# Federated Prompting and Chain-of-Thought Reasoning for Improving LLMs Answering

Xiangyang Liu, Tianqi Pang, and Chenyou Fan

South China Normal University, Guangdong, China {2022024952,2022024954}@m.scnu.edu.cn, fanchenyou@scnu.edu.cn

**Abstract.** We investigate how to enhance answer precision in frequently asked questions posed by distributed users using cloud-based Large Language Models (LLMs). Our study focuses on a typical situations where users ask similar queries that involve identical mathematical reasoning steps and problem-solving procedures. Due to the unsatisfactory accuracy of LLMs' zero-shot prompting with standalone questions, we propose to improve the distributed synonymous questions using Self-Consistency (SC) and Chain-of-Thought (CoT) techniques. Specifically, we first retrieve synonymous questions from a crowd-sourced database and create a federated question pool. We call these federated synonymous questions with the same or different parameters SP-questions or DP-questions, respectively. We refer to our methods as Fed-SP-SC and Fed-DP-CoT, which can generate significantly more accurate answers for all user queries without requiring sophisticated model-tuning. Through extensive experiments, we demonstrate that our proposed methods can significantly enhance question accuracy by fully exploring the synonymous nature of the questions and the consistency of the answers.

**Keywords:** Synonymous Question-answering · Federated Learning · Large Language Model · Prompt Learning · Chain-of-Thought.

## 1 Introduction

Recently, Large Language Models (LLMs) such as PaLM [2] and GPT family [1,13] have revolutionized the methodology of tackling natural language processing (NLP) tasks such as sentiment analysis [20], question answering [24], text summarization [23], and reasoning on arithmetic and common sense questions [25].

Large Language Models (LLMs) are highly over-parameterized with millions or billions of parameters, e.g., the GPT-3 model has about 175 Billion parameters. Due to this redundancy in model design, LLMs can represent language in a highly flexible and expressive manner by capturing the complex and structured patterns in human languages. In addition, LLMs can generate remarkably natural dialogues and accurate answers with contextual understanding, sometimes even surpassing human experts in certain tasks. For example, in arithmetic reasoning, GPT-4 achieved an accuracy rate of 92% on the GSM8K dataset [12]; in common sense reasoning, KEAR achieved an accuracy rate of 89.4% on the CSQA dataset [22].

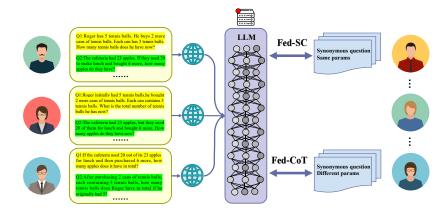


Fig. 1: A general overview of our approach to dealing with federated synonymous question-answering. Our approach is categorized into two user scenarios: synonymous questions that share the same parameters, and those that have different parameters. When the parameters are the same, we utilize self-consistency to select the most commonly voted answer as the consistent response. However, for cases where the parameters are different, we amalgamate each question's consistent answer to create a Chain-of-Thought, which makes it easier for the LLM to respond to new queries.

We consider a practical user scenario in which a large number of users can access a cloud-deployed LLM for solving personal tasks from all places over the world. For example, more and more primary school students and their parents rely on the capability of LLMs for solving mathematical problems. The users often access a LLM and ask common realistic questions. For example, primary school children might ask "Chickens and rabbits are in the same cage, a total of 35 heads, 94 feet, how many chickens and rabbits are there?", while computer science students often ask "How to write a QuickSort in Python?".

Due to the complexity in task understanding and reasoning, the LLMs often return the wrong answers even given seemingly simple questions. For example, on the GSM8K dataset, the fine-tuned GPT-3 (175B) with verifier only achieves an accuracy of 55.0%. Meanwhile, the PaLM-540B (Few-shot-CoT) only achieves an accuracy of 58.1%. [8] How to improve the question answering accuracy has become a serious challenge which decides whether LLMs can be accepted as a robust and reliable part in realistic applications.

One commonsense is that can we crowd-source many questions and aggregate those questions to better understand some common questions. A common question might be asked frequently as its variants in the concrete parameters or rephrased formulations. For example, the Chickens-and-rabbits questions can be asked with different number of heads and feet. Now we want to ask *Can we fully utilize those* 

similar questions to improve the question answering of the LLMs without tuning the model parameters or infringing user privacy?

Recent progressives in federated learning (**FL**) [10] have proved that utilizing distributed data sources can both preserve data privacy and enhance model training. In the FL paradigm, each client trains a local learning model with *own data*, while a central server regularly communicates with all agents to generate a better global model through the aggregation of the local models.

In this study, we consider improving the reasoning capacity of LLMs by better understanding crowd-sourced similar questions, from which we can explore the contextual information and improve the LLM answers substantially. Inspired by FL, we propose two typical scenarios when a user sends a QA request to the LLM and the LLM tries to answer with a collected question database.

- Synonymous Questions with Same Parameters (SP-questions). The cloud-deployed system retrieves from the database and finds several synonymous but rephrased questions with exactly the same parameters. For example, Q1:"If a farmer has a certain number of chickens and rabbits in a barn, and there are a total of 32 heads and 100 feet, how many chickens and how many rabbits does the farmer have?"
  - Q2:"In a barn, there are a certain number of chickens and rabbits that have a total of 32 heads and 100 feet. how many of each animal are in the barn?"
- Synonymous Questions with Different Parameters (DP-questions).
   This situation is harder as the question parameters mined in the database are different from each other. For example,
  - Q1:"If a farmer has a certain number of chickens and rabbits in a barn, and there are a total of 32 heads and 100 feet, how many chickens and how many rabbits does the farmer have?"
  - Q2:"A farmer has a total of 20 chickens and rabbits in his barn. If the total number of legs in the barn is 56, how many chickens and how many rabbits are in the barn?"

For **SP-questions**, we propose to leverage LLMs to directly generate answers first. Then we federate the answers and apply the self-consistency [19] technique to obtain the most voted answer for all synonymous questions in the federation. We call this method **Fed-SP-SC** (Fed-SP with Self-Consistency).

For **DP-questions**, we propose to leverage LLMs to generate consistent answers for each DP-questions first. Different from procedures of dealing SP-questions, we cannot directly agglomerate the answers since they are for different parameters. Instead, we federate them by forming the Chain-of-Thought (CoT) to provide hints to the LLMs. We append the original query to the CoT as the full prompt to obtain improved final answer. We call this technique **Fed-DP-CoT**.

Once the LLM has finished generating the answer using either Fed-SP-SC or Fed-DP-CoT, the system will store both the questions and answers into the database. This enables the system to collect all records and leverage past records to produce re-fined answers to new queries. For questions that have been asked before with wrong answers, the system can evolve itself by correcting the answers with self-consistency mechanism or with more comprehensive CoT prompts.

We extensively evaluate our methods on the GSM8K and SVAMP datasets and demonstrate that the Fed-SP-SC method achieves a notable improvement in accuracy of 14-18% over the standalone LLMs with Zero-Shot-CoT ("Let's think step by step"). Additionally, our Fed-DP-CoT method delivers an impressive increase of 10-15% over the standalone LLMs with Zero-Shot-CoT.

We summarize our contributions in this study as follows.

- 1. We consider a practical but under-studied scenario, which is the cloud-based LLMs are frequently asked similar and even synonymous common questions from large number distributed users.
- 2. We abstract two main user scenarios: distributed users are querying synonymous questions that share the same parameters (SP-questions), and those that have different parameters (DP-questions).
- 3. We design the system to firstly federate those SP- and DP-questions first by retrieving the database. Then we propose to utilize self-consistency methodology to select the most commonly voted answer to improve SP-question answering. All consistent answers and CoTs will be stored back into database for further reuse.
- 4. We also amalgamate consistent answers to create a chain-of-thought prompt that significantly improves DP-questions answering quality. We also design a simple disclaimer to handle noisy CoT generated from LLM answers better.
- 5. Inherited from Federated Learning, our Fed-SP-SC and Fed-DP-COT methods can collaboratively enhance the question-answering process of the LLM while preserving their anonymity. There would be no data exchange or leakage among distributed users.

#### 2 Related Work

Pre-trained Language Models (PLMs). Recent studies in Transformer-based language models such as ELMo [15] and BERT [4] have shown their capabilities in scaling up model sizes with pre-training methodology such as Masked Language Modeling [4]. Shortly after, several Large Language Models (LLMs), e.g., the GPT family [1,13], PaLM [2], Jurassic-X [9], Megatron-Turing [16], LaMDA [17], LLaMA [18], have been emerging with huge amount of parameters of up to 100B-5000B parameters. They have shown great advantages in language modeling tasks, such as arithmetic reasoning, commonsense reasoning, symbolic reasoning and natural language inference.

However, PLMs are still like black boxes which lack of explanation. Some recent studies made efforts towards unveiling the power of those LLMs. The proposal of the concept of the *Chain-of-Thought* (CoT) [21] indicates that incorporating intermediate reasoning steps can lead to a significant improvement in the performance of large language models on reasoning tasks. The proposal of the *Self-consistency* [19] suggest that aggregating multiple reasoning paths, rather than relying on greedy decoding, can lead to further improvements in the accuracy of models on reasoning tasks. LMSI [7] provides a demonstration of how

large language models can achieve self-improvement by utilizing only unlabelled datasets.

However, it is unknown how to apply proper pre-training to distributed learning scenarios, due to substantial differences between centralized large model deployment and distributed query demands. In this study, we adopt the recent popular distributed machine learning methodology called *Federated Learning* [5, 6, 10, 26] (FL) to fully explore the potentiality of Large Language Models to tackle frequently asked questions while preserving data privacy for the users. The FL provides a way of learning models over a collection of distributed devices while keeping data locality. However, classical FL studies assumed the agents in FL can own copies of local models while receiving updates from centralized model. In contrast, we focus on a practical scenario that the clients can only query answers from centralized Large Language Models without owning any local model, due to the practical situations that Large Language Models are simply too large and computational extensive to be deployed locally.

# 3 Scenarios and Approaches

In this section, we describe the federated scenarios that distributed users query the LLMs with similar (but not exact the same) questions. We identify two types of questions and discuss them in details.

## 3.1 Basic Concepts

Chain-of-Thought (CoT) [21] is a series of generated intermediate reasoning texts that can be added to the original prompts. CoT is proposed for enhancing the capability of language models to perform various reasoning tasks by allowing LLMs to decompose complex problems into intermediate steps that could be solved well step-by-step. Chain-of-thought prompting, i.e. prompting LLMs with CoT, is a simple and practical method for improving the reasoning tasks readily with no additional efforts of tuning the original LLMs. CoT prompting has shown improved reasoning results on arithmetic, commonsense, and symbolic reasoning tasks.

Self-Consistency (SC) [19] is a decoding strategy that enhances language model reasoning with voting ensemble. SC first samples a diverse set of answers as reasoning paths of a question, rather than only the greedy path. By exploring multiple paths, SC is capable of identifying the most consist answer as the final answer by majority voting, i.e., the most voted answer of the LLM is taken as the final answer. Compared with a single-path reasoning, SC ensembles answers to improve accuracy and filters out noises or outliers. SC has also been widely explored in reasoning and QA tasks [19].

Majority voting(MV) [11] is a commonly used method in statistical decision theory that involves aggregating the opinions or decisions of multiple individuals or models, typically by selecting the option with the highest frequency of agreement among the voters.

### 3.2 Synonymous Questions with Same Parameters (SP-questions)

We consider a cloud-based LLM system which accepts queries from distributed users. The first practical user scenario that we consider is as follows. Given a user query, we can retrieve from the cloud database several synonymous questions with same parameters (SP-questions).

Table 1: Examples of synonymous SP-questions and answers.

#### Example1:

Q1: "If a farmer has a certain number of chickens and rabbits in a barn, and there are a total of 32 heads and 100 feet, how many chickens and how many rabbits does the farmer have?"

A1: "The farmer has 24 chickens and 8 rabbits." (wrong)

**Q2:** "In a barn, there are a certain number of chickens and rabbits that have a total of 32 heads and 100 feet. how many of each animal are in the barn?" **A2:** "Let x = the number of chickens and y = the number of rabbits. We can set up the following system of equations:x + y = 32(heads), 2x + 4y = 100 (feet), Solving this system of equations, we get x = 20 and y = 12. Therefore, there are 20 chickens and 12 rabbits in the barn." (wrong)

### Example 2:

Q1: "James writes a 3-page letter to 2 different friends twice a week. How many pages does he write a year?"

A1: "James writes 3 pages to 2 different friends twice a week, which is 24 pages a month and 288 pages a year." (wrong)

Q2: "If James writes a 3-page letter to two different friends twice per week, what is the total number of pages he produces every year?"

**A2:** "James writes two 3-page letters twice per week. There are 52 weeks in a year. Therefore, James produces a total of 312 pages every year(2\*3\*52=312)." (wrong)

For complex reasoning tasks, the LLMs may provide unreliable answers to the questions. We provide two failure cases in Table. 1. We found that in both examples, Q1 and Q2 are synonymous while each of them gets a wrong answer from LLM. We summarize the difficulties of tackling the SP-questions as follows:

- 1. Most LLMs have unsatisfying accuracy in solving reasoning problems in zeroshot way, i.e., prompting the LLMs with questions directly without giving other information.
- 2. LLMs tend to under-perform when understanding complex problems involving multiple reasoning steps, such as the arithmetic problems given above.

Thus, our task is to fully explore the SP-questions as a federation which can enhance the answer quality together, instead of dealing them separately. To this

end, we propose a technique named **Fed-SP-SC** (Federated SP-questions with Self-Consistency) for answering the questions with the self-consistency technique mentioned above. Fed-SP-SC can improve the zero-shot accuracy of the answers by eliciting answers from multiple synonymous questions and make a majority vote to disclose the most likely answer.

Concretely, we query the database using the user's prompt to match SPquestions in the database. Note that here we assume we can retrieve the SPquestions which are just rephrased with synonymous and same parameters.

Next, we generate the answers with LLMs by zero-shot prompting. For SP-questions, these answers are presumably same. Assuming that we have generated a total of n answers of synonymous questions during the Fed-SP-SC process, we can ensure the consistency with SC procedure, i.e., we make a majority vote and select the most voted answer  $A^{SC}$  as the final answer of all SP-questions, as below.

$$A^{SC} \leftarrow \underset{A \in \mathcal{A}}{\operatorname{arg\,max}} \sum \mathbf{1}[A == A_i], \quad \forall i = 1, ..., n . \tag{1}$$

Intuitively, the majority voting filters out outliers and noisy rephrased questions. In addition, the most voted answer is the agreement of multiple reasoning paths from multiple rephrased SP-questions, thus is more likely to be better than a single prompted answer decoded from a single reasoning path.

In our experiments, we demonstrate that Fed-SP-SC achieves a 17.5% improvement in accuracy on the GSM8K dataset and a 14% improvement on the SVAMP dataset in Table 3. In a practical system, we can further store these user prompts and the SC-selected answer back into the database.

#### 3.3 Synonymous questions with Different Parameters (DP-questions)

We now describe the second scenario which is named **synonymous questions** with different parameters (**DP-questions**), which is broader and more practical. Based on the user query question, the cloud-deployed system searches and retrieves from the database for questions with same meanings but may have different parameters.

DP-questions are more practical yet harder than SP-questions as the question parameters retrieved from the database are different. We show two exemplary questions Q1 and Q2 below which have the same meaning yet with different parameters *heads* and *feet* in Table 2.

Note that tackling DP-questions would face all the difficulties of SP-questions, and would have additional obstacles as summarized below:

- 1. There is no uniform ground-truth for DP-questions in the database, since each one has different parameters.
- 2. Similarly, we cannot apply self-consistency (SC) directly to improve accuracy due to different parameters.
- 3. If we apply Chain-of-Thought (CoT) together to the original questions as enhanced prompts, we cannot guarantee the correctness of the CoT. Incorrect CoT may even harm the answering accuracy.

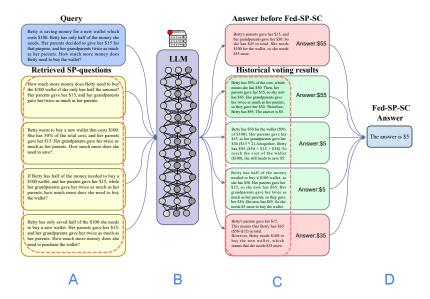


Fig. 2: The illustration of performing Fed-SP-SC for answering synonymous SP-questions.  $(A \to B)$ : When receiving the user's query, the LLM retrieves synonymous SP-questions from the centralized database.  $(B \to C)$ : The LLM generates the answers with zero-shot prompting for the query and combines the retrieved SP-questions' answers from the database for a majority vote to ensure self-consistency.  $(C \to D)$ : The most voted answer is returned to the user as the best answer. The database could store the query and answer pair back to the database, caching for later retrieval. This procedure can grow the database quickly by gathering distributed user queries.

Table 2: Two examples of DP-questions. Q1 and Q2 are synonymous but with different question parameters.

The specific description of Q1 and Q2:

Q1: "If a farmer has a certain number of chickens and rabbits in a barn and, there are a total of 32 heads and 100 feet, how many chickens and how many rabbits does the farmer have?"

Q2: "A farmer has a total of 20 chickens and rabbits in his barn. If the total number of legs in the barn is 56, how many chickens and how many rabbits are in the barn?"

To tackle the above challenges, we propose the **Federated questions of Different Parameters with Chain-of-Thought** (Fed-DP-CoT) technique to

leverage existing answers of DP-questions in CoT forms to improve new query answering.

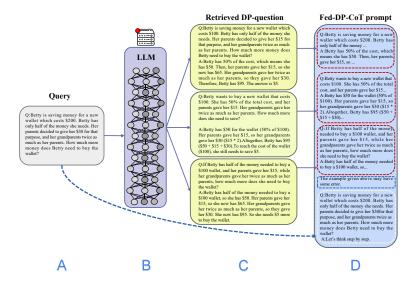


Fig. 3: The illustration of performing Fed-DP-CoT.  $(A \to B)$ : DP-questions are firstly retrieved from the centralized user query database.  $(B \to C)$ : The system selects SP-questions with consistent answers after applying Fed-SP-SC as DP-questions and retrieves consistent answers as "pseudo-labels".  $(C \to D)$ : The system concatenates questions and pseudo-labels, adds a pseudo-label disclaimer such as "The examples may have errors." after, and finally appends the original user query and Zero-Shot-CoT to form the complete CoT prompt.

When a user starts querying the cloud-based LLM service, the cloud system performs query-question retrieval first. The system matches several questions with the highest similarity in the database. Generally, these the retrieved questions are of different parameters (DP-questions). We design the system to perform Fed-DP-CoT for understanding DP-questions.

We consider a practical case that these DP-questions have *pseudo-labels* generated by self-consistency majority voting in the Fed-SP-SC processes. We call these labels "pseudo-labels" as they are not actual ground-truth labels.

Then we utilize these DP-questions with pseudo-labels together as CoT for the original query-question. To be specific, we concatenate DP-questions with their answers as a single prompt, followed by the error disclaimer "The examples given above may contain errors , please think more carefully." at the end of this prompt as the complete prompt. The error disclaimer reminds the LLMs that the answers in CoT are pseudo-labels and could be incorrect. We found this simple practice can boost performance by approximately 2%. Finally, we use the entire

disclaimed CoT as a prefix to the user's query prompt for the LLMs to provide the final answer.

# 4 Experiment

We evaluate our proposed **Fed-SP-SC** and **Fed-DP-CoT** methods on benchmark datasets with simulated user scenarios such that SP- and DP-questions are retrieved to improve over standalone question answering.

We compare our methods with Zero-Shot-CoT [1], which refers to adding "Let's think step by step." to prompt as a composite prompt, such as "[Question] A:Let's think step by step."

#### 4.1 Datasets

Grade School Math (GSM8K) is a math dataset with 7,473 training and 1,319 testing examples of grade-school-level word problems [3]. These math problems typically require two to eight calculation steps to arrive at a final answer, as shown in Fig.2. GSM8K is widely used as a benchmark dataset for testing the arithmetic and commonsense reasoning capacities of LLMs [7,8].

Simple Variations on Arithmetic Math word Problems (SVAMP) is a dataset of simple arithmetic math word problems [14] with around 6,000 samples. Each data instance has a short story and a question about unknown quantities. SVAMP provides a benchmark test set for comparing the textual understanding and reasoning abilities of LLMs, which is widely compared in recent studies [14, 19, 21].

In practice, we utilized the OpenAI's API <sup>1</sup> text-davinci-002 and text-davinci-003. We selected text-davinci-003 for the GSM8K dataset as text-davinci-002 performed very poorly. Similarly, we used text-davinci-002 for the SVAMP dataset as text-davinci-003 had an overly high accuracy rate on this dataset.

## 4.2 Results of Fed-SP-SC

As a kind reminder, in the following discussions, **SP-questions** stand for a set of rephrased synonymous questions with same parameters. Differently, **DP-questions** stand for a set of rephrased synonymous questions with different parameters. We now describe the experiment procedures shown in Fig. 4.

[SP-questions generation]. We first generate four SP-questions for each of the original question with both OpenAI GPT-3 [1] and GPT-3.5 [13], respectively. Concretely, we add each original question a same prompt prefix "Rephrase in 4 ways: [ORIGINAL QUESTION]", then we collect the generated answers.

[SP-questions answering]. We query the cloud-deployed LLM for answering rephrased questions generated as above. Specifically, we obtain the improved Zero-Shot-CoT answering with the magic sentence "Let's think step by step" as the prompt.

<sup>&</sup>lt;sup>1</sup> https://platform.openai.com/docs/models

We first examine the performance of our proposed Fed-SP-SC which deals with SP-questions with the Self-consistency technique. We conducted experiments on GSM8K and SVAMP and report the results below.

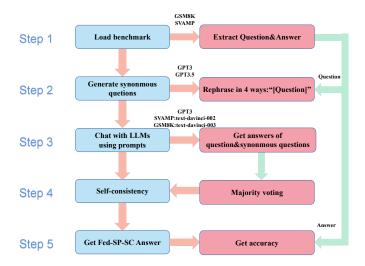


Fig. 4: The experiment of Fed-SP-SC contains five steps: (1) Load the GSM8K and SVAMP datasets as our benchmark and extract the questions and answers in the dataset; (2) Add each original question a same prompt prefix "Rephrase in 4 ways: [QUESTION]" to generate SP-questions; (3) Prompt both the original and rewritten questions to the LLMs to obtain their respective answers; (4) Use the majority vote for self-consistency; (5) Get the answer generated by Fed-SP-SC and compare it with the answer in the dataset to determine the accuracy rate.

Table 3: Fed-SP-SC results

Data\Method	Zero-Shot-CoT	Fed-SP-SC (GPT-3 Gen.)	Fed-SP-SC (GPT-3.5 Gen.)
GSM8K SVAMP	52.5% $77.2%$	$62.7\% \\ 86.3\%$	70.6% $91.1%$

We show accuracy of self-consistency after obtaining results from different phrasings of the synonymous question on GSM8K and SVAMP in Table 3. We have the following observations.

- Fed-SP-SC can improve answering accuracy of LLMs by federating multiple SP-questions through self-consistency.
- Fed-SP-SC(GPT-3.5 Gen.) performs best on the GSM8K and SVAMP datasets, improved the performance by 17.5% and 14% on the GSM8K and SVAMP datasets, respectively.
- The quality of the synonymous questions can affect the accuracy significantly, as seen in the larger improvement from the synonymous questions generated by GPT-3.5 compared to GPT-3.

#### 4.3 Results of Fed-DP-CoT

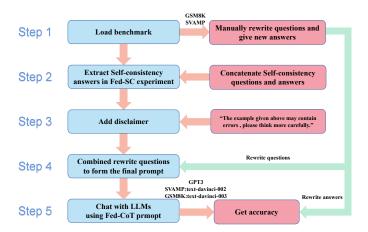


Fig. 5: The experiment of Fed-DP-CoT contains five steps:(1) Form a new test set by rephrasing the questions using different parameters in the benchmark manually and providing answers; (2) Extract consistent questions and answers in Fed-SP-SC experiment; (3) Add a disclaimer to form the CoT prompt; (4) Add the rephrased questions after the CoT prompt; (5) Prompt LLMs with entire CoT prompt and compared the answers with the rephrased answers for evaluation.

We report results of Fed-DP-CoT on GSM8K and SVAMP in Table 4, and compare with the baseline Zero-Shot-CoT.

 Fed-DP-CoT can improve the performance. Compared to Zero-Shot-CoT, CoT Prompt(GPT-3 Gen.) and CoT Prompt(GPT-3.5 Gen.) improve by approximately 10.9%-14.2% and 6.6%-10% respectively on the datasets GSM8K and SVAMP.

Setting\Method	Zero-Shot-CoT	Fed-DP-CoT (GPT-3 Gen.)	Fed-DP-CoT (GPT-3.5 Gen.)		
GSM8K SVAMP	$oxed{48.3\%} 76.5\%$	59.2% 82.4%	$62.5\% \ 85.7\%$		

Table 4: Fed-DP-CoT results.

- Fed-SP-SC performs better than Fed-DP-CoT. The results of Fed-SP-SC (GPT-3 Gen.) and Fed-SP-SC (GPT-3.5 Gen.) on the GSM8K and SVAMP datasets are both higher than Fed-DP-CoT (GPT-3 Gen.) and CoT Prompt (GPT-3.5 Gen.), with an approximate improvement of 5%.
- Less significant performance difference between GPT-3 Gen. and GPT-3.5
  Gen. compared to Fed-SP-SC experiment. The reason for this is the disparity
  in parameters employed, coupled with the lack of emphasis on synonym usage
  in the CoT prompt.

#### 4.4 Ablation studies

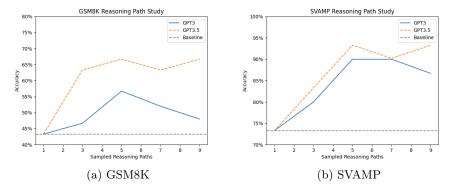


Fig. 6: Ablation study of choice of sampled reasoning paths.

The number of reasoning paths for self-consistency. We study the effect of using different number of sampled reasoning paths for Fed-SP-SC (Sec. 4.2) to apply self-consistency. We conduct hyper-parameter search with a subset of the data for this ablation study due to the limits of accesses of the OpenAI API.

We vary the number of sampled reasoning paths of synonymous questions from one to nine. Figure 6 shows that increasing the number of sampled reasoning paths of the synonymous questions does not always improve the accuracy of the model.

In the line chart, as the number of sampled reasoning paths increases from one to five, the accuracy rate gradually increases. However, when the number of synonymous questions exceeds five, the accuracy of the model starts to decrease.

We speculate that this is because introducing synonymous questions also introduces noisy phrases, causing a deviation in the semantic meaning of the original questions. This deviation is particularly evident in synonymous questions generated by GPT-3 (blue lines), while the generation results of GPT-3.5 (orange lines) exhibit stronger robustness.

Table 5: GSM8K disclaimer ablation.

Method\Setting	Zero-shot -CoT	Fed-DP-CoT (GPT-3 Gen.)	Fed-DP-CoT (GPT-3.5 Gen.)			
w/o disclaimer w/ disclaimer	48.3% NA	$57.7\% \ 59.2\%$	$60\% \\ 62.5\%$			

Disclaimer We investigate whether the disclaimer is effective of correcting noisy CoTs in this ablation experiment. As Zero-Shot-CoT does not employ pseudo-labels, we do not conduct disclaimer ablation on it. Table 5 compares the DP-questions answering accuracy with disclaimer or without disclaimer. We observe that the addition of a disclaimer in the questions and answers generated by GPT-3 resulted in an increase in accuracy from 57.7% to 59.2% for the Fed-DP-CoT task. Similarly, in the case of questions and answers generated by GPT-3.5, the accuracy increase from 60% to 62.5%. These results indicate that the use of a simple disclaimer can potentially improve the accuracy of LLMs by approximately 2% for the Fed-DP-CoT task. We postulate that the improvement in accuracy may be attributed to the fact that the disclaimer prompts LLMs to be careful of the pseudo-labels and self-examine the reasoning steps.

#### 5 Conclusion

We investigate the potential benefits of employing synonymous queries from distributed users to enhance question-answering beyond what is achievable by a single user. Specifically, we explore the use of such queries in a federated manner by extracting two common user scenarios whereby the cloud database retrieves either SP- or DP-questions. To address these scenarios, we propose the application of self-consistency to identify the most consistent answers for SP-questions and utilize them as CoT to improve the answers provided for DP-questions. Our experimental results demonstrate that this approach yields a significant boost in performance compared to standalone zero-shot QA.

Moving forward, future research may investigate the implementation of more realistic systems that can efficiently retrieve federated questions while also improving CoT correctness to further advance reasoning capabilities. In this study, we assumed the DP-questions have already been stored with their answers generated by LLMs, and the consistent answers have been generated. Future work can further extend to scenarios that part of the DP-questions have no answers or pseudo-answers.

### References

- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems 33, 1877–1901 (2020)
- 2. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
- 3. Cobbe, K., Kosaraju, V., et al.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
- Fan, C., Hu, J., Huang, J.: Private semi-supervised federated learning. In: IJCAI. pp. 2009–2015 (2022)
- Fan, C., Huang, J.: Federated few-shot learning with adversarial learning. In: 2021
  19th International Symposium on Modeling and Optimization in Mobile, Ad hoc,
  and Wireless Networks (WiOpt). pp. 1–8. IEEE (2021)
- 7. Huang, J., Gu, S.S., Hou, L., Wu, Y., Wang, X., Yu, H., Han, J.: Large language models can self-improve. arXiv preprint arXiv:2210.11610 (2022)
- 8. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. arXiv preprint arXiv:2205.11916 (2023)
- 9. Levine, Y., Dalmedigos, I., Ram, O., Zeldes, Y., Jannai, D., Muhlgay, D., Osin, Y., Lieber, O., Lenz, B., Shalev-Shwartz, S., et al.: Standing on the shoulders of giant frozen language models. arXiv preprint arXiv:2204.10019 (2022)
- McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS (2017)
- Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference. p. 305–320 (2014)
- 12. OpenAI: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155 (2022)
- Patel, A., Bhattamishra, S., Goyal, N.: Are NLP models really able to solve simple math word problems? In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 2080–2094 (2021)
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer,
   L.: Deep contextualized word representations. In: ACL. pp. 2227–2237 (2018).
   https://doi.org/10.18653/v1/N18-1202
- Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., Catanzaro, B.: Megatron-lm: Training multi-billion parameter language models using model parallelism. arXiv preprint arXiv:1909.08053 (2019)

- 17. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.T., Jin, A., Bos, T., Baker, L., Du, Y., et al.: Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239 (2022)
- 18. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models (2023)
- Wankhade, M., Rao, A.C.S., Kulkarni, C.: A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review 55(7), 5731–5780 (2022)
- 21. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903 (2022)
- 22. Xu, Y., Zhu, C., et al.: Human parity on commonsenseQA: Augmenting self-attention with external attention. arXiv preprint arXiv:2112.03254 (2022)
- Yadav, D., Desai, J., Yadav, A.K.: Automatic text summarization methods: A comprehensive review. arXiv preprint arXiv:2204.01849 (2022)
- Zaib, M., Zhang, W.E., Sheng, Q.Z., Mahmood, A., Zhang, Y.: Conversational question answering: A survey. Knowledge and Information Systems 64(12), 3151– 3195 (2022)
- 25. Zhao, W.X., Zhou, K., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
- Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)