Demonstration of FeVisQA: Free-Form Question Answering over Data Visualization

Yuanfeng Song*†, Jinwei Lu‡, Xuefang Zhao†, Raymond Chi-Wing Wong*, Haodi Zhang‡
*HKUST, Hong Kong, China †AI Group, WeBank Co., Ltd, Shenzhen, China ‡Shenzhen U., Shenzhen, China
*{songyf, raywong}@cse.ust.hk

***Abstract*—Question Answering (QA) systems play a vital role in knowledge acquisition. CodeQA refers to question answering (QA) over source code for code comprehension purpose. However, existing CodeQA studies mainly focus on questions related to general-purpose programming languages (GPLs) (e.g., Java and Python), and no study has been conducted on QA over declarative visualization languages (DVLs) (e.g., Vega-Lite), a kind of programming languages used for creating data visualization (DV). DVLs enjoys specific grammars that are instinct different from GPLs. This demonstration presents the first neural-based QA system for DVL, FEVISQASYSTEM. FeVisQASystem is based on a new task named FEVISQA, short for Free-form QA over data Visualization, which takes natural language questions and DV specification as inputs to predict the answers to the questions. As a particular case of the CodeQA task, FeVisQA enables people to better comprehend data and its DVs by conducting logical reasoning when answering these questions. Although research on question-answering and machine reading comprehension is progressing quickly, little attention has previously been paid to FeVisQA. This new system and the task can serve as a helpful pioneering study for DV comprehension. The video can be accessed via https://1drv.ms/f/s!Ah2vhbolPBFMhk6jTYOtaIRnLC2K?e=0kJqOq**

***Index Terms*—Question Answering, Data Visualization, FeVisQA, Declarative Visualization Language**

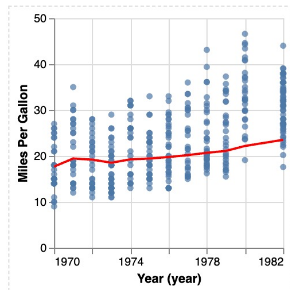
(a) A Car Dataset

Name	Miles_Per_Gallon	Cylinders	…	HorsePower
"chevrolet chevelle malibu"	18	8	…	130
…	…	…	…	…

(b) A DV in Vega-Lite for the Car Dataset

```
{
  "data": {"url": "data/cars.json"},
  "transform": [
    {"filter": "datum.Miles_per_Gallon !== null"},
    {"timeUnit": "year", "field": "Year", "as": "year"},
    {
      "window": [{
        "op": "mean",
        "field": "Miles_per_Gallon",
        "as": "Average_MPG"
      }],
      "sort": [{"field": "year", "order": "ascending"}],
      "ignorePeers": false,
      "frame": [null, 0]
    }
  ],
  "layer": [
    {
      "mark": "circle",
      "encoding": {
        "x": {"timeUnit": "year", "field": "Year"},
        "y": {"field": "Miles_per_Gallon", "type":
"quantitative"}
      }
    },
    {
      "mark": {"type":"line", "color": "red"},
      "encoding": {
        "x": {"timeUnit": "year", "field": "Year"},
        "y": {
          "field": "Average_MPG",
          "type": "quantitative",
          "title": "Miles Per Gallon"
        }
      }
    }
  ]
}
```

(c) The Corresponding DV Chart Rendered by Vega-Lite Engine.



Question 1: What is the meaning of this DV?
Answer 1: A line graph showing how the overall average Miles Per Gallon for how the MPG has changed over the years.
Question 2: It this DV suitable for this dataset?
Answer 2: Yes
Question 3: What is the meaning of the transform operation?
Answer 3: The window transform is used to calculate the averaged MPG for vehicles over the years.

Fig. 1. Examples of a Dataset, a DV Specification, and three QA Pairs of our proposed FeVisQA task.

I. Introduction

Designing Question Answering (QA) systems is an important research direction in database community [1]–[5], since these systems play vital roles in knowledge acquisition. CodeQA is an essential task that focuses on answering questions related to programming code for source code comprehension and educational purpose [6]. It significantly promotes the development of programming learning for educational purposes. However, limited efforts have been spent on question answering (QA) over Declarative Visualization Languages (DVLs) (e.g., Vega-Lite [7] and ECharts [8]) for Data Visualizations (DVs), which enjoy different grammars that are used to control the visual representations given a massive dataset.

In this demonstration, we design a novel QA system named FEVISQASYSTEM that focuses on automatically answering questions related to DV domain. FEVISQASYSTEM is based on a new task proposed by us named **FeVisQA** [9], referring to Free-form Question Answering over data Visualization. As shown in Fig. 1[1], given a DV specification (which is quite complicated) of a dataset and a question, the objective of the FeVisQA task is to predict the corresponding textual answer. FeVisQA could be considered as a particular case of the general CodeQA task [6]. Serving the same goal of facilitating programming learning in education, FeVisQA enables people to better comprehend DVs by conducting logical reasoning when answering questions. The underlying techniques behind the FEVISQASYSTEM system is a multi-modal neural network named FeVisQANet, which is composed of novel encoder and decoder structures that dedicated to specifications.

II. Related Work

In this section, we mainly introduce the closely related studies from two areas, DV and CodeQA.

Data Visualization (DV). Numerous institutions adopt DV to facilitate their strategic operations due to its excellent visual representation capacity. At the same time, a substantial amount of declarative visualization languages (DVLs) are released in the market, including Vega-Lite [7], ZQL [10], and ECharts [8]. Due to the difficulty of using these DVLs, various DV-related studies have also been carried out in the database community to provide user-friendly interfaces and

[1]The dataset and DV example are from the official website of Vega-Lite in *https://vega.github.io/vega-lite/docs/window.html*

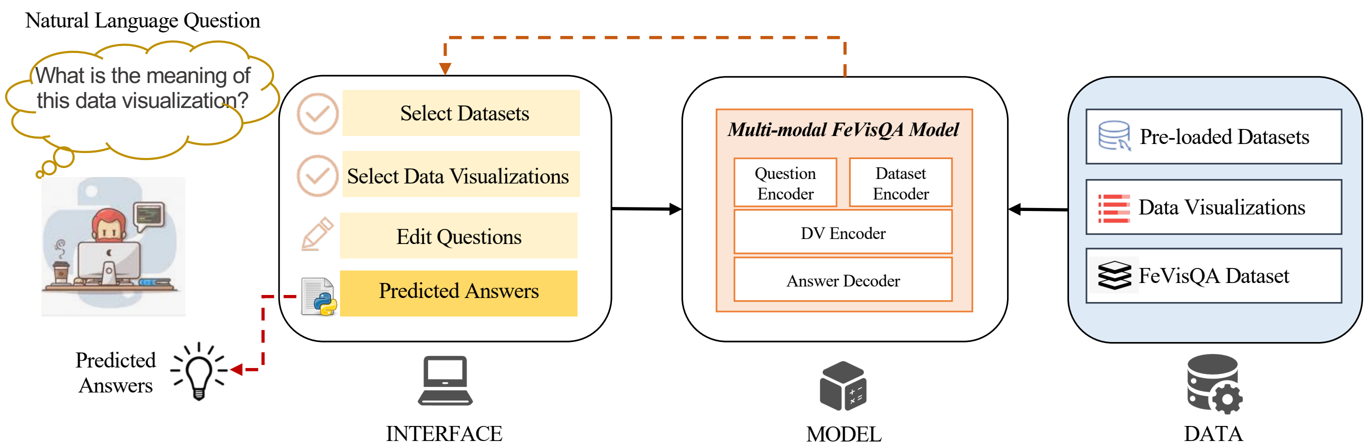


Fig. 2. The Overview of the Neural-based QA Architecture of FeVisQASystem.

tools,with representative studies like [11]–[13]. These studies focus on a task named text-to-vis [11], [14], which aims to translate natural language questions into DV automatically. Our proposed FeVisQA task is another DV-related task that significantly enlarges the DV family. Although research on question-answering and machine reading comprehension is progressing quickly, little attention has previously been paid to FeVisQA. This new task and system would be a pioneer study in this direction.

Question Answering (QA). Different kinds of QA systems have been designed and proposed in the database community recently, like [1]–[5]. CodeQA is a particular QA task that answers questions related to source code (e.g., Java, Python), with the purpose of source code comprehension. Representative studies in this field includes [6], [15]. For example, Liu *et al.* [6] released a CodeQA dataset that contains QA pairs on general-purpose programming languages (GPLs) like Java and Python. Then, they further designed several baseline methods to tackle this CodeQA task. Lee *et al.* [15] released another CS1QA dataset that aims to predict the question type and the relevant code snippet, given the question and the code and retrieving an answer from the annotated corpus. However, different from GPLs, composing DV requires defining specifications using the grammar of some DVLs. Moreover, the execution result of a specification is a chart, which differs from GPLs. These properties make FeVisQA instinct different from exsiting studies for the well-formed CodeQA task.

III. System Architecture

We are ready to briefly illustrate the overview of the proposed FeVisQASystem, mainly from the following aspects - the task, the system overview, the FeVisQA model, and the performance analysis.

A. The FeVisQA Task

FeVisQA refers to Free-form Question Answering over data Visualization. Specifically, given a dataset D, a data visualization v (a JSON object in the form of a specification in any DVL), and a question q, the FeVisQA task aims to automatically predict the textual answer a. FeVisQA enables users to get a better comprehension of DVs by conducting logic reasoning when answers these questions. Then, the complete training set can be represented as $\mathcal{T} = \{D^{(o)}, v^{(o)}, q^{(o)}, a^{(o)}\}_{o=1}^{N}$, where N is the dataset size. The desired model could be represented as $f(D, v, q) \rightarrow a$.

B. Overview of the proposed FeVisQASystem

As shown in Fig. 2, the designed FeVisQASystem could be categorized into three layers - *Interface*, *Model*, and *Data*. The interface layer focuses on the interaction between the users and the system, i.e., input the question, select the dataset and specification to be analyzed, and examine the answers. The model layer offers the fundamental neural models for FeVisQA in FeVisQASystem. Specifically, we designed a multi-modal neural network which predicts the answers to the given questions. The data layer saves the datasets (i) The QA dataset for building the FeVisQA models and (ii) the Datasets and Specifications to be analyzed.

C. The Multi-modal Neural Network

As shown in Fig. 3, the objective of a FeVisQANet model is to understand the DV specification and the question, and then predict the corresponding answers. As such, our proposed FeVisQANet model can roughly be divided into four components, namely a *dataset encoder*, a *question encoder*, a *DV specification encoder*, and an *answer decoder*. The dataset encoder incorporates the dataset header (i.e., schema of the database) and also samples some rows in the dataset and converts them into embeddings. To promote the performance, we also incorporate the pre-trained language models (PLMs, i.e., TaPas [16]) dedicated to tabular data as a part of the dataset encode. The question-encoder uses a Transformer-based architecture to convert the question into some hidden representations. At the same time, the most important part, the DV specification encoder, converts the specification into various format to preserve the structural information contained. The answer decoder aims to generate textual response to answer the question. A comprehensive introduction of this FeVisQANet model can be found in [9].

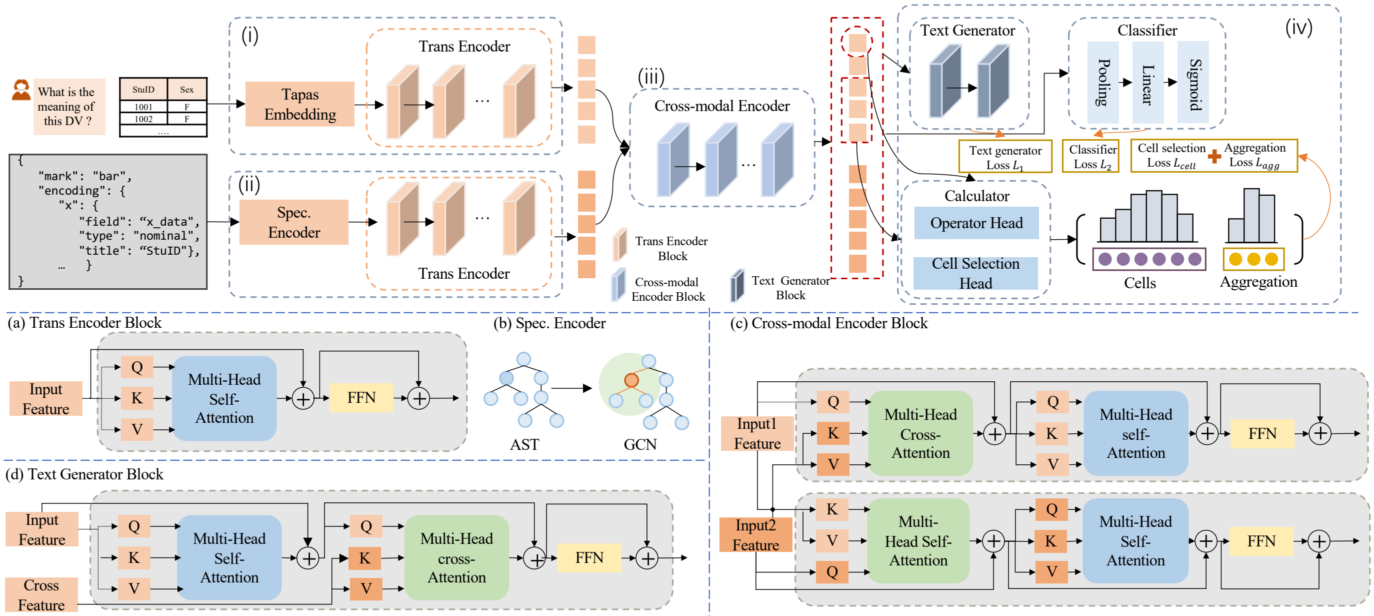


Fig. 3. Network Structure of our Proposed FeVisQANet Model, where (i) a *Question & Data Encoder* first loads Tapas embeddings for words and then encodes the dataset and question information using a Transformer-based structure, (ii) a *Specification Encoder* first converts the DV specification into AST and then encodes it with a GCN structure, (iii) a *Cross-modal Encoder* fuses the outputs from the previous two encoders, and (iv) an *Adaptive Decoder* generates corresponding answers to different question type.

D. Performance Analysis

We also conduct performance comparisons among several popular baselines in the CodeQA area, namely **Seq2Seq**, **Transformer** [17], **DualEncoder** [6], and **CodeBERT** [6]. The goal of the proposed FeVisQANet model is to predict answers to questions related to DV, we use the common metrics like *BLEU* and *Accuracy* as our primary indicators. The same training set is used to construct these models, and then we use the same testing set to test the performance, with the results shown in Table I.

TABLE I
PERFORMANCE COMPARISON

Method	BLEU	ROUGE	METEOR	EM
Seq2Seq	0.0100	0.3623	0.4187	53.71%
Transformer	0.0124	0.6203	0.3502	55.83%
DualEncoder	0.0129	0.5869	0.3297	52.68%
FeVisQANet	0.1233	0.7686	0.6091	70.59%

The performance of the vanilla Seq2Seq or Transformer approaches is not competitive compared with the advanced ones (i.e., CodeBERT and FeVisQANet), validating the necessity of exploring multi-modal information and techniques in this task. In a nutshell, among all the multi-modal methods, FeVisQANet could achieve the best performance since it remarkably preserves the rich context information in the dataset, question and DV specification and generate the accurate answers with advanced network structure.

IV. Demonstration Overview

In this demonstration, users could interact with FeVisQASystem to ask DV related questions. The system's user interface (UI) is shown in Fig. 4, with the main functionalities listed as follows.

Dataset Selection: The system preloads various datasets to be analyzed. The system enables the user to select the corresponding dataset that the user wants to query.

Data Visualization Selection: The system also includes various pre-defined DV specification to be visualized. The system enables the user to select some of them.

Natural Language Question Input: FeVisQASystem enables users to interact with the system using natural language questions.

Predicted Answer: After the dataset, the DV specification and the question are fixed, the FeVisQASystem will generate the predicted answer.

Case Study: A real case is also given in Table II for better illustration. This case contains the datasets (Fig. (a)), the DV specification (Fig. (b)) and the questions (Fig. (c)) and the predicted answer (Fig. (d)). We can see that our system

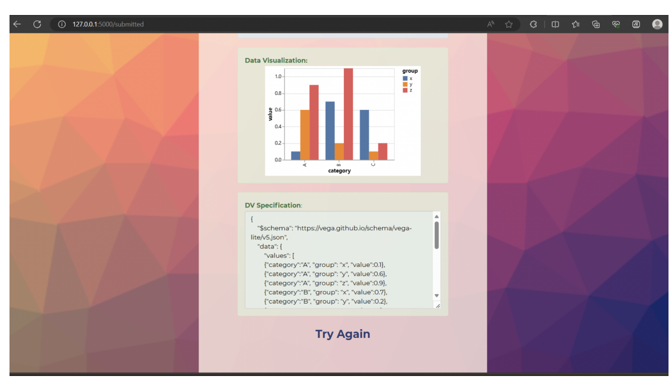


Fig. 4. The user interface of our proposed FeVisQASystem.

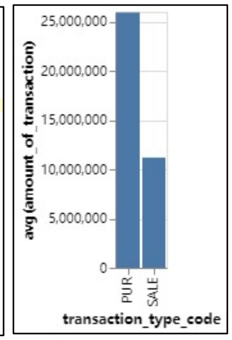
(a) The Dataset

transaction_id	investor_id	transaction_type_code	date_of_transaction	amount_of_transaction	share_count
1	6	SALE	1988/9/16 19:02	302507.7	8718572
2	18	PUR	1982/6/6 17:19	27.257	9
3	2	SALE	1979/4/27 06:03	48777.97	8580
			...		
10001	14	PUR	2001/11/28 15:06	4.5263	8040

(b) The Specification

```
{
  "data": {
    "values": [
      {
        "x_data": "PUR",
        "y_data": 25945880.3716125
      },
      ...
    ]
  },
  "mark": "bar",
  "encoding": {
    "x": {
      "field": "x_data",
      "type": "nominal",
      "title": "transaction_type_code",
      "sort": "ascending"
    },
    "y": {
      "field": "y_data",
      "type": "quantitative",
      "title": "avg(amount_of_transaction)"
    }
  }
}
```

(c) The Chart



(d) The QA Predicted by Various Methods

No.	Type	Question	Answer: Ground-Truth	Seq2Seq	Transformer	DualEncoder	FeVisQANet
1	type1	what is the meaning of this dv ?	create a bar chart showing mean amount of transaction across transaction type code , display by the transaction_typ_code in asc .	give me a bar chart to show the average experience working length of journalist working on different role type , show by the y - axis from low to high .	what are the average amount of transaction of transaction , and what is the average amount of amount for each ? , could you show by the name in ascending ?	show the average of amount for different statement detail in a bar chart , and could you show from low to high by the bar ?	show the transaction type code and the average amount of transaction with a bar chart , sort by the x - axis from low to high .
2	type2	is this dv a suitable one for the given dataset ?	yes	yes	yes	yes	yes
3	type3	what is the difference between the maximum value and minimum value of y - axis ?	14747713.13	2449.54	7.388888889	4696.69	14747713.13
4	type3	what is the total number of avg (amount _ of _ transaction) ?	37144047.61	5813.166667	64.72222222	815111.69	37144047.61
5	type3	"what is the value of transaction _ type _ code when avg (amount _ of _ transaction) is 25945880.3716125 ?	PUR	open	refund	result	PUR

TABLE II
EXAMPLES OF QUESTIONS ON A GIVEN DV SPECIFICATION & THE PREDICTED ANSWERS BY FEVISQASYSTEM.

correctly predicts the desired responses for various questions.

V. CONCLUSION

In this paper, we demonstrate a novel QA system named FEVISQASYSTEM to answer DV-related questions. We summarize our contributions as (i) a new QA task named FeVisQA with a constructed dataset is proposed to boost the development this field; (ii) a novel multi-modal neural network named FeVisQANet is proposed to validate the rationale of the proposed task and shows its superiority; (iii) a QA system named FEVISQASYSTEM is developed to demonstrate the practicability of the proposed FeVisQA task. We believe FeVisQA task would advance the field of intelligent DV and also inspire more Natural Language Processing for Database (NLP4DB) studies.

Acknowledgement. The research of Haodi Zhang is supported by the Guangdong Basic and Applied Basic Research Foundation (2022A1515010675). Haodi Zhang is the corresponding author.

REFERENCES

[1] W. Zheng, J. X. Yu, L. Zou, and H. Cheng, "Question answering over knowledge graphs: question understanding via template decomposition," *Proceedings of the VLDB Endowment*, pp. 1373–1386, 2018.

[2] C. Wu, E. Hu, K. Zhan, L. Luo, X. Zhang, H. Jiang, Q. Wang, Z. Cao, F. Yu, and L. Chen, "Triple-fact retriever: An explainable reasoning retrieval model for multi-hop qa problem," in *ICDE*, 2022, pp. 1206–1218.

[3] Z. Ji, E. Choi, and P. Gao, "A knowledge base question answering system for cyber threat knowledge acquisition," in *ICDE*, 2022, pp. 3158–3161.

[4] A. Orogat and A. El-Roby, "Cbench: Demonstrating comprehensive evaluation of question answering systems over knowledge graphs through deep analysis of benchmarks," *Proceedings of the VLDB Endowment*, pp. 2711–2714, 2021.

[5] X. Zhang and L. Zou, "Improve-qa: an interactive mechanism for rdf question/answering systems," in *SIGMOD*, 2018, pp. 1753–1756.

[6] C. Liu and X. Wan, "Codeqa: A question answering dataset for source code comprehension," in *Findings of EMNLP*, 2021, pp. 2618–2632.

[7] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *TVCG*, 2016.

[8] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang, M. Zu, and W. Chen, "Echarts: a declarative framework for rapid construction of web-based visualization," *Visual Informatics*, vol. 2, no. 2, pp. 136–146, 2018.

[9] Y. Song, X. Zhao, and R. C.-W. Wong, "Fevisqa: Free-form question answering over data visualizations," in *Tech. Report*, 2023. [Online]. Available: https://t.ly/V-jYd

[10] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, "Effortless data exploration with zenvisage: An expressive and interactive visual analytics system," *Proceedings of the VLDB Endowment*, 2016.

[11] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *SIGMOD*, 2021, pp. 1235–1247.

[12] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, "Towards visualization recommendation systems," *ACM SIGMOD Record*, vol. 45, no. 4, pp. 34–39, 2017.

[13] N. Tang, E. Wu, and G. Li, "Towards democratizing relational data visualization," in *SIGMOD*, 2019, pp. 2025–2030.

[14] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, "Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation," *KDD*, pp. 1646–1655, 2022.

[15] C. Lee, Y. Seonwoo, and A. Oh, "Cs1qa: A dataset for assisting code-based question answering in an introductory programming course," in *NAACL*, 2022, pp. 2026–2040.

[16] J. Herzig, P. K. Nowak, T. Mueller, F. Piccinno, and J. Eisenschlos, "Tapas: Weakly supervised table parsing via pre-training," in *ACL*, 2020.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.