PACIFIC: Towards Proactive Conversational Question Answering over Tabular and Textual Data in Finance *

Yang Deng[1], **Wenqiang Lei**[2,†], **Wenxuan Zhang**[3], **Wai Lam**[1], **Tat-Seng Chua**[4]
[1]The Chinese University of Hong Kong, [2]Sichuan University,
[3]DAMO Academy, Alibaba Group, [4]Sea-NExT Joint Lab, National University of Singapore
{ydeng,wlam}@se.cuhk.edu.hk, {wenqianglei,isakzhang}@gmail.com

Abstract

To facilitate conversational question answering (CQA) over hybrid contexts in finance, we present a new dataset, named PACIFIC. Compared with existing CQA datasets, PACIFIC exhibits three key features: (i) proactivity, (ii) numerical reasoning, and (iii) hybrid context of tables and text. A new task is defined accordingly to study Proactive Conversational Question Answering (PCQA), which combines clarification question generation and CQA. In addition, we propose a novel method, namely UniPCQA, to adapt a hybrid format of input and output content in PCQA into the Seq2Seq problem, including the reformulation of the numerical reasoning process as code generation. UniPCQA performs multi-task learning over all sub-tasks in PCQA and incorporates a simple ensemble strategy to alleviate the error propagation issue in the multi-task learning by cross-validating top-k sampled Seq2Seq outputs. We benchmark the PACIFIC dataset with extensive baselines and provide comprehensive evaluations on each sub-task of PCQA.

1 Introduction

Financial question answering (QA) systems aim to answer user's instant queries by selecting appropriate information from financial documents, which often contain a hybrid of tabular and textual content, and performing complex quantitative analysis. Existing studies on financial QA (Zhu et al., 2021; Chen et al., 2021b; Zhu et al., 2022; Li et al., 2022a) mainly focus on building *single-turn QA* systems to **passively** respond to user queries. However, in real-world information-seeking applications (Zamani et al., 2022), the system is expected to (i) answer highly context-dependent questions in a *multi-turn conversation*, and (ii) **proactively** assist users in performing complicated information seeks. In an interactive setting, users tend to ask follow-up or co-referencing questions (Kundu et al., 2020; Liu et al., 2021) without repeating previous information, and provide a succinct or brief query that may be ambiguous or lack the necessary content. Especially in financial QA, the user queries often contain multiple constraints from different aspects for the concerned objective, as the examples shown in Fig. 1. Even just missing one constraint may cause ambiguity. Therefore, a proactive conversational system that can help clarify the ambiguity is of great importance in financial QA.

To this end, this paper introduces a new dataset to promote research into **P**ro**A**ctive **C**onversat**I**onal question answering in **FI**nan**C**e, named PACIFIC. PACIFIC is constructed by using the QA pairs in an expert-annotated financial QA dataset, TAT-QA (Zhu et al., 2021), as guidance to build conversation sessions with consecutive topics. As shown in Fig. 1, we rewrite the original self-contained questions into conversational questions with anaphora (co-referencing among different turns) and ellipsis (omitting repeated words in the follow-up questions), as well as construct ambiguous questions that require clarification. Accordingly, we define a new task, named Proactive Conversational Question Answering (PCQA), which combines the problems of clarification question generation (CQG) (Aliannejadi et al., 2021) and conversational question answering (CQA) (Reddy et al., 2019). PCQA consists of three sub-tasks: (i) Given the user's query, the system first identifies whether the question is ambiguous (*i.e.*, clarification need prediction). (ii) If so, the system will proactively ask a clarifying question to clarify the uncertainty (*i.e.*, CQG). (iii) If not, it will directly answer the question (*i.e.*, CQA).

Compared with existing datasets listed in Table 1, PACIFIC exhibits three key challenges: (i) *proac-*

* This work is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719) and the Sea-NExT Joint Lab. Work done when Yang Deng was a visiting research scholar in the Sea-NExT Joint Lab.

† Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6970 - 6984
December 7-11, 2022

The following tables present the recorded investment by **portfolio segment** and by region.

($ in millions) At December 31, 2019:	Americas	EMEA	Asia Pacific	Total
Recorded investment:				
Lease receivables	3,419	1,186	963	5,567
Loan receivables	6,726	3,901	2,395	13,022
Allowance for credit losses:				
Balance at Jan. 1, 2019	158	65	56	279
Lease receivable	53	22	24	99
Loan receivables	105	43	32	179
Balance at Dec. 31, 2019	120	54	36	210
Lease receivables	33	23	16	72
Loan receivables	88	31	20	138

Write-offs of lease receivables and loan receivables were $16 million and $47 million, respectively, for the year ended December 31, 2019. The average **recorded investment** of impaired leases and loans for **Americas**, **EMEA** and **Asia Pacific** was $138 million, $49 million and $45 million, respectively.

#	Original Question in TAT-QA	#	Conversational Question in PACIFIC	Answer Type	Derivation (Python Code)	Answer
Q1	How many regions are recorded?	**T1**	How many regions are recorded?	Count.	len(["**Americas**","**Asia Pacific**","**EMEA**"])	3
Q2	What were the write-offs of lease and loan receivables in December 2019?	**T2**	What were the write-offs in December 2019?	Quest.	["Which **portfolio segment** are you asking about?"]	
		T3	***Write-offs*** of lease and loan receivables, respectively.	Spans	["**16 million**", "**47 million**"]	
Q3	What is the average recorded investment of lease and loan receivables for Americas in December 2019?	**T4**	What is the average recorded investment for Americas ***in that time***?	Quest.	["What kind of **recorded investment** are you asking about?"]	
		T5	The ***recorded investment*** of lease receivables and loan receivables.	Arith.	(3,419+6,726)/2	5072.5 million
Q4	What is the average recorded investment of lease and loan receivables for EMEA in December 2019?	**T6**	***How about that*** for EMEA?	Arith.	(1,186+3,901)/2	2543.5 million
Q5	What is the change in allowance for credit losses of loan receivables for EMEA during 2019?	**T7**	What is the change in allowance for credit losses of loan receivables for ***there*** during 2019?	Arith.	88-105	-17 million
Q6	What is the percentage change in allowance for credit losses of loan receivables for EMEA during 2019?	**T8**	What is ***its*** percentage change?	Arith.	(88-105)/105	-16.19 percent

Figure 1: An example of PACIFIC. The left dashed line box shows a hybrid context as the grounded document. The right solid line box shows the corresponding questions, responses with its answer types and derivation.

tivity: the system needs to proactively assist the user to clarify their question intent by asking clarifying questions; (ii) *numerical reasoning*: there are a large number of questions that require numerical reasoning to answer; and (iii) *hybrid context*: the grounded document is composed by both tabular and textual content.

To tackle these challenges, we propose a novel method, named UniPCQA, to unify all sub-tasks in PCQA as a sequence-to-sequence (Seq2Seq) problem. Specifically, we reformulate the numerical reasoning process in financial question answering as a code generation task, which captures the input knowledge (*e.g.*, figures or entities) and condenses their numerical reasoning relations (*e.g.*, arithmetic operators) into a piece of executable code (*e.g.*, `Python`). We further design specific input and output representations to adapt a hybrid of tabular, textual, and arithmetic content into the Seq2Seq framework. In addition, UniPCQA can perform multi-task learning over all sub-tasks to enable the proactive detection of the need for clarification. Finally, we propose an ensemble strategy, named Consensus Voting, to alleviate the error propagation issue in the multi-task learning by cross-validating the top-k sampled Seq2Seq outputs. The main contributions of this paper are:

- To study the proactivity in financial question answering, we propose a novel dataset, namely PACIFIC, for conversational question answering over tabular and textual contexts, and define the problem of PCQA.
- We reformulate the numerical reasoning process as code generation and propose a unified hybrid Seq2Seq framework, namely UniPCQA, to handle the hybrid contexts and diverse responses in PCQA.
- We benchmark the PACIFIC dataset with extensive baselines and provide comprehensive evaluations on each sub-task of PCQA. Despite the effectiveness of UniPCQA, the performance is far behind human experts, showing that PACIFIC presents a challenging problem for future studies.

Dataset	Domain	Turn	Modality	Proact.	NR
Hybrid-QA	General	Single	Table/Text	×	×
OTT-QA	General	Single	Table/Text	×	×
FinQA	Finance	Single	Table/Text	×	✓
TAT-QA	Finance	Single	Table/Text	×	✓
SQA	General	Multi	Table	×	✓
QuAC	General	Multi	Text	×	×
CoQA	General	Multi	Text	×	×
Abg-CoQA	General	Multi	Text	✓	×
HybriDial.	General	Multi	Table/Text	×	×
MMConvQA	General	Multi	Table/Text/Image	×	×
ConvMix	General	Multi	Table/Text/KB	×	×
PACIFIC	Finance	Multi	Table/Text	✓	✓

Table 1: Comparison of PACIFIC and related QA and CQA datasets. "NR" denotes Numerical Reasoning.

2 Related Works

Conversational Question Answering Evolving from single-turn QA tasks (Chen et al., 2020; Deng et al., 2022a), CQA aims at interactively answering multiple turns of information-seeking questions according to the given document (Reddy et al., 2019; Choi et al., 2018). Common challenges in CQA include the anaphora and ellipsis issue (Iyyer et al., 2017; Kundu et al., 2020; Liu et al., 2021). To this

end, several attempts have been made on developing end-to-end CQA models with dialogue history tracking (Qu et al., 2019; Qiu et al., 2021). Another group of works emphasizes the importance of query rewriting in CQA (Vakulenko et al., 2021; Raposo et al., 2022; Anantha et al., 2021; Kim et al., 2021), which generates self-contained questions for performing single-turn QA. In addition, beyond simply focusing on one kind of information source, it has received increasing attentions to investigate CQA over hetergeneous sources (Li et al., 2022b; Christmann et al., 2022).

Proactive Conversational Systems Early studies on conversational systems basically develop dialogue systems that passively respond to user queries, including all the CQA studies discussed above. As for conversational recommendation (Lei et al., 2020a,b; Deng et al., 2021) and goal-oriented dialogues (Lei et al., 2022; Deng et al., 2022b), policy learning or goal planning attaches great importance in building a proactive conversational system for promptly adjusting dialogue strategies or soliciting user intents. Recently, many efforts have been made on CQA systems that can proactively assist users to clarify the ambiguity or uncertainty in their queries by asking clarifying questions (Wang and Li, 2021; Zamani et al., 2020a; Sekulic et al., 2021; Gao and Lam, 2022). Several datasets such as ClariQ (Aliannejadi et al., 2021) and Abg-CoQA (Guo et al., 2021) have been constructed to facilitate this line of research. However, these datasets solely target at the clarification question generation (CQG) or clarification-based CQA problem. To stimulate progress of building the whole system for proactive CQA, we define the PCQA task, which unifies CQG and CQA.

Numerical Reasoning Numerical reasoning is the key to many NLP applications (Thawani et al., 2021; Pal and Baral, 2021), especially in QA, such as Mathematical QA (Dua et al., 2019; Amini et al., 2019) and Financial QA (Zhu et al., 2021; Chen et al., 2021b). Early works typically design specialized operation or reasoning modules for handling different types of questions (Andor et al., 2019; Hu et al., 2019; Ran et al., 2019). Despite the effectiveness, it is challenging for them to scale to different numerical reasoning scenarios due to their task-specific designs. Recent years have witnessed many advanced approaches to injecting the numerical reasoning skills into pre-trained language models (PLMs), by post-training (Geva et al., 2020; Pi et al., 2022) or prompt-based learning (Wei et al., 2022; Wang et al., 2022). However, these methods are developed to perform numerical reasoning over texts. Suadaa et al. (2021) investigate template-based table representations for numerical reasoning in PLMs-based table-to-text generation. In this paper, we propose to handle numerical reasoning as the code generation task over hybrid contexts.

3 PACIFIC Dataset Creation

3.1 Annotation & Quality Control

Similar to the dataset creation process of other CIS datasets, such as HybriDialogue (Nakamura et al., 2022) from OTT-QA (Chen et al., 2021a) and MMConvQA (Li et al., 2022b) from MMQA (Talmor et al., 2021), we build the PACIFIC dataset from the TAT-QA dataset by using its question-answer pairs as guidance for constructing conversation sessions. There are on average 6 individual question-answer pairs shared with the same grounded contexts in TAT-QA, which are integrally regarded as one conversation session. However, we construct the conversation session in a different way from the traditional manner where a complex single-turn question is decomposed into multiple context-dependent simple questions (Nakamura et al., 2022; Li et al., 2022b), since this manner may discard the nature of financial QA. Instead, we rewrite each question into one conversational question, which not only increases the efficiency of dataset construction, but also preserves the quality and difficulty of the dataset with expert-annotated answers and informative user queries.

Due to the space limitation, the overall pipeline for PACIFIC creation is presented in Appendix A. An example is presented in Figure 1 with its original sample in TAT-QA. For each conversation sample, two annotators are asked to build a natural and consecutive conversation session. They are well-educated postgraduate students majored in finance or similar disciplines. The first annotator serves as the seeker to perform the annotation tasks; while the second annotator plays the role of the agent to provide clarifying questions. The instructions given to the first annotator are as follows:

1) *Organize Conversation Sessions*. Given the same hybrid context, set up a conversation session with consecutive topics from multiple individual QA pairs. Two questions that share the same entities are regarded as talking about the same topic. For example, **Q1**, **Q2**, and **Q3** are concerned about

PACIFIC/TAT-QA	Train	Dev	Test
# Dialogues	2,201/-	278/-	278/-
# Turns (QA pairs)	15,087/13,215	1,982/1,668	1,939/1,669
# Clarifying turns	1,872/-	320/-	270/-
Avg. turns / dialogue	6.9/-	7.1/-	7.0/-
Avg. words / question	9.6/12.5	9.0/12.4	9.4/12.4
Avg. words / answer	4.6/4.1	4.6/4.1	4.8/4.3

Table 2: Data statistics of PACIFIC.

	Table	Text	Table-text	Total
Span	1,797	3,497	1,842	7,136
Spans	777	258	1,037	2,072
Counting	106	5	266	377
Arithmetic	4,744	143	2,074	6,961
Question	1,293	270	899	2,462
Total	8,717	4,173	6,118	19,008

Table 3: Number of questions regarding different answer types and sources in PACIFIC.

the same time (*December 2019*), while **Q4**, **Q5**, and **Q6** are asking about the same region (*EMEA*). We, thus, order these questions into adjacent turns.

2) *Rewrite Conversational Questions*. If consecutive questions share the same entities, rewrite the original self-contained questions to produce conversational questions with anaphora and ellipsis. For example, the only difference between **Q3** and **Q4** is the concerned region (*Americas* & *EMEA*). After the rewriting, **T6** becomes "*How about that for EMEA?*" without the repeated content in **T4**.

3) *Construct Ambiguous Questions*. If the question contains multiple entities, rewrite it to construct an ambiguous question by omitting one of the entities that can introduce ambiguity. For example, **Q3** is asking about the average value under multiple constraints. In **T4**, the portfolio segment (*Lease and loan receivables*) is omitted to construct an ambiguous question that required clarification.

Given the set of reconstructed questions, the second annotator is served as the agent to *Provide Clarification Questions*: *i.e.*, ask a clarification question in terms of the omitted entity. Subsequently, the omitted entity will be the seeker's query in the next turn, as **T3** and **T5** in Fig. 1.

To ensure the quality of annotation in PACIFIC, we ask two verifiers to validate each turn in the constructed conversations. If any mistake or problem is found, *e.g.*, the constructed conversation is incoherent, the annotator will be asked to fix it until the annotation passes the checks by the two verifiers. The first-round validation captures 212 mistakes (212/19,008=1.1%), and the inter-annotator agreement between the two verifiers is 0.62.

3.2 Statistical Analysis

Finally, we obtain a total of 2,757 conversations over the hybrid contexts, which contains 19,008 corresponding QA pairs in total and an average of 7 turns of QA in each conversation. The train-dev-test split is the same as TAT-QA. We present the data statistics of PACIFIC in Table 2, and the question distribution regarding different answer types and sources in Table 3. Compared with TAT-QA, PACIFIC contains 2,462 more QA turns for asking clarification questions (2,462/19,008=13.0%)[1]. The average length of the questions in PACIFIC is shorter than that in TAT-QA, which means that the conversational questions are more succinct and brief. Conversely, the average length of the answers in PACIFIC is longer than that in TAT-QA, due to the incorporation of clarification questions.

3.3 Problem Definition

We introduce the **Proactive Conversational Question Answering (PCQA)** task, which unifies two tasks: (I) Clarification Question Generation and (II) Conversational Question Answering. Given the conversation history $C_t = \{q_1, r_1, ..., q_t\}$ and the grounded document $D = \{E, T\}$ consisting of both textual contexts E and structured table T, the goal is to generate the response r_t at the current turn t. As shown in Fig. 2, the overall task can be decomposed into three sub-tasks:

1) *Clarification Need Prediction* (CNP) aims to predict the binary label y to determine whether to ask a question for clarifying the uncertainty. Otherwise the query q_t can be directly responded to.

2) *Clarification Question Generation* (CQG) will generate a clarification question as the response r_t, if CNP detects the need for clarification.

3) *Conversational Question Answering* (CQA) will directly produce the answer as the response r_t, if it is not required for clarification.

It is worth noting that the PACIFIC dataset can be adopted for the evaluation of both end-to-end and pipeline-based PCQA methods, as well as the separated evaluation on each sub-task.

4 Method

We introduce the UniPCQA model, which unifies all sub-tasks in PCQA as the Seq2Seq problem and performs multi-task learning among them.

[1]The proportion of clarification interactions is close to existing CQG datasets, *e.g.*, 16.1% in (Aliannejadi et al., 2021) and 11.5% in Guo et al. (2021)

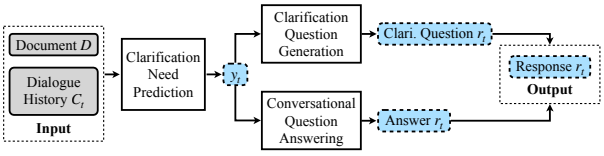

Figure 2: Proactive conversational question answering.

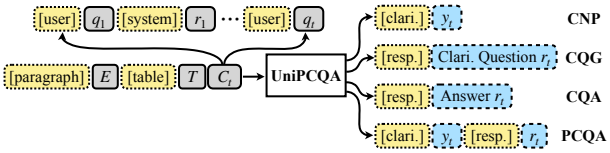

Figure 3: Overview of the input/output for UniPCQA. Note that the first three outputs denote the output in the single-task learning setting for each sub-task, while the last one denotes that in the multi-task learning setting.

4.1 Numerical Reasoning as Code Generation

One of the key challenges of PACIFIC is the requirement to conduct numerical reasoning, due to the large proportion of questions involving numerical calculation. However, existing methods proposed for financial question answering suffer from two main issues: 1) they rely heavily on handcrafted designs for numerical operators (Zhu et al., 2021) or symbolic programs (Chen et al., 2021b), which are hard to be generalized to complex numerical calculation; 2) the knowledge from large-scale PLMs cannot be fully utilized for the down-stream problem of numerical reasoning, due to the large gap between them.

In the light of these issues, we formulate the numerical reasoning process as the code generation task, which aims to capture the input knowledge (*e.g.*, figures or entities) and condense their numerical reasoning relations (*e.g.*, arithmetic operators) into a piece of executable code. Take `Python` as an example. `Python` can handle the derivation with different kinds of operations, such as arithmetic, counting, enumeration, etc. The addition, subtraction, multiplication and division operators are denoted by +, -, *, and /, respectively. The `len()` function that returns the number of items in an object can be used for the counting operation. To be consistent, we also regard span-based and question-based responses as a `list()` of items in `Python` for code generation. Examples of `Python` code for different types of answers are shown in Fig. 1.

Without the need for designing another execution algorithm (Zhu et al., 2021; Chen et al., 2021b), the generated `Python` code can be directly executed by the `eval()` function to derive the final answer r_t, as the following examples:

$$\texttt{eval}((36.6 - 20.5)/20.5) \rightarrow 0.7854$$
$$\texttt{eval}(\texttt{len}([\text{“2018”}, \text{“2019”}])) \rightarrow 2$$

Therefore, we can reconstruct the target `Python` code from the original answer derivation, according to the `Python` syntax, which can be easily generalized into different types of numerical calculation. The numerical reasoning process can not only get free of manually designed operators or programs, but also leverage the knowledge from PLMs, especially those code-related PLMs, such as CodeT5 (Wang et al., 2021).

4.2 Hybrid Seq2Seq Generation

In financial PCQA, the input sequence contains both textual and tabular content, while the output sequence can be a piece of code, a natural language sentence, or even a mix of code and text. In order to handle all sub-tasks in PCQA, we design the hybrid input/output representations for a unified Seq2Seq framework. Specifically, we add special tokens to indicate different types of information as well as specify each sub-task. Assuming that the grounded textual context is $E = \{p_1, ...p_k\}$ and the grounded structure table is $T = \{c_{11}, ..., c_{1n}, ..., c_{m1}, ..., c_{mn}\}$, then the input can be linearized as follows:

“`[paragraph]` p_1 `</p>` ... `</p>` p_k `</p>` `[table]` c_{11} : c_{12} | ... | c_{1n} `</t>` ... c_{m1} : c_{m2} | ... | c_{mn} `[user]` q_1 `[system]` r_1 ... `[user]` q_t”

As shown in Fig. 3, this Seq2Seq formulation can be applied to each sub-task, or perform multi-task learning of all sub-tasks in order. The output sequence for multi-task learning is represented as:

$$\text{“}\texttt{[clari.]}\ y\ \texttt{[resp.]}\ r_t\text{”}$$

where $y \in \{\texttt{True}, \texttt{False}\}$, and r_t will be a clarification question or a piece of code accordingly.

UniPCQA can be initialized with weights from any generative PLM, *e.g.*, T5 (Raffel et al., 2020). Given a training sample (C_t, D, o), the model is trained to maximize the sequential log-likelihood:

$$\mathcal{L}_\theta = \sum_{l=1}^{L} \log p_\theta(o_l | o_{<l}; C_t, D), \qquad (1)$$

where θ denote the model parameters, L is the maximum target sequence length, and o is the target sequence according to the target task.

4.3 Consensus Voting

As UniPCQA solves the end task using multi-task learning in sequential order, the error in the previous task may be propagated to the latter one. Specifically, if the model makes a wrong prediction in the CNP task, the model will generate an inappropriate response at the end.

Inspired by the Self-Consistency strategy (Wang et al., 2022) for improving the few-shot learning accuracy of PLMs, we investigate a similar ensemble-based strategy, namely Consensus Voting, to alleviate the error propagation issue in the multi-task learning. Specifically, Consensus Voting samples a set of candidate sequences $O = \{o_i : i \in 1, ..., N\}$ generated by the PLM, which contains a diverse set of multi-task results as well as different reasoning paths, instead of using Greedy Decode. We then select the final response by ensembling the derived responses from O based on plurality voting:

$$r_t = \arg\max_{o_i \in O} \sum_{j=1}^{N} \mathbb{I}(\sigma(o_j) = \sigma(o_i)), \quad (2)$$

where $\sigma(\cdot)$ denotes the execution of deriving the answer from the generated sequence, *e.g.*, `eval()`.

The motivation is that it will be difficult for the sampled outputs to reach a consensus if the user query is ambiguous, since the decoder will be confused about how to generate a correct derivation with incomplete information. At this time, the plurality vote will tend to ask a clarification question. In addition, the same answer can be obtained by executing different derivations in some cases. As shown in Fig. 1, different extraction orders of three regions can lead to the same answer in T1, *e.g.*, `len`([“Americas", “EMEA", “Asia Pacific"]) = `len`([“EMEA", “Americas", “Asia Pacific"]). So does the derivation in T8, *i.e.*, $(88 - 105)/105 = 88/105 - 1$. Therefore, if there are multiple generated derivations that lead to the same answer, this answer will get higher votes.

5 Experiments

We first evaluate methods on two widely-studied tasks in conversational information seeking, including (I) clarification question generation (CQG) and (II) conversational question answering (CQA). Then we benchmark the overall performance of proactive conversational question answering (PCQA) on PACIFIC.

Method	Dev			Test		
	P	R	F1	P	R	F1
$\text{BERT}_{\text{large}}$	84.5	85.4	84.9	80.0	83.9	81.7
$\text{RoBERTa}_{\text{large}}$	93.1	89.4	91.2	90.0	90.8	90.2
UniPCQA (T5)	**93.7**	**91.0**	**92.3**	**90.6**	**91.6**	**91.1**

Table 4: Results on Clarification Need Prediction.

5.1 Implementation

We evaluate UniPCQA with T5_{base} as the baseline. To study the effectiveness of the reformulation of code generation, we further adopt $\text{CodeT5}_{\text{base}}$ (Wang et al., 2021) for evaluation, which is a unified encoder-decoder model pre-trained with both code-related understanding and generation tasks. Following previous studies (Fan et al., 2018; Holtzman et al., 2020), we apply top-k sampling with temperature $T = 0.5$ and $k = 40$ to sample a diverse set of decoded sequences. For Consensus Voting, we sample $N = 40$ outputs, while the baseline is to apply Greedy Decode to generate a single output. More implementation details can be found in Appendix B.

5.2 Task I: Clarification Question Generation

The CQG task is commonly performed in two steps: 1) clarification need prediction (CNP), and 2) clarification question generation.

5.2.1 Baselines and Evaluation Metrics

Following ClariQ (Aliannejadi et al., 2021), a popular CQG challenge, we include $\text{BERT}_{\text{large}}$ (Devlin et al., 2019) and $\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019) based classifiers as baselines, and use Precision, Recall, and F1 for CNP evaluation. For CQG, we compare to several CQG baselines in latest studies, including Template-based Question Generation (TB) (Zamani et al., 2020a), CopyTrans. (Wang and Li, 2021), and Q-GPT (Sekulic et al., 2021), and adopt ROUGE-2 (F1), Exact Match (EM), and token-level F1 as evaluation metrics. Note that we simply flatten the table into a sequence by row followed by tokens from the paragraphs for all the baselines, which is also applied to the baselines in the following evaluation. More details about baselines can be found in Appendix C.

5.2.2 Experimental Results

Table 4 presents the experimental results on the CNP task, showing that a stronger PLM leads to better performance in this binary classification task.

Method	Dev			Test		
	ROUGE	EM	F1	ROUGE	EM	F1
BERT+TB	69.8	36.3	75.4	67.8	33.2	72.8
CopyTrans.	70.3	39.4	75.4	68.1	37.9	73.2
Q-GPT	86.5	67.8	90.5	83.9	63.4	87.8
UniPCQA (T5)	**90.7**	**76.9**	**93.4**	**87.8**	**71.1**	**91.1**

Table 5: Results on Clarification Question Generation.

QR Model	QA Model	Dev		Test	
		EM	F1	EM	F1
Gold	NumNet+ V2	38.1	48.3	37.0	46.9
	TAGOP	55.2	62.7	50.1	58.0
	TaCube	57.7	66.2	-	-
	POET-SQL	59.1	65.9	-	-
	UniPCQA (T5)	65.3	72.9	62.3	71.1
	UniPCQA (CodeT5)	**68.2**	**75.5**	**63.9**	**72.2**
Original	TAGOP	39.4	46.6	34.7	43.2
Trans.++		41.8	48.1	36.2	43.9
T5		42.0	48.4	36.6	44.2
T5*		50.0	56.6	46.2	54.2
End-to-end	NumNet+ V2	30.2	39.0	27.7	36.9
	TAGOP	45.6	53.2	43.3	50.4
	HAE ($BERT_{large}$)	20.3	30.6	18.2	25.4
End-to-end	UniPCQA (T5)	62.6	69.7	58.9	67.3
	UniPCQA (CodeT5)	**64.7**	**72.0**	**59.8**	**67.9**

Table 6: Results on Conversational QA. * denotes that the QR model is trained on the QR data from PACIFIC.

Table 5 summarizes the experimental results on the CQG task. Baseline methods can achieve relatively higher scores for ROUGE and F1, due to the similar expressions among different clarification questions. However, without using PLMs, CopyTrans. has a similar performance as the template-based method (BERT+TB). UniPCQA outperforms Q-GPT by a noticeable margin, indicating the effectiveness of the hybrid input sequence construction in such an CQG task based on hybrid contexts.

5.3 Task II: Conversational QA

Following previous studies (Vakulenko et al., 2021; Kim et al., 2021), we compare to both end-to-end and pipeline-based methods. End-to-end methods adopt a single QA or CQA model to encode the document and the whole conversation history, while pipeline-based methods decompose the CQA task into Query Rewriting (QR) and single-turn QA that are solved by different models.

5.3.1 Baselines and Evaluation Metrics

We adopt the following QR methods for comparisons: Original, Trans.++ (Vakulenko et al., 2021), and T5 (Lin et al., 2020; Kim et al., 2021). QR methods are trained on the QReCC dataset (Anantha et al., 2021). We include three QA/CQA models for comparisons: HAE (Qu et al., 2019), NumNet+ V2 (Ran et al., 2019), and TAGOP (Zhu et al., 2021). Details can be found in Appendix C.

In addition, we report the performance of using ground-truth self-contained questions (**Gold**) as input for single-turn QA models. This is equivalent to their performance on the TAT-QA dataset, including two latest results, *i.e.*, TaCube (Zhou et al., 2022) and POET-SQL (Pi et al., 2022).

Following previous studies on financial question answering (Zhu et al., 2021), we use EM and numeracy-focused F1 score (Dua et al., 2019) for the CQA evaluation.

5.3.2 Experimental Results

The CQA results are summarized in Table 6. There are several noticeable observations:

(1) A good QR model can lead to better performance on the CQA task for pipeline-based methods, where using ground-truth self-contained questions (Gold) can be regarded as an estimate of the upper bound for these methods. For a fair comparison, the QR models should not be trained on the QR data from PACIFIC. Therefore, pipeline-based methods barely work on PACIFIC when using an out-of-domain QR model. We also report the performance of each QR method in Appendix D.

(2) Conventional CQA methods, *e.g.*, HAE, fail to achieve promising results on PACIFIC, due to the inability of handling numerical reasoning.

(3) UniPCQA not only achieves the best performance on the original TAT-QA dataset, but also outperforms both pipeline-based and end-to-end methods on the CQA task, *i.e.*, PACIFIC. These results show the superiority of UniPCQA in handling both single-turn and conversational finance QA problems.

5.4 Overall Evaluation on PCQA

5.4.1 Baselines and Evaluation Metrics

Since this is a preliminary attempt on PCQA over hybrid contexts, we implement several alternative solutions for method comparisons, including two end-to-end generation methods, **DialoGPT** (Nakamura et al., 2022) and **FinQANet** (Chen et al., 2021b), as well as one pipeline-based method, **T5+TAGOP** (Zhu et al., 2021). As a union of CQG and CQA, we adopt their shared evaluation metrics for the evaluation of PCQA models, including EM

Method	Dev		Test	
	EM	F1	EM	F1
DialoGPT	25.2	32.1	22.6	30.7
FinQANet	40.3	47.2	38.0	45.5
T5+TagOp	49.6	56.1	46.6	52.3
UniPCQA (T5)	60.0	68.1	56.9	65.4
UniPCQA (T5) + MTL	61.6	70.3	58.7	67.5
UniPCQA (T5) + MTL + CV	62.1	70.7	59.4	68.3
UniPCQA (CodeT5)	63.2	71.4	60.4	68.5
UniPCQA (CodeT5) + MTL	64.0	72.1	61.0	69.4
UniPCQA (CodeT5) + MTL + CV	**64.5**	**72.6**	**61.9**	**70.2**
Human	-	-	86.3	92.1

Table 7: Overall evaluation on PCQA.

and numeracy-focused token-level F1. More details about baselines can be found in Appendix C.

5.4.2 Experimental Results

Table 7 presents the experimental results on the PCQA task. Among the baselines, due to the inability of performing numerical reasoning, DialoGPT performs much worse than FinQANet and T5+TagOp. T5+TagOp achieves a better performance than FinQANet, as its operators are specifically designed for the TAT-QA dataset, which can also be effectively applied to PACIFIC. Finally, UniPCQA substantially outperforms all these baselines, *i.e.*, 56.9/65.4 vs. 46.6/52.3 in EM/F1. In addition, UniPCQA is more flexible to different numerical calculations without the reliance on manually designed operators or programs and additional algorithms for executing the system outputs.

Among different variants of UniPCQA, CodeT5 achieves a better performance than T5 with the same size of model parameters, which indicates that UniPCQA effectively leverages the knowledge from the code-related pre-training tasks. The multi-task learning (MTL) improves the performance by explicitly learning from the clarification need labels. Further, the error propagation issue introduced by the MTL is alleviated by the Consensus Voting strategy (CV). However, compared with the performance of human experts (Human), there is still much room for improvement.

5.4.3 Detailed Analyses

Low-resource Evaluation Due to the high expenses in annotations, data is one of the largest bottlenecks for financial QA. We investigate how UniPCQA performs w.r.t different number of training data, by splitting 10% to 100% of training data for evaluation. As shown in Fig. 4, compared with

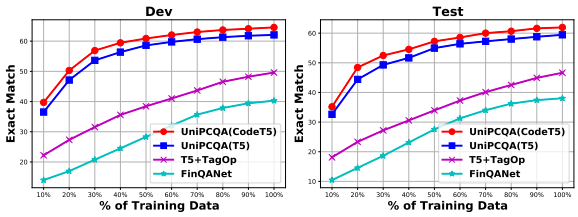

Figure 4: Performance w.r.t different % of training data.

	Table	Text	Table-text	Total
Span	**61.3**/57.1	**54.8**/50.8	70.8/**76.6**	**60.6**/59.2
Spans	61.4/**67.5**	**28.6**/23.8	**77.4**/72.6	**66.2**/65.7
Counting	**45.5**/36.4	-	**44.8**/41.4	**45.0**/40.0
Arithmetic	53.5/**58.7**	**27.3**/18.2	63.7/**67.4**	56.1/**60.7**
Question	55.6/**62.2**	**72.7**/63.6	64.8/**72.5**	61.5/**65.9**
Total	56.0/**59.5**	**54.6**/50.0	67.4/**70.6**	59.4/**61.9**

Table 8: Performance comparisons of UniPCQA intialized with $T5_{base}$/$CodeT5_{base}$, w.r.t different answer types and sources. The results are EM scores on test set.

FinQANet and T5+TagOp, UniPCQA can better transfer the knowledge from PLMs to achieve a much better performance in low-resource settings, especially from CodeT5.

Answer Type and Source Analysis We further compare the performance of UniPCQA w.r.t answer types and sources. As shown in Table 8, it can be observed that UniPCQA initialized with CodeT5 or T5 performs differently in terms of answer types and sources. CodeT5 performs much better than T5 on arithmetic questions (60.7 vs. 56.1), which indicates the effectiveness of reformulating the numerical reasoning process as code generation. This leads to better performance in the overall evaluation, since the majority in PACIFIC are arithmetic questions. Conversely, T5 has better performance on textual data as well as questions relying on span extraction, *e.g.*, Span and Spans.

Case Study for Consensus Voting Table 9 illustrates two examples where the Consensus Voting strategy remedies the mistakes made by greedy decode. In the first example, greedy decode generates a wrong formula, while Consensus Voting derives the correct answer by taking the plurality vote of the results from diverse numerical calculations. In the second example, the question is ambiguous as the period is not specified for the percentage change value. Greedy decode makes the wrong prediction on clarification needs, which affects the final answer. However, based on the plurality vote, most sampled outputs in Consensus Voting decide to ask a clarifying question, instead of directly calculating

Question 1	What is the average annual amount of it?		
Answer	2		
	#	Resp.	Sampled Outputs
Greedy	-	1.99	[clari.] False [resp.] (1.06+0.91+4.01)/3
CV 1	24	2	[clari.] False [resp.] (1.06+0.91+4.04)/3 [clari.] False [resp.] (1.06+4.04+0.91)/3
CV 2	12	1.99	[clari.] False [resp.] (1.06+0.91+4.01)/3
CV 3	4	3	[clari.] False [resp.] (1.06+0.91+4.04)/2
Question 2	What is the change in its amount as a percentage?		
Answer	Which period are you asking about?		
	#	Resp.	Sampled Outputs
Greedy	-	0.0	[clari.] False [resp.] (576523-576523)/576523
CV 1	22	[clari.] True [resp.] ['Which period are you asking about?']	
CV 2	10	0.0	[clari.] False [resp.] (576523-576523)/576523
CV 3	4	7.18	[clari.] False [resp.] (576523-537891)/537891
CV 4	2	-1.8	[clari.] False [resp.] (566523-576891)/576523

Table 9: Case study for Consensus Voting (CV). The underlined content denotes the mistake in the decoded output.

the percentage change value in a random period. More details about the case study can be found in Appendix F.

5.4.4 Error Analysis

In order to investigate the typical failure cases in UniPCQA, we randomly sample 100 error cases for analysis. As shown in Table 10, we categorize these failure cases into the following six groups:

- *Wrong Evidence* (34%): The model extracts wrong supporting evidences from the context.
- *Wrong Clarification Need Prediction* (18%): The model makes a wrong prediction on whether the user query requires clarification. More specific, 11% of all the failure cases are predicted to be unnecessary for clarification, while they are ambiguous in fact. And 7% of them vice versa.
- *Wrong Derivation* (13%): Although the model extracts all the necessary supporting evidences, the model fails to compute the answer with a correct derivation, *e.g.*, wrong formula or order.
- *Missing Evidence* (12%): Although the extracted evidences are correct, the model fails to extract all the required evidences from the context.
- *Wrong Clarification Question* (7%): The model generates a wrong clarification question that fails to clarify the ambiguity of the user query.
- *Other Errors* (18%): There are several other errors that are relatively acceptable, such as the scale error, missing symbols or missing punctuation marks.

Wrong Evidence (34%)	Q: What was the change in its amount in 2019 from 2018? G: 2.1 - 1.8 P: 2.1 - 1.3
Wrong Clarification Need Prediction (18%)	Q: In which year were the PSP payments larger? G: What kind of PSP payments are you asking about? P: 2019
Wrong Calculation (13%)	Q: How about their average salary? G: (1,000,000 + 650,000 + 440,000) / 3 P: (1,000,000 + 650,000 + 440,000) / 2
Missing Evidence (12%)	Q: What is the total stock-based compensation expense and unrecognized stock-based compensation expense in 2019? G: 3,711 + 4,801 + 1,882 P: 3,711 + 1,882
Wrong Clarification Question (7%)	Q: What is the total long-term debt due? G: Which period of payments due are you asking about? P: Which year are you asking about?
Other Errors (18%)	Q: What was the cash and cash equivalents in 2018? G: $148,502 P: 148,502

Table 10: Error Analysis (G: Ground-truth, P: Prediction).

Compared with the error analysis of the TAGOP model in the TAT-QA dataset (Zhu et al., 2021), it is worth noting that the percentage of errors that related to span extraction largely decreases from 84% to 46%. However, there are about 25% and 13% of errors that are related to the clarification question generation task and the numerical calculation, respectively.

6 Conclusions

In this paper, we present a new dataset, PACIFIC, for proactive conversational question answering over a hybrid context of tables and text. Accordingly, we define the problem of Proactive Conversational Question Answering that combines clarification question generation and conversational question answering. In addition, we reformulate the numerical reasoning process as code generation and recast all sub-tasks in PCQA into a Seq2Seq problem solved by a unified model, UniPCQA. Extensive experiments show that the PACIFIC dataset is very challenging and demonstrate the need to build models that can handle hybrid input and output formats as well as diverse numerical reasoning.

Ethical Considerations

The PACIFIC dataset was built from the TAT-QA dataset, which is publicly available. The authors of the TAT-QA dataset paper have allowed us to utilize the dataset for further construction. We will provide open access to our dataset and code for future studies via https://github.com/dengyang17/PACIFIC/.

Limitations

In this section, we analyze the limitations from the perspectives of both the constructed dataset and the proposed method.

Limitations of PACIFIC Dataset

Since PACIFIC is the first CIS dataset in finance domain as well as the first proactive CQA dataset, there are inevitably some limitations and room for further improvement.

- **Numerical Reasoning**. Similar to other popular NLP datasets that require numerical reasoning, such as DROP (Dua et al., 2019) and FinQA (Chen et al., 2021b), the questions in PACIFIC only require some basic numerical calculations, including arithmetic operations, counting, and comparison. In the future, with the advance in the model capability of numerical reasoning, it would be better to add questions that require more complicated numerical calculations.

- **Clarification Question**. In the clarification turn, PACIFIC only provides the clarification question. In some cases, it is beneficial to further provide the candidate options for better clarifying the uncertainty (Xu et al., 2019; Zamani et al., 2020b). Besides, in the data creation process, we construct ambiguous questions that contain only one missing information for guaranteeing the objectivity of the clarification question annotations. However, it is also worth studying the situation where there are multiple missing information for clarification.

- **Multimodality**. Although the PACIFIC dataset is based on a hybrid context of tables and text, there are more diverse information in the real-world financial documents with different modalities, such as images, charts, etc. It is necessary to consider more comprehensive QA or CQA datasets and problem settings for real-world applications in finance domain.

Limitations of UniPCQA

The error analysis in Section 5.4.4 reveals some limitations in the proposed method. Currently, the capability of numerical reasoning in UniPCQA relies on the pre-trained language models. In the future, we would like to investigate post-training strategies to transfer task-adaptive or domain-specific knowledge from other post-training tasks for further improving this capability. In addition, due to the heterogeneous input and output content, it would also be beneficial to investigate more robust prompt-based learning approaches for better learning the relationships among different types of information.

References

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail S. Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *EMNLP 2021*, pages 4473–4484.

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 *NAACL-HLT 2019*, pages 2357–2367.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *NAACL-HLT 2021*, pages 520–534.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *EMNLP-IJCNLP 2019*, pages 5946–5951.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021a. Open question answering over tables and text. In *ICLR 2021*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *EMNLP 2020*, Findings of ACL, pages 1026–1036.

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. 2021b. Finqa: A dataset of numerical reasoning over financial data. In *EMNLP 2021*, pages 3697–3711.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *EMNLP 2018*, pages 2174–2184.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational question answering on heterogeneous sources. In *SIGIR 2022*, pages 144–154.

Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *SIGIR 2021*, pages 1431–1441.

Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Wai Lam, and Ying Shen. 2022a. Contextualized knowledge-aware attentive neural network: Enhancing answer selection with knowledge. *ACM Trans. Inf. Syst.*, 40(1):2:1–2:33.

Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. 2022b. A unified multi-task learning framework for multi-goal conversational recommender systems. *CoRR*, abs/2204.06923.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019*, pages 4171–4186.

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 *NAACL-HLT 2019*, pages 2368–2378.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *ACL 2018*, pages 889–898.

Chang Gao and Wai Lam. 2022. Search clarification selection via query-intent-clarification graph attention. In *ECIR 2022*, pages 230–243.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *ACL 2020*, pages 946–958.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-coqa: Clarifying ambiguity in conversational question answering. In *AKBC 2021*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR 2020*.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *EMNLP-IJCNLP 2019*, pages 1596–1606.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *ACL 2017*, pages 1821–1831.

Gangwoo Kim, Hyunjae Kim, Jungsoo Park, and Jaewoo Kang. 2021. Learn to resolve conversational dependency: A consistency training framework for conversational question answering. In *ACL/IJCNLP 2021*, pages 6130–6141.

Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to identify follow-up questions in conversational question answering. In *ACL 2020*, pages 959–968.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020a. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *WSDM 2020*, pages 304–312.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020b. Interactive path reasoning on graph for conversational recommendation. In *KDD 2020*, pages 2073–2083.

Wenqiang Lei, Yao Zhang, Feifan Song, Hongru Liang, Jiaxin Mao, Jiancheng Lv, Zhenglu Yang, and Tat-Seng Chua. 2022. Interacting with non-cooperative user: A new paradigm for proactive dialogue policy. In *SIGIR 2022*, pages 212–222.

Moxin Li, Fuli Feng, Hanwang Zhang, Xiangnan He, Fengbin Zhu, and Tat-Seng Chua. 2022a. Learning to imagine: Integrating counterfactual thinking in neural discrete reasoning. In *ACL 2022*, pages 57–69.

Yongqi Li, Wenjie Li, and Liqiang Nie. 2022b. MMCoQA: Conversational question answering over text, tables, and images. In *ACL 2022*, pages 4220–4231.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *CoRR*, abs/2004.01909.

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.

Zhongkun Liu, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Maarten de Rijke, and Ming Zhou. 2021. Learning to ask conversational questions by optimizing levenshtein distance. In *ACL/IJCNLP 2021*, pages 5638–5650.

Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhu Chen, and William Yang Wang. 2022. HybriDialogue: An information-seeking dialogue dataset grounded on tabular and textual data. In *Findings of ACL: ACL 2022*, pages 481–492.

Kuntal Kumar Pal and Chitta Baral. 2021. Investigating numeracy learning ability of a text-to-text transfer model. In *Findings of ACL: EMNLP 2021*, pages 3095–3101.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Yan Gao, Qiang Fu, Jian-Guang Lou, and Weizhu Chen. 2022. Reasoning like program executors. *CoRR*, abs/2201.11473.

Minghui Qiu, Xinjing Huang, Cen Chen, Feng Ji, Chen Qu, Wei Wei, Jun Huang, and Yin Zhang. 2021. Reinforced history backtracking for conversational question answering. In *AAAI 2021*, pages 13718–13726.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with history answer embedding for conversational question answering. In *SIGIR 2019*, pages 1133–1136.

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.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. Numnet: Machine reading comprehension with numerical reasoning. In *EMNLP-IJCNLP 2019*, pages 2474–2484.

Gonçalo Raposo, Rui Ribeiro, Bruno Martins, and Luísa Coheur. 2022. Question rewriting? assessing its importance for conversational question answering. In *ECIR 2022*.

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

Ivan Sekulic, Mohammad Aliannejadi, and Fabio Crestani. 2021. Towards facet-driven generation of clarifying questions for conversational search. In *ICTIR 2021*, pages 167–175.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *ACL/IJCNLP 2021*, pages 1451–1465.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodal{qa}: complex question answering over text, tables and images. In *ICLR 2021*.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro A. Szekely. 2021. Representing numbers in NLP: a survey and a vision. In *NAACL-HLT 2021*, pages 644–656.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question rewriting for conversational question answering. In *WSDM 2021*, pages 355–363.

Jian Wang and Wenjie Li. 2021. Template-guided clarifying question generation for web search clarification. In *CIKM 2021*, pages 3468–3472.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171.

Yue Wang, Weishi Wang, Shafiq R. Joty, and Steven C. H. Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *EMNLP 2021*, pages 8696–8708.

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

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking clarification questions in knowledge-based question answering. In *EMNLP-IJCNLP 2019*, pages 1618–1629.

Hamed Zamani, Susan T. Dumais, Nick Craswell, Paul N. Bennett, and Gord Lueck. 2020a. Generating clarifying questions for information retrieval. In *WWW 2020*, pages 418–428.

Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020b. MIMICS: A large-scale data collection for search clarification. In *CIKM 2020*, pages 3189–3196.

Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. *CoRR*, abs/2201.08808.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *System Demonstrations, ACL 2020*, pages 270–278.

Fan Zhou, Mengkang Hu, Haoyu Dong, Zhoujun Cheng, Shi Han, and Dongmei Zhang. 2022. Tacube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. *CoRR*, abs/2205.12682.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. 2022. Towards complex document understanding by discrete reasoning. *CoRR*, abs/2207.11871.

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 *ACL/IJCNLP 2021*, pages 3277–3287.

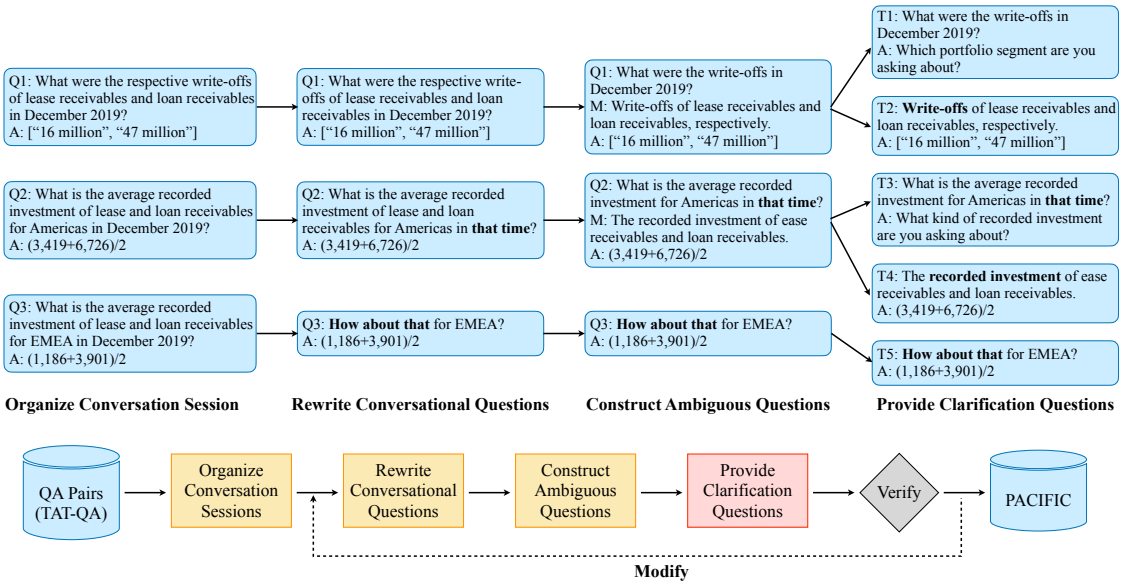

Figure 5: Overall Pipeline for PACIFIC Creation with Examples.

Appendix

A Pipeline of Dataset Creation

Fig. 5 presents the illustration of overall pipeline for the PACIFIC dataset creation with examples. In financial question answering (Zhu et al., 2021; Chen et al., 2021b), the user query is supposed to be informative and complicated with multiple constraints. Therefore, it is inappropriate to adopt the traditional way of decomposing a complex single-turn question into multiple conversational questions with limited information for constructing a financial conversational question answering dataset.

To this end, we employ a different pipeline to create the PACIFIC dataset. As described in Section 3.1, there are totally four steps for the creation of the PACIFIC dataset, including (1) Organize Conversation Sessions[2], (2) Rewrite Conversational Questions, (3) Construct Ambiguous Questions[3], and (4) Provide Clarification Questions[4]. This annotation pipeline not only increases efficiency in the dataset construction, but also guarantees the quality and preserves the difficulty of the dataset with expert-annotated answers and informative user queries for financial CQA.

B Implementation Details

The pre-trained weights of T5 and CodeT5 are initialized using HuggingFace[5]. We use the same hyper-parameter settings for different initialization. The learning rate and the weight decay rate are set to be 5e-5 and 0.01, respectively. The max source sequence length and the max target sequence length are 1280 and 128, respectively. We train the model up to 15 epochs with mini-batch size of 4, and select the best checkpoints based on the EM score on the validation set. We train the model on three NVIDIA Tesla V100 GPUs with 32GB RAM.

For a fair comparison, all the PLM-based baselines adopt the version of PLMs with a similar size of model parameters as $T5_{base}$ (220M) and $CodeT5_{base}$ (220M). For example, BERT, RoBERTa, and GPT-2 based methods adopt $BERT_{large}$ (340M), $RoBERTa_{large}$ (355M), and $GPT\text{-}2_{medium}$ (345M), respectively.

C Compared Baselines

Following Zhu et al. (2021), we simply flatten the table into a sequence by row followed by tokens from the paragraphs for all the baselines.

[2]Entities in the question will be automatically highlighted for the convenience of annotators, through lexical matching with the nouns in paragraphs and tables.

[3]Only one entity in the original question is randomly chosen to be omitted for the annotators to construct the ambiguous question.

[4]Similar to Zamani et al. (2020a), we provide several templates for the annotators to provide clarification questions using the omitted entity. This guarantees the objectivity of the clarification question annotations

[5]https://huggingface.co

C.1 Clarification Need Prediction

We fine-tune the vanilla $\text{BERT}_{\text{large}}$ (Devlin et al., 2019) and $\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019) based classifiers for the CNP task.

C.2 Clarification Question Generation

We compare to the following CQG baselines:

- **Template-based Question Generation (TB)**: A template-based approach (Zamani et al., 2020a) to generating clarifying questions produces a question by simply filling a slot in a pre-defined question, i.e., "What kind of _ are you asking about?". And we adopt a fine-tuned BERT-based span extraction model to extract the slot value from the document.
- **CopyTrans.** (Wang and Li, 2021) adopts the Transfomer-based encoder-decoder with the copy mechanism for the CQG task.
- **Q-GPT** (Sekulic et al., 2021) fine-tunes GPT-2 (Radford et al., 2019) to generate clarifying questions.

C.3 Query Rewriting

We adopt the following QR baselines for evaluation:

- **Original**: Use the original conversational question at the current conversation turn without query rewriting, which is often regarded as the lower bound for the pipeline-based CQA evaluation.
- **Trans.++**: A Transformer-based QR model (Vakulenko et al., 2021) initialized with the weights of pre-trained GPT-2 model (Radford et al., 2019).
- **T5**: Following (Lin et al., 2020; Kim et al., 2021), we adopt a T5-based sequence generator (Raffel et al., 2020) as a baseline QR model.

C.4 Conversational Question Answering

We adopt the following QA and CQA baselines for evaluation:

- **NumNet+ V2**[6]: A numerical QA model utilizes a numerically-aware graph neural network to consider the comparing information and performs numerical reasoning over texts (Ran et al., 2019).
- **TagOp**[7]: A RoBERTa-based QA model adopts sequence tagging to extract information and applies numerical reasoning over tables and texts with a set of aggregation operators (Zhu et al., 2021).
- **BERT+HAE**[8]: A BERT-based CQA model adds the history answer embeddings (HAE) to the BERT's word embeddings (Qu et al., 2019). Since there is no numerical reasoning module in this method, the output only contains the extracted spans from the documents.

C.5 Proactive Conversational Question Answering

We adapt the following methods for the evaluation of the overall PCQA problem:

- **DialoGPT** Following Nakamura et al. (2022), we fine-tuned a pre-trained DialoGPT model (Zhang et al., 2020) for dialogue response generation. Similar to BERT+HAE, the target sequence only contains the required spans from the document without numerical calculations.
- **FinQANet**[9] (Chen et al., 2021b) A retriever using BERT first retrieves the supporting facts from the document, then a generator combining RoBERTa and LSTM generates the response, which can be either a schema-based program for CQA or a natural language question for CQG.
- **T5+TagOp** A pipeline-based method first uses T5 (Raffel et al., 2020) for the sub-tasks of CNP and CQG. If it is not required for clarification, TagOp (Zhu et al., 2021) is adopted to produce the answer as an end-to-end CQA method.

Note that for two end-to-end methods, including DialoGPT and FinQANet, there is no sub-task of CNP, while the whole PCQA problem can be regarded as the response generation problem in dialogue systems.

D Evaluation on Query Rewriting

Following previous studies (Vakulenko et al., 2021), we adopt ROUGE-1 (Recall) and EM for the evaluation of QR models.

The performance of each Query Rewriting (QR) method is presented in Table 11. Due to the substantial difference between financial CQA and general

[6]https://github.com/llamazing/numnet_plus

[7]https://github.com/NExTplusplus/TAT-QA

[8]https://github.com/prdwb/bert_hae

[9]https://github.com/czyssrs/finqa

QR Model	Train Set	Dev		Test	
		ROUGE	EM	ROUGE	EM
Original	-	68.8	43.3	69.9	43.6
Trans.++	QReCC	78.2	28.9	77.4	30.2
T5	QReCC	79.5	28.7	80.2	29.7
Trans.++	PACIFIC	93.6	76.5	92.9	76.4
T5	PACIFIC	94.9	80.1	94.8	79.7

Table 11: Results on Query Rewriting.

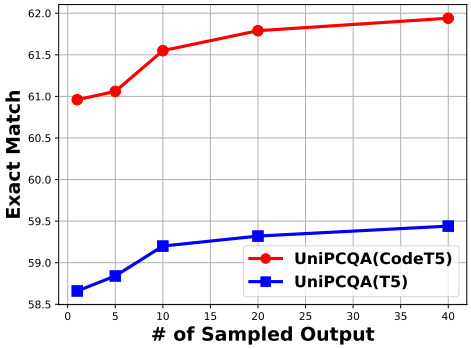

Figure 6: Performance w.r.t different number of sampled output.

CQA, *i.e.*, PACIFIC and QReCC, the QR models trained on QReCC perform poorly in PACIFIC.

E Effect of Sampling Number

Fig. 6 shows the performance of Consensus Voting in terms of different number of sampled outputs from the decoder, ranging from [1, 5, 10, 20, 40]. Due to the restriction of experimental environment, the maximum number of sample outputs is set to be 40. Experimental results show that a higher number of sampled outputs generally leads to a better performance for both T5 and CodeT5-based UniPCQA models, which indicates the effectiveness of the plurality voting in the Consensus Voting strategy for alleviating the error propagation issue.

F Detailed Case Study

Fig. 7 presents the details of the case study in Table 9, including the grounded document and the conversation history. At the current turn **T6**, the user query is "What is the change in its amount as a percentage?", where "it" refers to "*Total Revenue*" at the previous turn **T5**. As shown in the tabular context, the amount of "*Total Revenue*" is recorded in three years, from 2017 to 2019. Due to the uncertainty of the concerned period, the user query, "What is the change in its amount as a percentage?", is ambiguous under this context.

The following table sets forth our sources of revenue for each of the periods indicated (in thousands, except for percentages):

	Year Ended December 31,		
	2019	2018	2017
Subscription revenue	$542,968	$473,052	$396,764
Percentage of subscription revenue to total revenue	94.2%	87.9%	82.3%
Professional services revenue	$33,555	$64,839	$85,221
Percentage of professional services to total revenue	5.8%	12.1%	17.7%
Total revenue	$576,523	537,891	481,985

Subscription revenue increased by $69.9 million, or 15%, in 2019 when compared to 2018. Subscription revenue growth on a constant currency basis increased 16% in 2019 when compared to 2018. The increase was attributable to new business, which includes new customers, upsells, cross-sells, and renewals from existing customers. … (Omit for simplicity)

#	Question	Response	Derivation
T1	Why did subscription revenue increase by $76.3 million between 2018 and 2017?	["The increase was attributable to … from existing customers."]	
T2	What was its amount in 2019?	["$542,968"]	
T3	What was the change in its amount between 2018 and 2019?	69916 thousand	$542,968-$473,052
T4	How about that in Professional services revenue?	-31284 thousand	$33,555-$64,839
T5	What was Total Revenue in 2019?	["$576,523"]	
T6	What is the change in its amount as a percentage?	Which period are you asking about?	

	#	Response	Derivation
Greedy Decode	-	0.0	[clar.] False [resp.] ($576,523-$576,523)/$576,523
CV 1	22	Which period are you asking about?	[clar.] True [resp.] ['Which period are you asking about?']
CV 2	10	0.0	[clar.] False [resp.] ($576,523-$576,523)/$576,523
CV 3	4	7.18	[clar.] False [resp.] (576,523-537,891)/537,891
CV 4	2	-1.8	[clar.] False [resp.] ($566,523-$576,891)/$576,523
CV 5	1	-1.93	[clar.] False [resp.] ($576,523-$586,891)/$537,891
CV 6	1	-0.01	[clar.] False [resp.] ($576,523-$576,593)/$576,523

Figure 7: Detailed Case Study for Consensus Voting (CV).

In the inference, the decoder will be confused about which figures are supposed to be extracted from the grounded document. We can observe that Greedy Decode generates a wrong derivation, which may answer the query "What is the change in its amount as a percentage from 2019 to 2019?". Similarly, for CV 3, the generated derivation is supposed to answer the query "What is the change in its amount as a percentage from 2018 to 2019?". However, at this conversation turn, the system is not aware of the specific period that the user is asking. Therefore, all these derivations with a random period are incorrect. Overall, with the plurality voting, Consensus Voting effectively alleviates this kind of issue, since it would be difficult for the sampled derivation outputs to make a consensus for an ambiguous question.