Natural Language Generation Meets Data Visualization: Vis-to-Text and its Duality with Text-to-Vis

Yuanfeng Song*†, Xiaoling Huang†, Xuefang Zhao†, Raymond Chi-Wing Wong*
*HKUST, Hong Kong, China †AI Group, Webank Co., Ltd., Shenzhen, China
*{songyf, raywong}@cse.ust.hk †{smallhuang, summerzhao}@webank.com

***Abstract*—Data visualizations (DVs) refer to the methodologies and tools where visual elements like charts, bars and scatters are used to convey summaries behind the raw data. However, it usually takes much effort to master DV, even for data scientists and experts, not to mention beginners. Hence, a popular task named *Text-to-Vis*, focusing on automatically generating data visualizations (DVs) from natural language questions (NLQs), has been introduced and has recently been gaining great attention from both the database and the data mining communities. In this paper, we propose the reversed task *Vis-to-Text*, which aims to generate human-readable descriptions to explain complicated DVs for educational purposes.**

Intuitively, text-to-vis and vis-to-text are strongly correlated to one another, i.e., the input of text-to-vis is the output of vis-to-text, and vice versa. This relationship is generally known as duality and has been validated to be important for improving the performances of both tasks in machine translation, question answering, and dialogue systems. However, its effectiveness in text-to-vis and vis-to-text is under-explored. In this paper, we make use of the duality to optimize both tasks. We first design a Transformer-based network to tackle the vis-to-text task. Then, we further explore a dual training framework to simultaneously optimize the two tasks. More specifically, we analyze the duality and finally convert it into a corresponding regularization term to constrain the loss function to guide the model training process. Finally, we evaluate our approach on a public dataset, and the experimental results validate the rationale of this new proposed vis-to-text task and also show that this dual framework can boost the performance of the vis-to-text task over existing baselines.

***Index Terms*—data visualization, vis-to-text, dual learning**

I. Introduction

We are living in the era of Big Data, where web data is growing at a faster rate than ever before in the digital universe. Data visualizations (DVs), which use visual elements (e.g., charts, bars and maps) to represent information, provide an accessible and efficient way of communicating and conveying the trends and patterns behind the massive data [1]. As such, more and more industrial institutions have applied DV tools and technologies in their daily operations. At the same time, DV has also become a popular research area and has been extensively explored by the database [2]–[6] and the data mining [7]–[10] communities. In particular, in the data mining community, many papers about DVs [7]–[11] were published in KDD/ICDM over the past 4 years. For example, RGVisNet [10] in KDD'22 studied the problem of automatically translating natural language questions (NLQs) into DVs to avoid the difficulty of using DVs.

In spite of its great usefulness, manually creating suitable DVs is still quite hard since it requires the users mastering *declarative visualization languages* (DVLs) such as Vega-Lite [12], ggplot2 [13], ZQL [14], ECharts [15] and VizQL [5]. At the same time, the users should also have enough domain knowledge of and expertise with the data to choose the *specification* (see the example in Fig. 1) of the visualization details (e.g., variables of interest, data transformations and visual encodings). These prerequisites formulate a high barrier to entry into the DV field particularly for those who are new to the topic or lack technical background.

To further lower the barriers of mastering DVs, a popular *Text-to-Vis* (or NL2Vis, natural language to data visualization) task is proposed to automatically translate the user's visualization analysis intent, in the form of NLQs, into DVs [2], [3], [6], [16], [17]. Substantial effort has been made to tackle this text-to-vis task. For example, Seq2Vis [2] uses a sequence-to-sequence network to translate NLQs into visualization specification (in Vega-Lite), and then the visualization specification is rendered by the Vega-Lite engine to obtain the final DV chart. Data2Vis [18] is another work that employs recurrent neural networks (RNNs) to automatically generate DVs with the inputs of data sequences. These text-to-vis models and studies greatly promote the development of the DV area since they shed insights for more user-friendly natural language-based interfaces to manipulate DV systems.

In this paper, we propose a correlated and reversed task named *Vis-to-Text*, which focuses on generating human-readable descriptions to explain the complicated visualization specifications for educational purposes. This new proposed vis-to-text task is quite different from the well-known chart-to-text task [19] and the image caption generation task [20], [21], in terms of task objective, the input, and the corresponding methodology. More specifically, the natural language descriptions in vis-to-text play a similar role as the comments for general programming languages such as Python and Java, and vis-to-text could be considered as a special case of the general automatic comments generation task [22], [23]. The vis-to-text task could not only enhance the learning experience when users try to comprehend DV specifications since the generated explanations can be used for educational purposes but it could also be used as a critical component for existing DV recommendation systems [4], [7] to provide explanations

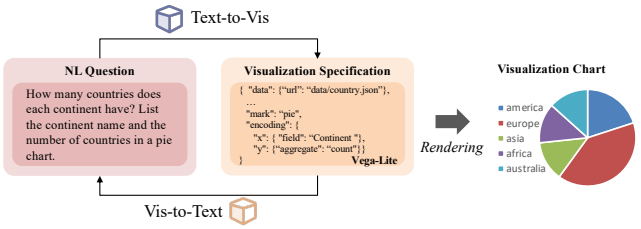

Fig. 1. An illustration of the text-to-vis and the vis-to-text tasks, with examples of an NLQ, its corresponding visualization specification in Vega-Lite, and the rendered visualization chart. Vis-to-text is a reversed task of the popular text-to-vis task.

for the recommended DV results.

Despite aforementioned strong motivations, existing studies mainly focus on text-to-vis, and limited effort has been made on the vis-to-text task. Furthermore, none of them try to explore the relationship between these two tasks and make use of their connections to direct the design of their approaches and enhance the performance of both tasks. Intuitively, text-to-vis and vis-to-text are strongly correlated to one another, i.e., the input of text-to-vis is the output of vis-to-text, and vice versa. This relationship is known as *duality* [24], [25], and inspired by recent progress in dual learning and their success in correlated tasks such as automatic machine translation [24], [25], question answering and question generation [26], and code generation and code summarization [27], we pioneer the exploration of the duality between the text-to-vis and vis-to-text tasks and thus improves the performance of both tasks. Fig. 1 gives a detailed example of these two tasks and shows the connections between them.

In this paper, we design a Transformer-based network structure to handle this vis-to-text task, and at the same time, also propose a dual learning framework to construct a text-to-vis model with this vis-to-text model simultaneously to fully utilize their duality. We consider the duality on probability and convert them into corresponding regularization terms to constrain the loss function to guide the model training process. Furthermore, the common part between the two tasks, the schema encoder, is also shared between these two tasks. We evaluate our approach on a public dataset *NVBench* [2], and extensive experimental results validate the rationale of this new proposed vis-to-text task. Furthermore, it shows that our dual framework can boost the performance of the vis-to-text tasks over existing baselines. For example, for vis-to-text task, our model surpasses a Transformer-based baseline by 3.6% in terms of BLEU, a measure reflecting the quality of the generated text. This study marries two very important fields from data mining (i.e., DV) and natural language processing (i.e., natural language generation, NLG in short) and we hope it will inspire more cross-field research.

In a nutshell, the main contributions of this work include:

- We propose a new task named vis-to-text to promote the development of the DV field. Vis-to-text aims to generate explanations for complicated DVs, and these explanations help users to easily understand the DVs, either from existing projects or from DV recommendation systems.
- To the best of our knowledge, this is the first work in the literature that jointly models text-to-vis and vis-to-text tasks. We make an effort to consider text-to-vis and vis-to-text as dual tasks and employ the dual framework to boost one another. This framework is so general that any text-to-vis and vis-to-text models can be incorporated.
- Inspired by other dual learning studies [27], we treat the duality as a probabilistic correlation between text-to-vis and vis-to-text task and then convert it into a regularization term in the loss function.
- We evaluate our approach on a public dataset in the DV area and the results validates that this dual framework can boost the performance of the vis-to-text task over baselines by more than 3.6% relative improvements. We hope that our work would enlarge the dual learning family and inspire more research on exploring similar connections between other correlated tasks.

II. Related Work

We briefly survey the related studies from three fields: text-to-vis, natural language generation, and dual learning.

Text-to-Vis. Text-to-vis aims to provide text-based interfaces to interact with DV systems, which is more user-friendly for non-experts. It takes the NLQs and database schema as the inputs and then automatically generates the DVs. Existing literature on this task can be broadly categorized into the rule-based approach and the DNN-based approach. The common practice of the DNN-based approach is to convert it into a machine translation problem and then employ advanced Seq2Seq models, with popular methods like Seq2Vis [2], ncNet [28], and RGVisNet [10].

Natural Language Generation. The vis-to-text task is intrinsically a natural language generation (NLG) problem, and it has been extensively studied in various applications in NLP. For example, in the programming code area, the generated text is also known as comments, which is used to enhance the learning experience of programmers when they maintain software projects [22], [23]. In the image area, the task aims at captioning the images with human-understandable descriptions [21], [29]. Another common area is recommendation system [30], [31]. To name a few, Zou *et al.* proposed a multi-modal approach to generate descriptions for Points of Interests in the digital map [31]. Li *et al.* aim to generate abstractive tips for recommendation systems [30].

Dual Learning. The concept of dual learning was first proposed by [25] for machine translation tasks, which considers the French-to-English translations as primal tasks and French-to-English translations as their dual-task. The dual correlation is validated to be effective in improving both tasks in this scenario. Dual learning has also been extended to supervised learning and is widely used in other tasks [24] such as image-to-image translation, [32] and open-domain information narration and extraction [33]. Furthermore, dual learning is also utilized in the question answering field [34], which treats

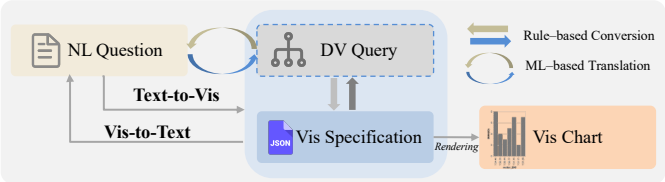

Fig. 2. The practical implementation of text-to-vis and the vis-to-text tasks. For both tasks, the NLQ is first translated into or from a intermedia named DV query via machine learning (ML)-based models. The transformation between the DV query and the visualization specification is easily achieved by some rule-based conversion.

the answer generation and question generation as reversed tasks. A similar idea is used in the dialogue generation task [35], code summarization and code generation tasks [27], [36]. We enlarge the dual learning family to the previously unexplored DV field.

III. Methodology

In this section, we are ready to introduce the details of our methodology. We first give an overview of the framework in Section III-A, then we introduce the vis-to-text and text-to-vis translation model in Section III-B and the details about the dual learning component in Section III-C.

A. Framework Overview

Following the common practice in text-to-vis [2], [10], we also utilize the concept of DV query to bridge the gap between the NLQ and the visualization specification. As shown in Fig. 2, a DV query can easily be converted into/from a visualization specification. Then, the overall training framework is composed of three main components: a text-to-vis model, a vis-to-text model and dual learning constraints, as illustrated in Fig. 3. The text-to-vis model focuses on automatically translating NLQs into DV queries, while the vis-to-text model reversely maps DV queries to NL descriptions. Similar encoder-decoder translation architecture is employed for these two models. The constraint from dual learning is mainly used as a regularization term in the loss function to guide the optimization process. It should be noticed that our proposed approach is general enough that other possible text-to-vis and vis-to-text models may also be optimized under this framework, besides our proposed Transformer-based one.

B. The Vis-to-Text and The Text-to-Vis Translation Model

A Transformer-based network is employed to implement the vis-to-text model to validate the rationale of this newly proposed task. Conversely, text-to-vis is a prevalent task with a series of existing studies like Seq2Vis [2]. Since the main focus of our work is to validate the proposed vis-to-text model and then explore the duality between vis-to-text and text-to-vis tasks, a similar Transformer-based structure rather than a more complicated one (e.g., RGVisNet [10]) is adopted for the text-to-vis part of the framework. The vis-to-text model includes three essential components: a *Vis Encoder*, a *Schema Encoder*, and a *NL Decoder*, while the text-to-vis model enjoys a similar structure, which is also denoted as the NL Encoder, the Schema Encoder, and the Vis Decoder. To simplify illustration and reduce redundancy, we only detail the vis-to-text model, and the text-to-vis model is constructed in a similar style.

1) Vis Encoder: The objective of Vis Encoder is to convert the DV query y into some hidden representations. We treat the query as a textual sequence of tokens, represented as $y = \{y_1, y_2, ..., y_n\}$, where n refers to the length of sequence. We map each token in y into its embedding using the pre-trained BERT model and then obtain $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2 ..., \mathbf{y}_n\}$. Then, this sequence of token embeddings is fed into bi-directional LSTMs (BiLSTMs) so that we can leverage the contextual information and get the hidden vectors for the DV query. The whole process can be represented as

$$\overrightarrow{\mathbf{h}}_t^y = \texttt{LSTM}(\mathbf{y}_t, \overrightarrow{\mathbf{h}}_{t-1}^y), \tag{1}$$

$$\overleftarrow{\mathbf{h}}_t^y = \texttt{LSTM}(\mathbf{y}_t, \overleftarrow{\mathbf{h}}_{t+1}^y), \tag{2}$$

where $\overrightarrow{\mathbf{h}}_t^y$ and $\overleftarrow{\mathbf{h}}_t^y$ denote the hidden states of time step t for forward and backward LSTM. We concatenate these two vectors as Eq. (3) and get the final embedding H_y for DV query $\mathbf{y}$ and $H_y = \{\mathbf{h}_1^y, \mathbf{h}_2^y ..., \mathbf{h}_n^y\}$.

$$\mathbf{h}_t^y = [\overrightarrow{\mathbf{h}}_t^y; \overleftarrow{\mathbf{h}}_t^y], \tag{3}$$

where $[;]$ represents the concatenation operation.

2) Schema Encoder: Given a schema s, it consists of a list of elements including table names and column names. We combine all the elements and also process them into a sequence of tokens. Given the processed schema sequence, we use the similar operations and network structure mentioned in the Section III-B1 to obtain the schema embedding H_s. Due to the importance of the schema linking strategy studied in previous research, we designed an attention module to integrate the schema information into the original query, as shown in the following equations. A context vector c_i^y is calculated by the weighted sum of schema embedding and we then obtain the final encoder outputs $H = \{\mathbf{h}_1, \mathbf{h}_2 ..., \mathbf{h}_n\}$.

$$e_{ij} = \frac{(\mathbf{h}_i^y)^T \mathbf{h}_j^s}{\|\mathbf{h}_i^y\| \, \|\mathbf{h}_j^s\|}, \tag{4}$$

$$c_i^y = \sum_{j=1}^{l} e_{ij} \mathbf{h}_j^s, \tag{5}$$

$$\mathbf{h}_i = \mathbf{h}_i^y + c_i^y, \tag{6}$$

where $\mathbf{h}_j^s$ represents the hidden states of schema s at time step j and the l is the total length of schema sequence.

3) NL Decoder: We employ a Transformer-based architecture which takes advantage of multi-head scaled dot-product attention as the NL decoder to generate the target NL description q. In Transformer-based architecture, given the query Q, the key K, and the value V, the attention mechanism performs the calculation as Eq. (7), which obtains attention weights between Q and K and assigns a new value for Q.

$$\texttt{Attention}(Q, K, V) = \texttt{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{7}$$

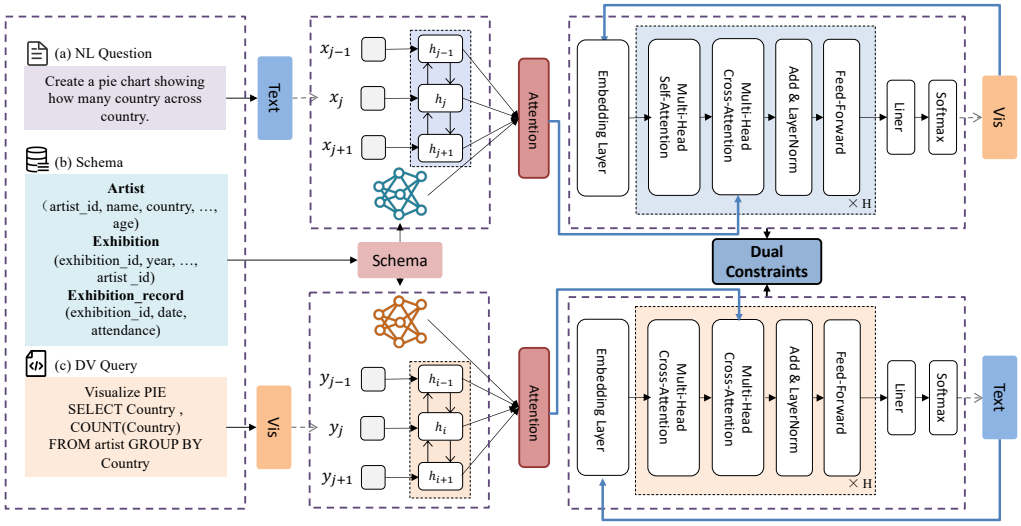

Fig. 3. The overall dual training framework with jointly training a Text-to-Vis model and a Vis-to-Text model, with a shared schema encoder.

where d_k is the dimension of the input key vector K. As for the multi-head attention, Q, K, and V are projected h times into different representation spaces, and then the concatenated result is the final output. Specifically, we have

$$head_i = \texttt{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{8}$$

$$\texttt{MultiHead}(Q,K,V) = [head_1; \cdots; head_h]W^O, \tag{9}$$

W_i^Q, W_i^K, W_i^V and W^O are trainable parameter matrices.

There are two multi-head attention sub-layers in our decoder layer, and a feed-forward network is put on the top of the attention module. The decoder is stacked by several identical described layers, and it takes target q as input to generate the decoder representation H_q. Specifically, the first multi-head attention sub-layer performs self-attention on H_q, that is, $Q = K = V = H_q$. The second one calculates cross-attention between the final encoder outputs H and the decoder representation H_q, namely $Q = H_q$ and $K = V = H$. Finally, the decoder outputs are fed into a linear network followed by a Softmax activation to give the probability distribution of word generation, and the loss function $\mathcal{L}_{vt}(\cdot)$ is formulated as

$$H_o = FFN(MultiHead(Q,K,V)), \tag{10}$$

$$p(q = q_t|y,s,q_{<t};\theta_{vt}) = Softmax(H_o), \tag{11}$$

$$\mathcal{L}_{vt}(g(y,s;\theta_{vt}),q) = \sum_{t=1}^{L} p(q = q_t|y,s,q_{<t};\theta_{vt}), \tag{12}$$

where $q_{<t}$ denotes the previous tokens before t.

C. Dual Training Framework

Many tasks such as machine translation, question answering and question generation, appear in dual form, where one is called the primal task while the other is called the dual one. Xia *et al.* [24] proposed the concept named *dual learning* to make use of this duality between two dual tasks. Inspired by the success of dual learning on these tasks like [27], we design our specific dual learning framework dedicated for text-to-vis and vis-to-text, which is more complicated due to the existence of schema in the input.

1) Dual Constrains as Regularization Terms: In our scenario, we take the text-to-vis as the primal task, and consider vis-to-text as the dual task. Given training instances in the form of $\{q,s,y\}$ from the dataset where $q \in \mathcal{Q}$, $s \in \mathcal{S}$, and $y \in \mathcal{Y}$, dual learning aims to jointly learn a text-to-vis model $f: \mathcal{Q},\mathcal{S} \to \mathcal{Y}$ and a vis-to-text model $g: \mathcal{Y},\mathcal{S} \to \mathcal{Q}$. For the text-to-vis task, we have the following constraint:

$$P(q,s,y) = P(q)P(s|q)P(y|q,s;\theta_{tv}), \forall (q,s,y), \tag{13}$$

where θ_{tv} is the parameter of the text-to-vis model $f(\cdot)$. Meanwhile, for the vis-to-text task, we could also get a similar constraint:

$$P(q,s,y) = P(y)P(s|y)P(x|y,s;\theta_{vt}), \forall (q,s,y), \tag{14}$$

where θ_{vt} is the parameter of the vis-to-text model $g(\cdot)$.

The traditional training objective is now transformed into a multi-objective optimization one as

$$\begin{aligned} &\min_{\theta_{tv}} \sum_{i=1}^{N} \mathcal{L}_{tv}(f(q_i,s_i;\theta_{tv}),y_i); \\ &\min_{\theta_{vt}} \sum_{i=1}^{N} \mathcal{L}_{vt}(g(y_i,s_i;\theta_{vt}),q_i); \\ &\text{s.t. } P(q)P(s|q)P(y|q,s;\theta_{tv}) \\ &\quad = P(y)P(s|y)P(q|y,s;\theta_{vt}), \forall i \in [1,N], \end{aligned} \tag{15}$$

where $\mathcal{L}_{tv}(\cdot)$ and $\mathcal{L}_{vt}(\cdot)$ are the loss functions used in optimizing $f(\cdot)$ and $g(\cdot)$. Finally, this constraint could be transformed into a regularization term to the original loss function as

$$\begin{aligned} \mathcal{L}_{dual} = &[\log P(q) + \log P(s|q) + \log P(y|q,s;\theta_{tv}) \\ &- \log P(y) - \log P(s|y) - \log P(q|y,s;\theta_{vt})]^2. \end{aligned} \tag{16}$$

ALGORITHM 1: Dual Learning Framework for Jointly Training Text-to-Vis and Vis-to-Text Models

Input: Language Models $P(t)$ and $P(s)$ for text and vis, respectively; hyper parameters λ_{tv} and λ_{vt};
Output: Text-to-Vis model $f_{tv}(\cdot)$ with its parameters θ_{tv}; Vis-to-Text model $g_{vt}(\cdot)$ with its parameters θ_{vt}
begin
  Randomly initialize θ_{tv} and θ_{vt};
  while *not reach convergence* **do**
    Get a batch of m pairs $\{(q_j, s_j, y_j)\}_{j=1}^{m}$;
    Obtain the gradients as:
    $$G_f = \nabla_{\theta_{tv}}(1/m)\sum_{j=1}^{m}\Big[\mathcal{L}_{tv}(f(q_j, s_i; \theta_{tv}), y_j) + \lambda_{tv}\mathcal{L}_{dual}(q_j, s_j, y_j; \theta_{tv}, \theta_{vt})\Big];$$
    $$G_g = \nabla_{\theta_{vt}}(1/m)\sum_{j=1}^{m}\Big[\mathcal{L}_{vt}(g(y_j, s_j; \theta_{vt}), q_j) + \lambda_{vt}\mathcal{L}_{dual}(q_j, s_j, y_j; \theta_{tv}, \theta_{vt})\Big]$$
    Update θ_{tv} and θ_{vt} with G_f and G_g;

In Eq. (16), the $P(q)$ and $P(y)$ could be obtained by two language models (LMs) constructed with the NLQ and the DV data in the training set. Rather than composing two new schema prediction models to calculate $P(s|q)$ and $P(s|y)$, we simply assume that $P(s|q)$ equals to $P(s|y)$ since NLQ q and DV query y are two representations expressing the same semantic. This assumption could greatly simplify our dual loss function and the whole framework. Them, we could obtain the whole algorithm, as illustrated in Algorithm 1.

IV. Experiments

In this section, we first introduce the experimental setup and then cover the key experimental results.

A. Experimental Setup

We use the public NL2Vis dataset *NVBench* [2] in our evaluation. Following [2], we also randomly split the NVBench dataset in a *question-based* style into the training set and the testing set. Our baselines for vis-to-text includes Graph2Seq [37] and Transformer [38]. The evaluation metrics include BLEU, ROUGE (1, 2, and L), and METEOR for vis-to-text. Due to space limitations, we mainly include and focus on the results of the proposed vis-to-text task.

B. Experimental Results

1) Performance Comparison: We mainly study the vis-to-text task, and Table I displays the automatic metrics values of our designed model alongside the baselines on NVBench. We can see that the baselines such as Graph2Seq is uncompetitive in generating descriptions for DVs. The main reason is that it is based on basic network structures and it has a limited ability to understand the information hidden in the NLQ. The Transformer-based approach performs better than the Graph2Seq baseline. Among all these models, our proposed dual framework could significantly outperforms all compared methods, proving the necessity of taking advantage of duality information for the DV description generation task. In particular, the dual framework achieves a BLEU score of 0.459, a ROUGE-L score of 0.638 and a METEOR score of 0.662, respectively. It outperforms the Transformer-based baseline by 3.6% in terms of BLEU, by 1.91% in terms of ROUGE-L and by 1.05% in terms of METEOR.

TABLE I
Performance Comparison of on the Vis-to-Text Task

Model	BLEU	ROUGH-1	ROUGH-2	ROUGH-L	METEOR
Transformer	0.443	0.680	0.528	0.626	0.655
Graph2Seq	0.260	0.543	0.347	0.485	0.498
Our Method	**0.459**	**0.692**	**0.541**	**0.638**	**0.662**

TABLE II
Ablation Study

Model	BLEU	ROUGH-1	ROUGH-2	ROUGH-L	METEOR
Our Method	**0.459**	**0.692**	**0.541**	**0.638**	**0.662**
w/o Dual	0.454	0.687	0.535	0.630	0.658

2) Ablation Study (Effect of Dual Learning): We performed ablation studies in this section to demonstrate the effectiveness and value of the proposed dual learning framework. As a baseline, we first obtain the result of the full model with all components. Then we remove the dual component and name it **w/o dual**. Table II presents the overall results.

Again, we take the BLEU score as the main indicators in our analysis, and the other metrics reflect similar observations. From the results, we can see that it improves the vis-to-text model by 1.08% relative improvement in terms of BLEU score. The improvement demonstrates the necessity of integrating a dual learning mechanism to jointly optimize the text-to-vis and the vis-to-text tasks.

3) Case Study: We also provide one example to vividly show the descriptions generated by the baselines and our proposed models for text-to-vis tasks in Table III. In this example, the vanilla Transformer baseline fails to capture a significant part of the meaning represented in the DV, especially about the visualization type (i.e., Bar chart). Compared with the baseline, our proposed framework accurately generates the keywords such as the visualization details "*bar chart*" and the data details "*number of the lot details of lots that belong to investors with details 'l'*".

V. Conclusion

In this paper, we propose a new task named vis-to-text that aims at generating human-readable descriptions to explain complicated DVs. We also explore the dual learning for simultaneously training a text-to-vis and a vis-to-text model. Compared with baseline methods, the proposed framework can exploit the duality between the two tasks and achieve better performance. Extensive experimental results on a public DV

TABLE III
AN EXAMPLE OF DV DESCRIPTIONS GENERATED BY VARIOUS VIS-TO-TEXT METHODS

Input	Vis Specification	{“data”:{“values”:[{“x_data”:“h”,“y_data”:1},{“x_data”:“m”,“y_data”:2},{“x_data”:“s”,“y_data”:1},{“x_data”:“z”, “y_data”: 1}]},“mark”:“bar”,“encoding”:{“x”:{“field”:“x_data”,“type”:“nominal”,“title”:“lot_details“,“sort”:{“op”:“sum”, “field”:“y_data”,“order”:“ascending”}},“y”:{“field”: “y_data”,“type”:“quantitative”,“title”:“COUNT(lot_details)”}}}
Intermedia	DV Query	visualize bar select lot_details, count_lot_details from investors as t1 join lots as t2 on t1 investor id = t2 investor id where t1 investor details = “l” group by lot details
Expected Output	Target Description	a bar chart for returning the number of the lot details of lots that belong to investors with details “l”
Predicted Output	Transformer (×)	return the number of the lot details of lots that belong to investors with details “l”
	Our Method (✓)	a bar chart for returning the number of the lot details of lots that belong to investors with details “l”

dataset validate that this designed framework can boost the performance of our proposed task. This work also validate the feasibility of the vis-to-text task. Next, we would like to explore other advanced neural structures to improve the vis-to-text performance, e.g., incorporating multi-modal information from specification structures as well as the generated charts. Furthermore, with the growing popularity and capabilities of large language models (LLMs) (e.g., GPT-4 [39]) in both the research community and the industrial circle, we would like to investigate the performance of LLMs in this new task.

Acknowledgement. We are grateful to anonymous reviewers for their constructive comments on this draft. The research of Yuanfeng Song and Raymond Chi-Wing Wong is supported by PRP/026/21FX.

REFERENCES

[1] N. Iliinsky and J. Steele, *Designing data visualizations: Representing informational Relationships*. ” O’Reilly Media, Inc.”, 2011.
[2] 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.
[3] Y. Luo, X. Qin, N. Tang, G. Li, and X. Wang, “Deepeye: Creating good data visualizations by keyword search,” in *ICDE*, 2018.
[4] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, “Towards visualization recommendation systems,” *SIGMOD Record*, 2017.
[5] P. Hanrahan, “Vizql: a language for query, analysis and visualization,” in *SIGMOD*, 2006.
[6] X. Qin, Y. Luo, N. Tang, and G. Li, “Making data visualization more efficient and effective: a survey,” *VLDBJ*, 2020.
[7] X. Qian, R. A. Rossi, F. Du, S. Kim, E. Koh, S. Malik, T. Y. Lee, and J. Chan, “Learning to recommend visualizations from data,” in *KDD*, 2021.
[8] R. Savvides, A. Henelius, E. Oikarinen, and K. Puolamäki, “Significance of patterns in data visualisations,” in *KDD*, 2019.
[9] S. An, S. Hong, and J. Sun, “Viva: semi-supervised visualization via variational autoencoders,” in *ICDM*, 2020.
[10] Y. Song, X. Zhao, R. C.-W. Wong, and D. Jiang, “Rgvisnet: A hybrid retrieval-generation neural framework towards automatic data visualization generation,” in *KDD*, 2022.
[11] C. Fu, Y. Zhang, D. Cai, and X. Ren, “Atsne: Efficient and robust visualization on gpu through hierarchical optimization,” in *KDD*, 2019.
[12] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-lite: A grammar of interactive graphics,” *TVCG*, 2016.
[13] R. A. M. Villanueva and Z. J. Chen, “ggplot2: elegant graphics for data analysis,” 2019.
[14] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, “Effortless data exploration with zenvisage: An expressive and interactive visual analytics system,” *VLDB*, 2016.
[15] 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*, 2018.
[16] Y. Luo, C. Chai, X. Qin, N. Tang, and G. Li, “Interactive cleaning for progressive visualization through composite questions,” in *ICDE*, 2020.
[17] Y. Luo, X. Qin, C. Chai, N. Tang, G. Li, and W. Li, “Steerable self-driving data visualization,” *TKDE*, 2020.
[18] V. Dibia and Ç. Demiralp, “Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks,” *IEEE computer graphics and applications*, 2019.
[19] J. Obeid and E. Hoque, “Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model,” in *INLG*, 2020.
[20] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, “Guiding the long-short term memory model for image caption generation,” in *ICCV*, 2015.
[21] S. Bai and S. An, “A survey on automatic image caption generation,” *Neurocomputing*, vol. 311, pp. 291–304, 2018.
[22] T. Haije, B. O. K. Intelligentie, E. Gavves, and H. Heuer, “Automatic comment generation using a neural translation model,” *Inf. Softw. Technol*, vol. 55, no. 3, pp. 258–268, 2016.
[23] X. Hu, G. Li, X. Xia, D. Lo, and Z. Jin, “Deep code comment generation,” in *ICPC*, 2018.
[24] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu, “Dual supervised learning,” in *ICML*, 2017.
[25] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, “Dual learning for machine translation,” *NIPS*, 2016.
[26] Y. Li, N. Duan, B. Zhou, X. Chu, W. Ouyang, X. Wang, and M. Zhou, “Visual question generation as dual task of visual question answering,” in *CVPR*, 2018.
[27] B. Wei, G. Li, X. Xia, Z. Fu, and Z. Jin, “Code generation as a dual task of code summarization,” *NIPS*, 2019.
[28] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, “Natural language to visualization by neural machine translation,” *TVCG*, 2021.
[29] A. Jaffe, “Generating image descriptions using multilingual data,” in *WMT*, 2017, pp. 458–464.
[30] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, “Neural rating regression with abstractive tips generation for recommendation,” in *SIGIR*, 2017.
[31] M. Zhou, J. Zhou, Y. Fu, Z. Ren, X. Wang, and H. Xiong, “Description generation for points of interest,” in *ICDE*, 2021.
[32] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *ICCV*, 2017.
[33] M. Sun, X. Li, and P. Li, “Logician and orator: Learning from the duality between language and knowledge in open domain,” in *EMNLP*, 2018.
[34] D. Tang, N. Duan, T. Qin, Z. Yan, and M. Zhou, “Question answering and question generation as dual tasks,” *arXiv preprint arXiv:1706.02027*, 2017.
[35] S. Cui, R. Lian, D. Jiang, Y. Song, S. Bao, and Y. Jiang, “Dal: Dual adversarial learning for dialogue generation,” in *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 2019.
[36] W. Ye, R. Xie, J. Zhang, T. Hu, X. Wang, and S. Zhang, “Leveraging code generation to improve code retrieval and summarization via dual learning,” in *The Web Conference*, 2020.
[37] K. Xu, L. Wu, Z. Wang, Y. Feng, and V. Sheinin, “Sql-to-text generation with graph-to-sequence model,” in *EMNLP*, 2018.
[38] 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.
[39] OpenAI, “Gpt-4 technical report,” 2023.