CONVFINQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering

Zhiyu Chen[1], **Shiyang Li**[1], **Charese Smiley**[2], **Zhiqiang Ma**[2], **Sameena Shah**[2] and **William Yang Wang**[1]
[1]University of California, Santa Barbara
[2]J.P. Morgan
{zhiyuchen,shiyangli,william}@cs.ucsb.edu,
{charese.h.smiley,zhiqiang.ma,sameena.shah}@jpmchase.com

Abstract

With the recent advance in large pre-trained language models, researchers have achieved record performances in NLP tasks that mostly focus on language pattern matching. The community is experiencing the shift of the challenge from how to model language to the imitation of complex reasoning abilities like human beings. In this work, we investigate the application domain of finance that involves real-world, complex numerical reasoning. We propose a new large-scale dataset, **CONVFINQA**, aiming to study the chain of numerical reasoning in conversational question answering. Our dataset poses great challenge in modeling long-range, complex numerical reasoning paths in real-world conversations. We conduct comprehensive experiments and analyses with both the neural symbolic methods and the prompting-based methods, to provide insights into the reasoning mechanisms of these two divisions. We believe our new dataset should serve as a valuable resource to push forward the exploration of real-world, complex reasoning tasks as the next research focus. Our dataset and code is publicly available[1].

Financial report:
… the total income tax benefit recognized for share-based compensation in the accompanying statements of income is also presented.

	2010	2009	2008
share-based compensation cost	$18.10	$14.60	$13.80
income tax benefit	-$6.30	-$5.20	-$4.90

Conversational QA:
Q1: In the year of 2010, what was the share-based compensation cost?
A1: 18.1

Q2: and what was the income tax benefit?
A2: -6.3

Q3: what was, then, the sum of both?
A3: **add**(18.1, -6.3) = 11.8

Q4: and what was that sum in 2009?
A4: **add**(14.6, -5.2) = 9.4

Q5: what, then, was the change in the sum of those amounts from 2009 to 2010?
A5: **add**(18.1, -6.3), **add**(14.6, -5.2), **subtract**(#0, #1) = 2.4

Figure 1: An example from **CONVFINQA**: each question may depend on previous questions to answer.

1 Introduction

The rapid advancement in developing large pre-trained language models (LM) has brought the natural language processing research into a new era. Based on the well-known transformer (Vaswani et al., 2017) architecture, such large pre-trained LMs (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020; Sanh et al., 2021; Wang et al., 2022) have set up new state-of-the-art results for many NLP tasks, with some of them approaching or even surpassing human performances, like on the SQuAD (Rajpurkar et al., 2016) dataset. We observe that the tasks with the essence of modeling language patterns can be well addressed by large pre-trained LMs. However, for the other kind of tasks requiring complex reasoning abilities, current researches are still away from satisfactory performances (Wei et al., 2022).

Traditional methods on reasoning tasks typically take neural symbolic models to encode the context, generate the reasoning program and do the execution (Liang et al., 2017; Chen et al., 2020). Most recently, it is shown that sufficiently large pre-trained LMs can excel at reasoning tasks given proper prompts (Wei et al., 2022). But their tasks being experimented with are relatively general and toy, such as simple math word problems. The form of the solutions and the reasoning explanations probably have been witnessed by the model during pre-training. This raises an interesting question: Which of the two directions is the fundamental way

[1]https://github.com/czyssrs/ConvFinQA

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6279 - 6292
December 7-11, 2022

to solve complex reasoning problems?

In this work, we go beyond the simple reasoning tasks and dive into the real application domain of finance to investigate the complex numerical reasoning ability of current modeling paradigms. The finance domain bears the natural requirements of realistic, complex numerical reasoning from human labor, such as quantitative analysis of financial reports. We seek to study the real-world scenario of **conversational question answering over financial reports** – investors or analysts would typically ask sequential questions to get insights into the numerical in the reports. The questions require extensive calculations and meanwhile often demonstrate cross dependency, forming the chains of numerical reasoning throughout the conversation.

To this end, we propose a new dataset, **CONVFINQA** (**Con**versational **Fin**ance **Q**uestion **A**nswering), with 3,892 conversations consisting 14,115 questions. To construct the dataset, we design a framework to simulate the conversation flow by decomposition and concatenation of the multi-hop questions from the FinQA (Chen et al., 2021) dataset. We then ask expert annotators to compose the question for each conversation turn based on the simulated conversing flow. Figure 1 shows one example conversation from our dataset. We conduct comprehensive experiments and analyses on our dataset using both the neural symbolic models and the prompting-based methods, and summarize the following insights: (1) Both kinds of approaches (with the execution accuracy less than 70.0%) fall far behind human performance (89.4%). The reasoning chains throughout the conversation pose great challenges for the models to learn when to refer to or discard the conversation history and how to assemble the reasoning path. (2) Though excelling at simple general reasoning tasks, prompting-based methods perform a lot worse for our task (less than 50.0% using GPT-3 175B). They either superficially mimic the given prompts or recall their own knowledge for simple general numerical reasoning. They tend to fail to understand new complex task paradigms for new domains. We believe our new dataset should serve as a challenging and valuable resource for the exploration of real-world, complex reasoning tasks as the next research focus.

2 Related Work

Conversational Question Answering Conversational question answering (ConvQA) (Zaib et al., 2021) has been gaining attentions in recent years. In ConvQA, the users can append multiple questions in addition to the first one to get more information. This also mitigates the need to ask a single complex multi-hop question at one time, making the information-seeking procedure more natural. For previous datasets, SQA (Iyyer et al., 2017) are built by decomposing multi-hop questions based on Wikitables. CSQA (Saha et al., 2018) questions require simple logical operations over knowledge graphs (KGs). CoQA (Reddy et al., 2019) focuses on co-references among the conversation turns to be more human-like. QuAC (Choi et al., 2018) focuses on open-ended, exploratory questions. In contrast, our dataset **CONVFINQA** targets complex numerical reasoning chains among the sequential questions in finance conversations.

Dataset	Size	Mode	Challenge	Domain
SQA	6k	ConvQA	table navigation	general
CSQA	200k	ConvQA	KG reasoning	general
CoQA	8k	ConvQA	co-reference	general
QuAC	14k	ConvQA	open-ended	general
DROP	96k	QA	numerical reasoning	general
MathQA	37k	QA	numerical reasoning	math
FinQA	8k	QA	numerical reasoning	finance
TAT-QA	17k	QA	numerical reasoning	finance
CONVFINQA	4k	ConvQA	numerical reasoning	finance

Table 1: Comparison of **CONVFINQA** with existing datasets.

Numerical Reasoning The numerical reasoning ability is often investigated in the form of question answering. The DROP dataset (Dua et al., 2019) explores simple calculations over texts in the general domain. MaWPS (Koncel-Kedziorski et al., 2016) and MathQA (Amini et al., 2019) focus on generating solutions for math word problems. Recently, Wei et al. (2022) demonstrate that large pre-trained LMs can excel at reasoning tasks given proper prompts with natural language explanations. However, their reasoning tasks are mostly simple and general. In this work, we explore complex numerical reasoning in a highly specialized domain.

Financial NLP Previous work in financial NLP mostly centers on sentiment analysis (Day and Lee, 2016; Akhtar et al., 2017), fraud detection (Han et al., 2018; Wang et al., 2019; Nourbakhsh and Bang, 2019), opinionated QA (Liu et al., 2020), such as the FiQA[2] dataset built based on social media. Most recently, Chen et al. (2021) propose the FinQA dataset with multi-hop numerical reasoning questions based on financial report. TAT-QA (Zhu

[2]https://sites.google.com/view/fiqa/home

et al., 2021) is another QA dataset with a similar focus. In **CONVFINQA**, we seek to construct question sequences in the conversational setting aiming at more natural experiences for real-world usages. Table 1 presents the comparison of our dataset with existing ones.

3 Task Formulation

Given a financial report containing both the textual content T and structured table B, the user asks a sequence of questions $\{Q_i\}_{i=0}^{n}$ where later questions may depend on previous questions to answer. The target is to generate the reasoning program G to be executed to get the answer A to the last question:

$$P(A|T, B, Q_n) = \sum P(G_i|T, B, Q_0, Q_1, ...Q_{n-1}) \quad (1)$$

Where $\{G_i\}$ is all the possible programs to evaluate to the correct answer. We follow the same domain specific language (DSL) in FinQA (Chen et al., 2021) to construct the reasoning programs as a sequence of operation-argument clauses (Appendix A for all operations):

$$\text{op}_1[\text{args}_\mathbf{1}], \text{op}_2[\text{args}_\mathbf{2}]..., \text{op}_n[\text{args}_\boldsymbol{n}] \quad (2)$$

We follow the same evaluation metric as in FinQA, the execution accuracy to evaluate the final execution result and program accuracy to evaluate program equivalence.

4 The CONVFINQA Dataset

4.1 Dataset Construction

The Overview The core challenge of building such a dataset is the construction of a natural, realistic conversational flow – what kinds of questions the queriers may ask and how these questions logically appear in a conversation. We consult financial experts to summarize the following key factors integrating a conversation when querying financial reports: *(i)* The questioner directly queries the surface content. *(ii)* The questioner asks something requiring calculations from the numbers in the report to answer. *(iii)* The questioner asks the above two kinds of questions sequentially to form the conversation, to cumulatively query more information or switch to other aspects.

Directly composing the conversations from scratch involving all the above factors is very heavy and costly. To tackle this challenge, we propose

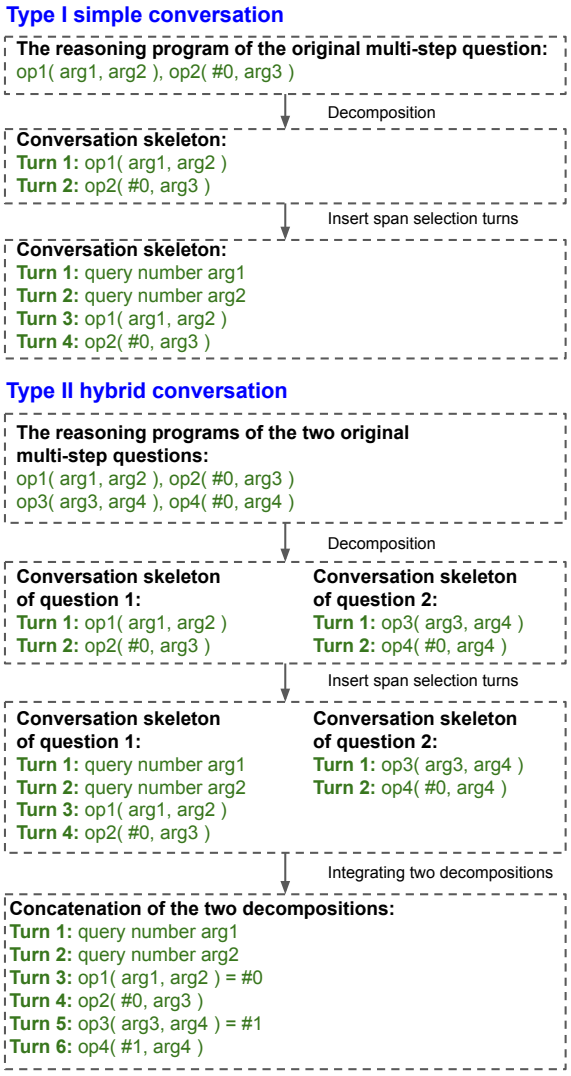

Figure 2: The simulation process of conversation skeletons.

a two-step construction framework: **(I): Conversational QA flow simulation** to produce the conversation skeleton with each turn filled with the reasoning semantics, and **(II): Question composition** to realize the reasoning semantics into textual questions.

Conversational QA Flow Simulation We build the conversation flow based on the decomposition and concatenation of the multi-step reasoning programs (the solutions of the multi-hop questions) in the existing FinQA (Chen et al., 2021) dataset. In FinQA, the authors construct two multi-hop questions for most of its reports. The two FinQA questions for the same report naturally query different but sometimes correlated aspects of the report, inspiring us to integrate them into a natural and realistic conversation. We simulate two types of conversations: **Type I: Simple conversation** from the

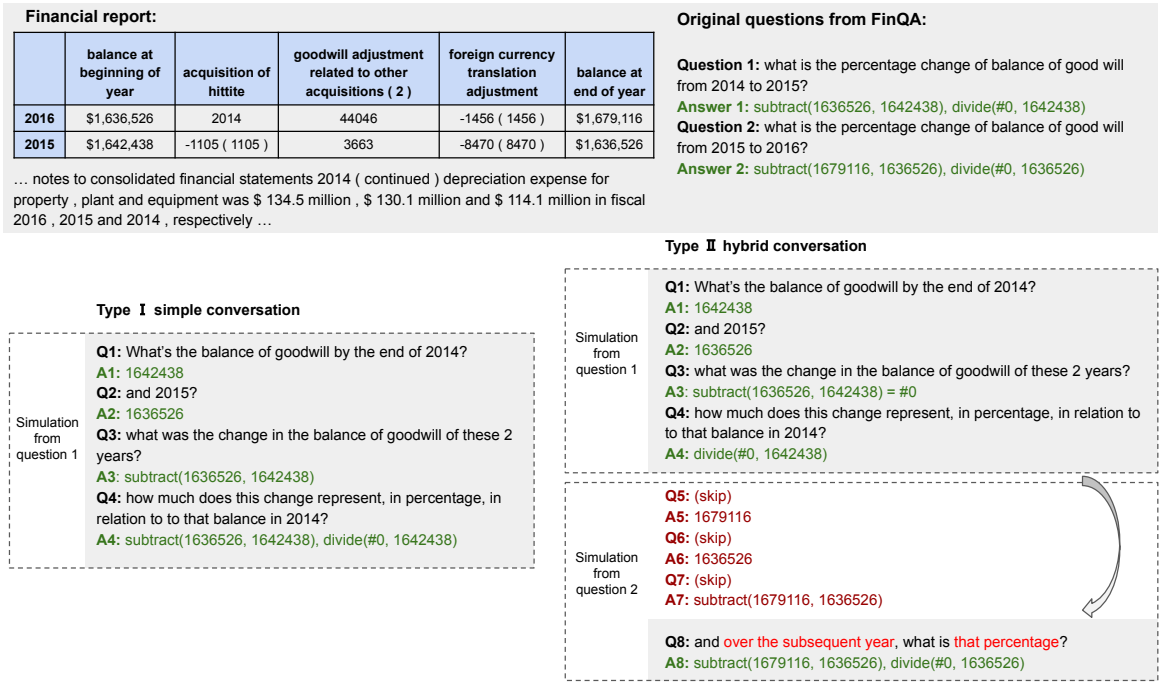

Figure 3: The question composition examples for the two types of conversations. For the hybrid conversation example, the annotator skips three turns and directly jumps to the last turn using references, making the conversation more natural.

decomposition of a single multi-hop question and **Type II: Hybrid conversation** from the decomposition and integration of two multi-hop questions. Figure 2 illustrates the simulation processes of the two types of conversation flows.

For **Type I simple conversations**, we take one muti-hop question and decompose its reasoning program into single steps – each reasoning step will then be realized into one question as one conversation turn. To consider the scenario that the questioner directly queries the surface content, every time there is a new number in a reasoning step, we randomly insert an additional turn before this turn with the semantic to query this new number.

For **Type II hybrid conversations**, we take two multi-hop questions based on the same report, decompose their reasoning programs and insert additional number selection turns similar to the type I conversation. Then we concatenate the decompositions of the two questions to integrate the full conversation skeleton – corresponding to the scenario where the questioner asks two different aspects of the same report. Since the two aspects of the same report often correlate with each other, the conversation flow constructed this way will involve longer dependencies among the turns.

Question Composition After we construct both types of conversation skeletons, we employ expert annotators to realize the skeletons into textual questions. We use the UpWork[3] platform to recruit expert annotators with finance backgrounds, such as CPAs, MBAs, etc. Figure 3 gives the composition examples of the two types of conversations.

Specifically, we present the financial report and the simulated conversation skeletons to the annotators, with each turn filled with the reasoning semantics (the decomposed reasoning program or a single number for the number selection turn). We instruct the annotators to: *(i)* Read the report and understand the reasoning flow of the whole conversation skeleton; *(ii)* Compose questions for the turns based on the given reasoning semantics;

Since our conversation skeletons are simulated, there must be many unnatural scenarios, e.g., unnatural decompositions, redundant or unnecessary turns, etc. Therefore, we emphasize the following key points: *(i)* The annotators can skip some turns and directly jump to a certain following turn with the goal of an overall natural conversation. The key is to *identify redundancies* in the given conversation flow and compress the unnecessary turns using references to the previous context. The right example in Figure 3 shows a scenario to skip unnecessary turns. *(ii)* If there is no way to compose a natural conversation with the given skeleton, the annotators can discard the example. We launch training

[3] www.upwork.com

sessions for the annotators to master task settings before working on the official large batches.

4.2 Dataset Analysis

Dataset Statistics We end up with 3,892 conversations containing 14,115 questions. We split the dataset into 3,037/421/434 for train/dev/test sets. 2,715 of the conversations are simple conversations, and the rest 1,177 are hybrid conversations. Table 2 summarizes the general statistics of our dataset.

In our **CONVFINQA** dataset, the major challenge is to learn the chain of numerical reasoning throughout the conversation turns. First, we sample 200 turns from our dataset and ask the expert annotators to count the longest dependency distance to answer the current question, i.e., how many previous questions need to be seen to answer the current one. Figure 4 shows the result distributions. Second, in **CONVFINQA**, we build two types of conversations – the simple conversation from the decomposition of one FinQA question and the hybrid conversation from the decompositions and concatenation of two FinQA questions. We are interested to see, for the second type of hybrid conversations, how the question set from the second FinQA question makes references to the first one. We split the hybrid conversations into the two sets – one from the first source FinQA question and one from the second, and ask the expert annotators to decide whether any questions from the second set depend on the questions from the first set to answer. Among 200 samples, 65.0% of them depend on the first question set to answer, which demonstrates the challenging reasoning chains in our dataset – the model may need to construct the reasoning chains crossing different aspects and long-range. At last, we also classify the type of questions based on the reasoning forms of the answers. 34.73% of the questions are number selection questions, 35.10%, 25.41%, and 4.75% of them have reasoning programs of 1, 2, and over 3 steps, respectively.

For the type of questions, 59.18% of the questions rely on supporting facts only from the table to answer, 25.56% of the questions rely on supporting facts only from the text, and the rest 15.26% rely on both. For the types of calculations, we have around 18.80% additions, 40.49% subtractions, 6.92% multiplications, 33.43% divisions.

Data Quality Assessment To evaluate the quality of **CONVFINQA** and establish human performance references, we sample 200 example questions and distribute to both the expert and laymen annotators to answer. The two expert annotators reach an average execution accuracy of 89.44% and program accuracy of 86.34%, with an agreement rate over 85.0% for both metrics. For the laymen performance, we distribute the samples to MTurk[4] and end up with an execution accuracy of 46.90% and program accuracy of 45.52% with agreement rates lower than 60.0%. This again demonstrates the great expertise required to solve our dataset.

Conversations	3,892
Questions	14,115
Report pages	2,066
Vocabulary	20k
Avg. # questions in one conversation	3.67
Avg. question length	10.59
Avg. # sentences in input text	23.65
Avg. # rows in input table	6.39
Avg. # tokens in all inputs (text & table)	675.61
Max. # tokens in all inputs (text & table)	2338.00

Table 2: Statistics of **CONVFINQA**.

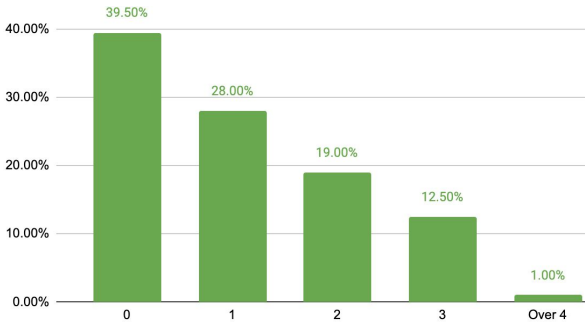

Figure 4: Distribution of the longest dependency distances of the questions in **CONVFINQA**. Over 60% of the questions have longer dependencies with previous questions.

5 Experiments on Neural Symbolic Approaches

In this section, we will first experiment with traditional neural symbolic approaches using the full training data and make detailed analyses.

5.1 Methods and Main Results

We take the FinQANet model from (Chen et al., 2021) and two generative models – the GPT-2 (Radford et al., 2019) and T5 (Raffel et al., 2020). FinQANet is a pipeline approach with a retriever to first retrieve the supporting facts from the financial report, then a generator taking the supporting

[4]Three built-in worker qualifications are used: HIT Approval Rate (≥95%), Number of Approved HITs (≥ 1000), and Locale (US Only) Qualification. We do not select any professional constraints. We pay $2.0 for each question.

Baselines	Exe Acc	Prog Acc
GPT-2(medium)	58.19	57.00
T-5(large)	58.66	57.05
FinQANet (BERT-base)	55.03	54.57
FinQANet (BERT-large)	61.14	60.55
FinQANet (RoBERTa-base)	64.95	64.16
FinQANet (RoBERTa-large)	**68.90**	**68.24**
FinQANet-Gold (RoBERTa-large)	77.32	76.46
Human Expert Performance	89.44	86.34
General Crowd Performance	46.90	45.52

Table 3: The execution accuracy (Exe Acc) and program accuracy (Prog Acc) for the models. We also experiment with using gold supporting facts, shown as FinQANet-Gold.

Methods	Exe Acc	Prog Acc
full results	68.90	68.24
Number selection turns		
Number selection questions	82.54	82.34
Program questions	62.14	61.26
Simple & hybrid conversations		
Simple conversations	72.37	72.00
Hybrid conversations	60.99	59.70
Hybrid conversations (first part)	68.11	66.54
Hybrid conversations (second part)	52.38	51.43

Table 4: Performance breakdown. The number selection questions are the easiest to answer. The hybrid conversations are harder than simple conversations, while the second part of them is even more difficult.

facts and the question as the input to decode the reasoning program. Structural information and constraints are also involved in the decoder. We adopt the same retrieving process from FinQANet and use the current conversation context, i.e., the questions up to the current turn, to retrieve the evidences from the input financial report. We end up with the retrieval results of 86.38% recall for the top 3 retrieved facts. For the program generation, we concatenate the retrieved facts with the conversation context as the input. We experiment with the encoder varied as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Table 3 shows the overall experiment results on **ConvFinQA**. Using a specially designed encoder-decoder with structural preservation of the program, FinQANet still outperforms the standalone generative models. While there is still a gap till the expert performance, the models already surpass the laymen performance. We can see that such neural approaches specially designed can learn better numerical reasoning ability for the specific domain than the *common sense numerical reasoning ability* of the general crowd.

5.2 Performance Breakdown

To gain a deeper understanding of the model insights, we analyze the performances of different types of questions. The results are shown in Table 4. We can see that the number selection turns are the easiest to answer. Considering different types of conversations, the hybrid conversations are harder to learn than simple conversations, especially the second part of the hybrid conversations where the question set comes from the decomposition of the second multi-hop question. In these questions, some of them are irrelevant to the questions in the first part, while some of them depend on the questions from the first part to answer. The model faces a stronger challenge of finding the correct reasoning chains. We also look into the performance breakdown by conversation turns, which is shown in Figure 5. Later turns in the conversations tend to be harder to answer due to longer reasoning dependencies.

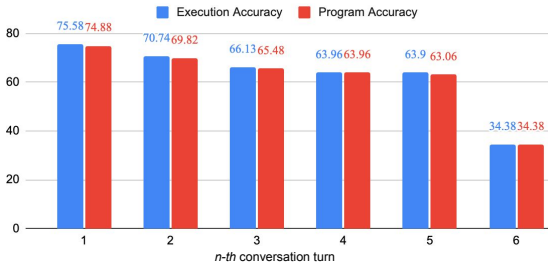

Figure 5: Performances for the *nth* conversation turn.

5.3 Analyses and Findings

We manually analyze a sample of the predictions from the FinQANet(RoBERTa-large) model and summarize the following findings:

The model excels at number selection questions. For the number selection questions depending on previous references, e.g., *what is that value in the subsequent year?*, the model is mostly able to answer. Also, the model is mostly clear on when to discard the previous context and make the transition to new questions.

The model suffers from the lack of domain knowledge. The lack of financial knowledge leads to many errors of missing retrieval facts, wrong value selections, and wrong mathematical

generations. Nonetheless, the current large pre-trained models do see financial corpus during pre-training; we still need to endow the system with stronger domain knowledge for tasks requiring high-level, complex domain reasoning abilities.

The model struggles with long reasoning chains. For the later question turns in a conversation that demonstrate longer reasoning dependencies to the previous context, the model often struggles with deducting the correct reasoning programs. If the prediction for any turn is wrong, then there is a very minor chance that the subsequent turns are correct. We provide two error case studies in Figure 6.

6 Experiments on Prompting-Based Approaches

In this section we attempt on few-shot learning with prompting-based methods and reveal the insights.

6.1 Methods and Main Results

We use the GPT-3 text-davinci-002 model[5] (Brown et al., 2020). Directly injecting the full financial report into the prompt is not realistic because of the length constraint. Therefore we still attempt the retriever-generator paradigm. Due to the high cost of using GPT-3, in this work we only run retrieval on a sample of the test set, and run program generation on the full test set using the gold retrieval results as the input. Nonetheless, we believe our experiments are sufficient to show many interesting and valuable insights into the prompting-based methods on **ConvFinQA**.

For the retrieval, we concatenate each sentence or linearized table row of the report with the conversation context, and let the model predict if the former is relevant for answering the last question. We use 16 exemplars and run GPT-3 on a sample of 300 examples of the test set. We end up with an average recall of 74.25% using 3 different sets of exemplars, which is much lower than the retriever trained with the full training data in §5.1.

For the program generation, the exemplar is formatted as `[supporting facts, conversation context, result to be generated]`. We experiment using the following settings: **(i) Answer-only**, to directly generate the execution results. **(ii) Program-original**, to generate the reasoning program with the original DSL. **(iii) Program-normal**, to generate the reasoning program with the normal DSL. We convert the programs into the normal form commonly used in the general domain, e.g., $add(a_1, a_2) \rightarrow a_1+a_2$. **(iv) the Chain of Thought (CoT) prompting.** CoT prompting (Wei et al., 2022) includes a natural language explanation of the reasoning steps before reaching the answer. We ask 3 expert annotators to compose the explanations for the exemplars. For each method, we run experiments using 3 sets of 10 different exemplars. Table 5 shows the overall results. Even with the gold retrieval results, GPT-3 still underperforms the neural symbolic approaches with full-training data in §5. See Appendix D for all the prompt details.

[5]OpenAI has released the model interface as a paid service

Baselines	Exe Acc	Prog Acc
Answer-only	$24.09_{0.61}$	-
Program-original	$40.81_{4.68}$	$36.62_{4.22}$
Program-normal	$45.15_{2.77}$	$38.88_{2.57}$
CoT prompting	$40.63_{1.25}$	$33.84_{2.19}$
Human Expert Performance	89.44	86.34
General Crowd Performance	46.90	45.52

Table 5: The results for all the prompting methods. We report the average and the standard deviation of different sets of exemplars or annotators.

6.2 Performance Breakdown

We take the results from the best-performing method, **Program-normal**, to investigate the detailed performances. Table 6 shows the performance breakdown for different types of turns. Surprisingly, GPT-3 even performs worse on number selection turns. We find that the model often makes errors for the number selection turns with references to the previous conversation context, e.g., for the question *what is that value in the subsequent year?*, the model still chooses the value in the previous year. Even if we specify the conversational QA setting in the prompt instructions and explicitly ask to answer the last question, the model likely does not understand this task paradigm and often fails to make correct references to the context. This further makes the performances in longer reasoning chains worse, as shown in Table 6. We also analyze the performances for conversation turn length and exemplar numbers in Appendix D.

6.3 Analyses and Findings

We analyze samples of the predictions for all the methods and summarize the following findings:

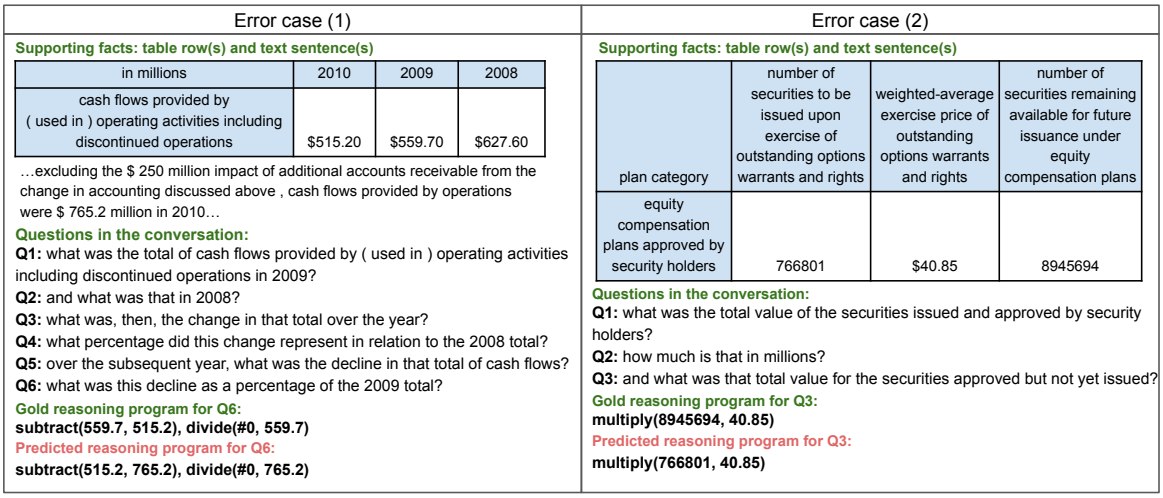

Figure 6: Error cases from the results of FinQANet(RoBERTa-large)

Methods	Exe Acc	Prog Acc
full results	48.85	42.14
Number selection turns		
Number selection questions	35.32	34.72
Program questions	55.56	45.82
Simple & hybrid conversations		
Simple conversations	52.22	46.64
Hybrid conversations	41.16	31.90
Hybrid conversations (first part)	56.30	48.03
Hybrid conversations (second part)	22.85	12.38

Table 6: Performance breakdown: it is hard for GPT-3 to learn the problem paradigm and correctly make references to the conversation context. Still, questions with longer context is harder to answer.

GPT-3 can do simple calculations by itself. For methods that generate the reasoning programs, compared with the results of the neural symbolic approaches in §5, the gap between the execution and program accuracy is much larger. We find that GPT-3 often directly generates the correct numerical results without generating the program. Though the given prompts always derive the programs first, GPT-3 tends to use its own knowledge acquired during pre-training. This is also the reason why Answer-only achieves certain correctness. However, GPT-3 still struggles with complex calculations, such as long digits and divisions.

GPT-3 performs better for its familiar program format. In Table 5, Program-normal outperforms Program-original, since we use the common form of calculation which is seen much more frequently by GPT-3 during its pre-training. GPT-3 makes many grammar errors for Program-original.

GPT-3 struggles with new complex task paradigms. Like stated in §6.2, GPT-3 probably has not seen a similar paradigm as our task setting during pre-training. We see many examples where GPT-3 simply mimics the reasoning steps given in one exemplar but ignores the actual context. This is also the reason that CoT prompting performs even worse than generating the program only. We explicitly explain our task setting in the prompt instructions about how the questions in the conversation are interrelated and the task goal to answer the current turn. However, in many cases, GPT-3 either mimics the reasoning steps given in the exemplars or comes up with incorrect reasoning based on its own knowledge in the general domain. See Appendix D for error cases from Program-normal.

7 Conclusion and Discussion

Our new dataset, **ConvFinQA**, targets one of the major directions to be explored as the next research focus – how to simulate human reasoning abilities in complex real-world settings. We experiment with the neural symbolic models with full training data and the prompting-based few-shot learning and find that: (1) Both approaches are still away from human expert performances, indicating the challenge of this task. (2) The neural symbolic approach uses specifically crafted architectures to learn co-occurrence patterns with large-scale training data. The prompting-based approach recalls its own memory of elaborating the reasoning process with the trigger of the prompts. However, this may not work well when encountering new complex task paradigms for new domains. (3) Theoretically, we may encode as many task paradigms into the

large LMs, as long as the reasoning process can be clearly illustrated by language. But for highly specialized domains or tasks, designing specific models also tend to be more realistic and effective. (4) We are also eager to see the actual *bound* between the reasoning tasks that can benefit from language modeling and the ones that can not. This should be the crucial factor in deciding the upper bound of what large LMs can solve with reasoning.

8 Limitations

In this work, we investigate two construction mechanisms for the conversation, the decomposition of single multi-hop questions and the decomposition and concatenation of two multi-hop questions regarding the same report. This definitely does not cover all possible cases in real-world conversations. We make this first attempt and hope for future work to continue exploration.

For prompting-based methods, we only experiment with the GPT-3 model, whose interface is released to the public as a paid service. Also, due to cost constraints, we do not conduct extensive experiments on complex prompt engineering. We believe our experiments can provide valuable insights into the task of complex reasoning over real-world specific domains, and meanwhile we do not exclude the possibility that there could be better performances for prompting-based methods if applying advanced prompt engineering or even larger pre-trained LMs, like the PaLM model (Chowdhery et al., 2022) which is not released. We leave this for future work.

9 Ethical Considerations

Dataset Collection Process and Conditions. For the annotation of our **CONVFINQA** dataset on Upwork, we first launch interviews of the task introduction with 2 example conversations, which is paid as $30. Then based on their consents to continue working on the large-scale job, we discuss with the workers to reach agreements on the compensation before starting the large-scale job. For the simple conversations from one FinQA question, we pay around $4.0 per conversation. For complex conversations from two FinQA questions, we pay around $7.0 per conversation. The hourly rates are discussed and agreed upon with both sides based on the working speed of different workers. Among all the US-based hires, the average hourly rate is $60.0, with the minimum hourly rate of $50.0. The evaluation tasks and prompt writing tasks follow the similar procedure and rates.

IRB (Institutional Review Board) Approval. The dataset annotation is classified as exempt by our Institutional Review Board (IRB). The systems trained using our dataset are primarily intended to be used as augmenting human decision-making in financial analysis, but not as a replacement of human experts.

Acknowledgment

We thank the anonymous reviewers for their thoughtful comments. This research was supported by the J.P. Morgan Faculty research award. The authors are solely responsible for the contents of the paper and the opinions expressed in this publication do not reflect those of the funding agencies.

References

Md. Shad Akhtar, Abhishek Kumar, Deepanway Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. 2017. A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 540–546. Association for Computational Linguistics.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020. Neural symbolic

reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R. Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3697–3711. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Min-Yuh Day and Chia-Chou Lee. 2016. Deep learning for financial sentiment analysis on finance news providers. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, pages 1127–1134. IEEE Computer Society.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2368–2378. Association for Computational Linguistics.

Jingguang Han, Utsab Barman, Jer Hayes, Jinhua Du, Edward Burgin, and Dadong Wan. 2018. Nextgen AML: distributed deep learning based language technologies to augment anti money laundering investigation. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1821–1831. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1152–1157. The Association for Computational Linguistics.

Chen Liang, Jonathan Berant, Quoc V. Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 23–33. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. Finbert: A pre-trained financial language representation model for financial text

mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4513–4519. ijcai.org.

Armineh Nourbakhsh and Grace Bang. 2019. A framework for anomaly detection using language modeling, and its applications to finance. *CoRR*, abs/1908.09156.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 705–713. AAAI Press.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Weikang Wang, Jiajun Zhang, Qian Li, Chengqing Zong, and Zhifei Li. 2019. Are you for real? detecting identity fraud via dialogue interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1762–1771. Association for Computational Linguistics.

Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, Shih-Fu Chang, Mohit Bansal, and Heng Ji. 2022. Language models with image descriptors are strong few-shot video-language learners. *CoRR*, abs/2205.10747.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. 2021. Conversational question answering: A survey. *CoRR*, abs/2106.00874.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3277–3287. Association for Computational Linguistics.

Appendix A: Operation Definitions

We describe all the operations in Table 7.

Appendix B: Annotation Interface

We use Turkle[6] to build our annotation platform, which is a Django-based web application that can run in a local server. Figure 7 shows our annotation interface. We present the financial report and the decomposed program list to the annotators and ask them to re-write each program step into a question.

Appendix C: Experiment Details

For the neural symbolic approaches, the training of all models are conducted on TITAN RTX GPUs.

[6]https://github.com/hltcoe/turkle

Name	Arguments	Output	Description
add	number1, number2	number	add two numbers: $number1 + number2$
subtract	number1, number2	number	subtract two numbers: $number1 - number2$
multiply	number1, number2	number	multiply two numbers: $number1 \cdot number2$
divide	number1, number2	number	multiply two numbers: $number1/number2$
exp	number1, number2	number	exponential: $number1^{number2}$
greater	number1, number2	bool	comparison: $number1 > number2$

Table 7: Definitions of all operations

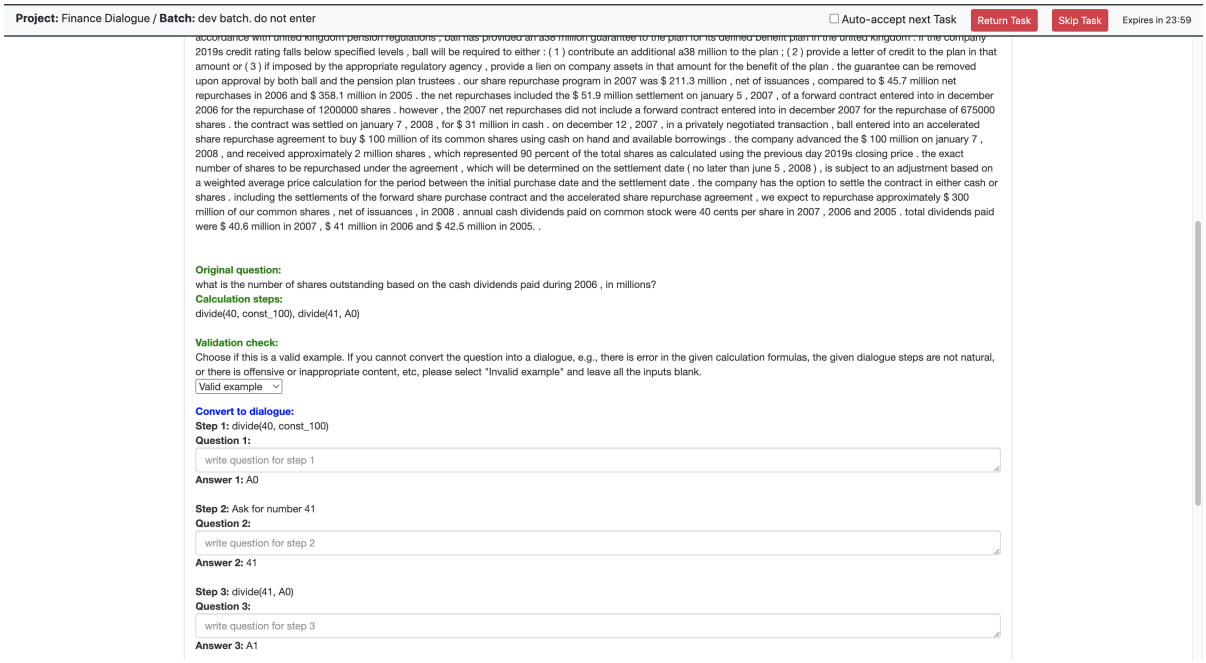

Figure 7: Annotation interface.

Baselines	Exe Acc	Prog Acc
GPT-2(medium)	59.12	57.52q
T-5(large)	58.38	56.71
FinQANet (BERT-base)	54.56	52.81
FinQANet (BERT-large)	60.67	58.99
FinQANet (RoBERTa-base)	64.90	63.15
FinQANet (RoBERTa-large)	68.32	67.87

Table 8: Validation results.

All the implementation and pre-trained models are based on the huggingface transformers library. We use the Adam optimizer (Kingma and Ba, 2015). The learning rate of all models varies at the level of 1e-5 (except for T-5 with 1e-4). We set the batch size as 16. Table 8 shows the results on validation set.

Appendix D: Prompt Details

For the experiments on GPT-3, here is the list of prompts we used:

Retriever Instruction: I am a highly intelligent bot. You need to provide me with context and a series of questions. I will respond yes if the context is needed to answer the last question, otherwise, I will respond with no.

Prompt format: context: (supporting fact candidate) questions: (the question sequence up to current question) answer: (yes or no)

Answer-only Instruction: I am a highly intelligent bot. I can have conversations with the user to answer a series of questions. Later questions may depend on previous questions to answer. You need to provide me with the series of questions as the context and I will answer the last question.

Prompt format: context: (supporting facts) questions: (the question sequence up to current question) answer: (the execution result)

Program-original & Program-normal Instruction: I am a highly intelligent bot. I can have conversations with the user to answer a series of

questions. Later questions may depend on previous questions to answer. You need to provide me with the series of questions as the context and I will answer the last question with a multi-step mathematical solution. We use symbols, such as #0, #1, to denote the results of the intermediate steps.

Prompt format: context: (supporting facts) questions: (the question sequence up to current question) solution: (the program)

CoT Prompt Instruction: I am a highly intelligent bot. I can have conversations with the user to answer a series of questions. Later questions may depend on previous questions to answer. You need to provide me with the series of questions as the context and I will answer the last question with a multi-step mathematical solution with step-by-step explanations. We use symbols, such as #0, #1, to denote the results of the intermediate steps.

Prompt format: context: (supporting facts) questions: (the question sequence up to current question) solution: (CoT explanation and the program).

For all prompts, we add the index of 'Q1', 'Q2', etc., before each question in the question sequence.

Table 9 shows the results of Program-normal for different number of exemplars. Figure 8 shows the performances for question of the nth conversation turn. Figure 9 gives two error cases of GPT-3 Program-normal.

Exemplar numbers	Exe Acc	Prog Acc
5	43.52	36.62
10	48.85	42.14
15	49.31	44.05
20	50.30	45.10
25	49.90	46.08

Table 9: Results of Program-normal for different number of exeamplers.

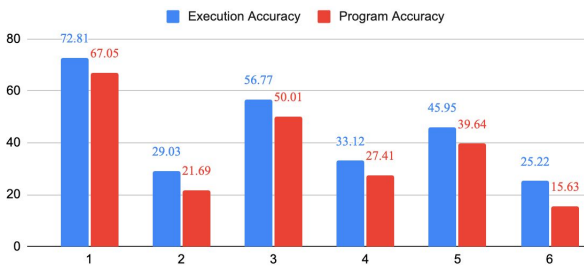

Figure 8: Performances for question of the *nth* conversation turn. The second turn mostly makes references to the first turn and GPT-3 often fails to understand it.

Error case (1)

Supporting facts: table row(s) and text sentence(s)

	2004	2005	2006	2007	2008	2009
s&p 500 index	100.00	104.91	121.48	128.16	80.74	102.11
loews common stock	100.00	135.92	179.47	219.01	123.7	160.62

Questions in the conversation:

Q1: what was the price performance of the loews common stock in 2009?

Q2: and by how much did it change since 2004?

Q3: and only between 2007 and 2008, what was that change for the s&p 500?

Gold reasoning program for Q3:

subtract(80.74, 128.16)

Predicted reasoning program for Q6:

subtract(102.11, 100.00), divide(#0, 100.00)

Error case (2)

Supporting facts: table row(s) and text sentence(s)

… the company recorded a liability for interest and penalties of $ 77 million , $ 55 million , and $ 48 million as of december 31 , 2018 , 2017 , and 2016 , respectively …

Questions in the conversation:

Q1: what was the difference in the liability for interest and penalties between 2017 and 2018?

Gold reasoning program for Q3:

subtract(77, 55)

Predicted reasoning program for Q3:

subtract(55, 48)

Figure 9: Error cases from the results of GPT-3 Program-normal.