Augmentation for Context in Financial Numerical Reasoning over Textual and Tabular Data with Large-Scale Language Model

Yechan Hwang* **Jinsu Lim*** **Young-Jun Lee** **Ho-Jin Choi**
School of Computing, KAIST
{yemintmint,j1n2u,yj2961,hojinc}@kaist.ac.kr

Abstract

Constructing large-scale datasets for numerical reasoning over tabular and textual data in the financial domain is particularly challenging. Moreover, even the commonly used augmentation techniques for dataset construction prove to be ineffective in augmenting financial dataset. To address this challenge, this paper proposes a context augmentation methodology for enhancing the financial dataset, which generates new contexts for the original question. To do this, we leverage the hallucination capability of large-scale generative language models. Specifically, by providing instructions with constraints for context generation with the original dataset's questions and arithmetic programs together as input to the language model's prompt, we create plausible contexts that provide evidence for the given questions. The experimental results showed that the reasoning performance improved when we augmented the FinQA dataset using our methodology and trained the model with it.

1 Introduction

FinQA[Chen et al., 2021] is a financial domain Question Answering dataset, where the task involves performing numerical reasoning on a reports that contain both text and table. Although there were various attempts to improve performance on FinQA, applying data augmentation techniques, which is a method commonly used in general Question Answering tasks, to numerical reasoning datasets like FinQA poses some challenges. This is primarily due to the nature of numerical reasoning, which requires making numerical inferences from the given context and then deriving the final answer through arithmetic calculations. Because of these characteristics, augmenting numerical reasoning dataset demands significant effort compared to other Question Answering dataset because it necessitates a comprehensive consideration of the relationships between operations and numerical values. Numerical reasoning on financial domain data is no exception in this regard.

One simple possible approach is EDA (Easy Data Augmentation)[Wei and Zou, 2019], which has been successful in augmenting text classification dataset. EDA contains very simple methods such as swapping the positions of two arbitrary words within a sentence or deleting random words. It has the advantage of being straightforward to implement and applicable to any natural language sentence, making it a popular augmentation method. However, EDA may not be effective for augmenting numerical reasoning dataset[Dua et al., 2019, Chen et al., 2021, Amini et al., 2019]. As a simple example, consider a question such as *"What is the difference in revenue between Company A and Company B?"* If the numerical values corresponding to the revenues of Company A or B are deleted through EDA from context, accurate inference is impossible. Although augmentation methods that modify only the numerical value parts within the context are commonly used, altering the values in

*Equal contribution

Table Representation Learning Workshop at NeurIPS 2023.

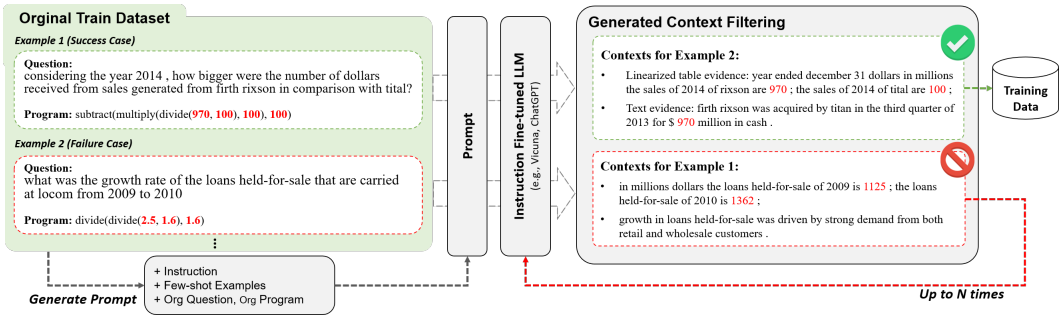

Figure 1: The overall process of our proposed augmentation technique. Generate Context by prompting based on Question and Program from FinQA Dataset. The blue example satisfies the condition of the argument, becoming the augmented Context. The red example is filtered due to missing arguments.

financial documents is not appropriate due to the domain-specific nature of reflecting real economic values. Therefore, in this paper, we propose a context augmentation methodology for augmenting financial numerical dataset. In this approach, the arithmetic program and operators (numerical values) that induce the answer are preserved while creating plausible contexts to augment the data.

Recently, there have been many attempts to augment dataset by creating synthetic data using the generation capabilities of large-scale language models in various domains[Whitehouse et al., 2023, Dai et al., 2023, Rosenbaum et al., 2022, Sarker et al., 2023]. Inspired by these works, we decided to utilize the ability of a generative language model to generate plausible financial contexts based on user instructions.

In this paper, we propose a prompt for augmenting data while preserving (numerical) arguments by constraining instructions with various conditions for context generation. With this prompt, original questions and arithmetic programs from the original dataset are fed to the LLM together. This framework allows us to generate synthetic contexts that can provide evidence for the given question. Subsequently, we trained a financial numerical reasoning model using the augmented dataset and analyzed the experimental results.

2 Numerical Dataset Augmentation

To generate plausible fictional evidence and use it as context augmentation. We employed LLMs to generate a Fictional Context by providing instructions and few-shot examples from the original training data, and use minimal rule-based filtering to preserve the quality and reasoning possibilities of the generated contexts. Finally, our augmentation process consists of two steps: (1) Context Generation and (2) Filtering.

2.1 Context Augmentation with Instructions

Figure 1 show the overall process of our augmentation Technique. In the proposed method, an augmentation method is configured to target FinQA in the Numerical Reasoning Task. However, by modifying the domain description of the instruction in the prompt, our proposed method can be applied to any dataset consisting of numerical processes. Our method is based on in-context learning without fine-tuning, so it is scalable to many different domains.

For an augmentation, we composed a prompt based on the question and reasoning program of the FinQA dataset to generate a context based on key reasoning arguments in numerical reasoning. we instruct the LLM to generate a context with linearized table data and free-form passages, which are the input types of the FinQA dataset's input. To align with the domain of the task, we added an instruction for the context to follow the format of a financial document or news article, described in the Appendix E.

Finally, we can generate contexts by instructing the LLM with our prompt, and by sampling N times using different decoding strategies, we can generate more various contexts. We use the open-access Instruction Fine-tuned LM for a synthetic generation. Specifically, We use Vicuna-13B[Chiang et al., 2023], RLHF fine-tuned LLaMA[Touvron et al., 2023] model, for generating context with instruction.

2.2 Context Filtering

In instruction-based in-context learning, there is no guarantee that generated sentences will adhere to the provided instructions. We implement three filtering methods to ensure numerical reasoning in the generated context. Figure 1 illustrates this filtering step.

Argument Filtering. This technique checks if the reasoning argument is present within the generated context. Any context where the required reasoning argument is absent is filtered out.

Template Filtering. To eliminate undesired content from the generation (e.g., additional explanation, request for feedback), we require the output to follow a specific "evidence: context" format. Any evidence not conforming to this format is discarded.

Context Length Filtering. This technique removes short passages based on the threshold. We set the threshold to 10.

3 Experiment

3.1 Dataset & Evaluation Definition

FinQA[Chen et al., 2021] is a numerical reasoning dataset created from financial reports. It contains 8,281 Reasoning QA Pairs and annotated numerical reasoning processes. The data split into train, eval, and test with 6,251, 883, and 1,147 pairs. Financial contexts consist of tables and text data with many financial terms. FinQA Task aims to find evidence in a dataset consisting of a question q and a financial document d consisting of textual and structured-table fact evidence and to generate a program that calculates the right answer. To evaluate methods, we adopt the evaluation metrics from the original FinQA paper, program accuracy (Prog Acc), and execution accuracy (Exe Acc). The program accuracy calculates the accuracy of the operators and arguments between the predicted program and the golden program. The execution accuracy calculates the accuracy between the executable results.

3.2 Statistics of Augmented Dataset

Table 1 shows the amount of our augmented dataset. We augmented the existing 6251 Train QA Pairs, which can be further expanded by sampling N times, but due to resource limitations, we generate and filter one context for each pair. The number of QA Pairs in the training dataset, FinQA + Ours, was augmented by 31.5% compared to FinQA's pairs. When counting QA Pairs according to the inclusion of each evidence type, there were 2,329 text type, 4,793 table type, and 871 combined cases in the original FinQA dataset. In comparison, our method has a more complex evidence pattern, with 1,312 cases of compound evidence. The average number of evidence is also higher in our method compared to the original. Finally, we obtained FinQA + Ours as our train dataset with a 31.5% augmentation over the original FinQA with a 1-time augmentation. It is noteworthy that although the size of acquired dataset may seem small, we can expand its size by performing our pipeline repeatedly.

Table 1: Statistics for our augmented datasets. Ours is the number of evidence generated, and FinQA is the evidence statistics of Gold fact evidences.

Properties	FinQA	Ours	FinQA + Ours
# of QA Pairs	6,251	1,971	8,222
w/ text evidence	2,329	1,902	4,231
w/ table evidence	4,793	1,381	6,174
# of Evidences	10,692	4,491	15,183
w/ text evidence	3,180	2,994	6,174
w/ table evidence	7,512	1,497	9,009
Avg of Evidences	1.7104	2.2785	1.8466
w/ text evidence	1.3654	1.5741	1.4592
w/ table evidence	1.5673	1.0840	1.4592

3.3 Baseline

We choose FinQANet as our baseline to assess our augmentation technique's performance in Numerical Reasoning. We compare our method with the general NLP augmentation technique, EDA Wei and Zou [2019]. Since the original FinQANet checkpoints were unavailable, we train using the original code and used performance scores from the original paper. Our Retriever is trained on the FinQA training set with Bert-base, while the Generator used each target dataset. To reduce Retriever's influence and highlight augmentation effectiveness, we test with Gold fact Evidence as direct input and contrast results against augmentation via EDA. We describe the detailed information in the Appendix A.

Table 2: **Performance on the FinQA test set.** The highest measures are in bold. † denote taken from the original FinQA. Average value obtained from 4 trials was used.

Methods	Exe Acc	Prog Acc
FinQANet (BERT-base)†	**50.00**	48.00
FinQANet (BERT-base) + **Ours**	49.76	**48.47**
FinQANet (BERT-large)†	53.52	51.62
FinQANet (BERT-large) + **Ours**	**54.14**	**52.05**
FinQANet-Gold (RoBERTa-large)†	70.00	68.76
FinQANet-Gold (RoBERTa-large) + **EDA**	70.71	69.31
FinQANet-Gold (RoBERTa-large) + **Ours**	**71.77**	**70.31**
Human Expert Performance	91.16	87.49
General Crowd Performance	50.68	48.17

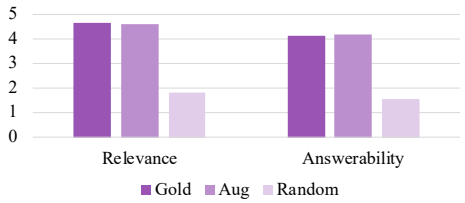

Figure 2: Results of evaluating relevance and answerability scores using G-Eval with GPT-4. The darkest color represents the scores of Gold Context, while the lightest color represents the scores of Randomly Selected Context.

3.4 Result

Main result: Table 2 presents the FinQANet generator performance in FinQA and our Augmented FinQA. Generator with BERT-Base shows that our model has a similar performance with the score reported in the Original FinQA paper. Program accuracy shows a performance improvement(+0.47%), but execution accuracy shows a performance decrease (-0.24%). However, when compared to larger models, our method shows performance improvements in both exe acc(+0.62%) and prog acc(+0.43%). In experiments using only Gold Fact Evidence without Retriever's errors (*FinQANet-Gold*), our proposed method increased by 1.3 points compared to the original paper's reported 68.76 in program acc, and the execution accuracy was 71.77, which is a 1.77% more increase. This shows that our method can generate Augmented Context well while preserving Numerical Reasoning, and can improve performance through the fictional context generated by LLM. The EDA results also showed performance improvement, but our proposed method showed a larger performance difference, which shows that our method is more effective in Numerical Reasoning.

Evaluation of Generated Context: We used G-Eval[Liu et al., 2023] with GPT4 to evaluate whether the generated and filtered contexts are of sufficient quality for numerical reasoning and whether the fictional context from the proposed method is reasonable for the question. We tested the quality of Augmented Context by comparing it with (1) Gold Context of the original document and (2) Random Context which is sampled randomly from the entire FinQA dataset. The evaluation criteria were based on the relevance between the question and the given context and the answerability of the question by referring to the context. We represent the detailed evaluation prompt in the Appendix B.

Fig 2 shows the result of the evaluation using G-Eval with GPT4. We can see that, in G-eval, our augmented contexts have higher relevance and answerability than Random Context in the same document, while assigning similar scores to the Gold Context. The result shows that our augmented context can generate complex contexts by utilizing LLM's generation capabilities.

4 Conclusion and Future Work

This paper proposes a novel context augmentation technique for the Numerical Reasoning Dataset. Our proposed method enables the generation of diverse textual and tabular evidence in the financial domain through in-context learning-based context generation. In the experiments results using FinQA dataset and case study, we demonstrated that the model trained on the augmented data using our approach showed superior performance compared to the model trained solely on the original dataset. Also in comparison to the commonly used EDA technique in NLP, our approach demonstrated significantly improved performance.

In the future, we plan to further enhance our approach by augmenting reasoning programs so that a broader range of contexts and deeper program augmentation could be possible. Furthermore, we will conduct additional experiments on various domains, not just limited to finance, to investigate how our approach can be generalized and applied to other domain's reasoning task.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 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, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*, 2023.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 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, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, et al. Report from the nsf future directions workshop on automatic evaluation of dialog: Research directions and challenges. *arXiv preprint arXiv:2203.10012*, 2022.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Amir Saffari, Marco Damonte, and Isabel Groves. Clasp: Few-shot cross-lingual data augmentation for semantic parsing. *AACL-IJCNLP 2022*, page 444, 2022.

Shouvon Sarker, Lijun Qian, and Xishuang Dong. Medical data augmentation via chatgpt: A case study on medication identification and medication event classification. *arXiv preprint arXiv:2306.07297*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, pages 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1670. URL https://aclanthology.org/D19-1670.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. Llm-powered data augmentation for enhanced crosslingual performance. *arXiv preprint arXiv:2305.14288*, 2023.

A Implementation Details

A.1 Retriever

We train the retriever model using a default hyperparameter setting from the official FinQA repository [2]. This retriever is built on the Bert-base model.

A.2 Generator

For each model, we set a learning rate of 1e-5 and a batch size of 12. All experiments were conducted on A100 40GB, and the test performance was chosen based on the highest validation score from 300 training epochs. We follow the same configuration of the model provided in the official GitHub repository.

with Retriever: In Table 2, the experiments of BERT-base and BERT-large were evaluated using inputs made up of evidence with our trained Retriever. For our Train, Test, and Validation, we used the evidence found in the given financial documents as input.

with Gold Fact Evidence: We used "gold_inds" from the original train, test, and validation dataset.

A.3 EDA

EDA augmented evidence of gold_inds. For the augmentation hyperparameter in EDA, we applied the 0.1 value recommended in the paper to the four methods, and the augmented size was set to 1. In our experiments, we use the same size train dataset and EDA.

B Detailed result

Table 3: Result for Augmented Dataset with Multiple Trials.

% of Augmented Sample	10%		50%		100%	
Metric	exe acc	prog acc	exe acc	prog acc	exe acc	prog acc
Mean	71.29	69.62	71.71	70.25	**71.77**	70.31
Min	70.53	68.79	71.32	69.75	70.97	69.40
Max	71.93	70.18	72.01	70.62	**73.15**	71.84
SD	0.62	0.65	0.32	0.38	0.95	1.07
Lower Limit (95%)	70.99	69.29	71.55	70.06	71.31	69.79

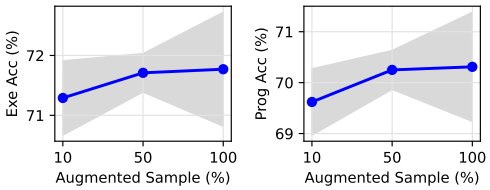

Figure 3: **Effect of Percent of Augmented Sample.**

In order to verify the significance of our experimental results and the effect of the size of the augmented data on the results, we conducted 4 trials each on the sampled augmented dataset. In Table 3 and Fig 3, Our augmented dataset using only 10% of the augmented QA pairs showed a higher performance on average compared to the baseline model's performance of 70.00, and the performance tended to increase as the augmented data size increased. Finally, we achieved an average exe acc of 71.77 using all of our augmented datasets. The results show that the proposed method can augment numerical reasoning over tabular and text datasets using LLM.

[2] https://github.com/czyssrs/FinQA

Table 4: **Error Analysis.** We compare error cases between FinQANet-Gold and ours-Gold.; Case 1) The financial knowledge for "credit lines" Case 2) Complex reasoning of 4 steps Case 3) Number unit conversion between "billion" and "million".

Case 1) what is the amount of credit lines that has been drawn in millions as of year-end 2016?

Gold: [1] additionally , we have other **committed and uncommitted credit lines of $ 746 million** with major international banks and financial institutions to support our general global funding needs , including with respect to bank supported letters of credit, performance bonds and guarantees .
[2] approximately **$ 554 million of these credit lines were available for use** as of year-end 2016

Answer: subtract(746, 554) **FinQA:** multiply(554, const_1000000) **Ours:** subtract(746, 554)

Case 2) what is the percentage change in the total fair value of non-vested shares from 2009 to 2010?

Gold:

	shares	weighted average grant-date fair value
non-vested at may 31 2009	**762**	**42**
non-vested at may 31 2010	**713**	**42**

Answer: multiply(762,42), multiply(713,42), subtract(#1,#0), divide(#2,#0)
FinQA: subtract(713,762), divide(#0,762)
Ours: multiply(762,42), multiply(713,42), subtract(#1,#0), divide(#2,#0)

Case 3) what is the percentage change in the total fair value of non-vested shares from 2009 to 2010?

Gold: [1] we maintained a **$ 1.4 billion senior credit facility** with various financial institutions , including the $ 420.5 million term loan and a **$ 945.5 million revolving credit facility.**

Answer: multiply(1.4, const_1000), divide(945.5, #0) **FinQA:** divide(945.5, const_1000)
Ours: divide(945.5, 1.4)

C Case Study

In the table 4, we compare the results of our method with representative error cases reported in the original FinQA. For two of the three representative cases, our model got the answer correct, and for the last case, an error occurred. In case 3, our model is wrong due to the difficulty in number unit conversion between billion and million. However, in cases 1 and 2, we think that our proposed method can alleviate the problem of misunderstandings of financial knowledge by augmenting the context with LLM based on various financial domain keywords that LLM knows.

D An Example of Prompt for Evaluating Answerability

We used G-eval to evaluate whether the contexts generated and filtered by our framework were of sufficient quality. In order to evaluate the ability of the given context, we randomly selected the same number of sentences as gold evidence. and, we defined criteria called "Relevance" and "Answerability" as follows:

- Relevance (1-5) - contexts are on-topic with the question (which is based on [Mehri et al., 2022])
- Answerability (1-5) - The ability of the given context to provide a clear and relevant answer to the given question.

then, we proceeded with the evaluation with the following prompt. Figure 4 shows an example of a prompt for evaluating Answerability.

You will be given a question for numerical question answering. You will then be given supporting contexts written for solving this question. Your task is to rate the contexts on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:
Answerability (1-5) - The ability of the given context to provide a clear and relevant answer to the given question.

Evaluation Steps:
1. Read the question and the supporting contexts carefully and identify the main facts and details it presents.
2. Read the context and compare it with the question. Check whether the context contains any supporting facts for the question or not.
3. Assign a score for answerability based on the Evaluation Criteria.

Example:
Question:
what is the net change in net revenue during 2015 for entergy corporation?

Supporting Contexts:
the 2014 net revenue of amount (in millions) is $ 5735 ;
the 2015 net revenue of amount (in millions) is $ 5829 ;

Evaluation Form (scores ONLY):

Figure 4: A prompt template for evaluating Answerability

E Few-shot Context Generation Prompt

Table 5 shows the full text of the prompts we use in our experiments. We used three few-shot examples and use </s> to separate the few-shot examples.

The Instruction part describes the task and constraints of the generation.

When generating the evidence context, we generated from all of the table and text sources. but, we excluded QA Pairs with the table operator in the reasoning program from the augmentation target. We excluded table operations that directly manipulate the table (e.g., table_max, table_average) because only the column name exists in the reasoning program, and the full arguments are not known.

Prompt with Few-shot	
Instruction	Compose a fictional context relevant to a Numerical QA Task in finance using the provided question, and arithmetic program for calculating the answer. The arithmetic program is a functional program code designed to calculate the answer. The context should be presented as a text evidence or linearized table evidence from a financial document or news article which provides information supporting the argument made by the arithmetic program. The table evidence is linearized with the form column name is cell value and a delimiter of ' ; '. You may include the name of a fictional or real organization or nation when generating context. The context must not reveal the answer. Generate contexts for solving question, consisting of one to five evidences based on a given arithmetic program.
Few-Shot Examples	Question: what is the interest expense in 2009? Arithmetic Program: divide(3.8, divide(100, 100)) Context: 1. text evidence: if libor changes by 100 basis points , our annual interest expense would change by $ 3.8 million . Question: what is the difference between the highest and average value of operating profit? Arithmetic Program: subtract(table_max(operating profit, none),table_average(operating profit, none)) Context: 1. linearized table evidence: in millions the operating profit of 2009 is 50 ; the operating profit of 2008 is 103 ; the operating profit of 2007 is 108 ; Question: for 2010 , was the after-tax gain on our sale of gis greater than overall net interest income? Arithmetic Program: greater(387, 9230) Context: 1. linearized table evidence: year ended december 31dollars in millions the net interest income of 2011 is $ 8700 ; the net interest income of 2010 is $ 9230 ; 2. text evidence: asset management group asset management group earned $ 141 million for 2011 compared with $ 137 million for 2010 . 3. text evidence: results for 2010 included the $ 328 million after-tax gain on our sale of gis , $ 387 million for integration costs , and $ 71 million of residential mortgage foreclosure-related expenses .
Input Template	Question: {question} Arithmetic Program: {program} Context:

Table 5: The prompt for the in-context learning to generate the Numerical Reasoning Context, and the few-shot example was taken from the train, and these 3-shot examples were removed from the Augmented Dataset.