8K dataset (Cobbe et al., 2021).

Configuration	LLaMA 7B	gemma-7b	GPT-4o
No FT.	11.0	51.9	93.1
FT. 500 (auto)	22.6	53.8	92.9
FT. 1000 (auto)	32.4	55.4	93.0
FT. 500 (GSM8K)	24.7	54.0	93.2
FT. 1000 (GSM8K)	32.9	55.7	93.1

Table 3: Test solve rates on GSM8K of raw LLMs (No FT.) or finetuned LLMs (FT.) with 500 or 1000 samples, either constructed automatically using the proposed method (auto) or sampled from GSM8K training data.

Table 3 shows performance with different configurations. On one hand, LLaMA 7B gained the most from finetuning. Although there was a narrow margin between "auto" and "GSM8K", performance improved significantly on both cases, indicating new information has been learned. On the other hand, GPT-4o gained little from finetuning. This is caused by two reasons: (1) the capability of GPT-4o is sufficient to solve problems in GSM8K with high accuracy, (2) the LLM used for automatic dataset construction was GPT-4o as well.

6.2 Few-shot Learning

Experiments of few-short learning were conducted on the task of the chatbot for hotel reservation. Specifically, the focus of this task is to test the capabilities of LLMs on tool utilization. The LLM was asked to achieve three different tasks related to tool utilization: (1) tool usage awareness, (2) tool selection, and (3) tool input generation, following previous researches (Huang et al., 2024a).

We constructed a small dataset for few-shot learning, using the proposed method. The dataset consisted of 20 dialogues related to hotel reservation, all of which contained utilization of external tools. We inserted k dialogues ($k = 1, 5, 10, 20$) at the end of prompts of LLMs.

We also constructed a larger dataset for evaluation in a semi-automated way, as follows. First, we generate a large number of dialogues using the proposed method. Then, dialogues that did not contain tool utilization were discarded. Last, we hired human workers to check tool utilization in remained dialogues, and correct possible errors. In this way, we obtained 100 dialogues.

Table 4 shows the results on three tasks. In this user-defined scenario aiming at approximating real business, GPT-4o performed bad. Few-shot learning with automatically constructed dataset im-

Configuration	Awareness	Selection	Input Generation
GPT-4o	0.45	0.39	0.51
+1-shot	0.53	0.45	0.60
+5-shot	0.58	0.49	0.63
+10-shot	0.62	0.52	0.66
+20-shot	0.67	0.55	0.72

Table 4: Accuracies of few-shot learning on tasks related to tool utilization in the scenario of hotel reservation.

proved accuracies significantly. The best performance was obtained by appending 20 new samples, yet only one new sample (i.e., 1-shot) also improved the performance.

7 Related Work

The development of advanced LLMs arouses emerging approaches to automatic dataset construction. Comparison with previous researches from various perspectives highlighted the advantages of the proposed method.

7.1 Perspective 1: Data Format

Data can take various forms in the area of NLP, and previous researches generally focus on one yet ignore others. Examples include studies on generating data with single labels (Chen et al., 2024; Martorana et al., 2024; Tekumalla and Banda, 2023), paired labels (Sun et al., 2024; Feng et al., 2024; Kim et al., 2023b), sentences (Wang et al., 2023a; Yu et al., 2023; Gupta et al., 2024), or dialogues (Kim et al., 2023a; Li et al., 2023; Liang et al., 2024).

Our method proposed an universal definition of data, which is independent from the apparent the variety of data format, making it possible to construct datasets in an universal manner.

7.2 Perspective 2: Task

Construction of datasets of traditional NLP tasks such as syntactic parsing (Marcus et al., 1993; Xue et al., 2005) or machine translation (Nakazawa et al., 2016) is undoubtedly meaningful, yet inadequate for real business, where special requirements are indispensable for making profits.

The proposed method allows user-defined tasks, making it possible to customize the dataset based on requirements of users themselves.

7.3 Perspective 3: Quality

Low-quality samples have to be removed from constructed datasets. Approaches generally resort to rules (Zheng et al., 2023; Ding et al., 2024; Guo et al., 2024), external feedback (Kim et al., 2023a; Dong et al., 2023; Wang et al., 2024c), or model scores (Wang et al., 2023b; Pace et al., 2024). These resources are resource-intensive (e.g., feedback from domain experts) and sometimes inaccessible (e.g., possibility distributions in closed-source LLMs).

The proposed method ensures the quality of constructed datasets by scoring with LLMs, which is highly aligned to human assessments as shown by experiment results.

7.4 Perspective 4: Cost

Cost of dataset construction generally originated from the participation of human experts, whose devotion can be either full (Xue et al., 2005; Nakazawa et al., 2016) or partial (Huang et al., 2024a,b).

Without loss of quality, the proposed method minimized the intervention of human experts, so that most of the cost originated from API calls of LLMs, which is generally much cheaper.

7.5 Perspective 5: Theory

Theory foundations of previous researches generally focus on application of probability theory on designing models (Wang et al., 2024a; Zhang et al., 2024) or statistics on analyzing experiment results (Yu et al., 2023). In contrast, we first formalized the process of dataset construction, built a solid mathematical theory based on algebra.

8 Conclusion

We presented the first formalization of the process of dataset construction, based on which we proposed a novel paradigm of fully-automated dataset construction, which succeed to construct high-quality datasets fully automatically given minimal information. By utilizing the automatically constructed datasets, performance on different tasks improved significantly, indicating the effectiveness of the proposed method. Compared with previous researches on dataset construction, the proposed method has advantages in many aspects.

Limitations

The quality of constructed datasets depends on the capability of LLMs. Throughout this paper, we conduct experiments using GPT-4o. We have not conducted experiments using other advanced LLMs

such as Llama-2, etc.

The proposed method involved nine types of LLMs, and the number of LLMs may be larger than ten, depending on the number of external programs. To construct a large-scale dataset, the expense may be large.

For each task, there are at most four external programs. However, in real applications, it is possible that there are hundreds of external programs. We do not know the effectiveness of the proposed method in such complicated scenarios.

Although quantitative assessment by human guaranteed the quality of constructed dataset, it is unknown whether the constructed dataset is helpful for downstream tasks. More experiments on the utilization of constructed datasets is needed.

References

Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1204–1207, New York, NY, USA. Association for Computing Machinery.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Julia Bonn, Harish Tayyar Madabushi, Jena D. Hwang, and Claire Bonial. 2024. Adjudicating LLMs as PropBank adjudicators. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 112–123, Torino, Italia. ELRA and ICCL.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17772–17780.

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*.

Linyi Ding, Sizhe Zhou, Jinfeng Xiao, and Jiawei Han. 2024. Automated construction of theme-specific knowledge graphs. *arXiv preprint arXiv:2404.19146*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

Yunlong Feng, Yang Xu, Libo Qin, Yasheng Wang, and Wanxiang Che. 2024. Improving language model reasoning with self-motivated learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8840–8852, Torino, Italia. ELRA and ICCL.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Hongyi Guo, Yuanshun Yao, Wei Shen, Jiaheng Wei, Xiaoying Zhang, Zhaoran Wang, and Yang Liu. 2024. Human-instruction-free llm self-alignment with limited samples. *arXiv preprint arXiv:2401.06785*.

Himanshu Gupta, Kevin Scaria, Ujjwala Anantheswaran, Shreyas Verma, Mihir Parmar, Saurabh Arjun Sawant, Chitta Baral, and Swaroop Mishra. 2024. TarGEN: Targeted data generation with large language models. In *First Conference on Language Modeling*.

Yuta Hayashibe. 2020. Japanese realistic textual entailment corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6827–6834, Marseille, France. European Language Resources Association.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruifeng Xu, and Qun Liu. 2024a. Planning, creation, usage: Benchmarking LLMs for comprehensive tool utilization in real-world complex scenarios. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4363–4400.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2024b. Metatool benchmark for large language models: Deciding whether to use tools and which to use. In *The Twelfth International Conference on Learning Representations*.

Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–30.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023a. SODA: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12930–12949, Singapore. Association for Computational Linguistics.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023b. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13677–13700, Singapore. Association for Computational Linguistics.

Gilbert W King and Hsien-Wu Chang. 1963. Machine translation of chinese. *Scientific American*, 208(6):124–137.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, hai zhao, and Pengfei Liu. 2024. Generative judge for evaluating alignment. In *The Twelfth International Conference on Learning Representations*.

Hao Liang, Linzhuang Sun, Jingxuan Wei, Xijie Huang, Linkun Sun, Bihui Yu, Conghui He, and Wentao Zhang. 2024. Synth-empathy: Towards high-quality synthetic empathy data. *arXiv preprint arXiv:2407.21669*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023a. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, pages 21558–21572. Curran Associates, Inc.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023b. Trustworthy LLMs: a survey and guideline for evaluating large language models' alignment. In *Socially Responsible Language Modelling Research*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Margherita Martorana, Tobias Kuhn, Lise Stork, and Jacco van Ossenbruggen. 2024. Zero-shot topic classification of column headers: Leveraging llms for metadata enrichment. In *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI*, pages 52–66. IOS Press.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.

Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2024. West-of-n: Synthetic preference generation for improved reward modeling. In *ICLR 2024 Workshop on Navigating*

and Addressing Data Problems for Foundation Models.

Farica Perdana Putri, Hira Meidia, and Dennis Gunawan. 2019. Designing intelligent personalized chatbot for hotel services. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 468–472.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Xuanhe Zhou, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Guoliang Li, Zhiyuan Liu, and Maosong Sun. 2024. Tool learning with foundation models. *ACM Comput. Surv.*, 57(4).

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. Pmg: Personalized multimodal generation with large language models. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3833–3843, New York, NY, USA. Association for Computing Machinery.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. 2024. SALMON: Self-alignment with instructable reward models. In *The Twelfth International Conference on Learning Representations*.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Ramya Tekumalla and Juan M Banda. 2023. Leveraging large language models and weak supervision for social media data annotation: an evaluation using covid-19 self-reported vaccination tweets. In *International Conference on Human-Computer Interaction*, pages 356–366. Springer.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *arXiv preprint arXiv:2407.13690*.

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

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Haoyu Wang, Guozheng Ma, Ziqiao Meng, Zeyu Qin, Li Shen, Zhong Zhang, Bingzhe Wu, Liu Liu, Yatao Bian, Tingyang Xu, et al. 2024a. Step-on-feet tuning: Scaling self-alignment of llms via bootstrapping. *arXiv preprint arXiv:2402.07610*.

Ruida Wang, Wangchunshu Zhou, and Mrinmaya Sachan. 2023a. Let's synthesize step by step: Iterative dataset synthesis with large language models by extrapolating errors from small models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11817–11831, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024b. What are tools anyway? a survey from the language model perspective. In *First Conference on Language Modeling*.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024c. CodecLM: Aligning language models with tailored synthetic data. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3712–3729, Mexico City, Mexico. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36:55734–55784.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1965, Bangkok, Thailand. Association for Computational Linguistics.

Chujie Zheng, Sahand Sabour, Jiaxin Wen, Zheng Zhang, and Minlie Huang. 2023. AugESC: Dialogue augmentation with large language models for emotional support conversation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1552–1568, Toronto, Canada. Association for Computational Linguistics.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Formalization of Dataset Construction: A Detailed Discussion

In this section, we give more information on the formalization of dataset construction, including detailed explanation, more examples, and proofs of all theorems.

A.1 Discussion on Dataset

In this paper, we distinguish the concepts of *data* and *dataset*. Intuitively, data consists of *any* possible sequences of tokens of the task. As long as the *format* of the sequence matches the requirement of the task, the sequence belongs to the data.

Given the definition of data, we can prove Theorem 1.

Theorem 1. *Any data $\mathcal{D}$ is topological space, and is Hausdorff.*

Proof. Select an arbitrary sequence $s = (t_1, t_2, \cdots) \in \mathcal{D}$. Define a series of sets $N_d(s)$ as below.

$$N_d(s) = \{s_d = (t_1^d, t_2^d, \cdots) \mid t_i{=}t_i^d \text{ for all } i \text{ except for at most } d \text{ elements.}\}.$$

Obviously,

$$\begin{aligned}
&\forall d \geq 0, s \in N_d(s), \\
&\forall d' \leq d, N_{d'}(s) \subseteq N_d(s), \\
&\forall d_1 \geq 0, d_2 \geq 0, \\
&\qquad N_{d_1}(s) \cap N_{d_2}(s) = N_{\min(d_1,d_2)}(s).
\end{aligned}$$

Furthermore, when $d' \leq d$, we select an arbitrary sequence

$$s_{d'} = (t_1^{d'}, t_2^{d'}, \cdots) \in N_{d'}(s).$$

According to the definition of $N_{d'}(s)$, there are at most d' elements that $t_i^{d'} \neq t_i$. Let the indices of these elements be $i_1, i_2, \cdots, i_{d'}$, then

$$\begin{aligned}
s_{d'} =&(t_1, \cdots, t_{i_1-1}, t_{i_1}^{d'}, t_{i_1+1}, \cdots, \\
&t_{i_2-1}, t_{i_2}^{d'}, t_{i_2+1}, \cdots, \\
&t_{i_{d'}-1}, t_{i_{d'}}^{d'}, t_{i_{d'}+1}, \cdots).
\end{aligned}$$

Now we select a sequence $s_d = (t_1^d, t_2^d, \cdots) \in N_d(s)$, such that

$$\begin{aligned}
&\forall i \in \{i_1, i_2, \cdots, i_{d'}\}, t_i^d = t_i^{d'}, \\
&\forall i \notin \{i_1, i_2, \cdots, i_{d'}\}, t_i^d = t_i, \\
&\qquad \text{except for at most } d - d' \text{ elements.}
\end{aligned}$$

This means that $s_d \in N_{d-d'}(s_{d'})$.

Let $\mathcal{N}(s) = \{N_0(s), N_1(s), \cdots\}$, from the discussions above, we can conclude that $\mathcal{N}$ is a neighborhood topology, and hence $\mathcal{D}$ is a topological space.

Now, we arbitrarily select two different sequence $s = (t_1, t_2, \cdots) \in \mathcal{D}$ and $s' = (t_1', t_2', \cdots) \in \mathcal{D}$, and assume that there are exactly m different elements.

When $m = 1$, $N_0(s) \cap N_0(s') = \emptyset$. When $m > 1$, it is obvious that $N_1(s) \cap N_1(s') = \emptyset$. Hence, $\mathcal{D}$ is Hausdorff. □

Because the subspace of a Hausdorff space is also Hausdorff, the *dataset* $\mathfrak{D}$, which is a finite subspace of data $\mathcal{D}$, is also a Hausdorff topological space, with the same neighborhood topology given above.

Data $\mathcal{D}$ consists of all sequences whose *format* match the task. On the other hand, dataset $\mathfrak{D}$ consists of all sequences that *semantically* match the task. Intuitively, when human experts construct datasets in NLP, they definitely understand the semantic meanings of sentences in the dataset. Human experts select or write finite number of sentences (or passages, question-answer pairs, parallel

sentence pairs, etc.) following the requirements of the given task.

Example 3. For English-French machine translation, the format of data should be a sentence pair, with a special token to separate two sentences. Therefore,

$$(Good, morning) \notin \mathcal{D},$$

because there is only on sentence.

$$(Goodbye, \texttt{[SEP]}, Bonjour) \in \mathcal{D}$$

because the data format matches the task of English-French machine translation. However, the English sentence "*Goodbye*" and the French sentence "*Bonjour*" are not equivalent semantically, therefore

$$(Goodbye, \texttt{[SEP]}, Bonjour) \notin \mathfrak{D},$$

yet

$$(Good, morning, \texttt{[SEP]}, Bonjour) \in \mathfrak{D}.$$

Now, we prove a theorem which states an important property of data $\mathcal{D}$.

Theorem 2. *$(\mathcal{D}, \cap)$ is a monoid, where $\cap$ is the intersection of subspaces.*

Proof. Because data $\mathcal{D}$ is a topological space, obviously for all of its subspaces, the closure property holds true for the intersection operation.

Select three datasets $\mathfrak{D}_1 \subseteq \mathcal{D}, \mathfrak{D}_2 \subseteq \mathcal{D}, \mathfrak{D}_3 \subseteq \mathcal{D}$ arbitrarily, it is obvious that

$$(\mathfrak{D}_1 \cap \mathfrak{D}_2) \cap \mathfrak{D}_3 = \mathfrak{D}_1 \cap (\mathfrak{D}_2 \cap \mathfrak{D}_3),$$

so the associativity property holds true.

Last, for any $\mathfrak{D} \subseteq \mathcal{D}$, we can see that

$$\mathfrak{D} \cap \mathcal{D} = \mathcal{D} \cap \mathfrak{D} = \mathfrak{D},$$

indicating that $\mathcal{D}$ is the identity element. Hence, $(\mathcal{D}, \cap)$ is a monoid. □

This property is used for proving the isomorphism relation between datasets and common word sequences (i.e., Theorem 7).

Last, the concept of *refinement* of dataset is introduced. Intuitively, when human experts construct a dataset, the core task is to select (or create) samples that satisfy the requirements. The requirements may be composed of multiple dimensions. For example, to construct a dataset for training a machine translation system focused on English-French sci-tech papers, the sentence pair $(e, f) \in \mathfrak{D}$ must satisfy the following conditions simultaneously at least.

- e is an English sentence.
- e is sampled from a paper.
- The paper that e is sampled is a paper in the sci-tech domain.
- f is a French sentence.
- f and e are equivalent semantically.
- Expression of f should not be colloquial.

For tasks in real business scenario, the number of conditions grows rapidly. Therefore, to construct a dataset satisfying all conditions, a practical approach is to refine the dataset step by step: filtering out (or refusing to create) examples that do not satisfy the first condition, then filtering out (or refusing to create) examples that do not satisfy the second condition, and so on. The formalization of this intuition is exactly the concept of refinement.

A refinement corresponds to a series of refinement mappings. Apparently, a *fake* refinement seems to be meaningless. However, it is indispensable both in theory and in practice.

In theory, the equivalence of dataset refinement and signed word sequence expansion is proved by Theorem 8, no information on datasets is provided. Allowing $\mathfrak{D}_{i+1} = \mathfrak{D}_i$ makes the proof more concise.

In practice, operations for dataset construction do not always correspond to real refinements, even for human experts. Generally, it is necessary to adopt intermediate steps such as data transformation, disambiguation, decision based on intermediate results, etc. These types of intermediate steps correspond to fake refinements.

Theorem 3. *For any dataset $\mathfrak{D}$, there is a data $\mathcal{D}$, and a refinement, such that $\mathfrak{D} \subseteq \cdots \subseteq \mathcal{D}$.*

Proof. Arbitrarily select a sequence $s = (t_1, t_2, \cdots) \in \mathfrak{D}$, and arbitrarily select an index i, then change t_i to any other token $t_i' \neq t_i$. This results in a new sequence

$$s' = (t_1, t_2, \cdots, t_{i-1}, t_i', t_{i+1}, \cdots).$$

If $s' \notin \mathfrak{D}$, we can define a new dataset

$$\mathfrak{D}' = \mathfrak{D} \cup \{s'\} \supset \mathfrak{D},$$

otherwise, we can define

$$\mathfrak{D}' = \mathfrak{D}.$$

In both cases, we have refinement

$$\mathfrak{D} \subseteq \mathfrak{D}'.$$

Repeat this process n times, we obtain the refinement

$$\mathfrak{D} \subseteq \mathfrak{D}' \subseteq \cdots \mathfrak{D}^{(n)}.$$

Let $\mathcal{D} = \mathfrak{D}^{(n)}$, we obtain the required refinement. □

This apparently trivial theorem is actually important. The focus of this research is on the existence of versatile approach. This theorem guarantees the existence of versatile approach in theory, and experiment results on three tasks (5) demonstrates the existence of versatile approach empirically.

A.2 Discussion on Signed Word Sequence

Intuitively, a signed word sequence can be regarded as a sentence consisted of words that *must* be presented and those that *must not* be presented. In the context of dataset construction, signed word sequences can be regarded as descriptions of datasets. For example, $(+A, +Chinese, +treebank)$ corresponds to a dataset consisted of constituent trees written in Chinese, while $(0, -Chinese, -treebank)$ corresponds to a dataset that does not containing any Chinese characters, and does not containing any constituent trees.

Furthermore, the fusion of signed word sequences can be regarded as the modification to dataset description, intuitively.

Example 4. The fusion of

$$W = (+An, +English, +sentence)$$
$$V = (0, 0, 0, +with, +ten, +words)$$

is

$$W \star V = (+An, +English, +sentence, +with, +ten, +words).$$

This can be regarded as the operation of adding more constraints to the dataset. Initially, the dataset consists of all English sentences. After the fusion operation, all sentences whose length are not ten words are filtered out, resulting in a dataset consisted of all English sentences with ten words.

Example 5. The fusion of

$$W = (+A, +Chinese, +treebank),$$
$$V = (0, -Chinese, -treebank),$$
$$U = (0, +French, +document)$$

is

$$W \star V \star U = (+A, +French, +sentence).$$

This can be regarded as two modifications to the dataset. Initially, the dataset is a treebank, and the language is Chinese. After the fusion operation, the dataset becomes a collection of documents, and the language is changed to French.

Given the definition of signed word sequence and fusion operation, we can prove the following theorem.

Theorem 4. *$(W, \star)$ is a abelian group, where W is a signed word sequence.*

Proof. According to the definition of the fusion operation, for any two signed word sequences W and V, their fusion $W \star V$ is also a signed word sequence. Therefore, the closure property holds true.

For any three signed word sequences

$$U = (\mathrm{sgn}(u_1) \cdot u_1, \cdots, \mathrm{sgn}(u_l) \cdot u_l),$$
$$V = (\mathrm{sgn}(v_1) \cdot v_1, \cdots, \mathrm{sgn}(v_m) \cdot v_m),$$
$$W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n),$$

that satisfy the conditions for fusion. We can see that

$$(U \star V) \star W = U \star (V \star W) = (x_1, \cdots, x_{\max\{l,m,n\}}),$$

where

$$x_i = \begin{cases} 0 & \text{if } \mathrm{sgn}(u_i) + \mathrm{sgn}(v_i) + \mathrm{sgn}(w_i) = 0, \\ \mathrm{sgn}(u_i) \cdot u_i & \text{if } \mathrm{sgn}(v_i) + \mathrm{sgn}(w_i) = 0, \\ \mathrm{sgn}(v_i) \cdot v_i & \text{if } \mathrm{sgn}(u_i) + \mathrm{sgn}(w_i) = 0, \\ \mathrm{sgn}(w_i) \cdot w_i & \text{otherwise.} \end{cases}$$

Therefore, the associativity property holds true.

For any signed word sequence W, we can see that

$$W \star \mathfrak{o} = \mathfrak{o} \star W = W,$$

so $\mathfrak{o}$ is the identity element.

Last, for any signed word sequence $W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$, we can define the following signed word sequence

$$W^{-1} = ((-1) \cdot \mathrm{sgn}(w_1) \cdot w_1, \cdots, (-1) \cdot \mathrm{sgn}(w_n) \cdot w_n).$$

Obviously, $W \star W^{-1} = \mathfrak{o}$, indicating W^{-1} is the inverse of W. Hence, $(W, \star)$ is a group.

Now, select two signed word sequences $V = (\mathrm{sgn}(v_1) \cdot v_1, \cdots, \mathrm{sgn}(v_m) \cdot v_m)$ and $W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$ arbitrarily. We have

$$V \star W = W \star V = (u_1, u_2, \cdots, u_{\max(m,n)}),$$

where u_i is defined in Definition 6. Hence, $(W, \star)$ is an abelian group. □

Certainly, $(W, \star)$ is also a monoid.

Subsequence of signed word sequence is an extension to the concept of subsequence in the sense of string. For example, given a signed word sequence

$$W = (+A, +Chinese, +sentence),$$

the collections of its subsequence is

$$\begin{aligned}
&\{\\
&\mathfrak{o}, (+A, 0, 0),\\
&(0, +Chinese, 0), (0, 0, +sentence),\\
&(+A, +Chinese, 0), (+A, 0, +sentence),\\
&(0, +Chinese, +sentence),\\
&(+A, +Chinese, +sentence)\\
&\},
\end{aligned}$$

which is an analogue to the concept of subset.

The following theorem is easy to prove.

Theorem 5. *The relation of subsequence is a partial order.*

Proof. Obviously, any signed word sequence is the subsequence of itself. Therefore the reflexivity property holds true.

Arbitrarily select two signed word sequences W and V. Without loss of generality, we assume the length of W is no longer than the length of V. Therefore

$$W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_m) \cdot w_m, \underbrace{0, \cdots, 0}_{n-m \text{ zeros in total}}),$$

and

$$V = (\mathrm{sgn}(v_1) \cdot v_1, \cdots, \mathrm{sgn}(v_n) \cdot v_n).$$

Now assume that $W \lhd V$ and $W \rhd V$. The first condition indicates that

$$\forall i \in [1, n], \mathrm{sgn}(w_i) = 0 \vee (\mathrm{sgn}(w_i) = \mathrm{sgn}(v_i) \wedge w_i = v_i),$$

while the second condition indicates that

$$\forall i \in [1, n], \mathrm{sgn}(v_i) = 0 \vee (\mathrm{sgn}(w_i) = \mathrm{sgn}(v_i) \wedge w_i = v_i).$$

Therefore,

$$\forall i \in [1, n], \mathrm{sgn}(w_i) \cdot w_i = \mathrm{sgn}(v_i) \cdot v_i,$$

which means that $W = V$. Hence, the antisymmetry property holds true.

Arbitrarily select three signed word sequences

$$\begin{aligned}
U &= (\mathrm{sgn}(u_1) \cdot u_1, \cdots, \mathrm{sgn}(u_l) \cdot u_l),\\
V &= (\mathrm{sgn}(v_1) \cdot v_1, \cdots, \mathrm{sgn}(v_m) \cdot v_m),\\
W &= (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n),
\end{aligned}$$

such that $U \lhd V$ and $V \lhd W$. Then, we have

$$\begin{aligned}
&\forall i \in [1, n], \mathrm{sgn}(u_i) = 0\\
&\qquad \vee (\mathrm{sgn}(u_i) = \mathrm{sgn}(v_i) \wedge u_i = v_i),\\
&\forall i \in [1, n], \mathrm{sgn}(v_i) = 0\\
&\qquad \vee (\mathrm{sgn}(w_i) = \mathrm{sgn}(v_i) \wedge w_i = v_i),
\end{aligned}$$

indicating that

$$\begin{aligned}
&\forall i \in [1, n], \mathrm{sgn}(u_i) = 0\\
&\qquad \vee (\mathrm{sgn}(w_i) = \mathrm{sgn}(u_i) \wedge w_i = u_i).
\end{aligned}$$

This means that $U \lhd W$. Therefore, the transitivity property holds true. Hence, the relation of subsequence is a partial order. □

This theorem about partial order is necessary to define *expansion*.

A.3 Discussion on Dialogue

One may wonder on what situations we can encounter signed word sequences defined above. Actually, signed word sequences are not real objects that we encountered in common tasks or corpus. Words, sentences, or passages in real NLP tasks do not have signs.

However, we can build the linkage between signed word sequences and dialogues. Dialogues become one of the most popular object in NLP thanks to the progress of LLMs. Definition 9 is the formalization of our intuitions.

The linkage between signed word sequences and dialogues is built using the following theorem.

Theorem 6. *The fusion of any numbers of signed word sequences can be approximated by a dialogue.*

Proof. We know that the fusion of any two signed word sequences is also a signed word sequence. Therefore, as long as we can approximate *one*

signed word sequence by a dialogue, we can trivially approximate the fusion of *any* numbers of signed word sequences.

Assume that there is only one signed word sequence $W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$. We can approximate in the following manner.

- Step 1: remove all elements in W whose sign is 0.
- Step 2: group all elements with the same sign. Formally, W is transformed to the following format.

$$W = (\mathrm{sgn}(g_1) \cdot (w_1^1, w_1^2, \cdots), \cdots, \mathrm{sgn}(g_i) \cdot (w_i^1, w_i^2, \cdots)),$$

  where $\mathrm{sgn}(g_1) \in \{+1, -1\}$ and $\forall i \geq 1, \mathrm{sgn}(g_i) \cdot \mathrm{sgn}(g_{i+1}) = -1$.
- Step 3: Paraphrase word sequences.
  - If $\mathrm{sgn}(g_i) = +1$, generate the paraphrase of word sequence $w_i^1, w_i^2, \cdots$, which is utterance u_i.
  - If $\mathrm{sgn}(g_i) = -1$, generate the negation of word sequence $w_i^1, w_i^2, \cdots$, which is utterance u_i.
- Step 4: assign all $\mathfrak{s}_i$ with the same value.

In this way, we succeed to construct all utterances in the dialogue. Hence the theorem holds true. □

At Step 4 of the proof above, we assign all $\mathfrak{s}_i$ with the same value. This means that there is only one speaker in the dialogue. A dialogue like this correspond to the case that all refinements of the dataset are real refinements. However, in practice, it is difficult to make all refinements real. Fake refinements are generally needed. Hence, there are at least *two* speakers in dialogues of real scenarios.

A.4 Discussion on Galois Equivalence

We are ready to build connections of the concepts that we have defined. The most important one is the equivalence of dataset and signed word sequence, which is formally represented as the following theorem.

Theorem 7. *$(\mathcal{D}, \cap)$ and $(W, \star)$ are isomorphic.*

Proof. We can construct mapping $f : \mathcal{D} \to W$ as below.

When the cardinality of $\mathcal{D}$ is infinite, we define

$$f(\mathcal{D}) = \mathfrak{o}.$$

Now we assume that the cardinality of $\mathcal{D}$ is finite. Arbitrarily select a subspace $\mathfrak{D} \subseteq \mathcal{D}$. We define

$$f(\mathfrak{D}) = \prod_{s \in \mathfrak{D}} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus \mathfrak{D}} F(s, -1),$$

where the $\prod$ operation is the $\star$ operation on signed word sequences. F is a mapping from a sequence of tokens to a signed word sequence. Let $s = (t_1, t_2, \cdots, t_m)$, and $\mathrm{sgn} \in \{+1, -1\}$, then mapping g is defined as

$$F(s, \mathrm{sgn}) = (\mathrm{sgn} \cdot t_1, \mathrm{sgn} \cdot t_2, \cdots, \mathrm{sgn} \cdot t_m).$$

First, we prove $f : \mathcal{D} \to W$ is a homomorphism. Select two datasets $\mathfrak{D}_1 \subseteq \mathcal{D}$ and $\mathfrak{D}_2 \subseteq \mathcal{D}$ arbitrarily. We can see that

$$f(\mathfrak{D}_1 \cap \mathfrak{D}_2) = \prod_{s \in \mathfrak{D}_1 \cap \mathfrak{D}_2} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus (\mathfrak{D}_1 \cap \mathfrak{D}_2)} F(s, -1),$$

and

$$f(\mathfrak{D}_1) \star f(\mathfrak{D}_2) = \prod_{s \in \mathfrak{D}_1} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus \mathfrak{D}_1} F(s, -1) \star \prod_{s \in \mathfrak{D}_2} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus \mathfrak{D}_2} F(s, -1).$$

Because $(W, \star)$ is abelian, the commutativity property holds true. Therefore,

$$\begin{aligned} f(\mathfrak{D}_1) \star f(\mathfrak{D}_2) &= \prod_{s \in \mathfrak{D}_1} F(s, +1) \star \prod_{s \in \mathfrak{D}_2} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus \mathfrak{D}_1} F(s, -1) \star \prod_{s \in \mathcal{D} \setminus \mathfrak{D}_2} F(s, -1) \\ &= \prod_{s \in \mathfrak{D}_1 \cap \mathfrak{D}_2} F(s, +1) \star \prod_{s \in \mathcal{D} \setminus (\mathfrak{D}_1 \cap \mathfrak{D}_2)} F(s, -1) \\ &= f(\mathfrak{D}_1 \cap \mathfrak{D}_2). \end{aligned}$$

Hence, $f : \mathcal{D} \to W$ is a homomorphism.

Next, we prove f is invertible. We construct mapping $h : W \to \mathcal{D}$ as below. First, the empty sequence $\mathfrak{o}$ is mapped to $\mathcal{D}$, i.e.,

$$h(\mathfrak{o}) = \mathcal{D}.$$

Next, select a non-empty signed word sequence $W = (\mathrm{sgn}(w_1) \cdot w_1, \cdots, \mathrm{sgn}(w_n) \cdot w_n)$. We group

elements in the sequence based on signs, so we can rewrite W as below.

$$W = (\mathrm{sgn}(g_1) \cdot (w_1^1, w_1^2, \cdots), \cdots, \\ \mathrm{sgn}(g_i) \cdot (w_i^1, w_i^2, \cdots))$$

Let

$$\mathfrak{D} = \{(w_i^1, w_i^2, \cdots) \mid \mathrm{sgn}(g_i) = +1\},$$

and define

$$h(W) = \mathfrak{D}.$$

We can see that

$$g \circ f = f \circ g = \mathrm{id}\,.$$

Therefore, h is the inverse mapping of f. Hence, $f : \mathcal{D} \to W$ is an isomorphism, and $(\mathcal{D}, \cap)$ and $(W, \star)$ are isomorphic. □

We can extend the theorem above to a series of datasets and a series of signed word sequences. Recall that a refinement of dataset $\mathfrak{D}$ is a tower

$$\mathfrak{D} \supseteq \mathfrak{D}_1 \supseteq \cdots \supseteq \mathfrak{D}_{n-1} \supseteq \mathfrak{D}_n\,,$$

and an expansion of a signed word sequence W is a tower

$$W \triangleleft W_1 \triangleleft \cdots \triangleleft W_{n-1} \triangleleft W_n\,.$$

By observing these two towers, we conjecture that there is a one-to-one correspondence between them, which is exactly the following theorem.

Theorem 8 (Galois Theorem on Dataset)**.** *For any dataset $\mathfrak{D}$ and signed word sequence W, there is a bijection between the refinement of $\mathfrak{D}$ and the expansion of W.*

Proof. Let mapping $f : \mathfrak{D} \to W$ be the isomorphism defined above. For an arbitrary dataset $\mathfrak{D}_1 \subseteq \mathfrak{D}$, we define the following signed word sequence

$$W_1 = W \star \prod_{s \in \mathfrak{D}_1} F(s, +1) \star \prod_{s \in \mathfrak{D} \setminus \mathfrak{D}_1} F(s, -1),$$

where F is also defined as above. It is obvious that $f(\mathfrak{D}_1) = W_1$. Repeat this process, we can construct two towers:

$$\mathfrak{D} = \mathfrak{D}_0 \supseteq \mathfrak{D}_1 \supseteq \cdots \supseteq \mathfrak{D}_{n-1} \supseteq \mathfrak{D}_n\,,$$
$$W = W_0 \triangleleft W_1 \triangleleft \cdots \triangleleft W_{n-1} \triangleleft W_n\,,$$

where for each $i \in [0, n]$, $f(\mathfrak{D}_i) = W_i$. This is the bijection that we are finding. □

Corollary 1 is simply a special case of this theorem, yet is important to dataset construction, which is discussed at the next subsection.

As an aside, we named Theorem 8 as *Galois Theorem on Dataset*. Galois theory is one of the most important theories in algebra. One of the main results is the following theorem.

Theorem (The Fundamental Theorem of Galois Theory)**.** *Let L/K be a finite Galois extension. Then there is an inclusion reversing bijection between the subgroups of the Galois group $\mathrm{Gal}(L/K)$ and intermediary subfields $L/M/K$.*

The apparent similarity to the theorem above results in the name of Theorem 8.

A.5 Discussion on Dataset Construction

After giving definitions of many terminologies and proving many theorem, we arrived at Corollary 2, which is the milestone of the discussion. This corollary is important, because it is the foundation of the proposed method. To achieve fully-automated dataset construction, the proposed method approximate the fusion of signed word sequence by dialogue between two LLMs, corresponding to two types of refinement mappings.

Example 6. To construct a English-French machine translation dataset $\mathfrak{D} \subseteq \mathcal{D}$, a feasible procedure is as below.

1. Ask an American to write an English sentence.
2. Ask a French native speaker to translate it to French.
3. ask the manager to decide whether to adopt the translation.

This procedure corresponds to a refinement

$$\mathfrak{D} = \mathfrak{D}_4 \subseteq \mathfrak{D}_3 \subseteq \mathfrak{D}_2 \subseteq \mathfrak{D}_1 = \mathcal{D}.$$

At each step, human workers received some instructions, based on which they do some operations on the dataset, which corresponds to real refinement mappings. Sometimes, clarifications of instructions or memorization of intermediate results are necessary, which corresponds to fake mappings.

The conversations at each step correspond to utterances in the dialogue. Specifically,

- $f_1 : \mathfrak{D}_1 \to \mathfrak{D}_2$ filters out all examples that did not start with the given English sentence.
- $f_2 : \mathfrak{D}_2 \to \mathfrak{D}_3$ filters out all examples that did not end with the given French translation.
- $f_3 : \mathfrak{D}_3 \to \mathfrak{D}_4$ decides whether this translation should be located in $\mathfrak{D}$ or $\mathcal{D} \setminus \mathfrak{D}$.

B Necessity of Components in Proposed Method: Ablation Studies

The formalization of dataset construction process shows the necessity of components in the proposed method theoretically. In this section, we demonstrate that these components are also necessary practically.

B.1 Importance of dialog-based construction

Table 5 shows the first several utterances of the dialogue for constructing the dataset of mathematical problem solver. The following points should be noticed.

Continuous operation of dataset refinement. At turn 2, 10 categories of problems are proposed, and at turn 3, 4 categories are selected. This corresponds to a real refinement $f_2 : \mathfrak{D}_2 \to \mathfrak{D}_3$, where $\mathfrak{D}_3 \subsetneq \mathcal{D}_2$.

Necessity of guidance LLM. At turn 6, after many turns of the operation utterances, the guidance LLM checks the dialog history, and gives new instructions. This corresponds to a fake refinement $f_6 : \mathfrak{D}_6 \to \mathfrak{D}_7$, where $\mathfrak{D}_6 = \mathfrak{D}_7$.

Invoke of external programs. Given the program of generating system of equations, the operation LLM invoked it at the appropriate timing. The program imitated by LLM without generating real codes, returns equations correctly.

B.2 Influence of Minimal Information

Recall that the minimal information from human consists of the information on (1) task description, and (2) data format.

Influence of task description. Table 6 shows the constructed datasets given different task information, for the task of mathematical problem solver. A vague description of task results in a generic dataset (Row 1), while more useful datasets can be constructed by adding specific information in the task description, such as difficulty (Row 2 and 3), domain (Row 4 and 5), etc.

Influence of data format. Table 7 shows the influence of data format. The quality of the constructed dataset seems not to be affected by data format.

B.3 Discussion on Programs

There are three types of programs in the external layer. The program written by human can be ex-

Awareness of program invoking: 9 Choice of program: 9 Arguments of program: 8 Guidance utterance: 9 Data utterance: 8
[TURN 1] Guidance: To effectively construct a dataset for mathematical problem-solving using systems of equations, let's start by identifying different types of problems that typically require systems of equations. This will help us ensure that our dataset covers a comprehensive range of scenarios. Some examples might include word problems involving mixtures, rate problems with distances over time, financial scenarios with investments, and problems dealing with geometry. Could we brainstorm a list of such problem types, and then, perhaps, select a few to focus on as starting points for our dataset? Additionally, what level of difficulty are we aiming for in these problems—basic, intermediate, or advanced? This will help tailor the problems to the appropriate complexity. **[TURN 2] Operation:** Certainly, here's a list of different types of problems that typically require systems of equations: 1. Mixture Problems: Involving the mixing of different substances with varying concentrations or prices. 2. Rate Problems: Addressing distances, speeds, and time. 3. Financial Problems: Including interest rates, investments, and profit calculations. 4. Geometry Problems: Dealing with dimensions, shapes, and configurations. 5. Work Problems: Involves multiple parties working together or separately at different rates. 6. Age Problems: Solving for ages based on given relationships and time differences. 7. Number Problems: Involving relationships between numbers, such as consecutive integers. 8. Production Problems: Related to manufacturing, supply, and demand. 9. Transportation Problems: Optimizing routes, costs, and logistics. 10. Chemical Solutions: Concentration and mixing of chemical solutions. For the complexity level, it would be beneficial to focus on intermediate to advanced problems to challenge the problem-solving capabilities of systems of equations. What are your thoughts on selecting some of these types and levels for our initial dataset? **[TURN 3] Operation:** To start constructing the dataset, let's select a few problem types to focus on: Mixture Problems, Rate Problems, Financial Problems, and Geometry Problems. We'll aim for an intermediate to advanced difficulty level for these problems. Please provide a mathematical problem for each of these categories, and we'll proceed to generate systems of equations for them. Here's a set of problems across the selected categories: 1. **Mixture Problem**: A chemist has a 30% saline solution and a 70% saline solution. How many liters of each should be mixed to obtain 10 liters of a 50% saline solution? 2. **Rate Problem**: A car travels 150 miles in the same amount of time a bicycle travels 30 miles. If the speed of the car is 20 mph faster than twice the speed of the bicycle, what are the speeds of the car and the bicycle? 3. **Financial Problem**: Sarah invests $10,000 in two parts: one part at 5% simple interest and the other at 8%. At the end of the year, the total interest earned is $700. How much did she invest at each rate? 4. **Geometry Problem**: The perimeter of a rectangle is 60 meters. If the length is 5 meters more than twice the width, what are the dimensions of the rectangle? Let's proceed with generating a system of equations for each problem. **[TURN 4] Operation:** `generate_system_of_equations(math_problem="A chemist has a 30% saline solution and a 70% saline solution. How many liters of each should be mixed to obtain 10 liters of a 50% saline solution?")` **[TURN 5] Program:** `equation_list:`$[x + y = 10, 0.3x + 0.7y = 0.5 * 10]$ **[TURN 6] Guidance:** Here is the system of equations for the Mixture Problem: 1. $x + y = 10$, where x is the liters of 30% saline solution, and y is the liters of 70% saline solution 2. $0.3x + 0.7y = 5$, since 50% of 10 liters should be saline Now, let's create systems of equations for the other three problem categories. If you'd like to try creating them or need assistance, please let me know! **[TURN 7] Operation:** `generate_system_of_equations(...)` .....

Table 5: First several utterances of the dialog for constructing the dataset of mathematical problem solver. Scores are given by the LLM for dialogue checking.

Task Description	Examples in Constructed Dataset
Mathematical problems solved by using systems of equations.	Problem: Find the values of x and y that satisfy the following conditions: The sum of twice x and y is 10, and the difference between x and twice y is 3. Equations: $2x + y = 10, x - 2y = 3$ Answer: $x = 4.6, y = 0.8$
Easy mathematical problems that primary school students can solve using equations.	Problem: Emma has 5 boxes of cupcakes. Each box contains 4 cupcakes. How many cupcakes does she have in total? Equations: Given: $x = 5$, Given: $y = 4$, Total cupcakes = $5 * 4$ Answer: 20
Difficult mathematical problems written in English, which can be solved by using systems of equations.	Problem: Emma has $20,000 to invest in two different types of bonds. Bond X pays an annual interest rate of 5%, while Bond Y pays an annual interest rate of 7%. Emma wants to earn a total annual interest of $1,180 from her investments. How much should she invest in each type of bond? Equations: $x + y = 20000, 0.05x + 0.07y = 1180$ Answer: Bond X = $6,000, Bond Y = $14,000
Problems of algebraic number theory, which can be solved using tens of equations.	Problem: Factorize the ideal $(6, \sqrt{-5})$ in the ring $\mathbb{Z}[\sqrt{-5}]$. Equations: $(6, \sqrt{-5}) = (2, 1 + \sqrt{-5})(3, 1 + \sqrt{-5})$ Answer: Factorization: $(6, \sqrt{-5})$ can be expressed as the product of prime ideals $(2, 1 + \sqrt{-5})$ and $(3, 1 + \sqrt{-5})$ in $\mathbb{Z}[\sqrt{-5}]$.
Physics problems in the area of fluid dynamics, which can be solved using partial differential equations.	Problem: Flow over a flat plate at a constant speed. Equations: Continuity Equation: $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$, Navier-Stokes Equation: $\rho(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) = -\frac{\partial p}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$, Energy Equation (if thermal effects are considered): $\rho c_p(u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}) = k(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2})$, Prandtl's Boundary Layer Equations (for simplified flat plate flow): Momentum Equation: $u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \nu\frac{\partial^2 u}{\partial y^2}$, Boundary Conditions: At $y = 0$, $u = 0$ (No-slip condition); As y approaches infinity, $u = U$ (Free stream velocity). Answer: The velocity profile over the flat plate can be analytically described using the Blasius solution, which is a similarity solution to the boundary layer equations for flow over a flat plate.

Table 6: Constructed datasets given different task information, for the task of mathematical problem solver.

Problem A car rental company charges a fixed fee of $50 for renting a car, plus $10 per hour. If a customer receives a bill of $120, how many hours did they rent the car?
Problem + Answer Problem: A car leaves town A traveling towards town B at a constant speed. Five hours later, another car leaves town A towards town B at a constant speed of 100 km/h. If the second car overtakes the first car 3 hours after it starts, what was the speed of the first car? Answer: 37.5 km/h
Problem + Equations + Answer Problem: An electrical circuit consists of two resistors in parallel, R_1 and R_2. The total resistance is 6 ohms. If R1 is 15 ohms, what is R_2? Equations: $1/R_{total} = 1/R_1 + 1/R_2, R_{total} = 6, R_1 = 15$ Answer: $R_2 = 10$ ohms.
Problem + Equations + Reasoning + Answer Problem: Two angles are complementary. The measure of one angle is 10 degrees more than three times the other. Find the measures of both angles. Equations: $x + y = 90, x = 3y + 10$ Reasoning: Let x be one angle and y be the other angle. Since the angles are complementary, they must sum to 90 degrees, hence the equation $x + y = 90$. The problem states that one angle is 10 degrees more than three times the other, which translates to the equation $x = 3y + 10$. Solve this system to find the values of x and y. Answer: The angles are 70 degrees and 20 degrees.

Table 7: Constructed datasets with different data format, for the task of mathematical problem solver.

pected to run with no abnormality. We compare the programs written by or imitated by LLM.

Table 8 shows the statistics on program invocation. The following facts are observed.

The frequency of invoking programs varies a lot. For the programs that obviously compensate the capability of LLMs (e.g., `generate_system_of_equations`) or obviously be helpful for dataset construction (e.g., `translate_to_chinese`), they are invoked more than once on one construction, on average. In contrast, for programs that are useless in the context of dataset construction (e.g., `hotel_reservation`), although they may be helpful in some other scenarios, they are invoked rarely.

Code generation is difficult with only the minimal information from human. We ask LLM to generate Python codes of solving system of equations (i.e., `solve_system_of_equations` (v1)), given the task descriptions on solving equations, and the formats of arguments and return values. The pass rate is approximately zero, and bug-free codes can hardly be generated.

Programs imitated by LLMs work reasonably well. Although the LLM struggles on generating bug-free codes with minimal information, it can imitate the behaviors of programs well enough (e.g., `solve_system_of_equations` (v2)). To achieve fully-automated dataset construction, in cases where human-written programs are not available, using LLM to imitate programs is a rational choice.

B.4 Necessity of Data Checking

The LLM for data checking in the peripheral layer is used to control the quality of constructed datasets, which is generally conduct by human experts in previous research. Our concern is: given the dataset constructed by the collaboration of other LLMs, is it possible to remove this LLM for data checking?

Table 9 shows the pass rate of data checking on dialog and/or data. The pass rate on data checking is extremely high, meaning that the dataset can be constructed successfully given a dialog for dataset construction. However, The pass rate on dialog checking is around 90% for all tasks. This means that checking the quality of dialogs is necessary.

C Prompts of LLMs

In this section, we gives the prompts of LLMs in the proposed method.

Task	Program	Type	Avg. #Invoke	Pass Rate
Translation	`translate_to_chinese`	Human Written	1.43	99.3%
Chatbot	`extract_information`	LLM Imitated	0.07	100.0%
	`check_information`	LLM Imitated	0.05	100.0%
	`search_hotels`	LLM Imitated	0.01	100.0%
	`hotel_reservation`	LLM Imitated	0	-
Math	`generate_system_of_equations`	LLM Imitated	1.72	96.7%
	`solve_system_of_equations` (v1)	LLM Generated	0.52	1.7%
	`solve_system_of_equations` (v2)	LLM Imitated	0.50	87.9%
	`solution_to_final_answer`	LLM Imitated	0.22	84.6%

Table 8: Statistics on program invocation. The Avg. #Invoke is the average number of program invocations per dialog. The pass rate is defined as the percentage that running result passed the check of the LLM for result checking.

Task	Pass Rate (Dialog)	Pass Rate (Data)	Pass Rate (Both)
Translation	94.1%	99.0%	93.1%
Chatbot	91.2%	98.0%	89.2%
Math	87.1%	99.0%	87.1%

Table 9: Pass rate of data checking for three tasks.

C.1 Prompts of LLMs in the Kernal Layer

Prompts of the LLM for human utterance, the LLM for AI utterance, the orchestrator LLM, the LLM for data transformation, can be found in Tables 10, 11, 12, 13, respectively.

C.2 Prompts of LLMs in the External Layer

Prompts of the LLMs for code generation and program imitation can be found in Tables 14 and 15 respectively.

C.3 Prompts of LLMs in the Peripheral Layer

Prompts of the LLMs for result checking, dialog checking, final checking can be found in Tables 16, 17, 18, respectively.

System Description
You are an expert on conversation-based dataset construction. The goal of this conversation is to construct a dataset of the given task. You need to guide the conversation so that the dataset can be constructed successfully.

Instruction
Continue the conversation by generating a guidance, based on the following information about task.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Output
Response to continue the conversation.

Constraints
1. The response should written in English.
2. The response should guide the conversation to a suitable direction.
3. The response should related to dataset construction.

Table 10: The prompt of the LLM for human utterance.

System Description
You are an expert on conversation-based dataset construction. The goal of this conversation is to construct a dataset of the given task. To construct the dataset, you need to follow the guidance in the conversation.

Instruction
Construct data based on the following information about task. Please follow the guidance in the conversation. You can resort to external tools.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Output
Data of the given task.

Constraints
1. You can either modify the data previously generated, or generate new data.
2. No other operations are allowed.
3. No explanation is needed.

Table 11: The prompt of the LLM for AI utterance.

System Description
You are an expert on managing the flow of conversation. Here is a conversation aiming at constructing a dataset of the given task.

Instruction
You need to decide whether to continue the conversation or not. If the conversation should be continued, you need to decide the speaker of next turn.
You have three choices:
1. The speaker should be the guidance model which guide the conversation so that the dataset can be constructed successfully.
2. The speaker should be the data model which either modify the data previously generated in the dialogue, or generate new data.
3. The dialog should not be continued.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Constraints
1. If the conversation is shorter than {min_turns} turns, the conversation must be continued.
2. If the conversation is longer than {max_turns} turns, the conversation must not be continued.
3. The guidance model cannot be invoked continuously.

Output
The index of choice (1, 2, or 3).

Table 12: The prompt of orchestrator LLM.

System Description
You are an expert on data transformation. Here is a conversation aiming at constructing a dataset of the given task.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Instruction
Transform the conversation to data in the following format.
{data_format}

Output
Transformed dataset.

Table 13: The prompt of the LLM for data transformation.

System Description
You are a professional programmer of Python. You can write a function in Python, given the information of the function.

Information of Function
{function_information}

Instruction
Write the code of this function in Python.

Constraints
1. The language is Python.
2. The name of the function should match the given information.
3. The input arguments should match the given information.
4. The return values should match the given information.
5. The code should work perfectly to implement the description in the given information.

Output
a function written in Python

Table 14: The prompt of the LLM for writing programs (i.e., code generation).

System Description
You are a high-performance computer. Given the description of a program, and the input arguments, you can predict the running result.

Instruction
Here is a program.
- Function of this program: {prog_description}
- Arguments: {prog_input}
- Outputs: {prog_output}
Given this program, and the following inputs:
- Inputs: {prog_input_instance}
Predict the running result.

Output
The running result of the program.

Table 15: The prompt of the LLM for imitating programs.

System Description
You are an expert on reading programs. Given the description of a program, you can understand how it works.

Instruction
Here is a program.
- Function of this program: {prog_description}
- Arguments: {prog_input}
- Outputs: {prog_output}
Given this program, and the following inputs:
- Inputs: {prog_input_instance}
The outputs of the program are:
- Outputs: {prog_output_instance}
Was the program executed normally?

You have two choices:
1. The program was executed normally.
2. The program was not executed normally.

Output
The index of choice (1 or 2).

Table 16: The prompt of the LLM for result checking.

System Description
You are an expert on managing the quality of conversation.
Here is a conversation aiming at constructing a dataset of the given task.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Background
You need to check the quality of the conversation in the following dimensions.
1. Awareness of program invoking. The timing of program invoking is appropriate.
2. Choice of program. Among given programs, the correct program has been chosen.
3. Arguments of program. The input arguments of the chosen program are correct.
4. Guidance utterance. The guidance given by the user is useful for dataset construction.
5. Operation utterance. The data generated or modify by the assistant matches the given task.

Instruction
For each dimension, give a score, ranging from 1 to 10. The higher the quality, the higher the score.

Output
1. Score of awareness of program invoking: an integer between 1 and 10.
2. Score of choice of program: an integer between 1 and 10.
3. Score of arguments of program: an integer between 1 and 10.
4. Score of guidance utterance: an integer between 1 and 10.
5. Score of data utterance: an integer between 1 and 10.

Table 17: The prompt of the LLM for dialog checking.

System Description
You are an expert on managing the quality of dataset. Here is a dataset of the given task.

Task Information
Abstract: {task_abstract}
Description: {task_description}

Dataset
{dataset_constructed}

Background
You need to check the quality of the dataset in the following dimensions.
1. Agreement. The dataset matches the given task perfectly.
2. Reality. The dataset matches scenarios in real world.

Instruction
For each dimension, give a score, ranging from 1 to 10. The higher the quality, the higher the score.

Output
1. Score of agreement: an integer between 1 and 10.
2. Score of reality: an integer between 1 and 10.

Table 18: The prompt of the LLM for final checking.

D Information from Human

We provide necessary information for dataset construction in the format of `json`.

D.1 Minimal Information from Human

As introduced in Section 3.1, the minimal information consists of task description and data format. The minimal information for tasks of English-Chinese machine translation, chatbot for hotel reservation, mathematical problem solver can be found in Listings 1, 2, 3, respectively.

```
{
    "name": "English-Chinese machine
        translation",
    "description": "Translate English
        sentences to Chinese.",
    "constraints": {
        "min_turns": 2,
        "max_turns": 10
    },
    "data_format": {
        "list_of_english_sentences": {
            "description": "a list of
                English sentences",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "list_of_chinese_translations":
            {
            "description": "a list of
                Chinese translations",
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    }
}
```

Listing 1: Minimal information from human for the task of English-Chinese machine translation.

```
{
    "name": "Chatbot for hotel
        reservation",
    "description": "The customer
        reserves a hotel by chatting
        with the chatbot",
    "constraints": {
        "min_turns": 3,
        "max_turns": 10
    },
    "data_format": {
        "dialog_history": {
            "description": "conversation
                between the customer
                and the chatbot",
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    }
}
```

Listing 2: Minimal information from human for the task of chatbot for hotel reservation.

```
{
    "name": "Mathematical problem solver
        ",
    "description": "Difficult
        mathematical problems written in
         English, which can be solved by
         using systems of equations.",
    "constraints": {
        "min_turns": 2,
        "max_turns": 15
    },
    "data_format": {
        "list_of_mathematical_problem":
            {
            "description": "A list of
                mathematical problems
                written in English.",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "list_of_system_of_equations": {
            "description": "A list of
                equations to solve the
                given mathematical
                problems.",
            "type": "array",
            "items": {
                "type": "array",
                "items": {
                    "type": "string"
                }
            }
        },
        "list_of_final_answers": {
            "description": "A list of
                final answer to the
                given mathematical
                problem.",
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    }
}
```

Listing 3: Minimal information from human for the task of mathematical problem solver.

D.2 Additional Information on External Programs

Besides the minimal information, in the case that external programs are available, providing additional information is helpful for dataset construction. Additional Information for tasks of English-Chinese machine translation, chatbot for hotel reservation, mathematical problem solver can be found in Listings 4, 5, 6, respectively.

```
[
 {
  "meta_info": {
   "code_exist": true,
   "code_name": "mt_tools"
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "translate_to_chinese",
    "description": "Translate an English
        sentence to Chinese.",
    "parameters": {
     "type": "object",
     "properties": {
      "list_of_english_sentences": {
       "type": "array",
       "description": "a list of English
           sentences",
       "items": {
        "type": "string"
       }
      }
     },
     "required": [
      "english_sentence"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "list_of_chinese_translations": {
       "type": "array",
       "description": "a list of Chinese
           translations",
       "items": {
        "type": "string"
       }
      }
     }
    }
   }
  }
 }
]
```

Listing 4: Additional information about external programs for the task of English-Chinese machine translation.

```
[
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "extract_information",
    "description": "Extract necessary
        information from customer's
        utterance.",
    "parameters": {
     "type": "object",
     "properties": {
      "utterance_from_customer": {
       "type": "string",
       "description": "utterance from
           customer"
      }
     },
     "required": [
      "utterance_from_customer"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "number_of_people": {
       "type": "integer",
       "description": "number of people
           to stay"
      },
      "name_of_representative": {
       "type": "string",
       "description": "name of the
           representative people"
      },
      "checkin_date": {
       "type": "string",
       "format": "mm/dd",
       "description": "date of check in
           "
      },
      "checkout_date": {
       "type": "string",
       "format": "mm/dd",
       "description": "date of check out
           "
      }
     }
    }
   }
  }
 },
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "check_information",
    "description": "Check whether the
        information is enough for hotel
        reservation. If not, return the
        list of unknown information.",
    "parameters": {
     "type": "object",
     "properties": {
      "number_of_people": {
       "type": "integer",
       "description": "number of people
           to stay"
      },
      "name_of_representative": {
       "type": "string",
       "description": "name of the
           representative people"
      },
      "checkin_date": {
       "type": "string",
       "format": "mm/dd",
       "description": "date of check in
           "
      },
      "checkout_date": {
       "type": "string",
       "format": "mm/dd",
```

```
      "description": "date of check out
          "
     }
    },
    "required": [
     "number_of_people",
     "name_of_representative",
     "checkin_date",
     "checkout_date"
    ]
   },
   "results": {
    "type": "object",
    "properties": {
     "unknown_information_list": {
      "description": "list of unknown
          information",
      "type": "array",
      "items": {
       "type": "string",
       "enum": [
        "number_of_people",
        "name_of_representative",
        "checkin_date",
        "checkout_date"
       ]
      }
     }
    }
   }
  }
 },
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "search_hotels",
    "description": "Search hotels that
        satisfy customer's requirements
        .",
    "parameters": {
     "type": "object",
     "properties": {
      "number_of_people": {
       "type": "integer",
       "description": "number of people
          to stay"
      },
      "name_of_representative": {
       "type": "string",
       "description": "name of the
          representative people"
      },
      "checkin_date": {
       "type": "string",
       "format": "mm/dd",
       "description": "date of check in
          "
      },
      "checkout_date": {
       "type": "string",
       "format": "mm/dd",
       "description": "date of check out
          "
      }
     },
     "required": [
      "number_of_people",
      "name_of_representative",
      "checkin_date",
      "checkout_date"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "hotel_list": {
       "type": "array",
       "description": "list of hotels",
       "items": {
        "type": "string"
       }
      }
     }
    }
   }
  }
 },
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "hotel_reservation",
    "description": "Reserve the given
        hotel.",
    "parameters": {
     "type": "object",
     "properties": {
      "name_of_hotel": {
       "type": "string",
       "description": "name of the hotel
           to reserve"
      }
     },
     "required": [
      "name_of_hotel"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "status_of_reservation": {
       "type": "boolean",
       "description": "whether the hotel
           is reserved successfully or
          not"
      }
     }
    }
   }
  }
 }
]
```

Listing 5: Additional information about external programs for the task of chatbot for hotel reservation.

```
[
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
```

```
  "type": "function",
  "function": {
   "name": "
       generate_system_of_equations",
   "description": "Given a mathematical
        problem, generate a system of
       equations that can solve this
       problem.",
   "parameters": {
    "type": "object",
    "properties": {
     "math_problem": {
      "type": "string",
      "description": "mathematical
          problem"
     }
    },
    "required": [
     "math_problem"
    ]
   },
   "results": {
    "type": "object",
    "properties": {
     "equation_list": {
      "type": "array",
      "description": "list of equations
          ",
      "items": {
       "type": "string"
      }
     }
    }
   }
  }
 },
 {
  "meta_info": {
   "code_exist": true,
   "code_name": "math_tools"
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "solve_system_of_equations",
    "description": "Solve the system of
       equations",
    "parameters": {
     "type": "object",
     "properties": {
      "system_of_equations": {
       "type": "array",
       "description": "list of equations
           ",
       "items": {
        "type": "string"
       }
      }
     },
     "required": [
      "system_of_equations"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "solution_to_equations": {
       "description": "solution to the
          system of equations",
       "type": "array",
       "items": {
        "type": "number"
       }
      }
     }
    }
   }
  }
 },
 {
  "meta_info": {
   "code_exist": false
  },
  "tool_info": {
   "type": "function",
   "function": {
    "name": "solution_to_final_answer",
    "description": "Generate the final
       answer of the problem based on
       the solution to the system of
       equations.",
    "parameters": {
     "type": "object",
     "properties": {
      "solution_of_equations": {
       "description": "solution to the
          system of equations",
       "type": "array",
       "items": {
        "type": "number"
       }
      }
     },
     "required": [
      "solution_of_equations"
     ]
    },
    "results": {
     "type": "object",
     "properties": {
      "final_answer_to_problem": {
       "description": "the final answer
          of the problem",
       "type": "string"
      }
     }
    }
   }
  }
 }
]
```

Listing 6: Additional information about external programs for the task of mathematical problem solver.

E More Examples of Constructed Dataset

In this section, we give more examples of the constructed dataset of three tasks.

E.1 English-Chinese Machine Translation

Table 19 shows some examples in the constructed English-Chinese machine translation dataset. Note that although there are only 100 sentence pairs in the dataset, various domains are covered.

Greetings

How was your day? 你今天过得怎么样?
Hey, are you coming to the party tonight? 嘿，你今晚来参加聚会吗?
Hey, how's it going? Long time no see! 嘿，最近怎么样？好久不见了！

Technology

Artificial Intelligence is revolutionizing the way we live and work. 人工智能正在彻底改变我们的生活和工作方式。
Press the power button for three seconds to turn on the device. 按住电源按钮三秒钟以打开设备。
The processor speed exceeds 3 GHz. 处理器速度超过3GHz。
Introducing the new ultra-thin smartphone with a revolutionary design and cutting-edge features. 推出具有革命性设计和尖端功能的新型超薄智能手机。

Science

Photosynthesis is the process by which green plants use sunlight to synthesize foods with the help of chlorophyll. 光合作用是绿色植物在叶绿素的帮助下利用阳光合成食物的过程。
Astronauts aboard the International Space Station conduct groundbreaking research. 国际空间站上的宇航员进行开创性的研究。
This book provides a comprehensive guide to understanding quantum physics. 本书提供了理解量子物理学的全面指南。
The theory of relativity revolutionized the way we understand space and time. 相对论彻底改变了我们理解空间和时间的方式。

Economy

The economic outlook for the next year appears to be uncertain. 明年的经济前景似乎并不确定。
The study explores the impact of renewable energy adoption on economic growth. 该研究探讨了可再生能源的应用对经济增长的影响。
Stock market fluctuations have created uncertainty among investors. 股市波动给投资者带来了不确定性。

Politics

The government has announced new regulations to improve public health. 政府已宣布改善公共卫生的新法规。
The government announced new policies to combat climate change. 政府宣布了应对气候变化的新政策。

Business

The meeting is scheduled for 10 AM tomorrow. 会议定于明天上午10点举行。
We should discuss the quarterly performance report in today's session. 我们应该在今天的会议上讨论季度业绩报告。
Given the current market trends and analysis patterns, our team needs to devise a new marketing strategy. 鉴于当前的市场趋势和分析模式，我们的团队需要制定新的营销策略。

Culture

Cultural heritage plays a crucial role in shaping the identity and values of a community. 文化遗产在塑造一个社区的身份和价值观方面发挥着至关重要的作用。
The Great Wall of China was originally built to protect Chinese states against invasions. 中国的长城最初是为了保护中国各州免遭入侵而修建的。

Climate

Global warming is one of the biggest challenges of our time. 全球变暖是我们这个时代面临的最大挑战之一。
The international community is working together to address climate change challenges. 国际社会正在携手应对气候变化挑战。
The economic implications of climate change are vast and far-reaching. 气候变化的经济影响巨大而深远。

Conversation

Can you recommend a good book I can read over the weekend? 你能推荐一本我周末可以读的好书吗?
Can you help me with this project next week? 下周你能帮我完成这个项目吗?
Can we meet at the coffee shop? 我们可以在咖啡店见面吗?

Literary

He looked at her with a hint of suspicion in his eyes. 他看着她，眼神里带着一丝怀疑。
The rain poured down relentlessly, drenching everything in sight. 大雨无情地倾盆而下，将眼前的一切都淋湿了。
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness. 这是最好的时代，这是最坏的时代；这是智慧的时代，这是愚昧的时代。

Cooking

First, preheat the oven to 350 degrees Fahrenheit. 首先，将烤箱预热至 350 华氏度。

Social Media

Can't believe the weekend is almost over already! #TimeFlies 难以置信周末就快结束了！ #TimeFlies

Law

The agreement shall be governed by and construed in accordance with the laws of California. 本协议应受加利福尼亚州法律管辖并依其解释。

Table 19: Examples in the constructed English-Chinese machine translation dataset, grouped by domains.

Tables 20, 21, and 22 show three dialogues for constructing datasets. Note that several sentence pairs are constructed simultaneously within one dialogue.

E.2 Chatbot for Hotel Reservation

Tables 23 and 24 show some examples in the constructed chatbot for hotel reservation dataset. Requests from customers include many aspects, and chatbots generate appropriate responses to these requests, making the dialogues realistic.

Tables 25, 26, and 27 show three dialogues for constructing datasets. Datasets can be constructed reasonably well, even without using external programs.

E.3 Mathematical Problem Solver

Table 28 shows some examples in the constructed mathematical problem solver dataset. Problems covered many domains, and various difficulties.

Tables 29, 30, and 31 show three dialogues for constructing datasets. Variety of domains and difficulties is considered by the guidance LLM.

[Guidance] To construct a high-quality dataset for English-Chinese machine translation, it's essential to focus on collecting diverse and well-structured sentence pairs. Let's begin by discussing the types of sentences we should include. We want a good mix of sentence complexities, such as simple sentences, compound sentences, and complex sentences. Additionally, it's beneficial to cover various topics like daily conversation, technology, travel, and so on. This will help the translation model generalize well across different contexts. Could we start by brainstorming some sources or methods to collect such diverse sentence pairs?
[Operation] Certainly, here are some methods and sources to collect diverse English-Chinese sentence pairs:
1. **Literature and Books**: Extract sentences from bilingual books or translated literature. These often contain a mix of sentence complexities and rich vocabulary.
2. **News Articles**: Utilize bilingual news websites. They cover various topics including politics, travel, technology, and more.
3. **Online Forums and Social Media**: Collect sentences from translation sub-forums or groups where people request translations, providing both simple and complex real-life usage.
4. **Educational Resources**: Leverage language learning websites or textbooks that provide translated exercises and examples.
5. **Public Domain Datasets**: Use already available translation datasets which could be licensed under creative commons or public domain.
6. **TV and Movie Subtitles**: Subtitle databases contain informal and contextual dialogue that can be beneficial for conversational translation.
7. **Crowdsourcing**: Engage language communities to translate specific sentences, ensuring a mix of various complexities and topics.
These methods can help create a diverse and well-rounded dataset. Let me know if you want to proceed with generating some translation data samples or discuss further!
[Guidance] Perhaps we can also explore industry-specific sources, like user manuals or technical documents, to collect English-Chinese sentence pairs. What industries do you think would be most valuable to include in our dataset, considering current demand for machine translation services?
[Operation] Focusing on industry-specific sources can greatly enhance the quality of a translation dataset. Here are some industries that are particularly valuable given the current demand for machine translation services:
1. **Technology & IT**: User manuals, API documentation, and software guides are crucial as the tech industry frequently requires accurate translations.
2. **Healthcare & Pharmaceuticals**: This includes medical research papers, patient information leaflets, and drug manuals, where precise translations are essential.
3. **Finance & Banking**: Financial reports, investment documents, and banking terms need specialized vocabulary and accuracy.
4. **Legal**: Contracts, terms and conditions, privacy policies, and legal documents demand high precision in translation due to legal implications.
5. **E-commerce & Retail**: Product descriptions, customer service communications, and marketing materials need localization for different markets.
6. **Automotive**: Technical specifications, user guides, and maintenance manuals for vehicles can benefit from accurate translations.
7. **Travel & Hospitality**: Travel guides, hotel brochures, and online booking platforms require translations for tourism purposes.
Including these industries can help build an encompassing dataset that addresses current translation needs across various specialized fields. Let me know how you'd like to proceed!
[Operation] Let's proceed by generating some sample English sentences across different contexts and complexities, which we can then translate into Chinese to create a part of our dataset. I'll prepare some sentences for translation.
[Operation] `translate_to_chinese(list_of_english_sentences=["The user manual for the software contains detailed installation instructions.","Recent advancements in medical research have opened new pathways for treatment.","The financial report for the last quarter shows a significant increase in revenue.","Please read the terms and conditions carefully before signing the contract.","The latest smartphone model features cutting-edge technology and superior performance.","The hotel offers a complimentary breakfast with a wide variety of options.","The car maintenance manual includes a section on troubleshooting common issues.","Our e-commerce platform provides a seamless shopping experience.","The API documentation offers comprehensive guides for developers.","Traveling by train is one of the most scenic ways to explore the countryside."])`
[Program] ['该软件的用户手册包含详细的安装说明。','医学研究的最新进展为治疗开辟了新的途径。','上一季度的财务报告显示收入大幅增加。','签署合同前请仔细阅读条款和条件。','最新款智能手机采用了尖端技术和卓越性能。','酒店提供多种选择的免费早餐。','汽车维修手册中有一节关于常见问题的故障排除部分。','我们的电子商务平台提供无缝的购物体验。','API 文档为开发人员提供了全面的指南。','乘坐火车旅行是探索乡村风景最优美的方式之一。']
[Operation] Here are the translated English-Chinese sentence pairs for our dataset:
1. English: The user manual for the software contains detailed installation instructions.
Chinese: 该软件的用户手册包含详细的安装说明。
2. English: Recent advancements in medical research have opened new pathways for treatment.
Chinese: 医学研究的最新进展为治疗开辟了新的途径。
3. English: The financial report for the last quarter shows a significant increase in revenue.
Chinese: 上一季度的财务报告显示收入大幅增加。
4. English: Please read the terms and conditions carefully before signing the contract.
Chinese: 签署合同前请仔细阅读条款和条件。
5. English: The latest smartphone model features cutting-edge technology and superior performance.
Chinese: 最新款智能手机采用了尖端技术和卓越性能。
6. English: The hotel offers a complimentary breakfast with a wide variety of options.
Chinese: 酒店提供多种选择的免费早餐。
7. English: The car maintenance manual includes a section on troubleshooting common issues.
Chinese: 汽车维修手册中有一节关于常见问题的故障排除部分。
8. English: Our e-commerce platform provides a seamless shopping experience.
Chinese: 我们的电子商务平台提供无缝的购物体验。
9. English: The API documentation offers comprehensive guides for developers.
Chinese: API 文档为开发人员提供了全面的指南。
10. English: Traveling by train is one of the most scenic ways to explore the countryside.
Chinese: 乘坐火车旅行是探索乡村风景最优美的方式之一。
These sentence pairs can be part of a diverse and industry-relevant machine translation dataset. Let me know if you need more examples or any further adjustments!

Table 20: A dialogue for constructing English-Chinese Machine Translation dataset.

[Guidance] To construct the dataset for English-Chinese machine translation, let's start by discussing the types of sentences we need. It's important to have a diverse set of sentences, including different sentence structures, vocabulary, and contexts. This ensures that the machine translation system can handle a wide range of scenarios.
Can we talk about the sources we might use to gather these sentences? For example, we could consider books, news articles, dialogues, or user-generated content. Each source provides a unique style and structure that can enrich our dataset.
[Operation] Let's focus on constructing a diverse dataset by gathering sentences from a variety of sources:
1. **Books**: Literary works provide rich language, complex structures, and diverse vocabulary. This type of content can include both classic literature and contemporary novels to cover different styles and time periods.
2. **News Articles**: These provide formal language and are useful for current events, politics, economics, and technical topics. They can help ensure that the translation system is up-to-date with contemporary usage.
3. **Dialogues**: Conversational data from sources like scripts, interviews, or direct conversations can cover informal language, slang, and colloquialisms. This is crucial for making translations more natural and relatable in everyday contexts.
4. **User-generated Content**: Online forums, social media posts, and blog comments can introduce a variety of tones, from formal to informal, and include user-specific jargon, abbreviations, and neologisms.
By incorporating sentences from these diverse sources, we can achieve a well-rounded dataset that aids in developing a robust translation system. Let's now proceed with gathering some example sentences across these categories to begin the dataset construction. Please provide some sentences, or I can suggest some for translation to start building our dataset.
[Operation] Here are some example English sentences from various sources that we can use to start building our dataset:
1. **Books**:
- She wandered into the forest, where the trees whispered secrets.
- His courage in the face of adversity inspired everyone around him.
2. **News Articles**:
- The government announced new policies to combat climate change.
- Stock market fluctuations have created uncertainty among investors.
3. **Dialogues**:
- Hey, are you coming to the party tonight?
- I don't think that's a good idea right now.
4. **User-generated Content**:
- OMG, this new song is a total banger!
- Can anyone recommend a good restaurant nearby for dinner?
Let's translate these sentences into Chinese.
[Operation] `translate_to_chinese(list_of_english_sentences=["She wandered into the forest, where the trees whispered secrets.", "His courage in the face of adversity inspired everyone around him."])`
[Program] ['她漫步走进森林，那里的树木低声诉说着秘密。','他面对逆境的勇气激励了周围的每一个人。']
[Operation] `translate_to_chinese(list_of_english_sentences= ["The government announced new policies to combat climate change.", "Stock market fluctuations have created uncertainty among investors."])`
[Program] ['政府宣布了应对气候变化的新政策。','股市波动给投资者带来了不确定性。']
[Operation] `translate_to_chinese(list_of_english_sentences=["Hey, are you coming to the party tonight?","I don't think that's a good idea right now."])`
[Program] ['嘿，你今晚会来参加聚会吗？','我认为现在这不是一个好主意。']

Table 21: A dialogue for constructing English-Chinese Machine Translation dataset.

[Guidance] To construct a high-quality dataset for English-Chinese machine translation, we first need to gather a diverse set of English sentences from various sources. These should cover different topics, styles, and levels of complexity to ensure the translation model can learn effectively.
Next, we need to focus on obtaining accurate Chinese translations for these English sentences. It's crucial to ensure the translations are done by native speakers or professional translators to maintain linguistic accuracy and cultural relevance.
Let's start by identifying the sources for our English sentences. We could consider using literary works, news articles, dialogue from movies or TV shows, and technical documents to diversify our dataset. Once we have a list of potential sources, we should discuss the best ways to acquire translations. Would crowdsourcing, professional translators, or a combination of both be suitable for our needs?
[Operation] Let's start by translating a diverse set of English sentences to Chinese. I'll use sentences from different categories to simulate diversity in topics, styles, and complexity.
1. "The economic implications of climate change are vast and far-reaching."
2. "To be, or not to be, that is the question."
3. "The latest smartphone release has taken the tech world by storm."
4. "In the field of artificial intelligence, machine learning has become a crucial tool."
5. "The sun set behind the mountains, painting the sky with hues of orange and pink."
6. "The quick brown fox jumps over the lazy dog."
7. "Astronauts aboard the International Space Station conduct groundbreaking research."
8. "This book provides a comprehensive guide to understanding quantum physics."
9. "As the city slept, the streets were silent, save for the occasional rustle of leaves."
10. "The novel explores themes of identity, love, and the passage of time."
I will now translate these sentences into Chinese.
[Operation] `translate_to_chinese(list_of_english_sentences=["The economic implications of climate change are vast and far-reaching.","To be, or not to be, that is the question.","The latest smartphone release has taken the tech world by storm.","In the field of artificial intelligence, machine learning has become a crucial tool.","The sun set behind the mountains, painting the sky with hues of orange and pink.","The quick brown fox jumps over the lazy dog.","Astronauts aboard the International Space Station conduct groundbreaking research.","This book provides a comprehensive guide to understanding quantum physics.","As the city slept, the streets were silent, save for the occasional rustle of leaves.","The novel explores themes of identity, love, and the passage of time."])`
[Program] ['气候变化的经济影响巨大而深远。','生存，还是毁灭，这是一个问题。','最新款智能手机的发布引起了科技界的轰动。','在人工智能领域，机器学习已经成为一个至关重要的工具。','太阳落入山后，天空呈现出橙色和粉红色的色调。','这只敏捷的棕色狐狸跳过了那只懒狗。','国际空间站上的宇航员进行开创性的研究。','本书提供了理解量子物理学的全面指南。','当城市沉睡时，街道一片寂静，除了偶尔传来树叶的沙沙声。','这部小说探讨了身份、爱情和时间流逝的主题。']
[Operation] Here are the translated Chinese sentences:
1. 气候变化的经济影响巨大而深远。
2. 生存，还是毁灭，这是一个问题。
3. 最新款智能手机的发布引起了科技界的轰动。
4. 在人工智能领域，机器学习已经成为一个至关重要的工具。
5. 太阳落入山后，天空呈现出橙色和粉红色的色调。
6. 这只敏捷的棕色狐狸跳过了那只懒狗。
7. 国际空间站上的宇航员进行开创性的研究。
8. 本书提供了理解量子物理学的全面指南。
9. 当城市沉睡时，街道一片寂静，除了偶尔传来树叶的沙沙声。
10. 这部小说探讨了身份、爱情和时间流逝的主题。

Table 22: A dialogue for constructing English-Chinese Machine Translation dataset.

Customer: Can you find hotels in downtown for two people from March 5th to March 10th?
Chatbot: Sure, let me search for available hotels in downtown for your specified dates.
Customer: I would like to reserve a room at the Grand Hotel from April 1st to April 5th.
Chatbot: Great choice! I'll proceed with the reservation at Grand Hotel for you.
Customer: I need to change my check-in date to March 6th.
Chatbot: I can help with that. Let me update your check-in date to March 6th.
Customer: Cancel my reservation at the Star Hotel.
Chatbot: Your reservation at Star Hotel has been canceled.
Customer: Does the hotel offer airport shuttle service?
Chatbot: Yes, the hotel does offer airport shuttle service.
Customer: Does the room include free WiFi and complimentary breakfast?
Chatbot: Yes, all rooms include free WiFi and complimentary breakfast.
Customer: What is your cancellation policy?
Chatbot: Our cancellation policy allows you to cancel up to 24 hours before check-in without penalties.
Customer: Can you confirm my reservation details for March 8th at the City Inn?
Chatbot: Certainly! Your reservation at City Inn is confirmed for March 8th.

Customer: Can you show me hotels in New York?
Bot: Sure! Here are some options in New York: The Plaza, Hilton Midtown, and Marriott Marquis. Do you have a preference or need more options?
Customer: Are there pet-friendly hotels?
Bot: Yes, we have pet-friendly hotels like Kimpton Hotel Eventi and Hotel Pennsylvania.
Customer: I need a room from December 1st to December 5th.
Bot: Great! Let me check the availability for those dates.
Customer: Can I extend my stay until December 7th?
Bot: Let me check if the extension is possible for your selected hotel and dates.
Customer: I'm looking for a single room.
Bot: We have single rooms available at The Plaza and Hilton Midtown. Would you like to proceed with a booking?
Customer: Do you have wheelchair-accessible rooms?
Bot: Yes, we have wheelchair-accessible rooms at several hotels. Would you like more information?
Customer: I'd like to book the Queen room from December 1st to 5th.
Bot: The Queen room is available for those dates at Hilton Midtown. Shall I confirm the booking?
Customer: Can you send me a confirmation email?
Bot: Sure, please provide your email address and I'll send the confirmation right away.
Customer: What's available in the city center with a pool?
Bot: In the city center, we have available hotels like The Grand and City Square Hotel, both with pool facilities. Would you like to know more?
Customer: Are there any rooms available for tonight that offer late check-in?
Bot: Yes, several hotels offer late check-in options for tonight. The Riverside Hotel and Park Inn are two such choices.
Customer: Can you show me rooms with a sea view and a king-size bed?
Bot: Yes, The Oceanfront Hotel offers rooms with a sea view and king-size beds. Would you like to book one?
Customer: I need to update my reservation to include a breakfast package.
Bot: Certainly, I can add the breakfast package to your reservation. Would you like me to proceed?
Customer: Do you have any options near the beach for this weekend?
Bot: Yes, Beachside Resort and Ocean View Hotel are available near the beach this weekend. Would you like to see their rates?
Customer: What's available for check-in on November 10th for a 3-night stay?
Bot: For those dates, we have availability at Central Park Hotel and Downtown Inn. Shall I show you the room options?
Customer: Do you offer any rooms that include a spa bath?
Bot: Yes, our Luxurious Spa Suite includes a spa bath. Would you like to check the availability?
Customer: Can you confirm my booking for December 15th at Beachside Resort?
Bot: Your booking for December 15th at Beachside Resort is confirmed. Is there anything else I can assist you with?
Customer: I have dietary restrictions; can the hotel accommodate a gluten-free menu?
Bot: The hotel can certainly accommodate gluten-free menus. Would you like this added to your reservation notes?

Customer: I need a room from March 5th to March 8th.
Chatbot: Great! I can assist with that. Would you like a confirmation email?
Customer: I'm thinking of staying sometime in the last week of March. What days are available?
Chatbot: Let me check that for you. Could you specify which days you're interested in more specifically?
Customer: I need a king suite.
Chatbot: Sure! A king suite is available. Will you require any additional services?
Customer: I'm looking for something with a view, but I'm not sure what room type that would be.
Chatbot: We have several options with views. Would you like a city view or a garden view?
Customer: I need a room for two, with separate beds.
Chatbot: I have a twin room available for you. When would you like to check in?
Customer: I need space for 3 adults and 2 kids. Do you have connecting rooms?
Chatbot: Yes, we offer connecting rooms. I can arrange that for your stay.
Customer: Is a late checkout option available?
Chatbot: Late checkout is available upon request for a small fee. Shall I arrange this for you?
Customer: Can we get a crib, and is there airport shuttle service?
Chatbot: Certainly! A crib can be arranged, and we do have shuttle services. Would you like me to book these for you?
Customer: I'll pay with my card upon arrival.
Chatbot: That's perfectly fine. We accept all major credit cards.
Customer: Can I use cryptocurrency to settle the bill?
Chatbot: I can confirm whether we accept cryptocurrency for payments and get back to you.
Customer: I need to cancel my reservation for March 5th to March 8th.
Chatbot: I can assist with that. Your reservation has been canceled and a confirmation email has been sent.
Customer: I need to cancel my reservation for tonight.
Chatbot: We can process your cancellation, but there may be a fee as per our policy. Shall I proceed?
Customer: Je voudrais réserver une chambre.
Chatbot: I see you prefer French. Let me guide you through the process. Quelle date préférez-vous pour l'enregistrement?
Customer: Hablas español, necesito más información sobre las habitaciones.
Chatbot: We can assist in Spanish. What details would you like to know about our rooms?
Customer: I booked a suite, but the confirmation says a double room.
Chatbot: I apologize for the confusion. Let me rectify this for you. Would you like to upgrade to a suite?
Customer: I thought my booking was for the 5th, but it's showing the 15th.
Chatbot: Let me correct this for you right away. I'll update your reservation to March 5th.

Customer: Hi, I'm looking to book a deluxe room from 10/15 to 10/18. Is it available?
Chatbot: Let me check that for you. Could you please tell me the number of people staying with you?
Customer: Just me.
Chatbot: Thank you. Yes, the deluxe room is available for those dates. Would you like to proceed with the booking?
Customer: Hello, I have a reservation from 11/01 to 11/05, and I need a crib for my baby. Can you arrange that?
Chatbot: Of course! I can arrange a crib for your stay. Is there anything else you need assistance with, like a late check-out?
Customer: Yes, a late check-out would be great.
Chatbot: I'll note your request for a late check-out. We'll confirm it upon your arrival. Is there anything else I can help you with?
Customer: Hi, I recently stayed at your hotel, and I have some feedback about my stay.
Chatbot: We're sorry to hear that. Could you please provide more details about your experience so we can assist you better?
Customer: The room was noisy, and I had trouble sleeping.
Chatbot: We apologize for the inconvenience. We appreciate your feedback and will address the noise issue immediately. Would you like us to offer a discount on your next stay?

Customer: Hi, I'd like to book a hotel room.
Chatbot: Sure, could you please tell me the number of people who will be staying?
Customer: It's for two people.
Chatbot: Great! When would you like to check in and check out?
Customer: Check-in on 10/20 and checkout on 10/22.
Chatbot: Under whose name should I make the reservation?
Customer: John Doe.
Chatbot: Thank you! I will now search for available hotels for you.
Customer: I'd like to know what types of rooms are available.
Chatbot: We have single, double, and suite rooms available. Do you have a preference?
Customer: I'm interested in a suite. Does it come with free Wi-Fi?
Chatbot: Yes, our suites include free Wi-Fi. Are there any other amenities you're looking for?
Customer: I'd prefer a non-smoking room, please.
Customer: I'm looking for a hotel in New York City within a budget.
Chatbot: Could you please specify your budget range?
Customer: I'd like to stay under $200 per night.
Chatbot: Thank you! I will find hotels in New York City that fit your budget.
Customer: I need a room with wheelchair accessibility.
Chatbot: Certainly, I'll make a note of that special request. Can I confirm the booking now?
Customer: Yes, please confirm.
Chatbot: Your reservation is confirmed under the name Jane Doe. Your booking reference number is ABC1234.
Customer: I need a room for next weekend.
Chatbot: Could you please specify the check-in and check-out dates?
Customer: Check-in on 10/21 and check-out on 10/23.
Customer: I'd like to book a room for one person.
Chatbot: Sure, when would you like to check in and check out?
Customer: Actually, it's for two people. Check-in on 10/15 and check-out on 10/17.
Customer: I'm not sure what hotel to choose. Can you recommend a popular one?
Chatbot: Certainly! Would you like options based on ratings, location, or amenities?
Customer: Based on ratings, please.
Customer: I'd like a suite from 10/10 to 10/15.
Chatbot: I'm sorry, suites are fully booked for those dates. Would you like to see other room types?
Customer: Sure, what do you have available?
Customer: I want a three-night stay sometime next month.
Chatbot: Could you please specify the exact dates for your stay?
Customer: From 11/15 to 11/18, if possible.
Customer: Can I book a room for tonight?
Chatbot: Let me check for last-minute availability. Could you provide your name and number of guests?
Customer: John Smith, and it's for one guest.
Customer: I need to cancel my booking.
Chatbot: I'm sorry to hear that. Could you provide your booking reference number?
Customer: It's XYZ5678.
Customer: I need two rooms for a family of six.
Chatbot: How would you like the rooms to be arranged? For example, one room for four and another for two?
Customer: Yes, please. One for four and one for two.
Customer: I'm booking for a honeymoon. Do you offer any special packages?
Chatbot: Yes, we offer honeymoon packages including complimentary room service and spa access. Would you like to add this to your booking?
Customer: That sounds great, please add it.
Customer: What's your cancellation policy?
Chatbot: You can cancel for free until 24 hours before check-in. Would you like a reminder on this policy?
Customer: No, that's fine. Thank you.

Table 23: Examples in the constructed chatbot for hotel reservation dataset.

Customer: Hi! Can you tell me if there are any single rooms available next weekend?
Chatbot: Sure! Could you please confirm the check-in and check-out dates?
Customer: Hello, do you have any family suites with a kitchen available in July?
Chatbot: Yes, we do! Family suites with kitchen facilities are available. Could you please provide the specific dates?
Customer: I need a room for a conference next week. Does your hotel offer business amenities and high-speed internet?
Chatbot: Absolutely, we offer business amenities including a conference room and high-speed internet. What are your intended check-in and check-out dates?
Customer: Hola, ¿tienen habitaciones disponibles para el próximo mes?
Chatbot: Sí, tenemos disponibilidad. ¿Cuáles son las fechas específicas de entrada y salida?
Customer: Could you tell me the price range for a double room from May 10th to May 15th?
Chatbot: Our double rooms range from $100 to $150 per night during those dates. Would you like to make a reservation?

User: Hi, I want to book a room.
Chatbot: Sure! Could you please provide the check-in and check-out dates?
User: Check-in will be on December 15th and check-out on December 18th.
Chatbot: Got it. How many people will be staying?
User: It will be two people.
Chatbot: What type of room would you like?
User: A double room, please.
Chatbot: Alright, let me check the availability for you.
Chatbot: We have a double room available for your dates. Would you like to book it?
User: Yes, please proceed with the booking.
Chatbot: Could I have the name of the person who will be the representative on the reservation?
User: The name is Alex Smith.
Chatbot: Thank you, Alex. Your room has been successfully booked from December 15th to December 18th for two people. Is there anything else I can help you with?
User: No, that's all. Thank you!
Chatbot: You're welcome! Have a great day!
User: Hi, do you have any rooms available?
Chatbot: I can help with that! Could you please provide the dates for your stay?
User: I'm planning to stay from June 1st to June 5th.
Chatbot: And how many guests will be staying?
User: Just me.
Chatbot: Let me check the availability for you.
Chatbot: We have several options available for those dates. Would you like to know more about the types of rooms or their prices?
User: Yes, can you tell me about the room types?
Chatbot: Certainly! We have single, double, and suite rooms available. The single room is a cozy option, the double offers more space, and the suite includes a living area. Would you like to proceed with booking any of these?
User: I'll think about it and get back to you. Thanks!
Chatbot: You're welcome! Feel free to reach out anytime.
User: What amenities do you offer with your rooms?
Chatbot: Our rooms come with complimentary Wi-Fi, breakfast, and access to our gym and pool. You can also request laundry service and room service for an additional fee. Is there anything specific you're interested in?
User: Do you have airport shuttle service?
Chatbot: Yes, we offer an airport shuttle service. It can be arranged by contacting the front desk. Would you like to know more about our services?
User: No, that will be enough. Thanks for the information!
Chatbot: You're welcome! Let me know if there's anything else I can assist you with.

Customer: I want to book a room from December 5th to December 10th.
Chatbot: Checking availability for those dates...
Customer: Are there any available rooms next weekend?
Chatbot: Searching for rooms available from December 7th to December 9th...
Customer: Can you suggest the cheapest dates to stay in early December?
Chatbot: The cheapest rates are available from December 3rd to 6th.
Customer: I need a room for two adults and one child.
Chatbot: We have several family rooms available. Would you like city view or garden view?
Customer: Do you have any suites available?
Chatbot: Yes, we have a variety of suites. Would you like a junior or executive suite? Customer: Can I upgrade to a business class room?
Chatbot: Sure, there's an additional charge of $50 per night for an upgrade.
Chatbot: Can I have the name of the main guest, please?
Customer: The reservation should be under John Smith.
Chatbot: May I have your contact number or email for the confirmation?
Customer: Sure, it's johnsmithexample.com.
Customer: Is there a hotel near the city center?
Chatbot: Yes, I can recommend Citylight Hotel, which is just 5 minutes from the center.
Customer: I prefer a 4-star hotel or higher.
Chatbot: We have several 4 and 5-star options. Would you like more information on ratings?
Customer: Does the hotel have a swimming pool and gym?
Chatbot: Yes, our hotel features both a heated pool and a 24-hour gym.
Customer: What are your pet policies?
Chatbot: We welcome pets up to 50 lbs with an additional cleaning fee of $30. Customer: What is the total cost for a 3-night stay in a standard room?
Chatbot: The total would be $450, including taxes and fees.
Customer: Can I pay upon arrival, or do I need to prepay?
Chatbot: You can choose either option. Prepaying comes with a 5% discount.
Chatbot: Please confirm your booking details for December 5th to 10th at City Hotel.
Customer: Yes, that's correct. Please send the confirmation to my email.
Chatbot: Your booking is confirmed. The confirmation number is 123456.
Customer: Thank you!
Customer: Can I change my check-in date to December 6th?
Chatbot: Your request is processed. Your new check-in date is December 6th.
Customer: I need to cancel my reservation for December 5th.
Chatbot: Your reservation has been cancelled. You will receive a confirmation email shortly.
Chatbot: How was your booking experience with us?
Customer: It was easy and fast, thank you!
Customer: I had some issues with the date selection process.
Chatbot: I'm sorry to hear that. We'll look into the issue for improvement.

Customer: Hi, I'd like to modify my reservation.
Chatbot: Sure, I can help with that. Could you please provide your reservation details, such as your name and reservation number?
Customer: My name is Alex Smith, and my reservation number is 98765ZX.
Chatbot: Thank you, Alex. What changes would you like to make to your reservation?
Customer: I need to change my check-out date from October 15th to October 17th.
Chatbot: Let me check the availability for those dates.
Chatbot: We have availability for those dates. I have updated your reservation to check-out on October 17th. Is there anything else you need to change?
Customer: No, that's all. Thank you!
Chatbot: You're welcome! Your reservation has been updated. Have a great day!

Customer: Hi, I'd like to book a hotel room.
Chatbot: Sure! Could you please provide the dates for your stay and the number of guests?
Customer: Hello, I'm planning a trip and need a hotel room.
Chatbot: Great! Could you let me know your check-in and check-out dates, as well as the number of guests?
Customer: Can you help me book a hotel for my upcoming vacation?
Chatbot: Of course! Please share the dates and the number of people who will be staying.
Customer: I'm looking for a hotel in downtown with a pool and free wifi.
Chatbot: I have found several options in downtown with a pool and free wifi. What is your budget range?
Customer: I prefer a hotel with a gym and breakfast included.
Chatbot: I've found some options with a gym and breakfast. Could you specify your preferred price range or star rating?
Customer: I need a hotel close to the airport with a shuttle service.
Chatbot: I can find hotels near the airport with a shuttle service. What's your maximum nightly budget?
Customer: Can you check if the Grand Hotel is available from February 10th to 15th for 2 people?
Chatbot: Yes, the Grand Hotel is available for your selected dates. Would you like to proceed with the booking?
Customer: Is there any availability at the Beach Resort from June 5th to 10th?
Chatbot: Yes, the Beach Resort has vacancies for those dates. Should I reserve a room for you?
Customer: Can I book a suite at the City Inn for next weekend?
Chatbot: The City Inn has suites available for next weekend. Would you like to confirm the booking?
Customer: I'd like to pay with my credit card.
Chatbot: Sure! Please provide your credit card details to complete the reservation.
Customer: Can I use PayPal for payment?
Chatbot: Yes, PayPal is accepted. Please follow the link to complete your payment.
Customer: I want to split the bill between two credit cards.
Chatbot: Sure, you can split the payment. Enter the first card details for half the amount, and then the second card.
Customer: Can I change my booking dates after reservation?
Chatbot: Yes, you can modify your reservation dates. Let me know the new dates to update your booking.
Customer: What if I need to cancel my reservation?
Chatbot: You can cancel your reservation from your account page. Please note the cancellation policy applies.
Customer: Is late check-out available?
Chatbot: Late check-out is possible depending on availability. Would you like me to request a late check-out for you?

Customer: Can you check the availability of rooms for me?
Chatbot: Certainly! Could you please provide the dates you are interested in?
Customer: What amenities does the hotel offer?
Chatbot: Our hotel offers free Wi-Fi, a fitness center, a pool, room service, and more. Do you have any specific amenities you're looking for?
Customer: What are the rates for a single room?
Chatbot: Single room rates start at $150 per night, depending on the time of year and availability. Would you like to book one?
Customer: How can I modify my existing reservation?
Chatbot: You can modify your reservation by providing your reservation number and the changes you wish to make. How can I assist you with that?

Table 24: Examples in the constructed chatbot for hotel reservation dataset.

[Guidance] To construct a comprehensive dataset for a hotel reservation chatbot, we need to identify various scenarios and interactions that might occur during a reservation process. Let's start by outlining some key components that should be included:
1. **User Intentions**: Identify specific user intents, such as checking availability, inquiring about amenities, making a reservation, canceling a reservation, and modifying a reservation.
2. **Essential Information**: Determine the essential information needed from the user, such as check-in and check-out dates, room type preferences, number of guests, and any special requests.
3. **Common User Queries**: List common user questions or concerns, such as price options, cancellation policies, check-in/check-out procedures, and location-related queries.
4. **Chatbot Responses**: Develop a variety of potential responses that the chatbot might give, which should cover both the successful handling of requests as well as error handling and clarifications.
5. **Contextual Variations**: Consider different ways users might phrase similar requests, including varying levels of formality and possible language use to reflect real-life scenarios.
Let's begin by focusing on these components and brainstorm initial user intents and potential user questions. Would you like to start with user intentions or something else?
[Guidance] Great start! To ensure we have a well-rounded dataset, tackling user intentions first is a logical step. Let's identify a list of specific user intents that users might have when interacting with the hotel reservation chatbot. Here are a few to consider:
- **Booking a Room**: The primary intention where users specify their preferences and confirm their reservation.
- **Room Availability Check**: Users inquire if rooms are available for specific dates.
- **Inquiring About Amenities**: Users asking about facilities like Wi-Fi, pools, gyms, etc.
- **Price Inquiries**: Users seeking information on room rates and any available deals or discounts.
- **Reservation Modification**: Users want to change their reservation details, such as dates or room type.
- **Reservation Cancellation**: Users need to cancel their booking and may inquire about refund policies.
- **General Hotel Information**: Information about hotel location, check-in/check-out times, or parking facilities.
For each of these intents, we should consider the types of information the user might provide and what they expect in return. Let's list out specific examples or typical questions under each intent to ensure comprehensive coverage. Which intent would you like to explore first, or do you have another area in mind to focus on initially?
[Operation] Let's explore the intent of **Booking a Room** first. We'll identify the specific information that users might provide and potential questions they might ask under this intent. Here are some aspects to consider:
- **Information Provided by the User**:
- Desired check-in and check-out dates.
- Number of guests, including adults and children.
- Room type preferences (e.g., single, double, suite).
- Special requests or needs (e.g., accessible room, late check-in).
- Name and contact details for reservation.
- **Potential User Questions**:
- What types of rooms are available?
- Can I book a room with a specific view or feature?
- Are there any current promotions or discounts available?
- Is breakfast included with the room booking?
- How can I confirm my reservation details?
For each of these aspects and questions, we need to create interaction scenarios and ensure that our chatbot can handle various phrasings and provide accurate responses. Do you want to create specific examples of conversation turns for these, or is there another part of the intention you'd like to develop further?
[Operation] Sure, let's create specific examples of conversation turns for the ""Booking a Room"" intent. Here are a few possible interactions:
1. **User**: ""I want to book a room from the 15th to the 18th of this month for two adults and one child.""
- **Chatbot**: ""Great! We have a range of rooms available. Would you prefer a standard room, a family room, or a suite?""
2. **User**: ""Do you have any rooms with a sea view available for next weekend?""
- **Chatbot**: ""Yes, we do have sea view rooms available for the upcoming weekend. Would you like to proceed with booking one?""
3. **User**: ""Are there any special deals if I book a room for a week?""
- **Chatbot**: ""Yes, we offer a 10% discount on weekly bookings. Would you like me to check availability for the whole week?""
4. **User**: ""Is breakfast included in the room rate?""
- **Chatbot**: ""Breakfast is complimentary with all room bookings. Would you like to book a room with us?""
5. **User**: ""Can I confirm my reservation details?""
- **Chatbot**: ""Sure, please provide your name or reservation number, and I will fetch the details for you.""
These examples cover initial booking queries and follow-up questions, providing a basis for developing the chatbot's understanding and response capabilities. Shall we proceed with a specific scenario or explore another intent?
[Operation] Let's proceed with specific scenarios and explore various user interactions regarding **Room Availability Check**. Here are some example interactions under this intent:
1. **User**: ""Can you check if there are any rooms available from March 5th to March 10th?""
- **Chatbot**: ""Certainly! We have several options available for those dates. Would you like a view or a particular room type?""
2. **User**: ""Are there any suites open this Friday and Saturday?""
- **Chatbot**: ""Yes, we have suites available this Friday and Saturday. Would you like to reserve one?""
3. **User**: ""Do you have any rooms with twin beds available next weekend?""
- **Chatbot**: ""Yes, we have rooms with twin beds available for next weekend. Can I assist with booking?""
4. **User**: ""Is there a presidential suite open for the entire month of June?""
- **Chatbot**: ""Let me check. We have a presidential suite available for the month of June. Would you like to make a reservation?""
5. **User**: ""I'm looking for a room on the 20th of this month. Anything available?""
- **Chatbot**: ""Yes, we have several rooms available on the 20th. Please choose your preferred type.""
These examples demonstrate how the chatbot manages availability inquiries, offering information and prompting the user for further action. Let me know if you would like to explore another intent or further develop these scenarios.
[Operation] Let's further explore interactions for the intent **Inquiring About Amenities**. Here are several example interactions:
1. **User**: ""Does your hotel have a gym?""
- **Chatbot**: ""Yes, we have a fully-equipped gym available for all guests. It's open from 6 AM to 10 PM daily.""
2. **User**: ""Is there free Wi-Fi in the rooms?""
- **Chatbot**: ""Yes, complimentary Wi-Fi is available in all rooms and public areas of the hotel.""
3. **User**: ""Do you have a swimming pool?""
- **Chatbot**: ""Yes, we have an outdoor swimming pool available from 8 AM to 9 PM every day.""
4. **User**: ""Are pets allowed in your hotel?""
- **Chatbot**: ""Yes, our hotel is pet-friendly. We do have certain pet policies, though. Would you like more details?""
5. **User**: ""Is breakfast included with the stay?""
- **Chatbot**: ""Yes, breakfast is included with every booking and is served from 7 AM to 10 AM.""
These interactions show how the chatbot can address inquiries related to hotel amenities, providing clear information to help users plan their stay. Would you like to continue with this intent or switch to another?

Table 25: A dialogue for constructing chatbot for hotel reservation dataset.

[Guidance] That's great! To construct a dataset for a hotel reservation chatbot, we should first consider some key components. We'll need to gather different types of sample interactions between a customer and the chatbot. Let's start by thinking about common scenarios that a customer might encounter when booking a hotel.
Can we brainstorm a list of possible intents or requests a customer might have during this interaction? For example, asking about room availability, inquiring about amenities, or requesting booking modifications. Once we have a clearer picture of these intents, we can work on creating sample dialogues for each.
[Operation] Here are some common intents or requests a customer might have during a hotel reservation interaction with a chatbot:
1. **Room Availability Inquiry**: Checking if there are rooms available for certain dates.
2. **Booking a Room**: Completing the process of reserving a room.
3. **Cancellation**: Requesting to cancel a reservation.
4. **Modification**: Changing details of an existing reservation (dates, room type, etc.).
5. **Amenities Inquiry**: Asking about available amenities (e.g., pool, gym, Wi-Fi).
6. **Price Check**: Inquiring about room rates for specific dates.
7. **Special Requests**: Adding special requirements like early check-in, late check-out, or room preferences.
8. **Payment Confirmation**: Asking for confirmation of payment or invoices.
9. **Location Inquiry**: Asking for details about the hotel location and nearby attractions.
10. **Check-in/Check-out Time Inquiry**: Clarifying the timing for check-in and check-out.
With these intents in mind, we can move on to creating sample dialogues for each. Let me know how you'd like to proceed!
[Guidance] Additionally, we could consider edge cases or less common interactions that might also be valuable for our dataset. For instance, handling requests for group bookings, dealing with language or communication barriers, or managing unavailability scenarios when rooms are fully booked.
Would you like to dive into creating sample dialogues for some of these specific intents, or perhaps explore any additional ones that might be relevant? Let's decide on our next step to ensure the dataset covers a broad range of customer interactions.
[Guidance] Great idea! Including edge cases and less common interactions will make the dataset more robust and comprehensive. Let's outline a few additional intents and scenarios to expand our dataset:
1. **Group Bookings**: Handling inquiries and reservations for larger groups or events.
2. **Waiting List**: Managing responses when the hotel is fully booked and offering alternatives like a waiting list.
3. **Language Assistance**: Responding to requests for language preferences or translation services.
4. **Feedback and Complaints**: Handling customer feedback or complaints regarding their stay.
5. **Accessibility Needs**: Catering to special requirements related to accessibility features.
6. **Discounts and Promotions**: Queries related to any available discounts, packages, or loyalty programs.
7. **Emergency Requests**: Dealing with emergencies or last-minute cancellations due to unexpected circumstances.
Let's start with creating sample dialogues for these less common scenarios. Please let me know if there are any specific intents you'd like to prioritize, or if there are additional scenarios you think we should consider.
[Operation] Here are a few sample dialogues for some of the less common scenarios:
1. **Group Bookings**:
- **Customer**: "Hi, I'd like to book several rooms for a business retreat. Do you offer any group rates or packages?"
- **Chatbot**: "Hello! Yes, we do offer special rates for group bookings. Could you please specify the dates and number of rooms you require?"
2. **Waiting List**:
- **Customer**: "I'm trying to book a room for next weekend, but it looks like you're full. Is there a waiting list option?"
- **Chatbot**: "Unfortunately, we are currently fully booked for that weekend. However, I can add you to our waiting list. May I have your contact details?"
3. **Language Assistance**:
- **Customer**: "Hola, no hablo inglés muy bien. ¿Hay alguien que hable español?"
- **Chatbot**: "¡Hola! Claro, puedo asistirte en español. ¿En qué puedo ayudarte hoy?"
4. **Feedback and Complaints**:
- **Customer**: "I had an issue with my room during my last stay. How can I file a complaint?"
- **Chatbot**: "I'm sorry to hear that. Could you please provide me with your reservation details, and I will assist you in filing a complaint?"
5. **Accessibility Needs**:
- **Customer**: "Do you have rooms that are wheelchair accessible?"
- **Chatbot**: "Yes, we do have wheelchair accessible rooms. Would you like to check availability for specific dates?"
6. **Discounts and Promotions**:
- **Customer**: "Are there any promotions or discounts available if I book directly through this chat?"
- **Chatbot**: "We offer a 10% discount for bookings made through our chat. When would you like to book your stay?"
7. **Emergency Requests**:
- **Customer**: "I need to cancel my reservation due to an emergency. What is your cancellation policy?"
- **Chatbot**: "I'm sorry to hear about your emergency. We have a 24-hour cancellation policy. Could you provide your reservation number, and I'll assist you with the cancellation?"
If there are other intents you'd like to explore or modify, please let me know!
[Operation] Furthermore, we can expand these scenarios with variations to further enrich the dataset:
1. **Group Bookings** (With Variations):
- **Customer**: "We're planning a wedding and will need about 20 rooms. Can you provide a group rate for us?"
- **Chatbot**: "Congratulations on your upcoming wedding! We offer special packages for events. Could you let me know your wedding date and specific requirements?"
2. **Waiting List** (Alternative Responses):
- **Customer**: "I'd like to book a suite for July 4th, but it appears unavailable. Can I join a waiting list?"
- **Chatbot**: "I apologize for the inconvenience. While we don't have a waiting list for suites, I can notify you if we have any cancellations. Would you like that?"
3. **Language Assistance** (Diverse Language Request):
- **Customer**: "Bonjour, je préfère communiquer en français. Pouvez-vous m'aider?"
- **Chatbot**: "Bonjour! Oui, je peux vous aider en français. Que puis-je faire pour vous aujourd'hui?"
4. **Feedback and Complaints** (Multiple Channels):
- **Customer**: "I'd like to leave feedback on my stay. Where can I do that?"
- **Chatbot**: "Thank you for your interest in providing feedback. You can share your experience directly here, or if you prefer, I can email you our feedback form."
5. **Accessibility Needs** (Specific Requirement):
- **Customer**: "I need a room with visual aid facilities. Is that available?"
- **Chatbot**: "Yes, we have rooms equipped with visual aid facilities. Would you like to proceed with a booking or have more questions?"
6. **Discounts and Promotions** (Different Types):
- **Customer**: "Do you have any loyalty programs or rewards for repeat customers?"
- **Chatbot**: "Yes, we offer a loyalty program with rewarding points for every stay. Would you like more details on how to enroll?"
7. **Emergency Requests** (Handling Documented Policies):
- **Customer**: "Due to unforeseen circumstances, I must cancel my booking. Am I eligible for a refund?"
- **Chatbot**: "I'm sorry to hear that. Our policy allows refunds if canceled 48 hours in advance. Could you provide your booking reference to proceed?"
These variations add depth and cover different aspects of each scenario, enriching the dataset further. Would you like to tweak these or explore additional segments?

Table 26: A dialogue for constructing chatbot for hotel reservation dataset.

[Guidance] To construct a dataset for a hotel reservation chatbot, let's consider what types of interactions typically occur in a reservation conversation. We should start by identifying the key components of the interaction, such as greeting, room inquiry, booking details, confirmation, and closing.
Could you provide some examples of typical questions or requests a customer might make during a hotel reservation chat? This will help us design a diverse and comprehensive dataset that covers all necessary aspects of hotel booking interactions.
[Operation] `extract_information(utterance_from_customer="Hi, I'd like to book a room for two people from March 15th to March 20th.")`
[Program] `{"number_of_people":2,"name_of_representative":"","checkin_date":"March 15th","checkout_date":"March 20th"}`
[Operation] `check_information(number_of_people=2,name_of_representative="",checkin_date="03/15",checkout_date="03/20")`
[Program] `unknown_information_list:["name_of_representative"]` **[Guidance]** Great start! It's important to have various sample interactions to cover different aspects of hotel bookings. Here are some typical questions or requests that the customer might have:
1. **Room Inquiry**: - "What types of rooms do you have available?" - "Do you have any suites?" - "Is there a room available with a sea view?"
2. **Booking Details**: - "I'd like to book a room for two people from March 15th to March 20th." - "Can I add breakfast to my booking?" - "Is there any discount available for a stay longer than 3 nights?"
3. **Special Requests**: - "Can I request a late checkout?" - "Do you have facilities for children?" - "Is the hotel pet-friendly?"
4. **Confirmation & Payment**: - "How do I confirm my booking?" - "What payment methods do you accept?" - "Can I get a refund if I cancel?"
5. **Additional Services or Information**: - "Do you offer airport pickup services?" - "Is there a gym or a pool at the hotel?"
These types of queries will help construct detailed datasets with multiple layers of reservation interaction. To build on these, let's consider various responses a chatbot might give and how to encode those into our dataset efficiently. Would you like to explore this next?
[Operation] Let's continue by developing sample interactions between the customer and the chatbot for each of the types of queries mentioned. This will include both customer queries and possible chatbot responses.
Room Inquiry
Customer: What types of rooms do you have available? **Chatbot**: We offer single rooms, double rooms, deluxe suites, and family suites. Would you like more information on any particular type? **Customer**: Do you have any suites? **Chatbot**: Yes, we have both deluxe suites and junior suites available. Would you like to book one or get more details? **Customer**: Is there a room available with a sea view? **Chatbot**: Yes, we have several rooms with sea views. Would you like to check availability for specific dates?
Booking Details
Customer: I'd like to book a room for two people from March 15th to March 20th. **Chatbot**: Sure! Can I have the name of the representative making the reservation? **Customer**: Can I add breakfast to my booking? **Chatbot**: Certainly! We offer a complimentary breakfast option. Would you like to include it in your reservation? **Customer**: Is there any discount available for a stay longer than 3 nights? **Chatbot**: Yes, we offer a 15% discount on stays longer than 3 nights. Would you like to proceed with this offer?
Special Requests
Customer: Can I request a late checkout? **Chatbot**: Late checkout is available based on request and availability. Please let us know your preferred checkout time. **Customer**: Do you have facilities for children? **Chatbot**: Yes, we have a play area for children and offer babysitting services. Would you like to know more? **Customer**: Is the hotel pet-friendly? **Chatbot**: Yes, we welcome pets and have pet-friendly rooms available. Shall I include a pet-friendly room in your booking?
Confirmation & Payment
Customer: How do I confirm my booking? **Chatbot**: Once you provide your details and preferred payment method, I will confirm your booking and send you a confirmation email. **Customer**: What payment methods do you accept? **Chatbot**: We accept credit and debit cards, PayPal, and bank transfers. Which method would you prefer? **Customer**: Can I get a refund if I cancel? **Chatbot**: Cancellations are refundable if made 48 hours prior to check-in. Would you like further details on our cancellation policy?
Additional Services or Information
Customer: Do you offer airport pickup services? **Chatbot**: Yes, we offer complimentary airport pickup for our guests. Would you like to arrange it for your arrival? **Customer**: Is there a gym or a pool at the hotel? **Chatbot**: Yes, our hotel features a fully-equipped gym and an indoor pool. Would you like to know about our opening hours?
These interactions should cover a broad range of customer inquiries and requests, enabling the construction of a well-rounded dataset for a hotel reservation chatbot.
[Operation] Having established the basic interactions, we can create specific examples of dialogues with the chatbot handling various customer requests. Here's how these might look:
Dialogues for Dataset
Dialogue 1: Room Inquiry
- **Customer**: "What types of rooms do you have available?" - **Chatbot**: "We have single rooms, double rooms, deluxe suites, and family suites. Are you interested in booking a specific type?"
Dialogue 2: Booking Details
- **Customer**: "I'd like to book a room for two people from March 15th to March 20th." - **Chatbot**: "Can I have the name of the representative for this reservation?"
Dialogue 3: Special Requests
- **Customer**: "Can I request a late checkout?" - **Chatbot**: "We can offer a late checkout based on availability. Please specify your desired time."
Dialogue 4: Confirmation & Payment
- **Customer**: "How do I confirm my booking?" - **Chatbot**: "I'll need your payment details, after which I'll confirm your reservation and send a confirmation email."
Dialogue 5: Additional Services
- **Customer**: "Do you offer airport pickup services?" - **Chatbot**: "Yes, we do. Would you like to arrange for an airport pickup at the time of your arrival?"
These examples reflect a diverse set of interactions that can be used to train a chatbot to handle various scenarios in hotel reservations, with each interaction targeting specific needs and responses.
[Guidance] Next, let's consider the different types of variations and user intents that should be included in our dataset. This will ensure that the chatbot can understand and respond to a wide range of input while maintaining effective communication. Here's how we can proceed:
1. **Language Variations**: Capture different ways of saying the same thing. For example:
- "What rooms are available?" vs. "Can you tell me about your rooms?"
2. **Intent Types**: Identify core intents like booking, querying services, making modifications, etc., with specific examples:
- Booking: "I need to reserve a room for two." - Querying: "What amenities do you offer?" - Modifications: "Can I change my booking dates?"
3. **Politeness Levels**: Incorporate varying degrees of formality or politeness.
- Formal: "Could you kindly inform me about your available suites?" - Informal: "Do you have any suites open?"
4. **Edge Cases and Clarification Requests**: Plan for ambiguous or incomplete questions.
- Ambiguity: "I'd like a room." (Needs more specifics) - Clarification: "What do you mean by 'deluxe suite'?"
By designing samples covering these aspects, our dataset will allow the chatbot to handle most, if not all, scenarios gracefully. Let's brainstorm use cases for these variations or start crafting sample interactions based on them. Which aspect would you like to delve into next?
[Operation] Let's delve into crafting sample interactions based on the variations and user intents outlined. We'll create examples for each category to ensure comprehensive coverage.
Language Variations
1. **Inquiry about Room Availability**
- **Customer**: "What rooms are available?"
- **Chatbot**: "We offer a variety of rooms including single, double, and suites. Would you like more details on any?"
- **Customer**: "Can you tell me about your rooms?"
- **Chatbot**: "Our hotel offers single, double, and luxury suites. Do you have a specific type in mind to book?"
Intent Types
1. **Booking**
- **Customer**: "I need to reserve a room for two."
- **Chatbot**: "Certainly! Could you please provide the check-in and check-out dates?"
2. **Querying Services**
- **Customer**: "What amenities do you offer?"
- **Chatbot**: "We offer a gym, pool, spa services, and a restaurant. Do you want to know more about any specific service?"
3. **Modifications**
- **Customer**: "Can I change my booking dates?"
- **Chatbot**: "Of course. Please provide your current booking details and the updated dates."
Politeness Levels
1. **Formal Request**
- **Customer**: "Could you kindly inform me about your available suites?"
- **Chatbot**: "Certainly, we have deluxe and executive suites. Would you like more detailed information?"
2. **Informal Request**
- **Customer**: "Do you have any suites open?"
- **Chatbot**: "Yes, we have some suites available. When would you like to book?"
Edge Cases and Clarification Requests
1. **Ambiguity Handling**
- **Customer**: "I'd like a room."
- **Chatbot**: "Sure, can you please specify the number of guests and your preferred dates?"
2. **Clarification Request**
- **Customer**: "What do you mean by 'deluxe suite'?"
- **Chatbot**: "A deluxe suite includes a spacious room with a king-sized bed, a seating area, and a balcony offering a great view. Is this what you're looking for?"
By incorporating these ideas into our dataset, we provide a robust foundation for training a chatbot to handle a wide array of customer interactions effectively.

Table 27: A dialogue for constructing chatbot for hotel reservation dataset.

Domain	Example
Kinetics	**[Problem]** A car travels 150 km at a certain speed and then 200 km at 10 km/h faster than the initial speed. If the total travel time is 7 hours, what is the initial speed of the car? **[Equations]** $150/x + 200/(x+10) = 7$ **[Answer]** 50 km/h
	[Problem] Two cars start from the same point. One car travels east at 60 km/h, while the other travels north at 80 km/h. How long will it take for them to be 200 km apart? **[Equations]** $x^2 = a^2 + b^2, 60t^2 + 80t^2 = 200^2, 3600t^2 + 6400t^2 = 40000, 10000t^2 = 40000$ **[Answer]** 2.5 hours
	[Problem] A ball is thrown vertically upward with an initial velocity of 20 m/s. How high will it go, and how long will it take to reach its maximum height? **[Equations]** $v_f = v_i + at, h = v_i t + 0.5at^2, v_f^2 = v_i^2 + 2ah, v_f = 0, v_i = 20, a = -9.8$ **[Answer]** 2.04 seconds to reach maximum height
Physics	**[Problem]** In a statics problem, a beam is supported by two pillars at its ends. If the total weight of the beam is 1000 N and is uniformly distributed along its length, what are the forces exerted by each pillar? **[Equations]** $R_1 + R_2 = 1000, R_1 * d_1 = R_2 * d_2$ **[Answer]** $R_1 = 500$ N, $R_2 = 500$ N
	[Problem] Calculate the tension forces in a system of two connected objects with different masses and friction coefficients on an incline. **[Equations]** $m_1 g \sin\theta - T - f_1 = m_1 a, T - m_2 g \sin\theta - f_2 = m_2 a, f_1 = \mu_1 m_1 g \cos\theta, f_2 = \mu_2 m_2 g \cos(\theta), a = (m_1 g \sin\theta - m_2 g \sin\theta - \mu_1 m_1 g \cos\theta) - (\mu_2 m_2 g \cos\theta))/(m_1 + m_2)$ **[Answer]** The tension in each segment is found to satisfy the equations considering all forces acting upon the system, yielding specific numerical results upon solving.
	[Problem] Calculate the stress and strain on different parts of a bridge structure, incorporating environmental factors like wind. **[Equations]** $\sigma = F/A, \epsilon = \Delta L/L_0, \sigma = E\epsilon, C_d = 0.5\rho v^2 A C_d, F_{wind} = C_d \rho v^2 A/2, E =$ (stress/strain) under linear elasticity, $\sigma_{total} = \sigma_{int} + \sigma_{ext}$ (considering wind loads, etc.), Equilibrium equations for bridge: $\sum F_x = 0, \sum F_y = 0, \sum M = 0$ **[Answer]** Calculated stress and strain meet the structure's design criteria, yielding the numerical evaluation from the solved theoretical model.
Chemistry	**[Problem]** Julia needs to create a 10-liter solution with a concentration of 30% acid using two solutions: one that is 20% acid and another that is 50% acid. How many liters of each solution should she mix to obtain the desired concentration? **[Equations]** $x + y = 10, 0.2x + 0.5y = 3$ **[Answer]** 5 liters of 20% solution and 5 liters of 50% solution
	[Problem] Determine the concentration of different solutions required to achieve a particular chemical reaction rate, considering temperature and pressure variables. **[Equations]** rate $= k[A]^x[B]^y, k = Ae^{-Ea/RT}, P_{total} = P_A + P_B + \cdots, d[P]/dt = -k[P]$ **[Answer]** Exact concentrations needed are determined by solving the rate and equilibrium equations, involving all affected conditions.
	[Problem] Balance the chemical reaction: Al + $O_2 \rightarrow Al_2O_3$. **[Equations]** 4Al + $3O_2$ = $2Al_2O_3$ **[Answer]** Reaction is balanced as 4Al + $3O_2$ = $2Al_2O_3$
Finance	**[Problem]** A company produces two products, A and B. The cost to produce product A is $5 per unit, and product B is $8 per unit. The company wants to produce a total of 1,000 units and spend exactly $6,400. How many units of each product should they produce to meet these conditions? **[Equations]** $x + y = 1000, 5x + 8y = 6400$ **[Answer]** 800 units of product A and 200 units of product B
	[Problem] In a small market, Company A supplies a product with the supply equation $P = 10 + 2Q$, and Company B supplies the same product with the supply equation $P = 15 + Q$. The demand equation for the product is $P = 50 - 3Q$. Find the equilibrium price and quantity in the market. **[Equations]** $10 + 2Q = 50 - 3Q, 15 + Q = 50 - 3Q$ **[Answer]** 8
	[Problem] An investor puts part of $1000 into an account earning 5% interest per annum, and the rest into another account earning 7% interest per annum. At the end of the year, the total interest from both accounts is $60. How much did the investor place in each account? **[Equations]** $x + y = 1000, 0.05x + 0.07y = 60$ **[Answer]** $400 at 5% and $600 at 7%.
Geometry	**[Problem]** Find the intersection point of the lines described by the equations $y = 2x + 3$ and $y = -x + 1$. **[Equations]** $y = 2x + 3, y = -x + 1$ **[Answer]** The intersection point is $(-2/3, 5/3)$
	[Problem] In a triangle, the second angle is twice as large as the first angle, and the third angle is 30 degrees more than the first angle. Find the measure of each angle. **[Equations]** $a + b + c = 180, b = 2a, c = a + 30$ **[Answer]** The first angle is 30 degrees, the second angle is 60 degrees, and the third angle is 90 degrees.
	[Problem] In a rectangle, the length is twice the width. If the perimeter is 36 units, find the dimensions of the rectangle. **[Equations]** $l = 2w, 2l + 2w = 36$ **[Answer]** Width = 6 units, Length = 12 units.
Algebra	**[Problem]** Find two numbers such that the sum of the two numbers is 10 and their difference is 4. **[Equations]** $x + y = 10, x - y = 4$ **[Answer]** The numbers are 7 and 3.
	[Problem] Find the values of x, y, and z that satisfy the equations: $2x + 3y - z = 1, 4x - y + 5z = 10$, and $6x + 2y + 3z = 7$. **[Equations]** $2x + 3y - z = 1, 4x - y + 5z = 10, 6x + 2y + 3z = 7$ **[Answer]** $x = 1, y = 2, z = 3$
	[Problem] Solve the system of linear equations: $2x + 3y = 6$ and $4x - y = 5$. **[Equations]** $2x + 3y = 6, 4x - y = 5$ **[Answer]** $x = 1, y = 1$

Table 28: Examples in the constructed mathematical problem solver dataset, grouped by domains.

[Guidance] To construct a robust dataset for solving mathematical problems using systems of equations, let's start by gathering a diverse set of problems. Consider incorporating different types of mathematical challenges across various contexts (e.g., real-world applications, theoretical scenarios) and levels of complexity.
What types of contexts or problem scenarios are you thinking about including to ensure the dataset is comprehensive and representative of a wide range of mathematical problems?
[Operation] To ensure the dataset is comprehensive and representative, the following types of contexts or problem scenarios could be included:
1. **Real-World Applications**:
- Budget planning and financial calculations.
- Distance-rate-time travel problems.
- Mixture problems involving concentrations or alloy compositions.
2. **Theoretical Scenarios**:
- Algebraic puzzles involving unknowns.
- Classic mathematical problems (e.g., calculating ages, work-rate problems).
- Geometry-related problems that involve finding dimensions.
3. **Scientific and Engineering Problems**:
- Calculating forces, velocities, or other physics-related quantities.
- Chemical reaction equations balancing.
- Structural engineering problems, such as load distributions.
4. **Data Analysis and Statistical Problems**:
- Linear regression problems.
- Predictive modeling with constraints.
5. **Business and Economics**:
- Profit maximization given constraints.
- Resource allocation problems.
- Inventory management with supply and demand constraints.
6. **Educational Scenarios**:
- Word problems typically found in textbooks.
- Exam-style questions to model student assessment challenges.
Including a variety of these contexts will help in creating a robust and challenging dataset.
[Guidance] Could you also consider the different levels of complexity we might need to cover? For instance, how might we categorize problems into beginner, intermediate, and advanced levels within these contexts? What characteristics should define each complexity level?
[Guidance] To categorize problems into beginner, intermediate, and advanced levels within these contexts, we can consider the following characteristics for each complexity level:
1. **Beginner Level**:
- Problems typically involve one or two variables.
- Direct and straightforward wording with minimal complexity in translation to equations.
- Solutions generally require basic algebraic techniques, such as substitution or elimination.
- Examples include basic solver problems involving ages, simple financial transactions, or straightforward distance-rate-time calculations.
2. **Intermediate Level**:
- Involves multiple variables (three or more) with a moderate level of intricacy in the relationship among them.
- Problems may require the use of slightly more advanced algebraic techniques such as simultaneous equations or incorporating graphical methods.
- Some word problems might weave in multiple concepts, requiring a step-by-step analysis to form the equations.
- Examples might include more complex mixture problems, multi-step budget calculations, or moderate difficulty physics-based problems.
3. **Advanced Level**:
- Highly complex problems involving several interconnected variables and potentially multiple equations or systems of equations.
- Requires deep analytical thinking to deconstruct the problem and correctly identify all necessary equations.
- Problems could involve higher-level mathematical concepts like non-linear systems, or complex contextual setups such as optimization problems.
- Examples include advanced engineering problems, complex financial modeling scenarios, or high-level competitive mathematical problems.
Defining problems according to these characteristics will help create a dataset with a clear gradient in difficulty, allowing solvers to apply different levels of strategy and problem-solving techniques.
[Operation] Let's create a set of problems that align with the complexity levels for a diverse dataset:
Beginner Level:
1. A person is three times as old as their child. In five years, the person will be twice as old as the child. How old is each person currently?
2. A car travels at 60 miles per hour for 2 hours. How far does it travel?
3. You buy three apples and two bananas for a total cost of $5. Apples cost $1 each. What is the cost of a banana?
Intermediate Level:
1. Two trains travel towards each other from two stations 300 miles apart. Train A travels at 60 mph, and Train B travels at 40 mph. How long does it take for the two trains to meet?
2. A solution contains 30% alcohol and another contains 50% alcohol. How much of each solution needs to be mixed to obtain 100 liters of a solution with 40% alcohol?
3. A company sells two products, A and B. It makes a profit of $5 on product A and $8 on product B. Selling 100 units of both products together yields a total profit of $700. How many units of each product were sold?
Advanced Level:
1. A farmer wants to plant two types of crops on a 100-acre farm. Crop A yields a profit of $200 per acre, while Crop B yields $300 per acre. The total cost of planting these crops should not exceed $10,000, and the farmer wants to maximize profit. How many acres of each crop should be planted?
2. A chemical company needs to produce a 2000-liter solution containing 25% compound X. They have solutions with 10% and 30% concentrations of compound X. What is the volume of each solution needed to achieve the desired concentration?
3. In a 2D coordinate plane, find the point of intersection for the lines represented by the equations $3x + 4y = 10$ and $5x - 2y = 3$. What does the point of intersection represent in a real-world scenario?
These problems span a range of contexts and complexity, suitable for constructing a comprehensive mathematical problem-solving dataset.
[Operation] If you would like, I can proceed to generate systems of equations for some of these problems and solve them to ensure they align with the intended complexity levels. Let me know if you would like me to continue with that process.
[Operation] Let's generate and solve systems of equations for one problem from each complexity level to illustrate the concept:
1. **Beginner Level Problem**:
Problem: A person is three times as old as their child. In five years, the person will be twice as old as the child. How old is each person currently?
2. **Intermediate Level Problem**:
Problem: Two trains travel towards each other from two stations 300 miles apart. Train A travels at 60 mph, and Train B travels at 40 mph. How long does it take for the two trains to meet?
3. **Advanced Level Problem**:
Problem: A farmer wants to plant two types of crops on a 100-acre farm. Crop A yields a profit of $200 per acre, while Crop B yields $300 per acre. The total cost of planting these crops should not exceed $10,000, and the farmer wants to maximize profit. How many acres of each crop should be planted?
Let's start by generating systems of equations for these problems.
[Operation] `generate_system_of_equations(math_problem="A person is three times as old as their child. In five years, the person will be twice as old as the child. How old is each person currently?")`
[Program] `equation_list:["$p = 3 * c$","$p + 5 = 2 * (c + 5)$"]`
[Operation] `generate_system_of_equations(math_problem="Two trains travel towards each other from two stations 300 miles apart. Train A travels at 60 mph, and Train B travels at 40 mph. How long does it take for the two trains to meet?")`
[Program] `equation_list:["Let the time taken for the trains to meet be T hours.","Distance covered by Train A in T hours = Speed of Train A * Time = 60 * T","Distance covered by Train B in T hours = Speed of Train B * Time = 40 * T","Total distance covered by both trains = Distance between stations = 300 miles","Equation: 60 * T + 40 * T = 300"]`
[Operation] `generate_system_of_equations(math_problem="A farmer wants to plant two types of crops on a 100-acre farm. Crop A yields a profit of $200 per acre, while Crop B yields $300 per acre. The total cost of planting these crops should not exceed $10,000, and the farmer wants to maximize profit. How many acres of each crop should be planted?")`
[Program] `equation_list:["Let x be the number of acres planted with Crop A.","Let y be the number of acres planted with Crop B.","Objective Function: Maximize $Z = 200x + 300y$ (Z is the total profit)","Constraint 1: $x + y <= 100$ (Total acres constraint)","Constraint 2: $40x + 50y <= 10000$ (Total cost constraint, assuming $40 per acre for Crop A and $50 per acre for Crop B as an example)","Constraint 3: $x >= 0$ (Non-negativity constraint for Crop A)","Constraint 4: $y >= 0$ (Non-negativity constraint for Crop B)"]`
[Operation] `solve_system_of_equations(system_of_equations=["$p = 3 * c$", "$p + 5 = 2 * (c + 5)$"])`
[Program] `solution_to_equations:[20,6.666666666666667]`

Table 29: A dialogue for constructing mathematical problem solver dataset.

[Guidance] Great! Now that we have established that we will be focusing on difficult mathematical problems that can be solved using systems of equations, let's delve into the details of constructing our dataset.
To start, we need to determine the various types of mathematical problems we want to include. These problems can range in complexity and application. We should consider including problems from different domains, such as physics, economics, or engineering, which naturally lend themselves to systems of equations.
To keep the dataset well-rounded and challenging, let's brainstorm a list of problem types that would be suitable. For instance, we might include rate problems, mixture problems, or optimization problems.
Once we have a list, we can decide on the level of difficulty we aim for in this dataset. Should we have a mix of intermediate and advanced problems, or focus solely on highly advanced topics?
Please share your thoughts or any initial problem ideas you have in mind!
[Operation] Here are some types of mathematical problems that could be suitable for inclusion in the dataset:
1. **Rate Problems**: Problems that involve finding speeds, flow rates, or other rates of change where multiple variables interact linearly.
2. **Mixture Problems**: These involve combining substances with different concentrations and solving for final mixtures.
3. **Optimization Problems**: These could involve maximizing or minimizing some quantity under given constraints, typical in operations research.
4. **Physics-Based Problems**: Such as those involving forces and motion, where multiple equations describe a system's dynamics.
5. **Economic Problems**: Supply and demand modeling, or cost-profit optimization, often require solving systems of equations.
6. **Engineering Problems**: Load distribution analysis in structures, fluid dynamics, or electrical circuit analysis.
7. **Geometric Problems**: Problems involving finding dimensions or angles within complex geometric shapes.
8. **Chemistry Problems**: Balancing chemical equations or reaction rate calculations.
As for the level of difficulty, we could include a mix of intermediate and advanced problems to cater to a wide range of learners while still maintaining the challenging nature desired for the dataset.
[Operation] Now that we have a list of suitable problem types, let's generate data for a specific mathematical problem as a sample for our dataset.
Consider a rate problem:
Problem: Two trains, A and B, are traveling towards each other from towns X and Y, respectively. Train A starts 50 miles away from Y and travels at a speed of 60 miles per hour. Train B starts 60 miles away from X and travels at a speed of 70 miles per hour. At what time will the two trains meet?
Let's proceed to generate a system of equations for this problem, solve it, and finally generate the answer.
[Operation] `generate_system_of_equations(math_problem="Two trains, A and B, are traveling towards each other from towns X and Y, respectively. Train A starts 50 miles away from Y and travels at a speed of 60 miles per hour. Train B starts 60 miles away from X and travels at a speed of 70 miles per hour. At what time will the two trains meet?)`
[Program] `equation_list:["Let t = time in hours when the trains meet.","Distance covered by Train A in time t = 60t miles.","Distance covered by Train B in time t = 70t miles.","The total distance between trains A and B = 50 + 60 = 110 miles.","Equation: 60t + 70t = 110"]`
[Operation] `solve_system_of_equations(system_of_equations=["`$60t + 70t = 110$`"])`
[Program] `solution_to_equations:[0.7857142857142857]`
[Operation] `solution_to_final_answer(solution_of_equations=[0.7857142857142857])`
[Program] `final_answer_to_problem:"The final answer to the problem is 0.7857, rounded to four decimal places."`
[Operation] Here is a sample problem and its resolution for the dataset:
Problem: Two trains, A and B, are traveling towards each other from towns X and Y, respectively. Train A starts 50 miles away from Y and travels at a speed of 60 miles per hour. Train B starts 60 miles away from X and travels at a speed of 70 miles per hour. At what time will the two trains meet?
System of Equations:
1. Let t = time in hours when the trains meet.
2. Distance covered by Train A in time t = $60t$ miles.
3. Distance covered by Train B in time t = $70t$ miles.
4. Total distance between trains A and B = 50 + 60 = 110 miles.
5. Equation: $60t + 70t = 110$
Solution: The time t when the trains meet is 0.7857 hours, rounded to four decimal places.

Table 30: A dialogue for constructing mathematical problem solver dataset.

[Guidance] To construct a robust dataset for solving mathematical problems using systems of equations, let's start by identifying the types of problems we want to include. It's important that these problems are diverse in complexity and context to ensure a comprehensive dataset. Here are a few guiding questions we can consider:
1. **Complexity Levels**: What range of difficulty should the problems span? For example, should we include simple, moderate, and complex problems?
2. **Real-life Contexts**: Should the problems relate to real-life scenarios, such as financial calculations, physics applications, or engineering problems?
3. **Types of Equations**: Are we focusing solely on linear systems, or should we also include non-linear systems of equations?
Let's discuss these aspects and decide on a strategy to gather or create problems that fit these criteria.
[Operation] To start constructing this dataset, let's address each of these aspects:
1. **Complexity Levels**:
- We should include a range of difficulty from simple, moderate, to complex problems.
- Simple problems might involve straightforward calculations with clear relationships.
- Moderate problems could include multiple steps of reasoning or require more than one equation.
- Complex problems should challenge the solver by involving multiple variables and interconnected relationships.
2. **Real-life Contexts**:
- Incorporating real-life contexts can make the problems more engaging and relevant.
- We can include scenarios from finance (e.g., loan interest calculations), physics (e.g., force and motion problems), and engineering (e.g., circuit analysis).
- This also ensures that the problems are practical and relatable.
3. **Types of Equations**:
- Starting with linear systems can provide a foundation, but we should also include non-linear systems to broaden the problem-solving skills.
- Linear systems are easier to solve and are a good starting point for learners.
- Non-linear systems introduce more complexity and require advanced methods for solving.
With these considerations in mind, let's begin by generating mathematical problems within these categories. Once we have these problems, we can create systems of equations for each and eventually solve them.
[Guidance] To systematically approach the creation of our dataset, let's begin by defining specific problem examples for each complexity level and context. Here's our plan:
1. **Complexity Levels and Examples**:
- For **simple problems**, think about situations involving 2-3 variables with clear relationships. Example: "Find the intersection point of two lines given by equations $y = 2x + 3$ and $y = -x + 5$."
- For **moderate problems**, consider scenarios with more depth or additional constraints. Example: "In a rectangle, the length is twice the width. If the perimeter is 36 units, find the dimensions of the rectangle."
- For **complex problems**, incorporate scenarios that need multiple equations and logical reasoning. Example: "In a group of chickens and cows, there are 30 heads and 74 legs. Determine how many chickens and cows are there."
2. **Real-life Contexts and Examples**:
- For financial contexts, create problems about investments or savings: "An investor puts a total of $1000 in two accounts, one earning 5% interest and the other 7%. If the total interest after one year is $62, how much is invested in each account?"
- For physics, base problems on motion or force calculations: "Two cars start from the same point. One travels east at 40 mph, and the other travels north at 30 mph. How far apart are they after 2 hours?"
- For engineering, consider circuit analysis or statics: "In a series circuit with a total voltage of 10V and resistors of 2Ω and 3Ω, find the current flowing through the circuit."
3. **Types of Equations and Examples**:
- Begin with systems that are linear: "Solve the system of equations: $3x + 2y = 6$ and $x - y = 1$."
- Introduce non-linear systems for added challenge: "Solve the system: $x^2 + y^2 = 25$ and $y = 2x$."
Once we have a list of potential problem scenarios and examples, the next step is to create more detailed problem statements in a clear and precise manner. Let's come up with a few problems for each category and share them for feedback. What do you think?
[Operation] Let's generate mathematical problem statements for each of the categories outlined in our plan:
Complexity Levels
1. **Simple Problems**:
- **Example 1**: Two lines are represented by the equations $y = 2x + 3$ and $y = -x + 5$. Find their intersection point.
- **Example 2**: A pen's price is $2 more than a pencil. If 5 pens and 7 pencils together cost $15, find the price of each.
2. **Moderate Problems**:
- **Example 3**: In a rectangle, the length is twice the width. If the perimeter is 36 units, find the dimensions of the rectangle.
- **Example 4**: The sum of two numbers is 10, and their product is 21. Find the numbers.
3. **Complex Problems**:
- **Example 5**: In a group of chickens and cows, there are 30 heads and 74 legs. Determine how many chickens and cows are there.
- **Example 6**: A farmer has goats and chickens. Together, they have 91 heads and 234 legs. How many goats are there?
Real-life Contexts
1. **Financial Contexts**:
- **Example 7**: An investor puts a total of $1000 in two accounts, one earning 5% interest and the other 7%. If the total interest after one year is $62, how much is invested in each account? - **Example 8**: A person deposits $500 in a savings account and $1000 in a checking account. The savings account earns 4% annually and the checking account 2%. If the total interest earned after one year is $40, how much was deposited in each?
2. **Physics Contexts**:
- **Example 9**: Two cars start from the same point. One travels east at 40 mph, and the other travels north at 30 mph. How far apart are they after 2 hours?
- **Example 10**: A ball is projected upwards with a velocity of 20 m/s. Simultaneously, another ball is dropped from a height of 80 m. When and where will the two balls meet in mid-air?
3. **Engineering Contexts**:
- **Example 11**: In a series circuit with a total voltage of 10V and resistors of 2Ω and 3Ω, find the current flowing through the circuit.
- **Example 12**: A beam is supported at two points and subjected to a uniform load. If the reactions at the supports are 20N and 40N, determine the load and its distribution.
Types of Equations
1. **Linear Systems**:
- **Example 13**: Solve the system of equations: $3x + 2y = 6$ and $x - y = 1$.
- **Example 14**: Solve: $5x - 3y = 9$ and $2x + y = 4$.
2. **Non-linear Systems**:
- **Example 15**: Solve the system: $x^2 + y^2 = 25$ and $y = 2x$.
- **Example 16**: Find the points on the curve $y = x^2$ that are 5 units away from the point $(0, 2)$.
These problems reflect a range of complexities, contexts, and equation types. Let me know how to proceed or if you require any modifications or additional examples.
[Operation] If everything seems appropriate, let's create a system of equations for one selected problem from each category using the available tools. We'll start by generating a system of equations for a simple, a moderate, and a complex problem. Here's what we'll do:
1. **Simple Problem (Example 1)**: Two lines are represented by the equations $y = 2x + 3$ and $y = -x + 5$. Find their intersection point.
2. **Moderate Problem (Example 3)**: In a rectangle, the length is twice the width. If the perimeter is 36 units, find the dimensions of the rectangle.
3. **Complex Problem (Example 5)**: In a group of chickens and cows, there are 30 heads and 74 legs. Determine how many chickens and cows are there.
Let's generate systems of equations for these problems.
[Operation] `generate_system_of_equations(math_problem="Two lines are represented by the equations $y = 2x + 3$ and $y = -x + 5$. Find their intersection point.")`
[Program] `equation_list:["$y = 2x + 3$","$y = -x + 5$"]`
[Operation] `generate_system_of_equations(math_problem="In a rectangle, the length is twice the width. If the perimeter is 36 units, find the dimensions of the rectangle.")`
[Program] `equation_list:["$l = 2w$","$2l + 2w = 36$"]`
[Operation] `generate_system_of_equations(math_problem="IIn a group of chickens and cows, there are 30 heads and 74 legs. Determine how many chickens and cows are there.")`
[Program] `equation_list:["$c + h = 30$","$2c + 4h = 74$"]`

Table 31: A dialogue for constructing mathematical problem solver dataset.