Automatic Data Visualization Generation from Chinese Natural Language Questions

Yan Ge[2†], **Victor Junqiu Wei**[1*†], **Yuanfeng Song**[1,3*]
Jason Chen Zhang[2], **Raymond Chi-Wing Wong**[1]
[1]The Hong Kong University of Science and Technology
[2]The Hong Kong Polytechnic University, [3]WeBank Co., Ltd
yangecn@hotmail.com, jason-c.zhang@polyu.edu.hk
{jweiad, songyf, raywong}@cse.ust.hk

Abstract

Data visualization has emerged as an effective tool for getting insights from massive datasets. Due to the hardness of manipulating the programming languages of data visualization, automatic data visualization generation from natural languages (*Text-to-Vis*) is becoming increasingly popular. Despite the plethora of research efforts on the English Text-to-Vis, studies have yet to be conducted on data visualization generation from questions in other languages like Chinese. Motivated by this, we propose the first Chinese Text-to-Vis dataset named *CNvBench* in the paper and then demonstrate our first attempt to tackle this problem. Our model integrates multilingual BERT as the encoder, boosts the cross-lingual ability, and infuses the n-gram information into our word representation learning. Our experimental results show that our dataset is challenging and deserves further research.

Keywords: Data visualization, Chinese Text-to-Vis, Dataset construction

1. Introduction

Data visualization (Qin et al., 2020; Wang et al., 2021; Allen et al., 2019; Waskom, 2021) has become increasingly popular since it provides insights into data of massive size. In the pipeline of data visualization, an inevitable and inherent component is the creation of the *specifications*, which is achieved through the *declarative visualization languages* (DVL), (e.g., Vega-Lite (Satyanarayan et al., 2016) and EChart (Li et al., 2018)). This DVL specifies what data is required and how the data is supposed to be visualized. It requires users to have expertise and knowledge of the data domain and also good programming skills of DVL, which is not quite practical, esp. for novices.

Motivated by this, automatic DVL generation from natural language, or *Text-to-Vis*, is becoming an emerging topic since it could provide a much more user-friendly interface. Many research studies have been invested in this problem, such as (Cui et al., 2019; Gao et al., 2015; Luo et al., 2020; Narechania et al., 2020). Given a natural language question and a database, Text-to-Vis aims to automatically translate the question into the specification in some DVLs for data visualization. Despite the variety of studies on this topic, we observe that all existing datasets for Text-to-Vis are for English only, and no previous studies have been conducted on Chinese Text-to-Vis datasets and methodology. Chinese is one of the languages that enjoy the most users worldwide. The lack of Chinese datasets prevents using Text-to-Vis services among these users.

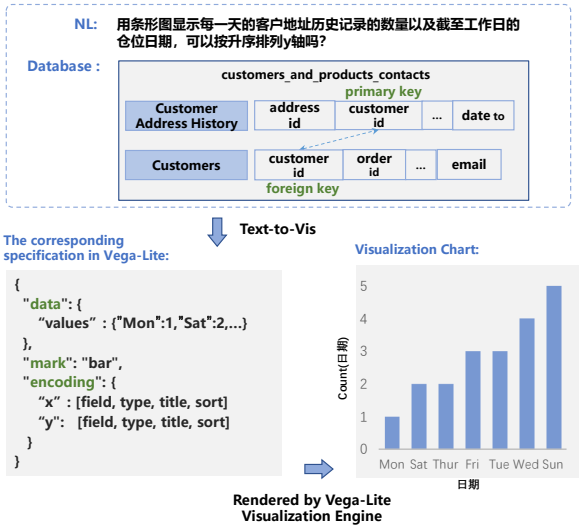

Figure 1: An example of Chinese Text-to-Vis. The input question is posed in Chinese, while the target chart that needs to be generated contains both Chinese linguistic elements as well as English linguistic elements, which makes it much harder than other parallel tasks like Chinese Text-to-SQL.

This work presents a Chinese Text-to-Vis dataset that imposes three challenges to the Text-to-Vis tasks. Firstly, the names of the attributes/columns in each table are typically represented in English, whereas the natural language questions are written in Chinese. This discrepancy requires the model to have cross-lingual ability. Secondly, the most basic

* Corresponding Author.
† Work done during their employment at The Hong Kong Polytechnic University.

units for denoting columns or cells can be Chinese characters, but the word segmentation can be erroneous. Third, the target chart that needs to be generated contains both Chinese and English linguistic elements. This requires the model to have multi-lingual generation capabilities, being able to produce outputs containing both Chinese and English words. This differs from existing parallel studies like Chinese Text-to-SQL (e.g., CSpider (Min et al., 2019)) , where the target SQL query usually only contains English keywords. The cross-lingual nature of generating charts based on Chinese text introduces an additional complexity not found in prior English-only Text-to-SQL tasks. Figure 1 is an example of the Chinese Text-to-Vis task[1]. Given a Chinese natural language question and a corresponding database, this task aims to generate a visualization based on the semantics of the question.

We present a novel neural model dedicated to this task. The model is designed to generate vivid and accurate visualizations directly from Chinese text descriptions. Specifically, the model contains the multilingual BERT (Kenton and Toutanova, 2019) as part of the encoder to boost the cross-lingual ability and also infuse n-gram information into the word representation learning process. In addition to introducing the first Chinese Text-to-Vis dataset, this work represents our initial effort to address the Chinese Text-to-Vis challenge. We demonstrate its capabilities on the proposed CNvBench dataset across various visualization types. The resource of this paper is available at `https://github.com/yangecn/CNvBench`.

In a nutshell, our contributions are summarized as follows.

- We propose a Chinese Text-to-Vis dataset named *CNvBench* in this paper. To our knowledge, this is the first Chinese Text-to-Vis dataset. We detail our construction method in this paper and release our dataset to promote the development of this field.
- We propose our model, the first attempt at this Chinese Text-to-Vis problem. It integrates the multilingual BERT and n-gram information to boost cross-lingual performance and word representation learning.
- We conduct extensive experiments on the proposed dataset, and the results show that our proposed Chinese Text-to-Vis task is challenging and the proposed model could achieve relatively good performance.

[1]The Vega-Lite language is primarily used in our discussion, due to its widespread usage and popularity (Song et al., 2022; Luo et al., 2021a; Qin et al., 2020; Luo et al., 2018). However, the proposed methodology is easily adaptable to other DVLs.

The rest of this paper is structured as follows: we first introduce some closely related work in Section 2. Then, we discuss the details of our proposed dataset in Section 3, followed by the discussion of our proposed model in Section 4. The experimental setup and results are listed in Section 5. Finally, we conclude the work in Section 6.

2. Related Work

This study is closely related to three fields, data visualization, Text-to-Vis methods, and Text-to-Vis datasets, as is briefly surveyed in the following.

2.1. Data Visualization

Data visualization, which converts abstract data into concrete, graphical representations, is naturally well-suited for providing an overview of large amounts of data. Data visualization can highlight patterns, trends, and relationships in the data that may not be immediately apparent from looking at raw data. To help data analysts gain more intuitive insights from their data, the researchers in this area have done a lot of work to make it easier to convert data into visualizations. For example, Data-Driven Documents (D^3) (Bostock et al., 2011) is a unique approach to creating visualizations for the web that focuses on transparency and direct manipulation of the underlying data. Vega-lite (Satyanarayan et al., 2016) is a high-level language for creating interactive graphics and visualizations. It is designed to be easy to use and understand, even for users without previous experience in data visualization. VizQL (Hanrahan, 2006) is a domain-specific language for data analysis and visualization. It supports an extensive range of visual expressions, allowing for easy customization and control over visualizations.

2.2. Text-to-Vis Methods

Text-to-Vis focuses on using Natural Language Processing(NLP) techniques to automatically generate visualizations from text data, this technique requires both natural language understanding for machine comprehension of natural language questions and translation algorithms for generating target visualizations using visualization language. DeepEye (Luo et al., 2018) is such a rule-based method that enables users to express their query intent using non-specific or ambiguous statements. Then the natural language input by the user is converted into an internal visualization language to generate potential visualizations. Recently, some Text-to-Vis methods based on state-of-the-art NLP techniques have been proposed. NcNet (Luo et al., 2021b) is an end-to-end solution that employs a Transformer-based model to translate natural language questions to visualization. The authors proposed a novel

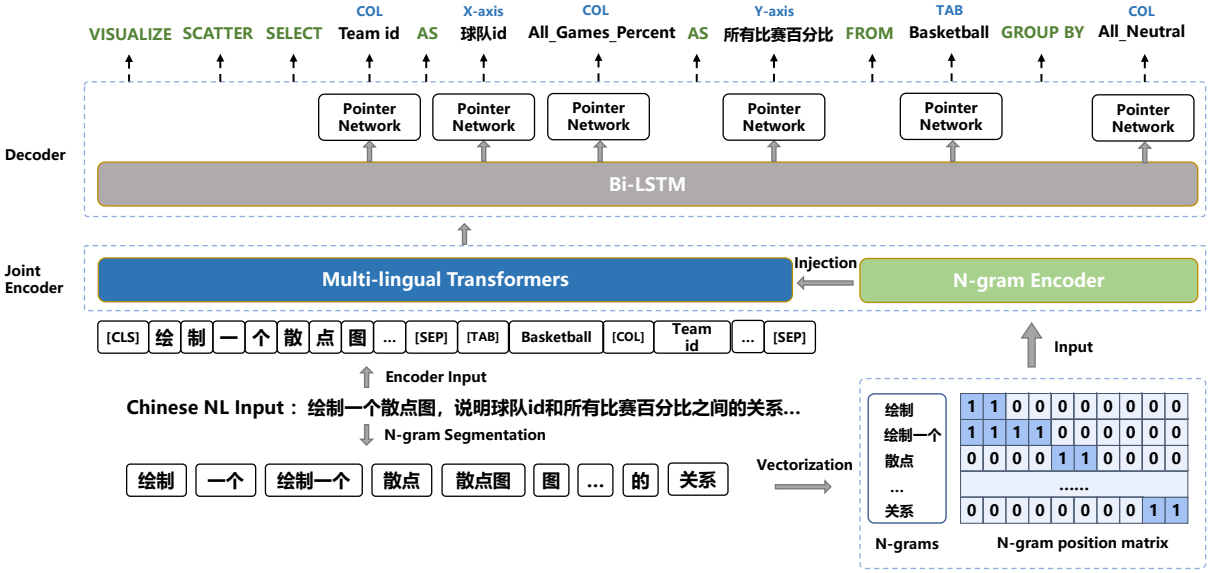

Figure 2: The overall structure of the proposed model, we use a cross-lingual pre-trained encoder to solve the language mismatch problem between natural language questions and database schema, and also integrate Chinese n-grams in the model, making it better able to encode Chinese semantics.

and concise visualization grammar that enables Text-to-Vis to be performed in a machine translation way. Different from the end-to-end models, RGVisNet (Song et al., 2022) resolves the task in two phases: retrieval and revision. The authors first construct a Data Visualization (DV) codebase in advance. When a new natural language question comes, the model retrieves the codebase to find the most relevant DV query candidate as a prototype, and then based on the prototype, the model revises to generate the most appropriate query.

2.3. Text-to-Vis Datasets

The emergence of deep learning technology has greatly benefited the field of NLP. However, the biggest obstacle currently hindering the development of deep learning in Text-to-Vis technology is not the existence of corresponding NLP techniques, but the lack of massive data for training deep learning models. To alleviate this issue, Luo et al. (2021a) released a public Text-to-Vis benchmark named NvBench, which contains 25,750 NL-Vis pairs across 105 domains, making it possible to use learning-based methods to solve the Text-to-Vis problem. In addition, another recent study (Srinivasan et al., 2021) also released a curated dataset containing 893 natural language questions distributed across three datasets. However, the relatively small amount of data means that its significance is more in the field of human-computer interaction rather than constructing learning-based methods.

Although some researchers have proposed datasets in this field, to the best of our knowledge, there is currently no Chinese Text-to-Vis dataset available. This absence hinders the development of Text-to-Vis research in the Chinese context.

3. Dataset: CNvBench

In this section, we describe the construction and splitting of CNvBench, the first Chinese Text-to-Vis dataset.

3.1. Dataset Construction

We manually translated the NvBench dataset (Luo et al., 2021a) into Chinese. It should be noted that, in NvBench, both the natural language questions, the visualizations, and the databases (including table names, column names, and the stored values) are represented in English, but we only translated the questions and the x and y axis title of the visualizations into Chinese, which is shown as 3. This approach is based on the fact that professionals often construct databases using English to represent the database schema, as it adheres to programming conventions and facilitates database maintenance. In addition, the construction of this dataset aims to explore the capability of models to comprehend the semantic structure of Chinese questions and transform them into corresponding *visualization query language (VQL)* queries. This objective remains detached from the specific data languages stored within the database. The NvBench dataset includes 25,750 pairs of natural language questions and visualizations, with a total of 7,247 unique visualizations in four levels of hardness. Statistics for different types of visualization are shown as 1. We

Original: For those records from the products and each product's manufacturer, draw a bar chart about the distribution of name and code, and group by attribute founder, rank from low to high by the x axis.

Translated: 对于产品和每个产品的制造商的记录，绘制一个关于名称和代码分布的条形图，按创始人属性分组，并按x轴从低到高排列。

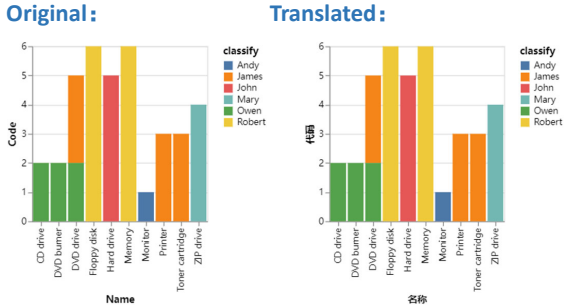

Figure 3: A comparison of the original data of NvBench and our translated data, We manually translate English natural language queries into Chinese, and, to better suit the Chinese application context, we also translate the names of the x and y axis within the visualization to Chinese.

translated all English questions in NvBench, and we named the final Chinese dataset CNvBench. The translation work was completed by two NLP researchers and a computer science student. The questions were first translated by one annotator, then reviewed and revised by a second annotator. Finally, a third annotator compared the original and revised versions to ensure accuracy. This process was carried out for each question to ensure the highest level of accuracy and thoroughness. When translating the questions, the translators are asked to preserve the style and structure of the original sentence if a literal translation is possible. Otherwise, If the question is complex, the translators are asked to rephrase it based on the semantic meaning of the VQL query, which is an intermediate representation of natural language question and DVL, to produce a more natural Chinese translation.

3.2. Dataset Split

To properly assess the model's performance, it is important to ensure that the data used for training is not visible to the model during evaluation. As described in Text-to-SQL task (Iacob et al., 2020), we believe that there are also three aspects to be considered when splitting our dataset since both of these tasks involve retrieving data from a database.

In the question-based split, the same VQLs are allowed to appear in different sets (e.g., training, development, or test), but the precondition is that the questions corresponding to these VQLs should not be the same. In other words, the question statements should not overlap between the different sets, this ensures that the model is not biased towards a specific question during evaluation and can generalize to new, unseen results. A query-based split method makes that identical VQLs do not appear in the same subset. Finally, in a database-split method, all questions related to a particular database are required to appear in different subsets. This way of splitting aims to test how well the model performs when applied to new domains, rather than just those it has seen during training. In our experiments, we only use a question-based split to evaluate the performance of our proposed baseline model.

Vis Type	Vis	Vis & query pairs
Pie	520	1750
Bar	5523	19407
Stacked Bar	359	1172
Scatter	266	1041
Grouping Scatter	127	547
Line	380	1562
Grouping Line	72	271
total	7247	25750

Table 1: Statistics of different types of visualizations in the dataset.

4. Method

In this section, we present our baseline model in response to the aforementioned cross-lingual Text-to-Vis challenges, which is inspired and inherited from the BRIDGE model (Lin et al., 2020) due to its simple yet efficient architecture. Our model contains a BERT-based question-schema encoder for cross-lingual encoding, followed by a sequential pointer-generator to generate the corresponding VQL, which will be executed to obtain the visualization of the data. The overall structure of our model is shown in Figure 2.

4.1. Task Formulation

The Chinese Text-to-Vis task can be formally defined as follows: given a Chinese natural language question $Q = [q_1, q_2, ..., q_N]$, where q_i represents the i^{th} word in the question, and the corresponding English database schema S, the model is required to generate a corresponding VQL Y. The schema is composed of a set of tables denoted as $T = [t_1, t_2, ..., t_N]$ and a set of columns represented as $C = [c_{11}, c_{12}, ..., c_{1|t_1|}, ..., c_{N|t_N|}]$, t_i represents the i^{th} table, and c_{ij} denotes the j^{th} column within

the i^{th} table.

4.2. Injecting N-grams information for Chinese Encoding

In the Chinese Text-to-Vis task, it is possible that the WordPiece(Kenton and Toutanova, 2019) segmentation (which treats each Chinese character as a token and is unaware of the boundaries of Chinese words) could cause the encoder to overlook potential database schema mentioned in Chinese questions, preventing the model from establishing connections between them and leading to the generation of incorrect table or column names during the decoding phase.

To address or alleviate this issue, we adopted a method akin to the ZEN model (Diao et al., 2020). We extracted n-grams from the Chinese question and utilized an external encoder to encode these n-grams. Subsequently, we injected the representations of the n-grams into the original cross-lingual question-schema encoder. Specifically, for encoding the input n-grams, we employed a multi-layer Transformer(Vaswani et al., 2017a) as the n-gram encoder. The n-gram encoder processed the embedding vectors of the n-grams to produce their representations. These representations of each character and its associated n-grams were then combined to form an enhanced representation, which was further passed to the subsequent layer of the original encoder. This process was iterated layer-by-layer in conjunction with the original encoder.

We adopted a BERT-style input structure for structuring natural language questions and their corresponding schema. To represent each table name and its associated column names, we utilized special tokens, denoted as "$[T]$" and "$[C]$" respectively.

$$\begin{aligned} X = &[CLS], Q, [SEP], [T], t_1, [C], c_{11}, \ldots, c_{1|t_1|}, \\ &[T], t_i, [C], c_{i1}, \ldots, c_{i|t_i|}, [SEP]. \end{aligned}$$

We input X, along with the n-gram and n-gram position matrix corresponding to the natural language question Q, into both the multi-lingual Transformer and the n-gram encoder separately. With this encoding approach, we can establish a mapping between the Chinese natural language question and the English schema, while also enhancing the representation of the Chinese text by leveraging n-grams. For more details, please refer to ZEN (Diao et al., 2020).

4.3. LSTM-based Pointer-Generator Decoder

To generate the final VQL statements, we use an attention-based (Vaswani et al., 2017b) LSTM with pointer-generator(See et al., 2017) as the decoder. During the generation phase, the decoder has the ability to selectively incorporate specific parts of the input sequence into the output by "pointing" to them. The decoder is initialized using the hidden state from the encoder. Then at each time step, the decoder has two options: one is generating a VQL keyword from the vocabulary V; the other is using the pointer network to copy a component from the schema S or to copy a token from the natural language question Q. These options allow the decoder to create a VQL query while also incorporating relevant information from the schema.

To generate the VQL, at each decoding step t, the decoder calculates the multi-head attention as described in Vaswani et al. (2017b):

$$e_{ti}^{(h)} = \frac{s_t W_U^{(h)} (h_i W_V^{(h)})^\top}{\sqrt{n/H}} \tag{1}$$

$$\alpha_{ti}^{(h)} = \underset{i}{softmax}\left\{e_{ti}^{(h)}\right\} \tag{2}$$

$$z_t^{(h)} = \sum_{i=1}^{L} \alpha_{ti}^{(h)} (\boldsymbol{h_i} W_V^{(h)}) \tag{3}$$

$$z_t = \left[z_t^{(1)}; z_t^{(2)}; ...; z_t^{(H)}\right] \tag{4}$$

where h represents the head number and H represents the total number of attention heads. L is the sum of the number of tokens in question Q and the number of components in Schema S. $\boldsymbol{h_i}$ stands for the i^{th} vector of the encoder representation $\boldsymbol{h} \in \mathbb{R}^{L \times n}$.

To decide whether to generate from the vocabulary or copy the token, the model defines the probabilities in the following form:

$$p_{gen}^t = sigmoid\left(s_t W_{gen}^s + z_t W_{gen}^z + b_{gen}\right) \tag{5}$$

$$p_{copy}^t = 1 - p_{gen}^t \tag{6}$$

where p_{gen}^t represents the probability of generating from the vocabulary, while p_{copy}^t represents the probability of copying from Q or S.

5. Experiments

5.1. Experimental Setup

We conducted several quantitative experiments to evaluate our method. In addition to the approach proposed in this paper, we also conducted experiments under a variety of settings, mainly focusing on the impact of the performance on different encoding methods in this cross-lingual task.

In the experiment, we test the model presented in Section 4, denoted by **CT2V**$_{MN}$ (Cross-lingual

	Easy	Medium	Hard	Extra Hard	All
LSTM	0.511	0.497	0.471	0.455	0.463
CT2V$_M$	0.854	0.782	0.754	0.710	0.791
CT2V$_{MN}$	0.863	0.795	0.761	0.753	0.798

Table 2: The overall Vis tree matching accuracy of three different encoding methods on the CNvBench.

Text-to-Vis), which integrates both the multilingual BERT and our proposed n-gram injection method in the encoder. We also test its variant **CT2V**$_M$ which only utilizes multilingual BERT but does not use the n-gram encoder.

To assess the effectiveness of our proposed joint-encoder method, we also test our model with an LSTM as the encoder instead. It adopts the Tencent multilingual embeddings[2] as the pre-trained word embedding. We use two different word segmentation tools, Jieba[3]and HanNLP[4] to investigate the effect of Chinese word segmentation methods on the final results. Correspondingly, the models utilizing Jieba and HanNLP for word segmentation are named **LSTM**$_J$ and **LSTM**$_H$, respectively.

5.2. Evaluation Metrics

We evaluate our proposed model with the metrics described in Luo et al. 2021a.

We employ the **tree matching accuracy** to assess the overall results. This evaluation method necessitates the transformation of VQL into an Abstract Syntax Tree (AST) and comparing it with the ground truth. The calculation method for tree accuracy is denoted as $Acc_{tree} = N_{tree}/N$, where N_{tree} denotes the number of generated VQL ASTs identical to the ground truth, and N represents the total number of VQL ASTs in the test data.

In addition, we also utilize the **vis component matching accuracy** to offer a more comprehensive evaluation of the model's performance regarding each specific component of the visualization. This metric allows for a fine-grained analysis of the model's capabilities. This metric breaks down as follows: evaluating visualization types involves the "Visualize" part of the generated VQL query; assessing the x/y/z-axis component relates to the "Select" part of the query; and analyzing data includes aspects such as "Group," "Filter," "Order," and "Superlative" components. The metric is defined as $Acc_{com} = N_{com}/N$, where N_{com} denotes the number of components correctly matched with ground truth N.

[2] https://ai.tencent.com/ailab/nlp/en/embedding.html

[3] https://github.com/fxsjy/jieba

[4] https://github.com/hankcs/HanLP

5.3. Overall Results

Table 2 shows the overall VQL tree matching accuracy of our baseline model in different hardness levels.

Our proposed model that combines Chinese n-grams performed the best and achieved 79.8% VQL tree matching accuracy overall. It also performed the best on different hardness levels. On the other hand, the model employing the LSTM as the encoder only achieved an accuracy of 46.3%, which reflects the advantage of using current popular pre-trained language models as the encoder. Compared to LSTM, pre-trained language models are experts in modeling the context within the question and the relationship between the question and schema.

	Top1	Top3	Top5	All
LSTM$_J$	0.463	0.489	0.494	0.562
LSTM$_H$	0.452	0.503	0.529	0.553
CT2V$_M$	0.791	0.822	0.864	0.907
CT2V$_{MN}$	0.798	0.829	0.857	0.891

Table 3: Results on CNvBench with different model settings.

Table 3 summarizes the model performance in different settings. Notably, the **CT2V**$_{MN}$ method stands out as the most effective in capturing the semantic relationships between text and visualization. Its implementation yields the best performance with a Top1 accuracy (we use a beam search when decoding) of 0.798. Additionally, the performance of **CT2V**$_{MN}$ surpasses **CT2V**$_M$ at Top-1 and Top-3 accuracies. This observation signifies that the N-gram injection approach enables a more comprehensive understanding of the text's underlying semantics by taking into account not only individual Chinese characters but also the contextual relationships between consecutive sequences of characters.

Furthermore, considering that Chinese sentences require segmentation prior to LSTM processing, we examined how the choice of segmentation tool for the questions impacts the performance. Our findings reveal that different Chinese word segmentation methods can influence the model's outcomes due to the potential accumulation of errors. However, employing a pre-trained model as the encoder

	Vis							Axis	Data				
	Bar	**Pie**	**Line**	**Scatter**	**SB**	**GL**	**GS**	**Select**	**Where**	**Join**	**Group**	**Binning**	**Order**
LSTM	0.955	0.961	0.892	0.925	0.861	0.920	0.883	0.711	0.723	0.509	0.642	0.808	0.677
CT2V$_M$	0.993	0.977	0.931	0.974	0.919	0.926	0.981	0.831	0.912	0.904	0.877	0.912	0.878
CT2V$_{MN}$	0.995	0.969	0.982	0.948	0.932	0.948	0.959	0.851	0.879	0.930	0.862	0.928	0.919

Table 4: Vis component matching accuracy on CNvBench. SB, GL, and GS stand for Stacked Bar, Grouping Line, and Grouping Scatter, respectively.

mitigates this issue.

5.4. Results on Different Parts of the Visualization Component

Table 4 reports the vis component matching accuracy on different encoders. Overall, the n-gram based encoder performs well in each visualization component prediction task. When predicting the visualization types, all three models obtained good performance, especially on the "Bar" charts. However, for stacked bar(SB) predictions, performance drops across all models due to stacked bars sometimes being implicitly referenced in the questions, which requires the model to infer from the sentence context. For predicting the axis part, only the LSTM encoder model obtains a poor result, possibly because there are often some corresponding aggregate functions occurring in the "Select" clause in the VQL, and the LSTM encoder is not able to well capture this type of information in the question. For data parts, both models utilizing pre-trained encoders achieved good results, with LSTM still performing the worst in this part.

5.5. Error Analysis and Future Work

To identify the causes of errors, we conducted an error analysis on our test set of 2562 VQL examples. Utilizing **CT2V**$_{MN}$, we identified several sources of errors from the 519 failed examples out of 2562, we discuss some typical cases below.

For about 36 examples, the model produces wrong predictions for the visualization part. For example, the model produced a wrong VQL for the question " 对于产品和每个产品的制造商的记录，制造商和代码之间的关系是什么，并按总部属性分组？ ", the model incorrectly predicted the visualization type as "bar" when it is actually "scatter", this is due to the lack of explicit mention of the visualization type in the question. Additionally, due to the uneven distribution of visualization types among the total train samples, the model may perform well on the majority of types but poorly on others.

For about 314 examples, the model generates wrong column names or table names in the axis part. For example, considering the question " 展示来自产品和每个产品的制造商的记录，返回一个关于价格和收入相关性的散点图，并按总部属性进行分组。 ", the model made a wrong prediction on column name "Price" as "Manufacturer", in addition to errors in predicting the column or table name, the model may also make wrong predictions on the number of column names or table names and insert extra ones into the VQL.

Errors in the data part of the VQL mean that the model makes mistakes in predicting the keywords "where", "group", "bin", and "order" of the VQL. There were a total of 169 samples with errors in this part. For the question " 使用柱状图展示每天最高温度大于或等于80度的日期有多少个, 并按照Y轴从高到低排序。", the model made the mistake of predicting "bin by weekday" as "bin by month".

After a comprehensive analysis of the errors, we believe that improving the model's ability to understand and interpret the nuances of Chinese natural language questions is crucial. This includes addressing cases where the model makes incorrect predictions due to the absence of explicit information in the question. Future work should focus on enhancing the natural language understanding component, which could involve more advanced language models, fine-tuning, and domain-specific training.

6. Conclusion

We construct the first large-scale Chinese Text-to-Vis dataset. We also present a strong baseline model and conduct extensive experiments in different configurations. We find that Chinese semantic parsing and cross-lingual question-schema linking are important factors affecting the experimental results. We hope that our dataset can play an active role in addressing Chinese Text-to-Vis with a data-driven approach.

7. Acknowledgements

Chen Jason Zhang's work acknowledges partial support from the following funding sources: ITF (PRP/009/22FX), PolyU-MinshangCT Generative AI Laboratory (Fund No: P0046453), Research Matching Grant Scheme (Fund No: P0048191, P0048183), PolyU Start-up Fund (Fund No: P0046703), Project P0043294, ITS/028/22FP, Project P0045695 by PolyU (UGC) and Project

P0047204, 25504823 by RGC. The research of Victor Junqiu WEI was supported in part by the HKUST-WeBank Joint Laboratory Project (Ref. Code: HWJL-2023.003, Project No.: WEB24EG01-A).

8. Ethical Considerations

We have considered the potential ethical issues when conducting this study. The purpose of the translation of the dataset into Chinese is to advance research and development in the field of converting Chinese text into VQL. Our primary objective is to facilitate and promote research, analysis, and innovation in this field.

The original dataset (Luo et al., 2021a) is from publicly available sources on the internet. Our data collection process is in strict compliance with all relevant laws, regulations, and ethical guidelines governing data usage, translation, and dissemination. We have ensured that we are not infringing upon any copyrights, privacy rights, or intellectual property rights during the dataset's translation process. This commitment to legal and ethical compliance underscores the legitimacy and responsible handling of the data throughout its translation and subsequent usage.

9. Bibliographical References

Micah Allen, Davide Poggiali, Kirstie Whitaker, Tom Rhys Marshall, and Rogier A Kievit. 2019. Raincloud plots: a multi-platform tool for robust data visualization. *Wellcome open research*, 4.

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

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):906–916.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. Zen: Pre-training chinese text encoder enhanced by n-gram representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4729–4740.

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):33–46.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G Karahalios. 2015. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th annual acm symposium on user interface software & technology*, pages 489–500.

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

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

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*, pages 381–395.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

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

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):136–146.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888.

Yuyu Luo, Xuedi Qin, Chengliang Chai, Nan Tang, Guoliang Li, and Wenbo Li. 2020. Steerable self-driving data visualization. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):475–490.

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)*, pages 101–112. IEEE.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021a. Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1235–1247.

Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2021b. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for chinese sql semantic parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3652–3658.

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):438–448.

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):369–379.

Xuedi Qin, Yuyu Luo, Nan Tang, and Guoliang Li. 2020. Making data visualization more efficient and effective: a survey. *The VLDB Journal*, 29(1):93–117.

Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2020. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1020–1034.

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):341–350.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

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*, pages 1646–1655.

Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M Drucker, and John Stasko. 2021. Collecting and characterizing natural language utterances for specifying data visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural*

Information Processing Systems, volume 30. Curran Associates, Inc.

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

Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu. 2021. A survey on ml4vis: Applying machinelearning advances to data visualization. *IEEE Transactions on Visualization and Computer Graphics*.

Michael L Waskom. 2021. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.