Marrying Dialogue Systems with Data Visualization: Interactive Data Visualization Generation from Natural Language Conversations

Yuanfeng Song
AI Group
WeBank Co., Ltd
Shenzhen, China
yfsong@webank.com

Xuefang Zhao
AI Group
WeBank Co., Ltd
Shenzhen, China
summerzhao@webank.com

Raymond Chi-Wing Wong
The Hong Kong University of Science and Technology
Hong Kong, China
raywong@cse.ust.hk

ABSTRACT

Data visualization (DV) has become the prevailing tool in the market due to its effectiveness into illustrating insights in vast amounts of data. To lower the barrier of using DVs, automatic DV tasks, such as natural language question (NLQ) to visualization translation (formally called *text-to-vis*), have been investigated in the research community. However, text-to-vis assumes the NLQ to be well-organized and expressed in a *single sentence*. However, in real-world settings, complex DV is needed through *consecutive exchanges* between the DV system and the users. In this paper, we propose a new task named **CoVis**, short for Conversational text-to-Visualization, aiming at constructing DVs through a series of interactions between users and the system. Since it is the task which has not been studied in the literature, we first build a benchmark dataset named **Dial-NVBench**, including dialogue sessions with a sequence of queries from a user and responses from the system. The ultimate goal of each dialogue session is to create a suitable DV. However, this process can contain diverse dialogue queries, such as seeking information about the dataset, manipulating parts of the data, and visualizing the data. Then, we propose a multi-modal neural network named **MMCoVisNet** to answer these DV-related queries. In particular, MMCoVisNet first fully understands the dialogue context and determines the corresponding responses. Then, it uses adaptive decoders to provide the appropriate replies: (i) a *straightforward text decoder* is used to produce general responses, (ii) an *SQL-form decoder* is applied to synthesize data querying responses, and (iii) a *DV-form decoder* tries to construct the appropriate DVs. We comparatively evaluate MMCoVisNet with other baselines over our proposed benchmark dataset. Experimental results validate that MMCoVisNet performs better than existing baselines and achieves a state-of-the-art performance.

CCS CONCEPTS

• **Human-centered computing → Visualization**; • **Computing methodologies → Natural language generation**.

KEYWORDS

Conversational Text-to-Visualization, Multi-modal Data Mining

ACM Reference Format:
Yuanfeng Song, Xuefang Zhao, and Raymond Chi-Wing Wong. 2024. Marrying Dialogue Systems with Data Visualization: Interactive Data Visualization Generation from Natural Language Conversations. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671935

1 INTRODUCTION

> *"The future of BI (Business Intelligence) is Conversational!"* – Gartner Analyst Report

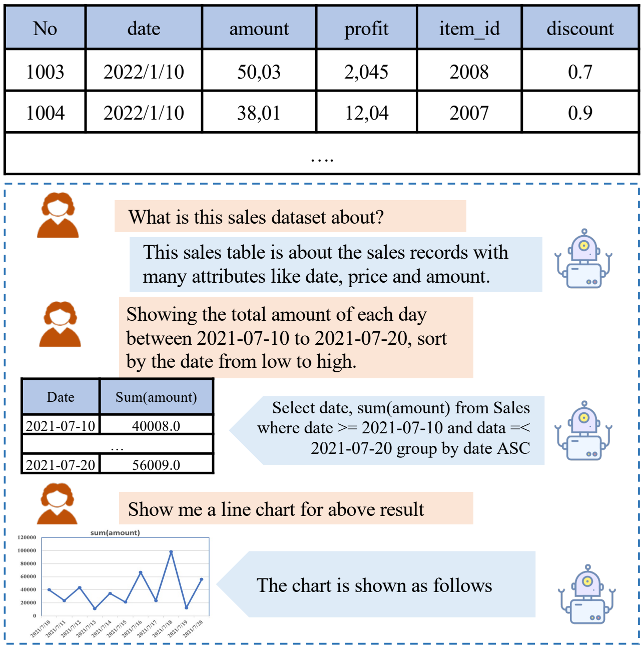
Data visualization (DV) has emerged as the prevailing tool in the industry in summarizing the insight behind raw data. Given a massive dataset, DV could vividly and efficiently convey the essence behind the raw data than a verbal description [20]. Due to its great usage, DV has been widely adopted and explored by many commercial vendors (e.g., ThoughtSpot [3] and Microsoft Power BI [2]), as well as academic researchers [13, 33, 43, 44].

To lower the barrier of using DVs, automatic DV tasks, such as *text-to-vis* (i.e., natural language question (NLQ) to visualization translation), have been extensively explored in the data mining community [30, 31, 43]. For example, Song *et al.* [43] published a novel text-to-vis model in KDD'22 that employs a hybrid retrieval and generation framework towards accurate DV generation. However, all these existing studies expect the input NLQ to be well-organized and expressed in a *single sentence* for existing text-to-vis methods. In real-world settings, complex DV is needed through *successive exchanges* between the DV system and the users. Under this scenario, *multi-turn interactions* between users and the DV system are frequently required to (1) clarify any ambiguous questions, (2) query the information about the dataset, (3) visualize the data, and (4) notify users about the returned data or charts.

In this paper, we propose a new task named **CoVis**, short for Conversational text-to-Visualization, aiming to construct DVs via a series of interactions between the system and the user. An illustrative example can be found in Figure 1, which shows that the analyst expresses his data querying and visualization needs gradually when the conversation progresses. The dialogue system generates different responses according to the context, including text, data (by SQL queries) and DV (by Vega-Lite [39], one popular programming language dedicated for DV). The ultimate goal of each dialogue session is to create a suitable DV. However, this process can contain

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0490-1/24/08…$15.00
https://doi.org/10.1145/3637528.3671935



No	date	amount	profit	item_id	discount
1003	2022/1/10	50,03	2,045	2008	0.7
1004	2022/1/10	38,01	12,04	2007	0.9
….					

What is this sales dataset about?

This sales table is about the sales records with many attributes like date, price and amount.

Showing the total amount of each day between 2021-07-10 to 2021-07-20, sort by the date from low to high.

Date	Sum(amount)
2021-07-10	40008.0
…	
2021-07-20	56009.0

Select date, sum(amount) from Sales where date >= 2021-07-10 and data =< 2021-07-20 group by date ASC

Show me a line chart for above result

The chart is shown as follows

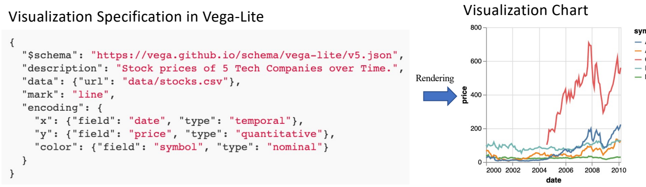
Figure 1: A conversation between a user and the data visualization dialogue system. The analyst gradually expresses his/her data querying and visualization needs gradually when the conversation progresses. Then, the dialogue system generates different responses according to the context, including text, data and DV Charts.

diverse user dialogue queries, like asking the information about the dataset, querying statistical information about the data, and visualizing parts of the data.

Since CoVis has not been studied in the literature, we first construct a benchmark dataset, including a sequence of dialogues with queries and responses. The dataset is named **Dial-NVBench**, modified from NVBench [30], a text-to-vis dataset that contains single-turn (NLQ, DV) pairs. Since each sample in NVBench could be considered as a single-turn dialogue between the user and the chatbot, we greatly extend each sample from the NVBench dataset by adding other queries and responses aside from only querying about the desired DVs. Specifically, we add queries like greetings, asking for the details of the dataset, and the general response.

To construct a conversational text-to-vis system, we propose a multi-modal conversational network to answer these DV-related queries, and we name it **MMCoVisNet**. In particular, MMCoVisNet first fully understands the dialogue context and determines the corresponding responses. Then, it uses adaptive decoders to provide the appropriate replies: (i) a *straightforward text decoder* is used to produce general responses, (ii) an *SQL-form decoder* is applied to synthesize data querying responses, and (iii) a *DV-form decoder* tries to construct the appropriate DVs. We comparatively evaluate MMCoVisNet with other baselines over our proposed benchmark dataset. Extensive experiments validate that our method could considerably improve the baseline models.

To sum up, the main contributions of this paper are as follows.



Visualization Specification in Vega-Lite

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Stock prices of 5 Tech Companies over Time.",
  "data": {"url": "data/stocks.csv"},
  "mark": "line",
  "encoding": {
    "x": {"field": "date", "type": "temporal"},
    "y": {"field": "price", "type": "quantitative"},
    "color": {"field": "symbol", "type": "nominal"}
  }
}
```

Figure 2: An example of the visualization specification in Vega-Lite, and its corresponding DV chart.

- We are the first to propose the CoVis task, which focuses on interactively constructing DVs given the massive dataset. CoVis could help people better comprehend the data and master how to create suitable DVs. We believe that this new CoVis task will enlarge the family of intelligent dialogue and DV systems.
- To validate the rationale of this new task, we construct a benchmark dataset Dial-NVBench, which could also promote the development of this field. Each dialogue session contains queries like querying the information about the dataset, querying parts of the data, and visualizing the data.
- We design the first multi-modal conversational DV network for this CoVis task - MMCoVisNet. MMCoVisNet is equipped with advanced encoders and adaptive decoders, which guarantees the final inference performance of the whole network.
- We conducted extensive experiments on our proposed dataset and compared it with other popular baselines. The results show that our proposed MMCoVisNet can perform better than several strong counterparts.

The rest of this paper is structured as follows: we first introduce basic concepts and the problem definition in Section 2. Then, we discuss our proposed MMCoVisNet model in Section 3. The experimental results are listed in Section 5, followed by the related work in Section 6. Finally, we conclude the work in Section 7.

2 CONCEPTS AND PROBLEM DEFINITION

This section introduces some foundational concepts that will facilitate a better comprehension of the subsequent work, and these concepts can be divided into two categories, DV and CoVis.

2.1 Concepts on Data Visualization

Declarative Visualization Language (DVL). A DVL specifies the visualization construction details, encompassing chart type, color, size, mapping function, and properties for marks like canvas size and legend. The most common DVLs include Vega-Lite [39], ggplot2 [50], ZQL [41], ECharts [22] and VizQL [16], each of which has its own grammar. Our work mainly uses Vega-Lite due to its popularity and wide usage [30, 31, 43, 46]. However, the methodology proposed can easily be applied to other DVLs.

Visualization Specification. A visualization specification defines the exact properties of a DV using a DVL. In Figure 2[1], the specification in Vega-Lite is a JSON file specifying properties such as data path, mark and encoding. A specification could be rather complicated, and creating a suitable specification for a given dataset requires experience in DVL as well as familiarity with the raw data.

DV Query. To abstract all the DVLs, a inter-media form named DV query is proposed and widely used in the DV tasks [30, 31,

---

[1] More complicated examples, including this one, can be found in the official website of Vega-Lite via *https://vega.github.io/vega-lite/examples/*.

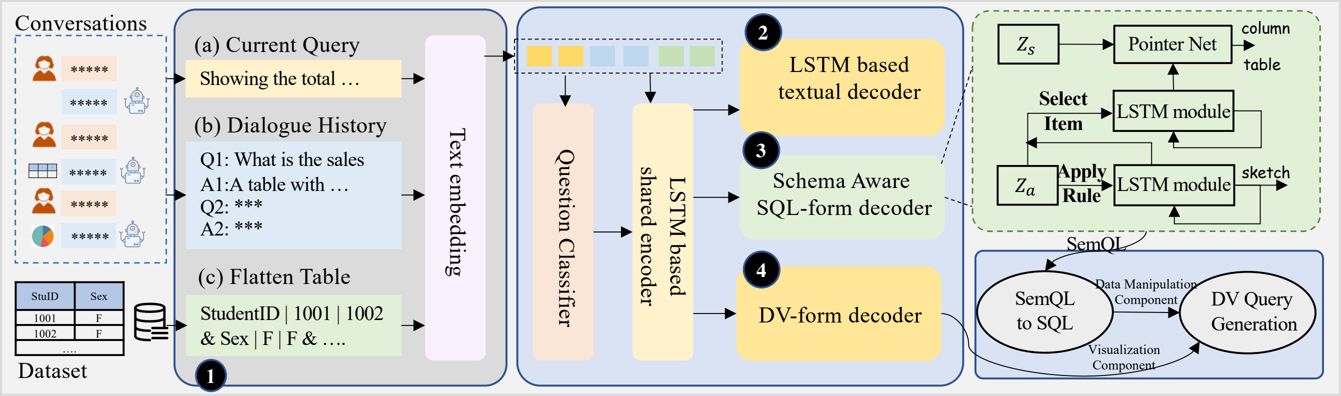


Figure 3: The network structure of the MMCoVisNet model, which is comprised of ❶ a Data-aware Dialogue Session Encoder, ❷ a Textual Decoder, ❸ a SQL-form Decoder, and ❹ a DV-form Decoder. To answer the data-related query, MMCoVisNet first generates SemQL via the SQL-form Decoder, then converts the SemQL into an SQL query, and finally obtains the data by executing the SQL query. Similarly, to answer the DV-related queries, MMCoVisNet first generates DV queries via the DV-form Decoder, then converts the DV queries to get the Vega-Lite specifications, and finally renders the specifications to get the charts.

Table 1: The Statistics of the proposed Dial-NVBench Dataset

Attributes	Train	Val.	Test	Total
No. of Dialogue Sessions	3115	484	896	4495
No. of QRs	15044	2359	4332	21725
Avg. No. of QRs per Session	4.83	4.874	4.824	4.833
No. of Datasets	137	69	107	143
No. of General Queries	6197	977	1797	8971
No. of Data-related Queries	5732	898	1629	8259
No. of DV-related Queries	3115	484	896	4495

43, 46]. Existing text-to-vis studies first synthesizes the DV query from the NLQ, and then executes the DV query to obtain the Vega-Lite specification. DV query enjoys a SQL-alike grammar with *visualization component* and *data manipulation component*. The final chart is obtained by rendering the specification file via the visualization engine [39].

2.2 CoVis Problem Definition

The definition of the *CoVis* (Conversational Text-to-Visualization) task is relatively straightforward once the previous concepts are defined. Specifically, given a dataset D, a query q from the user with its context S, the CoVis task aims to predict the response r automatically. CoVis enables users to create suitable DVs interactively. To fulfill the above objective, the context S includes all the previous queries and responses in the same session and they are usually in the DV domain. The query q can be a diverse dialogue query, like seeking the information about the dataset, manipulating parts of the data, and visualizing the data. A desired CoVis model could be represented as $f(D, S, q) \longrightarrow r$. Subsequently, the whole training set can be formally defined as $\mathcal{T} = \{D^{(o)}, S^{(o)}, q^{(o)}, r^{(o)}\}_{o=1}^{N}$, where N is the dataset size.

2.3 CoVis Dataset Construction

Since CoVis is a new kind of DV-related dialogue tasks, there is currently no literature or public datasets for it. Thus, we first construct and release a benchmark dataset to promote the development of this field. In particular, we synthesize the CoVis dataset by piggybacking the *NVBench* dataset [30] from the correlated text-to-vis task. Each sample in the NVBench contains a *single-turn* NLQ with the corresponding DV (in terms of a DV query). We convert each of them into a *dialogue session* by (i) randomly deciding the number of rounds (or *query-responses (QRs)*) for the dialogue; (ii) sampling from some pre-defined general queries about the statistics of the dataset; (iii) generating some SQL-related queries about data manipulation, including ones involving complicated data operations like nested queries and joins, with the method similar to [56]; and (iv) ending each session with a query (the same as the one contained in NVBench) about a DV. To validate this CoVis task, in this new dataset, we mainly focus on data querying and visualization on single tables. Finally, we obtained a dialogue dataset dedicated to DV, and we named it *Dial-NVBench* ("Dial" is short for "Dialogue"). The statistics of the dataset is shown in Table 1.

3 OUR PROPOSED APPROACH

In this section, we mainly discuss the proposed MMCoVisNet model from its main components.

3.1 Model Overview

As shown in Figure 3, MMCoVisNet mainly comprises four components, namely a *Dialogue Session Encoder*, a *Textual Decoder*, a *SQL decoder*, and a *DV Decoder*. The dialogue session encoder takes the dataset, dialogue history and current query as input and then maps them into some hidden representation. The dialogue history is essential in the CoVis scenario since it helps identify item references in multi-turn dialogue sessions. If the desired response belongs to the general textual category, the textual decoder is used to generate a typical textual response, similar to existing dialogue systems. However, if the query is about manipulating the data, then the SQL decoder is triggered to generate SQL queries to obtain the data. To increase the conversion accuracy, MMCoVisNet employs SemQL [15] as an inter-media form for each SQL query. Similarly, MMCoVisNet uses a DV decoder to obtain the final charts to answer DV-related queries. Following the common practice in text-to-vis

[30, 31, 43, 46], MMCoVisNet also first synthesizes the intermedia form - the DV query, and then executes the DV query to obtain the Vega-Lite specification. The final chart is obtained by rendering the specification via the visualization engine [39].

3.2 Data-aware Dialogue Session Encoder

The objective of the dialogue session encoder is to convert the current query q, incorporated with a dialogue history and the dataset information, into a hidden representation. Given the dialogue history, we first concatenate them by a separate character '|', and then consider it a textual sequence. The whole process can be denoted as:

$$s_d = d_1|d_2|\cdots|d_n. \tag{1}$$

where d_i is the sequence of the i-th dialogue session and n is the total number of the dialogue sessions. Since the dataset information dramatically affects the desired response, we must also preserve the schema information. Moreover, we sample some data records from the dataset as a snapshot for this dataset. To restrict the input sequence length, a certain number of rows are first selected randomly, and after we retain all the names of the schema items, we add the table values from the selected rows word by word until we reach the limit. Finally, the dataset sequence is formatted as

$$s_t = c_1|v_{11}|v_{12}|...\&c_2|v_{21}|..., \tag{2}$$

where c_i is the name of the column i, v_{ij} is the j-th value of column c_i, '|' and '&' are the separate characters for the items and rows, respectively.

The input sequence Q of this session encoder is composed of the current query q, the history sequence s_d, and the dataset sequence s_t. We first map each token in Q into its embedding (all these embeddings forms $\mathbf{Q}$) by a pre-trained Glove [36] and then a bidirectional LSTM network (BiLSTM) [60] is employed to deal with the input embedding to obtain the hidden vectors for the current input. The whole process is represented as

$$\mathbf{h}_t^Q = \texttt{BiLSTM}(\mathbf{Q}_t, \mathbf{h}_{t-1}^Q), \tag{3}$$

where $\mathbf{Q}_t$ is the item from $\mathbf{Q}$ at time step t and $\mathbf{h}_t^Q$ represents the hidden states at time step t for BiLSTM. Finally, we obtain the output $\boldsymbol{H}_Q = \{\mathbf{h}_1^Q, \mathbf{h}_2^Q ..., \mathbf{h}_l^Q\}$ from the session encoder, where l is the number of tokens in sequence Q.

3.3 Adaptive Response Decoder

Since the output of the CoVis task is multi-modal, we design an adaptive decoder to generate diverse responses. The adaptive decoder first uses a classifier to judge the desired response type. Then, three specific decoders are proposed for the downstream process: (i) a textual one for the textual output, (ii) an SQL-form one for the data query, and (iii) a DV-form one for the visualization query.

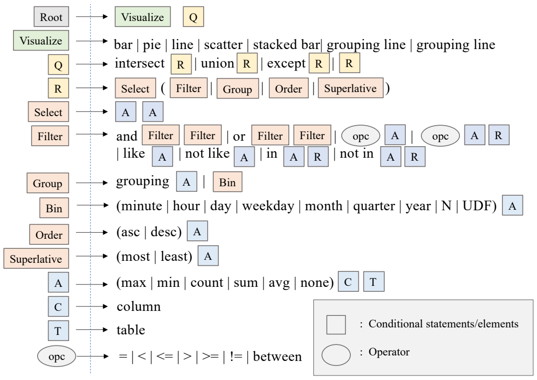

Root
Visualize
Q
Visualize
bar | pie | line | scatter | stacked bar| grouping line | grouping line
Q
intersect R | union R | except R | R
R
Select (Filter | Group | Order | Superlative)
Select
A A
Filter
and Filter Filter | or Filter Filter | opc A | opc A R
| like A | not like A | in A R | not in A R
Group
grouping A | Bin
Bin
(minute | hour | day | weekday | month | quarter | year | N | UDF) A
Order
(asc | desc) A
Superlative
(most | least) A
A
(max | min | count | sum | avg | none) C T
C
column
T
table
opc
= | < | <= | > | >= | != | between
: Conditional statements/elements
: Operator

Figure 4: Grammar used to decode SQL-form response.

3.3.1 *General Textual Decoder.*

The objective of this decoder is to generate textual sequences to answer questions about basic information. A Transformer-based decoder that takes the advantage of a multi-head scaled dot-product attention [48] is employed here, where multi-head attention can be formulated as the following equation. Specifically, given the query $\hat{Q}$, the key $\hat{K}$, and the value $\hat{V}$, they are first injected into different high-dimensional spaces[2]. Then, they perform an attention calculation to obtain a hidden representation for each head, and finally combine them to produce a final attention output.

$$head_i = \texttt{Attention}(\hat{Q}W_i^{\hat{Q}}, \hat{K}W_i^{\hat{K}}, \hat{V}W_i^{\hat{V}}), \tag{4}$$

$$\texttt{MultiHead}(\hat{Q}, \hat{K}, \hat{V}) = [head_1; \cdots; head_g]W^O, \tag{5}$$

where $W_i^{\hat{Q}}, W_i^{\hat{K}}, W_i^{\hat{V}} \in \mathbb{R}^{d_k \times d_\mathrm{m}}$ and $W^O \in \mathbb{R}^{d_m \times d_\mathrm{m}}$ are trainable parameter matrices. d_k is the dimensionality of the input key vector $\hat{K}$, d_m is the hidden size of the model and g is the number of the heads. The attention process consists of the similarity score calculation obtained by a dot-product operation between $\hat{Q}$ and $\hat{K}$ and a re-assignment according to the similarity score and the given $\hat{V}$, as shown in Equation 6.

$$\texttt{Attention}(\hat{Q}, \hat{K}, \hat{V}) = \texttt{Softmax}(\frac{\hat{Q}\hat{K}^T}{\sqrt{d_k}})\hat{V}, \tag{6}$$

Furthermore, the decoder is stacked with K identical blocks composed of a self-attention layer, a cross-attention layer, and a feed-forward layer. Both self-attention and cross-attention follow the process of the aforementioned multi-head attention but have different inputs and objectives. We use Z_i^k and Z_o^k to denote the input and output for the k-th block ($0 \le k \le K$), and Z_s^k to denote the output of self-attention. The word embedding R for the target textual response r is set as the initial input (i.e., $Z_o^0 = R$). Since it contains multiple layers, the output of previous one is set as the input for the following one (i.e., $Z_i^{(k)} = Z_o^{(k-1)}$, $\hat{Q} = \hat{K} = \hat{V} = Z_i^k$). It thus makes sure that the target textual response can go through and pay attention to itself and finally generate $Z_s^{(k)}$. In contrast, the cross-attention component takes the outputs from both the self-attention layer and the encoder as the inputs (i.e., $\hat{Q} = Z_s^k, \hat{K} = \hat{V} = \boldsymbol{H}_Q$) to produce a representation for each target sequence word that captures the influence from the input sequence.

3.3.2 *SQL-form Decoder.*

An important kind of query from the users is data-related, and the essential way of answering them is through an SQL query. Due to the strict grammar that the SQL-form response has, which differs from the textual response, we design an SQL grammar-aware decoder here. Inspired by the common practice in a paralleled task named *text-to-SQL* [19], we also employ an intermedia SemQL grammar [15] to represent each SQL query.

---

[2] The query $\hat{Q}$ in the attention mechanism is different from the previously defined Q in CoVis, and thus, we use two different variables.

The grammar of SemQL is represented in Figure 4, where each row in the figure corresponds to an *action*, and the process of generating SemQL (which could be represented by an Abstract Structure Tree (AST) $\tilde{r}$) is achieved by selecting a series of actions.

In particular, each step in the action selection process can be regarded as a classification problem, where the candidates and the final choice are determinied by the inputs and outputs of the previous step, represented mathematically as follows.

$$p(\tilde{r}|q, D, S) = \prod_{i=1}^{T} p(a_i|D, S, q, a_{<i}), \tag{7}$$

where a_i refers to an action performed at step i, $a_{<i}$ includes all the previous actions of step i, and T indicates the total number of actions needed to obtain the desired SemQL $\tilde{r}$. The actions are further divided into two types: *ApplyRule* and *SelectItem*. The former generates the *sketch* of the AST, the latter selects column and table items to finalize the sketch, and *finally*, we obtain a complete SemSQL AST. The mathematical details of ApplyRule can be denoted as follows:

$$\boldsymbol{h}_i = \mathtt{LSTM}([\boldsymbol{a}_{i-1} \oplus \boldsymbol{e}_{i-1} \oplus \boldsymbol{v}_{i-1}], \boldsymbol{h}_{i-1}), \tag{8}$$

$$\boldsymbol{v}_i = \mathtt{Softmax}(\boldsymbol{h}_i^T \boldsymbol{W}_h \boldsymbol{H}_Q^T) \boldsymbol{H}_Q, \tag{9}$$

$$\boldsymbol{u}_i = \mathtt{tanh}(\boldsymbol{W}_u [\boldsymbol{h}_i \oplus \boldsymbol{v}_i] + \boldsymbol{b}_u), \tag{10}$$

$$p(\tilde{r}_i = a_i|q, S, D, a_{<i}) = \mathtt{Softmax}(\mathtt{tanh}(\boldsymbol{W}_p \boldsymbol{u}_i + \boldsymbol{b}_p)), \tag{11}$$

where $\oplus$ denotes the concatenation operation. $\boldsymbol{h}_i$ is the LSTM hidden state of the current step i, $\boldsymbol{a}_{i-1}$ is the previous action embedding, $\boldsymbol{e}_{i-1}$ is the previous action type (i.e., ApplyRule or SelectItem) embedding and $\boldsymbol{v}_{i-1}$ is the previous contextual representation of LSTM. $\boldsymbol{W}_h \in \mathbb{R}^{d_m \times d_m}$, $\boldsymbol{W}_u \in \mathbb{R}^{d_m \times 2d_m}$ and $\boldsymbol{W}_p \in \mathbb{R}^{n_a \times d_m}$ (n_a is the number of actions associated with the specified grammar) are the trainable weights, $\boldsymbol{b}_u \in \mathbb{R}^{d_m}$ and $\boldsymbol{b}_p \in \mathbb{R}^{n_a}$ are trainable biases, and the initial state $\boldsymbol{h}_0$ is calculated by an average-pooling operation on the output $\boldsymbol{H}_Q$ given by the encoder (in Section 3.2).

Given the sketch, what we need to do next is to select column items. We first calculate an *NLQ-aware representation* for column c_k, and then a *pointer network* is used to obtain the selection probability. The mathematical representation of this progress is shown as follows:

$$\beta_{k,j} = \frac{\boldsymbol{s}_{c_k}^T \boldsymbol{q}_j}{\left\|\boldsymbol{s}_{c_k}\right\| \left\|\boldsymbol{q}_j\right\|}, \tag{12}$$

$$\tilde{\boldsymbol{s}}_{c_k} = \boldsymbol{s}_{c_k} + \textstyle\sum_{j=1}^{|Q|} \beta_{k,j} \boldsymbol{h}_j^Q, \tag{13}$$

$$\gamma_{k,i} = (\tilde{\boldsymbol{s}}_{c_k})^T \boldsymbol{W}_c \boldsymbol{u}_i, \tag{14}$$

$$p(\tilde{r}_i = \text{SelectColumn}(c_k)|D, S, q, a_{<i}) = \frac{exp(\gamma_{k,i})}{\sum_{j=1}^{n_c} exp(\gamma_{j,i})}, \tag{15}$$

where $\boldsymbol{s}_{c_k}$ is the embedding for column c_k obtained from Glove embedding [36], $\boldsymbol{q}_j$ and $\boldsymbol{h}_j^Q$ are the corresponding vectors of the j-th question token extracted from $\mathbf{Q}$ and $\boldsymbol{H}_Q$, respectively, and $\boldsymbol{W}_c \in \mathbb{R}^{d_e \times d_m}$ is the learnable weights.

ALGORITHM 1: The Pipeline for Obtaining the DV Charts

```
Input: Current query $q$, Dialogue Context $S$, MMCoVisNet Model
  $M$, SQL Response in the Last Turn $r_{sql}$, Dataset $D$
Output: Visualization Chart $c$
begin
    Obtain the DV Query from the model output $r_{dvq} \leftarrow$
      $M(q, D, S)$;
    Execute the SQL query to retrieval data $d \leftarrow$ execute($D$, $r_{sql}$);
    Get Vega-Lite specification $c_{vl} \leftarrow$ transform($r_{dvq}$, $d$) ;
     // Following previous studies like [30, 43].
    DV Chart $c \leftarrow$ render($c_{vl}$);
    return $c$;
end
```

3.3.3 DV-form Decoder. The ultimate goal of the DV-form decoder aims to show the desired DV chart to the user. Rather than directly generating the charts with the neural network, we design the decoder first to synthesize the DV query. The grammar of the DV query is quite similar to the SQL query, and it was proposed and widely used in the text-to-vis task [30, 31, 43]. We use r_{sql} to represent the SQL query predicted in the last turn by the MMCoVisNet model and r_{dvq} to denote the DV query predicted by the MMCoVisNet model. If the model synthesizes an accurate DV query r_{dvq}, it yields an accurate DV chart, denoted as c. Specifically, after the DV query is obtained, a pipeline approach shown in Algorithm 1 is triggered to generate the final chart. A DV query is composed of the *visualization component* and the *data manipulation component*. The visualization component of the DV query is generated by a *Transformer-based decoder* in MMCoVisNet, with a shared structure to the textual decoder described in Section 3.3.1. Combined with the previous SQL response and the data retrieved by the SQL query, we can synthesize the visualization specifications c_{vl} in Vega-Lite.

3.4 Model Training

Since there exists different types of responses in one continuous multi-turn dialogue and a similar natural language question, we make sure that they share one session encoder but have different decoders. To train them simultaneously, different QR pairs appear in the same batch and are divided into mini-batches by the question classifier. The mini-batch of various types is fed into the network, and the decoder parameters are updated sequentially (textual-form decoder, data-form decoder and DV-form decoder). Then, the encoder is updated only once in the final phase. The widely-used cross-entropy loss is used in our experiments. For the textual response ($\mathcal{L}_1$) and DV-form response ($\mathcal{L}_3$), we maximize the probability of each word appearing in the target sequence by the following function:

$$\mathcal{L}_3(r, \hat{r}) = \mathcal{L}_1(r, \hat{r}) = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{N_i} \sum_{j=1}^{N_i} \log p(r = \hat{r}_{ij}), \tag{16}$$

where $\hat{r}$ is the predicted response, r is the target response, N is the total number of training samples, and N_i is the length of the response of the i-th sample. $p(r = \hat{r}_{ij})$ represents the probability of the predicted j-th token in the i-th sample being equal to $\hat{r}_{ij}$.

For the SQL-form response, unlike above mentioned textual and DV-form responses, it works in a two-step generation style. To train the model, we employ the objective of maximizing the log-likelihood of the ground truth action sequences in the training data, which is defined as follows:

$$\mathcal{L}_2(r,\hat{r}) = \max \sum_{(D,S,q,r)\in\mathcal{T}} \left[\sum_{a_i\in\text{ApplyRule}} \log p(\tilde{r}_i = a_i | D, S, q, a_{<i}) + \sum_{a_i\in\text{SelectColumn}} \log p(\tilde{r}_i = a_i | D, S, q, a_{<i}) \right]. \tag{17}$$

4 EXPERIMENTAL SETUP

This section aims to provide a detailed performance evaluation of the proposed model. We first describe the experimental setup, the evaluation measurements, and the baselines. Then, we compare the performance of our proposed model with these strong baselines.

4.1 Models

Several popular dialogue models and methods were utilized to facilitate a comprehensive performance comparison.

- Seq2Seq: Seq2Seq [7] is a fundamental model that treats the dialogue task as a translation that maps from the queries to the responses.
- Transformer: Transformer [48] has shown a dominant performance in many NLP tasks including machine translation [12], dialogue system [63] and speech recognition [61]. Like Seq2Seq, we modify it as a baseline method.
- HRED: HRED [40] is a prevalent and representative model in the field of multi-turn dialog systems.
- MMCoVisNet: This is our proposed method, which employs a data-aware session encoder as well as advanced adaptive decoders to generate responses in various forms.

To ensure a fair evaluation, all the aforementioned methods were trained on the same training set and subsequently assessed on the same testing set. We try our best to optimize the performance of these models to achieve their best performance.

4.2 Evaluation Metrics

Since our task includes three kinds of outputs, i.e., the textual form response, the SQL-form data response and the DV chart response, we employ various metrics to compare the performance. For the textual response, we employ the classical *BLEU* [35], *ROUGE* [27] and *METEOR* [9] scores in a textual-form dialogue system as our main metrics. For SQL-form response, we employ the widely-used *Sketch Accuracy* and *Query Matching Accuracy* [58] in the text-to-SQL task as the main indicators. If a model can generate an accurate SQL query, it will also retrieve the correct data. Similarly, we also employ *Vis Accuracy*, *Axis Accuracy*, *Data Accuracy* and *DV Accuracy* used in text-to-vis [30] to reflect the performance of the generated DV query. For all these metrics, including the ones for evaluating the textual, the SQL-form and the DV-form responses, a larger value of each metric means a better result. A correctly predicted DV query leads to a correctly generated Vega-Lite configuration and the final chart.

4.3 Implementation Details

Our networks are constructed using the Adam optimizer, with a mini-batch size set to 32 for dialogues and a learning rate set to 1×10^{-4}. The dimensionality of word embedding of Glove is 300, the size of vocabulary used in this experiment is 15,894, and each unseen word is initialized by '<unk>' denoting unknown word. For the session encoder, we set the number of layers to 1, the number of heads to 4, and the hidden size to 512. Both textual and DV-form decoders constructed by the Transformer have the same parameters as the encoder. In addition, the dimensionality of action embedding and type embedding in the SQL-form decoder are all set to 128 and the hidden size to 512.

5 EXPERIMENTAL RESULTS

5.1 Performance Evaluation

We compare the performance in the three forms of the response generated in Table 2. Since all the baseline methods directly generate the SQL query rather than first predicting a sketch, only our proposed MMCoVisNet contains the value in sketch accuracy. From the result, we report the following observations.

Since the data-related queries are critical and significantly affect the accuracy of the following generated DV charts, we first evaluate the performance of SQL-from response generation. The vanilla end-to-end model, Seq2Seq, performs poorly and is not quite competitive as a baseline, due to two factors: (i) its capability is limited in capturing the semantics contained in the dialogue session and the dataset; (ii) its ability is also limited in generating the SQL queries, since SQL enjoys a very strict and complicated grammar. Other advanced models, such as Transformer and HRED, could improve the vanilla Seq2Seq model from these two aspects. For example, the HRED model employs a hierarchical decoder to generate responses for different inputs, leading to a 32.37% relative improvement in the DV accuracy. The RGVisNet achieves superior performance on this task owing to its advanced retrieval-generation design, but it is only capable of producing the final DV output, without response for the intermediate steps. Prompted LLMs exhibit limited performance on this task, as they typically work in few-shot in-context learning style and they are constructed on datasets with limited DV examples.

Our designed MMCoVisNet could make full use of all the information (i.e., dialogue history, query and dataset) out of all possible models, and at the same time, employ adaptive decoders to generate accurate responses. Thus, it remarkably beats all the baseline methods, including advanced models such as HRED and Transformer, thus proving the efficacy and validating the necessity of designing an advanced encoder and decoder in the CoVis scenario.

For the final DV-form responses, the basic end-to-end approach, Seq2Seq, performs poorly and is not quite competitive as a baseline. Other models (e.g., Transformer and HRED) could beat the vanilla Seq2Seq model. Even though our proposed MMCoVisNet model is slightly weaker in the metric *Vis. Accuracy* ($\approx 0.1\%$), it achieves the best performance over all other metrics. Most importantly, it largely surpasses all other baselines in the final *DV Accuracy* (e.g., a 21.8% relative improvement over the Transformer). We also notice that the accuracy of the DV-form response is much higher than the SQL-form query. The reason is also apparent. The proposed

Table 2: Performance Comparison.

	Textual Response			SQL-form Response		DV-form Response			
Method	BLEU	ROUGH	METEOR	Sketch Acc.	SQL Acc.	Vis. Acc.	Axis Acc.	Data Acc.	DV Acc.
Seq2Seq	0.7437	0.8499	0.7954	-	14.92%	100.00%	31.47%	33.59%	18.97%
Transformer	0.7436	0.8451	0.7953	-	46.72%	100.00%	67.30%	60.49%	55.80%
HRED	0.7200	0.7732	0.7206	-	18.72%	99.67%	39.73%	39.73%	25.11%
MMCoVisNet	0.7381	0.8556	0.7958	87.11%	62.37%	99.89%	73.55%	67.75%	67.97%

Table 3: Ablation Study Results.

	Textual Response			SQL-form Response		DV-form Response			
Method	BLEU	ROUGH	METEOR	Sketch Acc.	SQL Acc.	Vis. Acc.	Axis Acc.	Data Acc.	DV Acc.
MMCoVisNet	0.7381	0.8556	0.7958	87.11%	62.37%	99.89%	73.55%	67.75%	67.97%
w/o session	0.7480	0.8594	0.8021	58.81%	33.33%	100.00%	43.19%	17.97%	16.52%
w/o dataset	0.5106	0.8059	0.5806	89.50%	61.45%	100.00%	74.44%	66.29%	66.63%
w/o both	0.5050	0.7831	0.5607	58.44%	32.78%	100.00%	40.40%	18.08%	17.08%
w. basic decoder	0.7511	0.8593	0.8057	-	45.67%	99.89%	66.85%	61.94%	54.35%

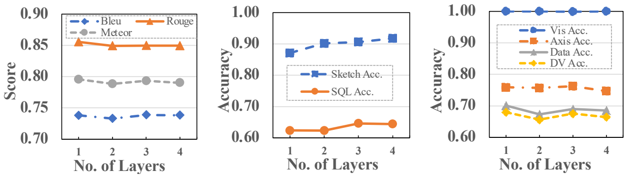
Dial-NVBench dataset contains more SQL-related queries than the DV-related queries, and a large portion of these SQL-related queries are complicated queries (e.g., nested operations), resulting to an overall 62.37% SQL accuracy by MMCoVisNet. Among all these SQL queries, only parts of them are being derived into the final DV and this part is relatively easier, leading to a higher SQL query accuracy compared with the *overall* SQL query accuracy and thus a higher DV accuracy (i.e., 67.97%).

Finally, we also analyze the quality of the textual form responses generated by various methods. Since this dataset is mainly for DV generation, the textual responses generation part in this dataset mainly focuses on talking about the dataset and not very diverse. And thus, all these models could achieve relative good results. For some metric like BLEU, the difference between these method is slight (e.g., < 0.76% between Seq2Seq, Transformer and MMCoVisNet). From the result, we could see that our model could still generate comparable results even if baselines such as HRED are dedicated to and specially designed for textual dialogue generation. The main reason for this is that our model can understand the dialogue context, which leads to accurate textual responses.

5.2 Ablation Studies

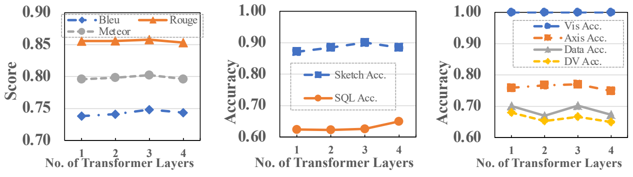
This section presents ablation studies to show the efficacy and contribution of each designed component within the MMCoVisNet architecture. Firstly, we examine the performance of the MMCoVisNet model with all its proposed components. Then, we investigate the performance of the MMCoVisNet model by removing or replacing its components. We first check each part in the encoder. To evaluate the efficacy of the dialogue session component, we removed it and named this baseline **w/o session**. Similarly, to analyze the effectiveness of the dataset, we removed it and called it the **w/o dataset**. We also remove both the dialogue session and the dataset components, and we named it **w/o both**. For the decoder part, we replace our designed adaptive decoder with a vanilla LSTM-based one (**w. basic decoder**). The results are shown in Table 3.

We take the DV accuracy as the primary indicator, and the other metrics reflect similar conclusions. First, integrating the session information leads to 51.45% absolute performance improvement (67.97% v.s 16.52%). This ablation study set and its substantial improvement demonstrate the necessity of incorporating dialogue session information in this CoVis task. Other components, like involving dataset information, show similar observations. Then, for the decoder part, using a basic LSTM-based decoder directly generated the SQL query rather than a sketch first, and it has no value in sketch accuracy. Compared with a basic LSTM-based model, our designed adaptive decoder shows about 16.37% relative improvement, proving the necessity of using advanced NN-based models in generating diverse responses.



(a) Textual-form Perf. (b) SQL-form Perf. (c) DV-form Perf.

Figure 5: Vary No. of Layers in Encoder



(a) Textual-form Perf. (b) SQL-form Perf. (c) DV-form Perf.

Figure 6: Vary No. of Transformer Layers

5.3 Hyper-parameter Study

This set of experiments analyze the influence of parameter variations on the performance of our proposed MMCoVisNet model. We first identify three main parameters that affect the performance of the model, namely the number of layers in the encoder (Figure 5), the number of heads (Figure 6), and the number of layers in the

Product_id	Product_Name	Product_Price	Product_Description	Other_Product_Service_Details
11	photo	4448536	...	...
154	film	2302	...	...
...				

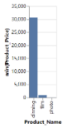
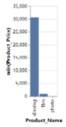
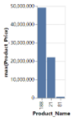
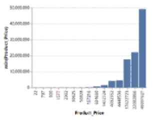
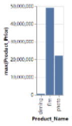
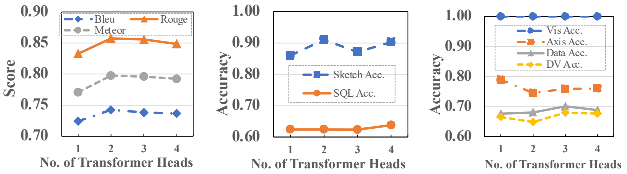
Turn	Query	Respone				
		Ground-Truth	CoVis	Seq2seq	Transformer	HRED
1	give me a short description about the table	a product table with many attribute like product id , product name , product price , product description , other product service detail	a product table with many attribute like product id , product name , product price , product description , other product service detail	a product table with many attribute like product id , product name , product price , product description , other product service detail	a product table with many attribute like product id , product name , product price , product description , other product service detail	a product table with many attribute like product id , product name , product description , product price
2	What are the different product names and the average product price for each of them	SELECT Product_Name , avg(Product_Price) FROM PRODUCTS GROUP BY Product_Name	SELECT Product_Name , avg(Product_Price) FROM PRODUCTS GROUP BY Product_Name	select product_id , avg (product_price) from products group by product_name	select product_name , avg (product_price) from products group by product_name	select product_description , avg (product_price) from products group by product_description
		[('dinning', 238704.33333333334), ('film', 11385515.333333334), ('photo', 5102691.666666667)]	[('dinning', 238704.33333333334), ('film', 11385515.333333334), ('photo', 5102691.666666667)]	[('300', 238704.33333333334), ('154', 11385515.333333334), ('11', 5102691.666666667)]	[('dinning', 238704.33333333334), ('film', 11385515.333333334), ('photo', 5102691.666666667)]	[(None, 6643023.666666667)]
3	how about the minimum product price	SELECT Product_Name , min(Product_Price) FROM PRODUCTS GROUP BY Product_Name	SELECT Product_Name , min(Product_Price) FROM PRODUCTS GROUP BY Product_Name	select product_id , max (product_price) from products group by product_name	select product_price , min (product_price) from products group by product_price	select product_name , max (product_price) from products group by product_name
		[('dinning', 30625), ('film', 900), ('photo', 22)]	[('dinning', 30625), ('film', 900), ('photo', 22)]	[('81', 634660), ('188', 49097627), ('21', 22082866)]	[(22, 22), (797, 797), (900, 900), (1577, 1577), ... (22082866, 22082866), (49097627, 49097627)]	[('dinning', 634660), ('film', 49097627), ('photo', 22082866)]
4	show me a bar for above result		✓ 	× 	× 	× 

Table 4: Examples of Queries & Responses Returned by Various CoVis Methods (errors are marked with red color and ×).



(a) Textual-form Perf.

(b) SQL-form Perf.

(c) DV-form Perf.

Figure 7: Vary No. of Transformer Heads

Transformer (Figure 7). We change the value of one parameter and fix the remaining ones each time.

We utilize the DV accuracy in Figure 5c as the main indicator in the following analysis. The highest accuracies are obtained when the encoder comprises three layers, as the performance declines when the number of layers deviates from this optimal value, whether higher or lower. In particular, we observe that increasing this value does not always lead to a better performance. This phenomenon is also observed in previous studies like [24], since too many layers usually result in the smoothing problem. For other parameters, i.e., the number of Transformer layers (Figure 6) and the number of Transformer heads (Figure 7), we also observe similar performance trends.

5.4 Case Study

We also list a real case shown in Table 4 to give a vivid illustration, which includes the original datasets to be analyzed, the queries and the predicted responses (text, SQL query & data, and DV charts) by various methods. In this example, the user directly queries the details about a "Product" dataset by asking general question like *"give me a short description about the table"*, data-related question like *"what are the different product names and the average product price for each of them"*, and DV-related question like *"show me a bar for the above result"*. From this example, we could see that MMCoVisNet could accurately understand the meaning of all the queries and generate correct answers for text, data and DV. Other baselines, including the advanced Transformer and HRED, fail in answering most of the queries and also obtained incorrect DV charts finally. It is noted that MMCoVisNet also makes full use of the meaning in the whole dialogue session rather than a solely current query, which is very important for understanding queries like *"how about the minimum product price"* (Turn 3) that refers to the previous context (Turn 2).

6 RELATED WORK

This work is relevant to the research areas of data visualization, natural languages to SQL, and dialogue systems.

6.1 Data Visualization

Recent years have witnessed an enormous rise in DV in the data mining [14, 18, 37, 43] and the database communities [16, 29, 30, 46, 47]. There are many intelligent tasks proposed to lower the barriers to use DV, including text-to-vis and DV recommendation.

Text-to-vis focuses on automatically translating a natural language question (NLQ) into its corresponding DV. It also enables non-experts to manipulate the DV system easily. The most popular method right now is to treat it as a machine translation problem and then utilize deep neural networks to map from the input (i.e., NLQ) to the output (i.e., DV). For example, Cui *et al.* introduces text-to-viz and utilizes rule-based methods to translate text statements to infographics [11]. Draco-Learn [32] views the visualization design as a selection process from a set of constraints and then use experimental data to turn the weights for these soft constraints. Data2Vis [13] also attempts to convert the data series to visualization specifications in some declarative languages in a machine translation style. NL4DV [33] is a Python toolkit that offers a number of high-level operations to assist users in building DV systems with NL-interfaces (NLIs). To address the challenge of limited dataset in this field, Luo *et al.* additionally describes an approach to synthesize the NLQ-DV dataset called NVBench based on the popular NL2SQL benchmark Spider [58]. A Seq2Seq model is further trained on this dataset to validate the feasibility of DV query generation from NLQs on this benchmark [30]. RGVisNet [43] is another representative study in KDD'22 that employs a DNN-based approach for converting NLQs into DVs. However, it beats other methods by combining the retrieval-based method with the generative-based one. In contrast, automatic DV recommendation directly outputs top-N potential DVs given a dataset without any NL queries from the users. To name a few, DataEye [28] tackles the DV recommendation problem in a three-step style, namely visualization recognition, ranking and selection. Qian *et al.* [37] employ an end-to-end learning-based method to construct DVs, giving a massive dataset.

Different from these studies, we propose a novel DV-related task named CoVis (i.e., Conversational text-to-Vis), which focuses on *interactive exchange* between the user and the DV system. With the benchmark and all the baseline methods proposed in this paper, CoVis would become a prevalent DV-related task and inspire more studies in *NLP for Data Mining* direction.

6.2 Natural Language to SQL

Natural language to SQL (NL2SQL or text-to-SQL) is another closely related area that motivates text-to-vis development. Its ultimate goal is to enable databases with NL-based interfaces (NLIs). Past literature in this field first handles the *single-turn* text-to-SQL translation with representative studies such as Seq2SQL [49], SQLNet [52], TypeSQL [55], Syntax SQL [45], IRNet [15]. [6] and [19] are two representative surveys about these studies. Moreover, numerous open-source text-to-SQL benchmarks have been released to promote research in this field, such as Spider [58] and WikiSQL [49]. Similarly, researchers also extend the vanilla text-to-SQL task to a multi-turn scenario with a public dataset such as SparC [59] and CoSQL [57]. The multi-turn interaction between the user and the NLI for DB systems would greatly improve the accuracy and further reduce the translation errors when converting NLQs to SQL queries, because it provides more chances to correct the errors in a sequence of conversations.

The success of the dialogue system and CoSQL also validates the necessity of *multi-turn exchanges* between machines and users in database scenarios. Our work is the first to study multi-turn exchanges between machines and users in the DV area.

6.3 Dialogue System

Tremendous efforts have been made by the community in designing various dialogue systems. The existing dialogue systems could be roughly divided into *open-domain* and *task-oriented ones.* The former focuses on providing chitchat or general responses to the users with typical systems like Microsoft's XiaoIce [64], Alibaba's AliMe [23], Google's Meena [4, 5], and Meta's BlenderBot [1]. Regarding implementation, retrieval-based and generation-based approaches were proposed for this task. The retrieval-based approach usually *chooses* the most appropriate instances from a library as the responses, whereas the generation-based method employs learning-based models to *generate* responses [21]. Recently, studies on combining both approaches were conducted to enjoy the merits of both methods [42, 53]. In contrast, the objective of task-oriented dialogue systems is typically to facilitate users to accomplish specific tasks in specific domains [10, 62]. Typical approaches to the task-oriented dialogue systems includes hybrid solutions [8] and the end-to-end ones [25, 51, 54]. In addition to textual dialogue systems, multi-modal dialogue systems are another direction that shows excellent usefulness in scenarios such as product recommendation in e-commerce [17, 26, 34, 38]. However, these multi-modal dialogue systems (e.g., MAGIC [34]) usually treat the product recommendation task as an image selection one from a candidate pool. Our task requires the model to *generate the responses from scratch* without any image candidate pool, which is much harder than selecting appropriate images from a large set of candidates. Thus, these exsiting multi-modal dialogue systems cannot perform well on this new proposed CoVis task. Our proposed CoVis task marries the popular DV task with dialogue system, and the proposed MMCoVisNet model enlarges the family of multi-modal dialogue systems and would greatly alleviate the barriers to using DV systems for beginners and non-technical background users.

7 CONCLUSION

We introduce a new task named CoVis, aiming to generate data visualizations from conversations, and it could be used to construct interactive DV dialogue systems. To promote this field, we first construct a benchmark dataset, which includes a sequence of dialogues with diverse queries and responses. Then, we propose a multi-modal neural network named MMCoVisNet to answer these DV-related queries. Experimental results have validated that MMCoVisNet achieves superior performance over other existing baselines. This study is a good start in this new CoVis task. After we validate the feasibility of this task, there are many potential studies following this line of research. For example, a potential direction would be designing advanced neural structures like graph neural networks for modeling the connections in the dialogue context.

Acknowledgments We thank reviewers for their valuable comments and effort to improve this manuscript. Yuanfeng Song and Raymond Chi-Wing Wong is supported by fund WEB24EG01-H.

REFERENCES

[1] Last accessed 16 Jan. 2023. BlenderBot 3: An AI Chatbot That Improves Through Conversation. https://about.fb.com/news/2022/08/blenderbot-ai-chatbot-improves-through-conversation/

[2] Last accessed 16 Jan. 2023. Microsoft Power BI. https://powerbi.microsoft.com/en-us/what-is-power-bi/

[3] Last accessed 16 Jan. 2023. SpotIQ: AI-Driven Insights. https://www.thoughtspot.com/spotiq

[4] Last accessed 16 Jan. 2023. Towards a Conversational Agent that Can Chat About... Anything. https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html

[5] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977* (2020).

[6] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal* 28, 5 (2019), 793–819.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[8] Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 239–251.

[9] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[10] Zheqian Chen, Rongqin Yang, Zhou Zhao, Deng Cai, and Xiaofei He. 2018. Dialogue act recognition via crf-attentive structured network. In *The 41st international acm sigir conference on research & development in information retrieval*. 225–234.

[11] Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guan Lou, and Dongmei Zhang. 2019. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 906–916.

[12] Anna Currey and Kenneth Heafield. 2019. Incorporating source syntax into transformer-based neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. 24–33.

[13] Victor Dibia and Çağatay Demiralp. 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications* 39, 5 (2019), 33–46.

[14] Usama M Fayyad, Georges G Grinstein, and Andreas Wierse. 2002. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann.

[15] Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4524–4535.

[16] Pat Hanrahan. 2006. Vizql: a language for query, analysis and visualization. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. 721–721.

[17] Weidong He, Zhi Li, Dongcai Lu, Enhong Chen, Tong Xu, Baoxing Huai, and Jing Yuan. 2020. Multimodal dialogue systems via capturing context-aware dependencies of semantic elements. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2755–2764.

[18] TuBao Ho, TrongDung Nguyen, and DungDuc Nguyen. 2002. Visualization support for a user-centered KDD process. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 519–524.

[19] Radu Cristian Alexandru Iacob, Florin Brad, Elena-Simona Apostol, Ciprian-Octavian Truică, Ionel Alexandru Hosu, and Traian Rebedea. 2020. Neural approaches for natural language interfaces to databases: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. 381–395.

[20] Noah Iliinsky and Julie Steele. 2011. *Designing data visualizations: Representing informational Relationships*. " O'Reilly Media, Inc.".

[21] Kristiina Jokinen and Michael McTear. 2009. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies* 2, 1 (2009), 1–151.

[22] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. 2018. ECharts: a declarative framework for rapid construction of web-based visualization. *Visual Informatics* 2, 2 (2018), 136–146.

[23] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al. 2017. Alime assist: An intelligent assistant for creating an innovative e-commerce experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2495–2498.

[24] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9267–9276.

[25] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 733–743.

[26] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-seng Chua. 2018. Knowledge-aware multimodal dialogue systems. In *Proceedings of the 26th ACM international conference on Multimedia*. 801–809.

[27] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[28] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 101–112.

[29] Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018. Deepeye: Creating good data visualizations by keyword search. In *Proceedings of the 2018 International Conference on Management of Data*. 1733–1736.

[30] Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021. Synthesizing Natural Language to Visualization (NL2VIS) Benchmarks from NL2SQL Benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*. 1235–1247.

[31] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2021. Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics* (2021).

[32] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe, and Jeffrey Heer. 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 438–448.

[33] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2020. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 369–379.

[34] Liqiang Nie, Wenjie Wang, Richang Hong, Meng Wang, and Qi Tian. 2019. Multimodal dialog system: Generating responses via adaptive decoders. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1098–1106.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

[36] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[37] Xin Qian, Ryan A Rossi, Fan Du, Sungchul Kim, Eunyee Koh, Sana Malik, Tak Yeon Lee, and Joel Chan. 2021. Learning to Recommend Visualizations from Data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1359–1369.

[38] Amrita Saha, Mitesh Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[39] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 341–350.

[40] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[41] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System. *Proceedings of the VLDB Endowment* 10, 4 (2016).

[42] Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. 2018. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4382–4388.

[43] Yuanfeng Song, Xuefang Zhao, Raymond Chi-Wing Wong, and Di Jiang. 2022. RGVisNet: A Hybrid Retrieval-Generation Neural Framework Towards Automatic Data Visualization Generation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1646–1655.

[44] Arjun Srinivasan and John Stasko. 2017. Natural language interfaces for data analysis with visualization: Considering what has and could be asked. In *Proceedings of the Eurographics/IEEE VGTC conference on visualization: Short papers*. 55–59.

[45] Yibo Sun, Duyu Tang, Nan Duan, Jianshu Ji, Guihong Cao, Xiaocheng Feng, Bing Qin, Ting Liu, and Ming Zhou. 2018. Semantic Parsing with Syntax-and Table-Aware SQL Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 361–372.

[46] Jiawei Tang, Yuyu Luo, Mourad Ouzzani, Guoliang Li, and Hongyang Chen. 2022. Sevi: Speech-to-visualization through neural machine translation. In *Proceedings*

of the 2022 International Conference on Management of Data. 2353–2356.

[47] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. 2017. Towards visualization recommendation systems. *ACM SIGMOD Record* 45, 4 (2017), 34–39.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[49] Caiming Xiong Victor Zhong and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR* abs/1709.00103 (2017).

[50] Randle Aaron M Villanueva and Zhuo Job Chen. 2019. ggplot2: elegant graphics for data analysis.

[51] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 438–449.

[52] Xiaojun Xu, Chang Liu, and Dawn Song. 2017. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436* (2017).

[53] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1341–1350.

[54] Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14230–14238.

[55] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. 2018. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 588–594.

[56] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir R Radev, Richard Socher, and Caiming Xiong. 2021. GraPPa: Grammar-Augmented Pre-Training for Table Semantic Parsing. In *International Conference on Learning Representations*.

[57] Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. 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)*. 1962–1979.

[58] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3911–3921.

[59] Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, et al. 2019. SParC: Cross-Domain Semantic Parsing in Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4511–4523.

[60] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* 31, 7 (2019), 1235–1270.

[61] Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2019. A comparison of Transformer and LSTM encoder decoder models for ASR. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 8–15.

[62] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences* 63, 10 (2020), 2011–2027.

[63] Xiangyu Zhao, Longbiao Wang, Ruifang He, Ting Yang, Jinxin Chang, and Ruifang Wang. 2020. Multiple knowledge syncretic transformer for natural dialogue generation. In *Proceedings of The Web Conference 2020*. 752–762.

[64] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics* 46, 1 (2020), 53–93.

APPENDIX

A EVALUATION METRICS

We have a detailed definition about the evaluation metrics. The first set is to evaluate the quality of the textual-form response.

- **BLEU**: BLEU, short for Bilingual Evaluation Understudy, is the most widely used metric in NLG tasks. In BLEU, the calculation of n-gram precision and recall are modified, which counts the percentage of N-gram co-occurrences between candidate and reference texts. The formal definition of BLEU is:

$$\mathtt{BLEU} = \mathtt{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), w_n = 1/n \tag{18}$$

  where p_n is the modified n-gram precision. The brevity penalty BP is calculated as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}, \tag{19}$$

  where c is the length of candidate text and r is the effective reference text length. In this paper, we reported BLEU-4 for all our experiments.
- **ROUGE**: ROUGE is short for Recall Oriented Understudy for Gisting Evaluation. Based on different features used to compute recall, ROUGE includes types including ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-L is based on the longest common subsequence (LCS), and uses F-score instead of using only recall. We report the ROUGE-L score in our experiment.
- **METEOR**: METEOR stands for Metric for Evaluation for Translation with Explicit Ordering, which is another popular evaluation metric for text generation tasks. Compared with BLEU, METEOR could reflect a better correlation with human judgment. METEOR is formally defined as:

$$\mathtt{METEOR} = F_{mean}(1-p), \tag{20}$$

  where F_{mean} is the *harmonic mean* of the unigram's precision P and recall R,

$$F_{mean} = \frac{PR}{\alpha P + (1-\alpha)R}, \tag{21}$$

  where α is a parameter between 0 and 1 with default value 0.9. The text segment's penalty p function is defined as

$$p = \gamma(\frac{c}{m})^{\beta}, \ \mathtt{where}\ 0 \leq \gamma \leq 1, \tag{22}$$

  where c is the number of chunks and m is the number of mapped unigrams between two texts, β is a parameter with default value set to be 3.

The second set is to evaluate the quality of the SQL-form response.

- **Sketch Accuracy**: This metric measures the percentage of matches between the structure of the predicted AST with the ground-truth one.
- **Overall Accuracy**: This metric reflects the percentage of matches between the final generated SQL query and the ground truth. To mitigate the impact of condition order, we further decompose each SQL statement into distinct components like "select column", "aggregator", and "condition". Rather than directly comparing the SQL string, we treat the condition part as a set and compare each element within the set.

The last set is to evaluate the quality of the DV-form response.

- **DV Accuracy:** This metric measures whether the whole predicted DV query sentence matches the ground truth DV query, which gives the overall comprehensive performance of the text-to-vis model. The overall accuracy is calculated as $Acc = N_c/N$, where N_c refers the number of DV queries predicted correctly and N is the total number of ground truth.
- **Vis Accuracy:** This metrics measures the match of the visual types between the predicted DV query and ground truth query. The accuracy is calculated as $Vis\ Acc = N_{vis}/N$, where N_{vis} denotes the number of DV queries in which visual type is predicted correctly.
- **Data Accuracy:** The data parts in DV query are composed of data values with transformation from database. The accuracy of data components is denoted as $Data\ Acc = N_{data}/N$, where N_{data} is the number of DV queries in which data components are predicted correctly.
- **Axis Accuracy:** We measure the match of select component of the DV query for the *x/y/z-axis* part. The axis accuracy, denoted as Axis Acc, is formally defined as the ratio of correctly predicted axis components to the total number of axis components, expressed as: $Axis\ Acc = N_{axis}/N$, where N_{axis} represents the number of x/y/z-axis components that match the ground truth labels, and N is the total number of axis components being evaluated.

B LIMITATIONS

We also identify several limitations that may worth further exploration in future studies: (i) *Extension to more DV types*: This study focused on a specific types of DVs, such as bar or scatter charts. However, there are many other important types of DVs like heatmaps. Extending the dataset and approach to handle more DV types would increase its practical utility. We expect studies on more comprehensive datasets covering diverse DV types could be conducted by the community soon. (ii) *High-quality dataset from real-human interactions*: The dataset used in this work consisted of examples generated automatically using rules and algorithms from single-turn text-to-vis scenarios. While this allows for preliminary research to validate the feasibility of this task, it may not fully capture the complexities of how humans naturally describe data insights in conversational settings. Collecting a dataset of real dialogues where humans describe insights and request visualizations could lead to more realistic benchmarks and promote advancements in handling the pragmatics of conversational text-to-vis. (iii) *LLM-based approach for conversational text-to-vis*: The current methods employ traditional neural network-based techniques on DV task. Recent progress in large language models (LLMs) has demonstrated their strong capabilities in understanding natural language. An interesting direction is to explore advanced multi-step or multi-turn prompting methods to map from conversational inputs to DV outputs.