



Intent-Aware Data Visualization Recommendation

Atsuki Maruta¹ · Makoto P. Kato^{1,2}

Received: 7 March 2022 / Revised: 28 June 2022 / Accepted: 1 August 2022 / Published online: 14 September 2022
© The Author(s) 2022

Abstract

This paper proposes a visualization recommender system for tabular data given visualization intents (e.g., “population trends in Italy” and “smartphone market share”). The proposed method predicts the most suitable *visualization type* (e.g., line, pie, or bar chart) and *visualized columns* (columns used for visualization) based on statistical features extracted from the tabular data as well as semantic features derived from the visualization intent. To predict the appropriate visualization type, we propose a bi-directional attention (BiDA) model that identifies important table columns using the visualization intent and important parts of the intent using the table headers. To determine the visualized columns, we employ a pre-trained neural language model to encode both visualization intents and table columns and predict which columns are the most likely to be used for visualization. Since there was no available dataset for this task, we created a new dataset consisting of over 100 K tables and their appropriate visualization. Experiments revealed that our proposed methods accurately predicted suitable visualization types and visualized columns.

Keywords Data visualization · Tabular data · Bi-directional attention · BERT

1 Introduction

This paper is an extension of our earlier report entitled “Intent-aware Visualization Recommendation for Tabular Data” presented at. Visualization is an effective means of gaining insight into and demonstrating trends in statistical data. Furthermore, properly visualizing data can help find specific data from large-scale data collections. However, because effective visualization requires special skills, knowledge, and deep data analysis, it is sometimes difficult for end users to produce appropriate data visualization for their purposes. Therefore, *visualization recommendation*, which predicts the appropriate *visualization type* (e.g., line, pie, or bar chart) and *visualized columns* (columns used for visualization) for tabular data, has recently attracted increased attention [1–5]. Visualization recommendation is often achieved by a machine learning (ML) approach using features extracted from tabular data, such as the means and

variances of column values [2], to predict the visualization types and visualized columns.

However, existing studies on visualization recommender system have limitations. First, existing studies assume that only tabular data are given as input for visualization recommendation; however, appropriate visualization types are dependent on users’ *visualization intents*—specific characteristics or content of the data that users wish to represent. For example, selecting the appropriate visualization type for smartphone sales data is challenging; namely, a pie chart should be selected if the intent is “to illustrate the market share”, whereas a line chart should be selected if the intent is “to illustrate the market growth”. Therefore, it would be helpful to also input the visualization intent of users into the recommender system. The second limitation of existing studies is that they assume that the columns used for data visualization are input to the visualization recommender system. However, given tabular data and the users’ visualization intent, the system would ideally automatically predict the appropriate visualization type and visualized columns without requiring additional effort from users.

In this paper, addressing the limitations discussed above, we focus on identifying the appropriate visualization type and visualized columns for tabular data given a visualization intent. We propose a novel method based on a *bi-directional*

✉ Atsuki Maruta
s2121653@s.tsukuba.ac.jp

Makoto P. Kato
mpkato@acm.org

¹ University of Tsukuba, Tsukuba, Japan

² JST PRESTO, Kawaguchi, Japan

attention (BiDA) mechanism for predicting the visualization type. The BiDA mechanism can automatically identify the table columns required for visualization, based on the given visualization intent. Furthermore, as it is bi-directional, our proposed model can attend to important terms of the visualization intent based on table headers and to important columns in the tabular data based on the visualization intent, which are particularly useful for estimating the suitable visualization type. In addition, to identify the visualized columns, we propose three models that apply a pre-trained neural language model (i.e., BERT [6]) to tabular data. We use the BERT model as it has archived high performance in a variety of natural language processing tasks. Our three proposed models differ with respect to the input format. By inputting a pair of the visualization intent and column headers to BERT, we can encode their textual information and use it with the statistical features derived from the table to effectively predict the visualized columns.

Since there is no publicly available dataset for the problem setting addressed in this paper, we created a new dataset consisting of over 100 K tables with visualization intents and appropriate visualization types by crawling publicly available visualization of tabular data from Tableau Public, a Web service that hosts user-generated data and visualization. To ensure the quality of users' visualization, we manually examined a subset of datasets and found that most of the data visualization was appropriate. We conducted experiments with the new dataset and found that our BiDA model accurately predicted suitable visualization types and outperformed baseline models and models without BiDA. Furthermore, our BERT-based models outperformed baseline models in the visualized column identification.

The contributions of this paper are as follows:

- 1 We proposed methods of identifying the appropriate visualization type and visualized columns for tabular data with a visualization intent (Sects. 2 and 3).
- 2 We created a new dataset for visualization recommendation (Sect. 4).
- 3 We demonstrated that the BiDA model and BERT-based models are effective in determining the appropriate visualization types and visualized columns, respectively (Sect. 4).

2 Related Work

This section reviews existing approaches for visualization recommendation, which can be categorized into rule-based and ML-based approaches.

Rule-based approaches [7–11] are mainly based on rules defined by experts. An example of such rule is that a pie chart is unlikely to be suitable for a table with a large

number of rows. Therefore, within the scope of the rules, it is possible to create effective visualizations that appear to be created by experts. However, rule-based approaches are not flexible and are not effective for data with a large number of columns or rows. Nonetheless, they can be accurate when the columns used for visualization are provided as input. However, these approaches cannot be applied to our problem setting, where the columns to be visualized are not explicitly given. Moreover, applying rule-based approaches requires defining new rules for visualization types that are not used in existing studies such as multi-polygon charts, which requires high cost due to consultations with experts. Accordingly, it is difficult to compare rule-based approaches with our approach.

ML-based approaches primarily extract statistical features from tabular data and train a classifier based on the extracted features [1–4]. Examples of features include the means and variances of column values, and a binary feature indicating whether column values are categorical or numeric. Dibia and Demiralp proposed an encoder–decoder model that translates data specifications into visualization specifications in a declarative language [1]. In a different study, Luo et al. addressed the visualization recommendation problem by using both rule-based and ML-based (learning-to-rank) approaches [3]. Liu et al. also proposed a method for predicting visualization types and visualized columns in the context of a table QA task, in which a table and *question* are given as input [5]. This method predicts the columns to be used for visualization by identifying the correspondence between the header of the tabular data and question about the tabular data. However, the table QA task requires a specific question for tabular data, which is substantially different from a visualization intent we discussed in the present study. There have been studies on automatic data visualization from natural language [12–14], among which ncNet [14] took the most similar approach to ours. ncNet [14] is a machine translation model that takes a query for visualization as input and outputs a sentence that contains elements of visualization (e.g., visualization type and visualized columns). However, the nvBench dataset, which was used in their work and consisted of query and visualization pairs, was generated from an existing dataset of natural language-to-SQL (NL2SQL) query and SQL query pairs. As the queries were generated from SQL based on predefined rules, the vocabulary in the queries are highly limited. In contrast, our dataset was developed based on texts generated from real users and contained a variety of expressions in the input (or visualization intents).

Table 1 presents differences between existing ML-based models and our model. These ML-based approaches have different input and output from our proposed model. Therefore, it is not appropriate to train these models on our dataset and to apply the dataset used in them to the our proposed

Table 1 Differences between existing models and our model

Model	Input	Output
Data2Vis [1]	Table	Vega-Lite
Deepeye [3]	Table	Ranked visualizations
ncNet [13]	Table & natural language	Vega-Zero
Ours	Table & natural language	Visualization type & visualized columns

model, which makes comparison with the our proposed model difficult. A study given by Hu et al. [2] is closely related and comparable to ours: they extracted statistical features from tabular data and applied ML models to predict the appropriate visualization type.

There are several differences between our study and existing studies. Specifically, our study predicts the appropriate visualization type and visualized columns based on not only tabular data but also visualization intent. Since these inputs have different modalities, it is challenging to effectively incorporate both of them for visualization recommendation. Furthermore, our study faces the challenge of selectively using only parts of the tabular data, which is in contrast to existing studies, in which only several columns to be used for visualization are given as input for ML-based approach models with the statistical features in visualization recommendation.

3 Methodology

In this section, we describe methods for visualization recommendation for tabular data given a visualization intent. Figure 1 illustrates our proposed methods using BiDA and BERT, which consist of five components. For visualization

type prediction, (1) visualization intent embedding converts each token in the intent into a vector, while (2) tabular data embedding encodes table headers based on the word embeddings and extracts statistical features from each column. Then, (3) BiDA is applied to both embeddings to aggregate information from the visualization intent and tabular data. Finally, the (5) output layer predicts the most suitable visualization type based on the output of BiDA. Our BiDA model, inspired by a reading comprehension model [15], attends to important columns in the tabular data based on the visualization intent and to important terms in the intent based on the tabular data. Therefore, the proposed method allows us to focus only on important columns and tokens in the intent and tolerate redundant columns in a table.

For the visualized column prediction, the visualization intent and table headers are directly input into (4) BERT to estimate the relevance of columns to the given visualization intent. We propose three BERT-based models that differ with respect to the input format: Single-Column BERT, Multi-Column BERT and Pairwise-Column BERT. The output of BERT is combined with statistical features derived by (2) tabular data embedding and is then used to identify the most relevant columns. The details of each component of the proposed methods are described in the following subsections.

3.1 Visualization Intent Embedding

The visualization intent embedding component transforms T words in a given visualization intent into word embeddings. The t th word in the intent is represented by a one-hot representation: \mathbf{v}_t of size $|V|$, where V is an entire set of words, or a vocabulary. A word embedding matrix $\mathbf{E}_w \in \mathbb{R}^{d_e \times |V|}$ is used to obtain a word embedding $\mathbf{i}_t = \mathbf{E}_w \mathbf{v}_t$, where d_e is the word embedding dimension.

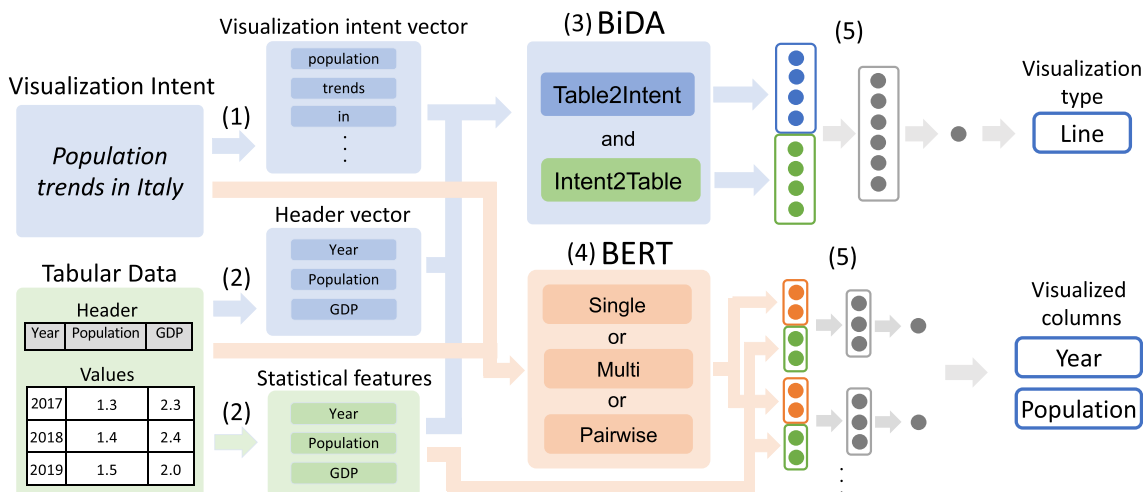
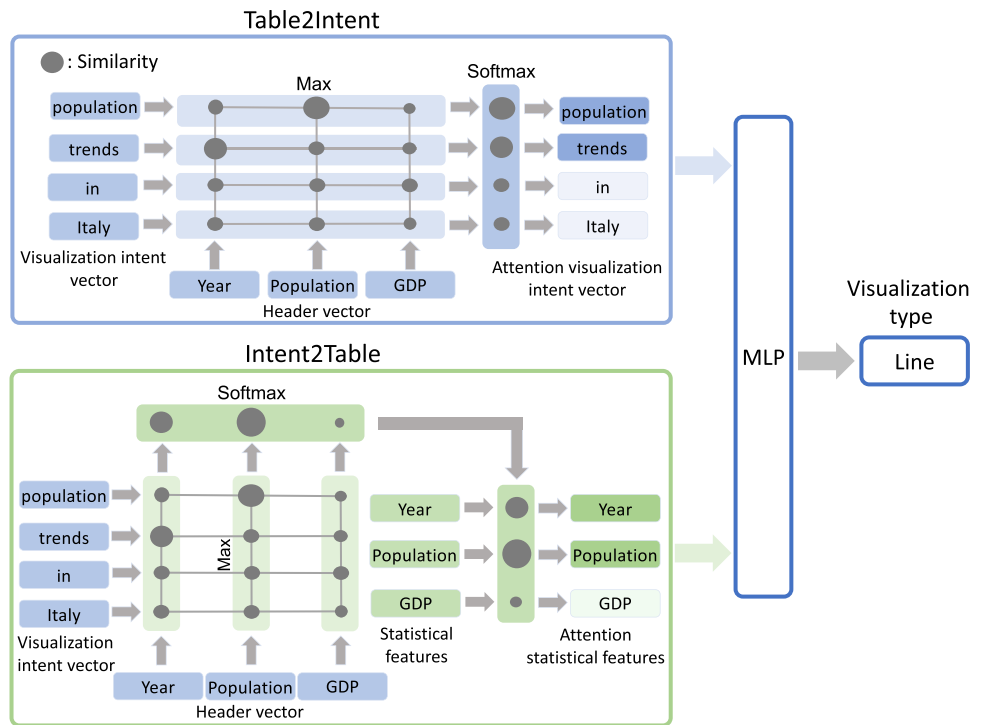


Fig. 1 Our proposed methods with BiDA and BERT-based models

Fig. 2 BiDA computes the weight of each component on one side based on information from the other side, and vice versa



3.2 Tabular Data Embedding

Given tabular data consisting of N columns, the tabular data embedding component produces a header embedding and extracts statistical features from each column. The header of the j th column consists of M_j words and is represented by the mean of their word embeddings; that is, $\mathbf{h}_j = \frac{1}{M_j} \sum_{t=1}^{M_j} \mathbf{i}_{j,t}$ where $\mathbf{i}_{j,t} \in \mathbb{R}^{d_c}$ is the word embedding of the t th term in the header of the j th column. Statistical features are extracted from each column and denoted by $\mathbf{c}_j \in \mathbb{R}^{d_c}$. We extract features following a previous study on visualization recommendation [2] ($d_c = 78$), which include the means and variances of column values and a binary feature indicating whether column values are categorical or numeric. These statistical features were used alone in the original study [2] to predict the appropriate visualization types. In our proposed methods, these statistical features are combined with textual features for improved performance in both the visualization type and visualized column prediction tasks.

3.3 Bi-directional Attention (BiDA)

Figure 2 presents our BiDA model, which consists of Table2Intent and Intent2Table for predicting the visualization types. BiDA computes the weight of each component on one side based on information from the other side, and vice versa. We compute the weight of each column based on the

visualization intent (Intent2Table), and the weight of each term in the visualization intent based on the headers of the tabular data (Table2Intent). Our model then aggregates the word embeddings of the visualization intent and the statistical features of the tabular data with the estimated weights.

The attentions from both directions are based on the similarity between a column header and a word in the visualization intent, which is defined as follows:

$$s_{ij} = \alpha(\mathbf{i}, \mathbf{h}_j) \tag{1}$$

where α is a trainable function defined as $\alpha(\mathbf{i}, \mathbf{h}) = \mathbf{w}^\top [\mathbf{i}; \mathbf{h}; \mathbf{i} \circ \mathbf{h}]$, \mathbf{w} denotes a trainable weight vector, $[\cdot]$ denotes vector concatenation across a row, and \circ denotes the Hadamard product.

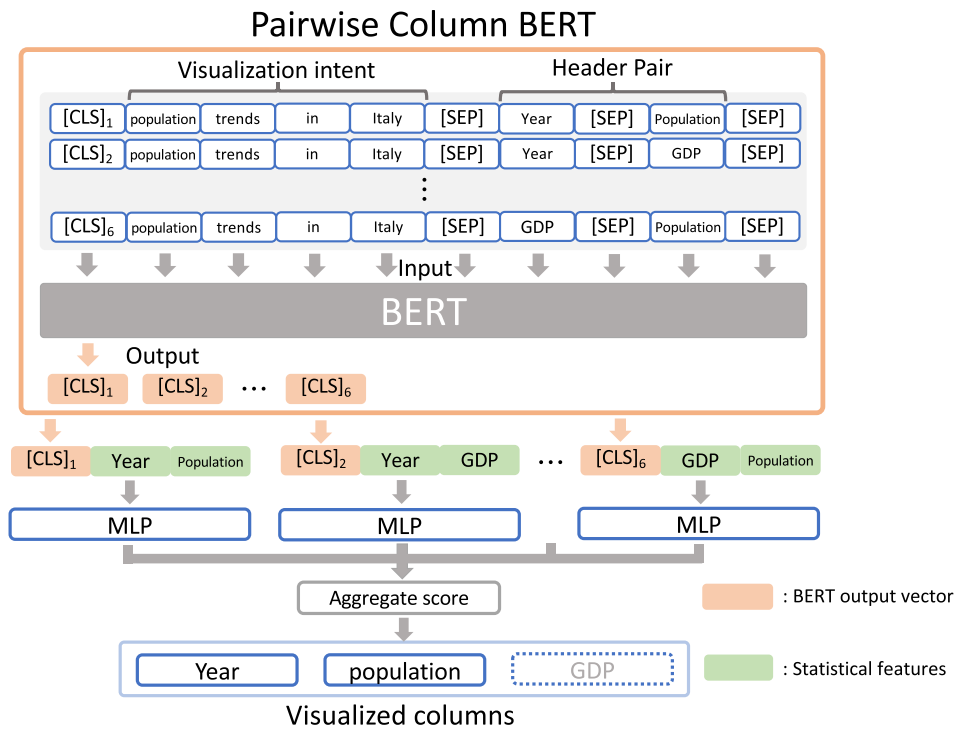
Intuitively, columns are considered important if their headers are similar to any of the visualization intent terms. This concept can be implemented as follows:

$$a_j^{(c)} = \text{softmax}(\max_t(s_{tj})) \tag{2}$$

where the maximum similarity between the j th header and the intent terms is used as the Intent2Table attention $a_j^{(c)}$. Formally, softmax function in this equation is defined as follows:

$$\text{softmax}(\max_t(s_{tj})) = \frac{\exp(\max_t(s_{tj}))}{\sum_{j'=1}^N \exp(\max_t(s_{tj'}))} \tag{3}$$

Fig. 3 Pairwise-column BERT model is inputs a visualization intent and a pair of column headers into BERT and identifies the visualized columns



The statistical features are then aggregated into a single vector with the Intent2Table attentions:

$$\mathbf{c} = \sum_{j=1}^N a_j^{(c)} \mathbf{c}_j \tag{4}$$

The word embeddings of the visualization intent are also aggregated in the same way as \mathbf{c} , namely,

$$\mathbf{i} = \sum_{t=1}^T a_t^{(i)} \mathbf{i}_t \tag{5}$$

where the Table2Intent attention $a_t^{(i)}$ is defined as $a_t^{(i)} = \text{softmax}(\max_j(s_{ij}))$. Finally, we obtain the concatenation of the visualization intent and tabular data embeddings, $\mathbf{x} = [\mathbf{i}; \mathbf{c}]$.

3.4 BERT

Figure 3 illustrates one of our proposed BERT-based models for visualized column identification. Although BERT was originally designed for text, it is used for the concatenation of the visualization intent and table headers in our model. As mentioned in the beginning of Sect. 3, three BERT models are proposed: Single-Column BERT, Multi-Column BERT and Pairwise-Column BERT. Single-Column BERT inputs a visualization intent and column header into BERT, while Multi-Column BERT inputs a visualization intent and all column headers into BERT. Pairwise-Column BERT inputs

a visualization intent and pair of column headers into BERT. These models then combine the output of BERT with the statistical features of the column to estimate which column should be used for visualization. We use the statistical features of columns because they have been reported to be useful for visualization type prediction in existing work [2]. We also use column statistical features in the prediction of visualized columns.

Letting $X = \{x_1, x_2, \dots, x_T\}$ be a visualization intent consisting of T tokens, and $Y = \{y_{j1}, y_{j2}, \dots, y_{jM_j}\}$ be the j th column header containing M_j header tokens, we formally describe each of the BERT-based models below.

3.4.1 Single-Column BERT

This model is inspired by an NL2SQL model based on BERT [16]. Single-Column BERT inputs a visualization intent and column header into BERT as follows:

$$[CLS]x_1x_2 \dots x_T[SEP]y_{j1}y_{j2} \dots y_{jM_j}[SEP] \tag{6}$$

where $[CLS]$ and $[SEP]$ are special tokens representing the entire sequence and a separator for BERT, respectively. Letting $\mathbf{y}_{CLS,j} \in \mathbb{R}^{d_{BERT}}$ be the output of BERT for the $[CLS]$ token ($d_{BERT} = 768$ in our experiment), we can obtain a vector \mathbf{y}_j as follows:

$$\mathbf{y}_j = [\mathbf{y}_{CLS,j}; \mathbf{c}_j] \tag{7}$$

which is the concatenation of the BERT output and statistical features, and is used to predict whether the j th column is a visualized column.

3.4.2 Multi-Column BERT

Multi-Column BERT inputs the visualization intent and all column headers in the tabular data into BERT and obtains the output of BERT for each header token. Given N columns of a table, Multi-Column BERT inputs the visualization intent and all column headers into BERT as follows:

$$[\text{CLS}]x_1x_2 \cdots x_T[\text{SEP}]y_{11}y_{12} \cdots y_{1M_1}[\text{SEP}] \cdots [\text{SEP}]y_{N1}y_{N2} \cdots y_{NM_N}[\text{SEP}] \quad (8)$$

BERT outputs $\mathbf{y}_{i,j} \in \mathbb{R}^{d_{\text{BERT}}}$ for the i th token of the j th header, which is averaged over all the header tokens of the header to embed the header; that is, $\bar{\mathbf{y}}_j = \frac{1}{M_j} \sum_{i=1}^{M_j} \mathbf{y}_{i,j}$. We then obtain the j th column vector \mathbf{u}_j by inputting the averaged vector $\bar{\mathbf{y}}_j$ into a linear layer as follows:

$$\mathbf{u}_j = \mathbf{W}_M \bar{\mathbf{y}}_j + \mathbf{b}_M \quad (9)$$

where $\mathbf{W}_M \in \mathbb{R}^{d_M \times d_{\text{BERT}}}$ and $\mathbf{b}_M \in \mathbb{R}^{d_M}$ are parameters of the linear layer ($m = 30$ in our experiment). We finally obtain the vector representation of the j th column by concatenating the BERT output and statistical features as follows:

$$\mathbf{y}_j = [\mathbf{u}_j; \mathbf{c}_j] \quad (10)$$

which is used to predict whether the j th column is a visualized column.

3.4.3 Pairwise-Column BERT

This approach is inspired by a document ranking model based on BERT [17]. Figure 3 illustrates Pairwise-Column BERT model, which takes a visualization intent and a pair of column headers as input and predicts which column is more appropriate for visualization. This model inputs the visualization intent and pair of i th and j th column headers into BERT as follows:

$$[\text{CLS}]x_1x_2 \cdots x_T[\text{SEP}]y_{i1}y_{i2} \cdots y_{iM_i}[\text{SEP}] y_{j1}y_{j2} \cdots y_{jM_j}[\text{SEP}] \quad (11)$$

When training the model, one of the columns is a visualized column while the other is a non-visualized column. Unlike the other models, Pairwise Column BERT predicts which column is more appropriate based on the BERT output for the [CLS] token, which includes information about both columns, and their statistical features. Thus, we use the following vector representation for prediction:

$$\mathbf{y}_{ij} = [\mathbf{y}_{\text{CLS},ij}; \mathbf{c}_i; \mathbf{c}_j] \quad (12)$$

where $\mathbf{y}_{\text{CLS},ij} \in \mathbb{R}^{d_{\text{BERT}}}$ is the output for the [CLS] token when a pair of i th and j th column headers is input into BERT.

3.5 Output Layer

Although they have similar architectures, the output layers for visualization type prediction and visualized column identification are slightly different. Given either the output of BiDA \mathbf{x} or that of BERT \mathbf{y}_j , we apply a multilayer perceptron with rectified linear unit activation. In addition, we use a softmax function to predict the visualization types and a sigmoid function to determine whether the j th column is a visualized column.

For Pairwise Column BERT, given a pair of columns, a sigmoid function is used to predict which column is a visualized column. When predicting visualized columns, we first apply the model to every pair of columns and obtain a probability $p_{i,j}$ that the i th column is more appropriate than the j th column. Following earlier work on document ranking [17], we then aggregate the probability as follows:

$$s_i = \sum_{j \neq i} (p_{i,j} + (1 - p_{j,i})) \quad (13)$$

where s_i is the score of the i th column, by which the most appropriate column is predicted.

4 Experiments

This section describes the dataset, experimental settings and experimental results.

4.1 Dataset

Our dataset consisted of quartets of tabular data, a visualization intent, the appropriate visualization type, and a set of visualized columns. To create this dataset, we crawled publicly available visualizations of tabular data from Tableau Public,¹ which is a Web service that hosts user-generated data and visualization. We used the title of each visualization as the visualization intent and the mark type of each visualization as the appropriate visualization type. The columns used in the charts were regarded as the visualized columns. Visualizations with three words or less in the visualization intent were excluded from the dataset, since short titles were usually insufficient to express an intent.

¹ <https://public.tableau.com/>.

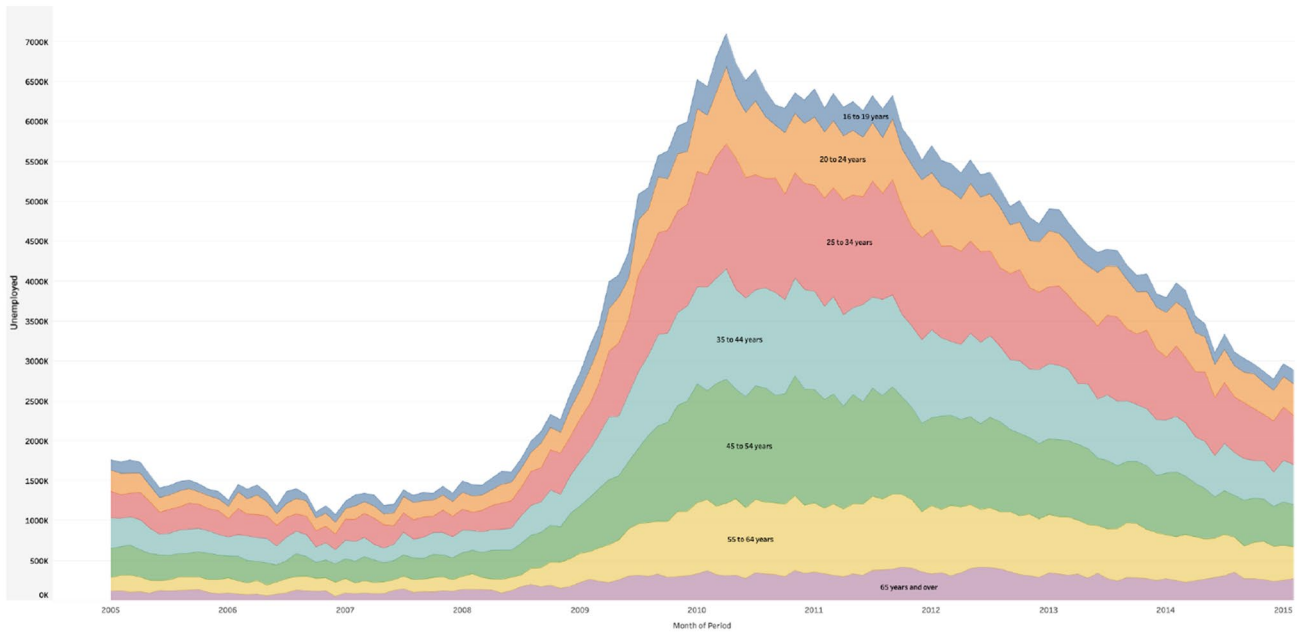


Fig. 4 Example of area chart

Only eight visualization types were used in the dataset, as other types were too infrequent. Each visualization type is described as follows:

- Area Figure 4 presents an example of Area chart.² This type of chart usually represents the evolution of multiple numeric variables, such as population growth.

- Bar This type of chart is often used to compare trends and multiple data values. Box-and-whisker plots are included in this category.

- Circle This type of chart is used to represent the distribution of data. Scatter plots and bubble charts are included in this category.

- Line Line chart is often used to illustrate the evolution of a variable, and can be used as an alternative to an area chart in many cases.

- Multi-Polygon Figure 5 presents an example of multi-polygon chart.³ This type of chart can represent

a region in a map, and is often used to indicate geographical regions.

- Pie Pie chart is often used to represent a fraction.

- Shape Figure 6 presents an example of Shape chart.⁴ Each instance is independent (not representing evolution of any kind), and is denoted by different *shapes* (e.g., circle, triangle, or any icon).

- Square Figure 7 presents an example of Square chart.⁵ This type of chart represents values by the size of the squares.

In our experiment, the mark type defined in Tableau Public was used as the visualization type. Therefore, although taxonomy of visualizations was partially different from the original visualization taxonomy, the mark types correspond to the original visualization types, and we used the mark types as visualization types. Table 2 presents the correspondence between the original visualization type and the mark type.

² https://public.tableau.com/app/profile/clive2277/viz/AreaChart_60/Area.

³ <https://public.tableau.com/app/profile/sukumar.roy.chowdhury/viz/Asia-PacificPolygonMap/AsiaPacificPolygonMap>.

⁴ <https://public.tableau.com/app/profile/naveen2129/viz/Shapesscat/terchart/Shapessscatterchart>.

⁵ https://public.tableau.com/app/profile/veita/viz/Recommendation_Square/Chart.

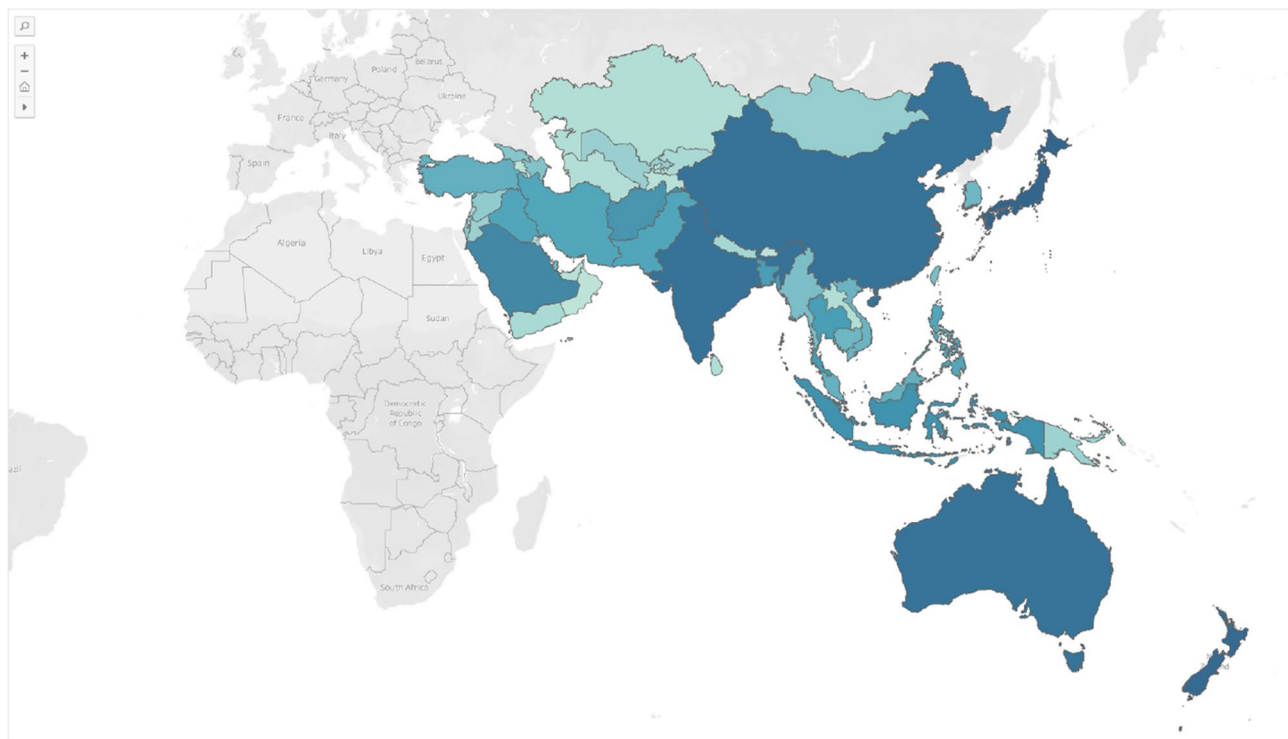


Fig. 5 Example of multi-polygon chart

Figure 8 and displays show the distributions of the length of the visualization intents, while Fig. 9 displays the number of columns. There were 7.23 words per visualization intent, and 18.10 columns and 3.76 visualized columns per table on average. After preprocessing the dataset, 115,183 data remained, which were split into training (93,297), validation (10,367), and test (11,519) sets.

Since the ground truth of the dataset (i.e., visualization types) was based on users' visualization, we investigated the quality of the ground truth by manual assessment. Two annotators, who were university graduates, were instructed to examine the visualization intent and tabular data and select appropriate visualization types independently. The annotators were first given an explanation of each visualization type and engaged in a practice session. They were then instructed to examine 215 cases. They were allowed to skip a case when they thought there were four or more appropriate visualization types. Otherwise, they were allowed to select up to three appropriate visualization types. We took the union and intersection of their answers for each case and examined whether the ground truth was in the union or intersection. As a result, we found that the union and intersection contained the ground truth

for 63.0% and 51.6% of cases, respectively. These results suggest that most of the ground truth in our dataset was reasonable.

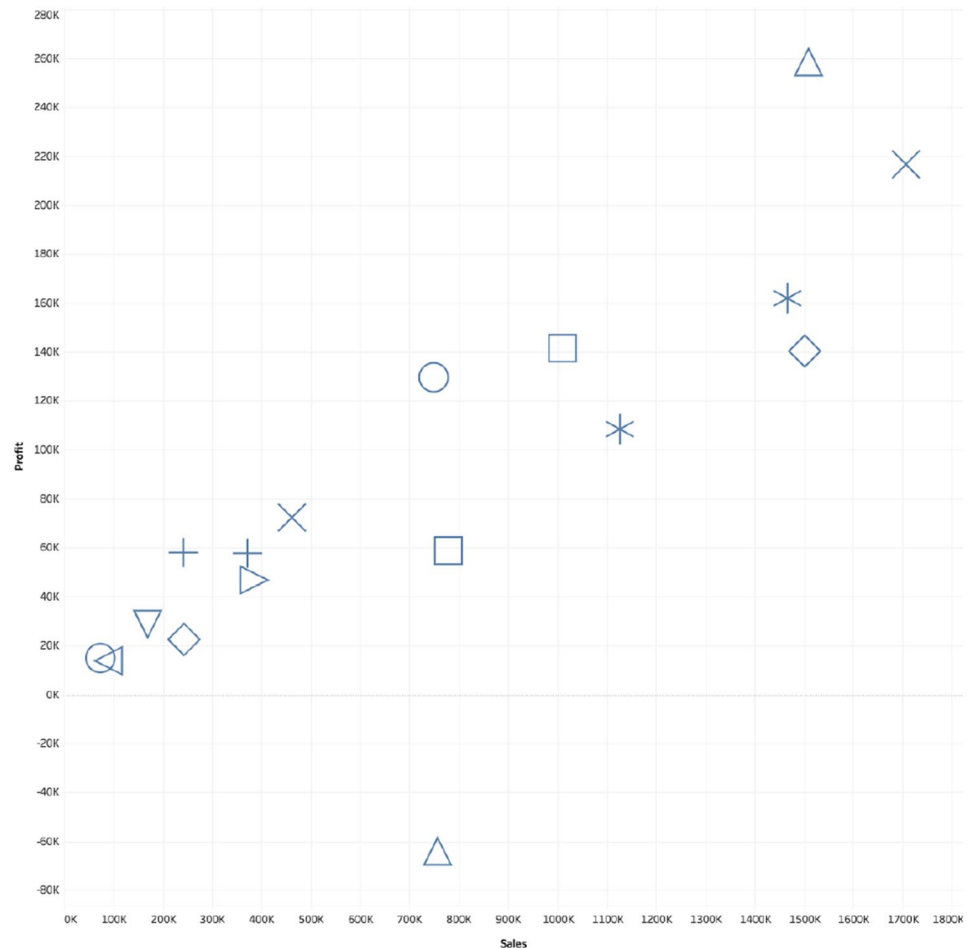
4.2 Experimental Settings

We used a pre-trained GloVe model [18] for word embeddings, which was trained by the Wikipedia 2014 dump and Gigaword 5 corpus. The size of embeddings was $d_e = 100$.

4.2.1 Visualization Type Prediction

In the BiDA model, the number of layers in the multilayer perceptron was set to six according to the results using the validation set. The model was trained with a cross-entropy loss function, and the Adam [19] optimizer was used.

We compared our proposed methods with existing ML-based approaches [2]; however, we could not compare our models with other approaches due to input and output incompatibility [1, 3, 4]. The compared ML-based approaches were based on 912 statistical features extracted from tabular data, and their hyperparameters were tuned using the validation set. We used Precision, Recall and F1 score as the evaluation metrics in this task.

Fig. 6 Example of shape chart

4.2.2 Visualized Column Identification

We use a pre-trained BERT model from Hugging Face⁶ and fine-tuned the three layers on the output side. The number of layers in the multilayer perceptron was set to two. The BERT model was trained using a binary cross-entropy loss function, and the Adam [19] optimizer was used.

We compared our proposed method with three baseline methods. “Random” is a weak baseline method that gives each column a random score. “Word similarity”, which calculates the cosine similarity between the mean vectors of words in a visualization intent and that of words in a column header. “BM25” uses the BM25 score between the visualization intent and column headers to rank columns. We treated the visualized column identification task as a ranking task rather than a classification task, and used R-precision and nDCG@10 as the evaluation metrics, where the visualized

columns were regarded as relevant items of relevance grade +1.

4.3 Experimental Results

4.3.1 Visualization Type Prediction

Table 3 presents the results of baseline models and our proposed models in the visualization type prediction task. Some of the baseline models trained with dedicated features performed well; however, they were not comparable to the proposed models with both visualization intents and table features. In Table 3, “Ours (without BiDA)” refers to simplified versions of the proposed model in which \mathbf{i} and \mathbf{c} are the mean vectors of word embeddings \mathbf{i}_i and statistical features \mathbf{c}_j , respectively. When BiDA was applied to those models, there was significant performance improvement. “Intent” denotes the model without the statistical features of the table, while “Table” denotes the model without word embeddings from the visualization intent. Comparing “Intent” and “Table”, “Intent” showed a exhibited higher performance

⁶ <https://github.com/huggingface/transformers>.

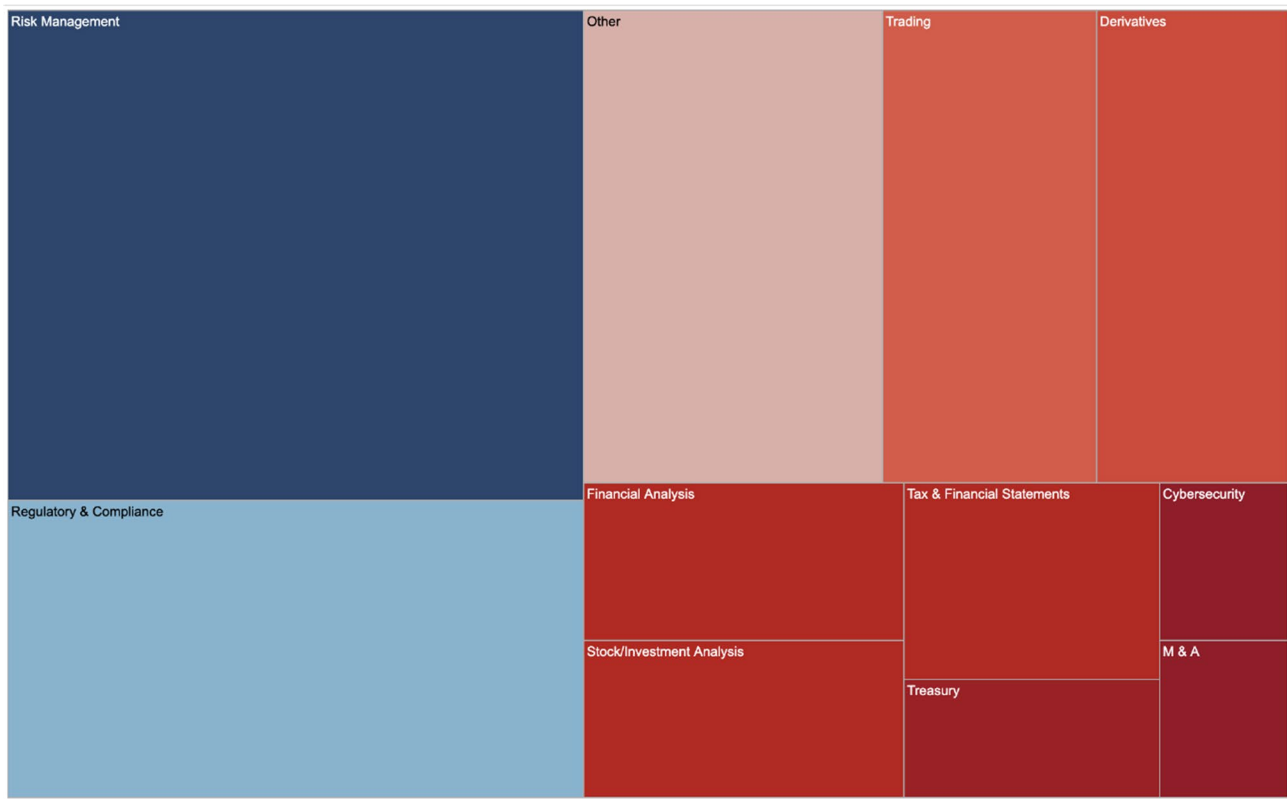


Fig. 7 Example of square chart

Table 2 Correspondence between mark type and visualization type

Mark type (this paper)	Original visualization type
Area	Area
Bar	Bar, Stacked Bar
Circle	Scatter, Bubble
Line	Line
Multi-polygon	Map
Pie	Pie
Shape	Scatter
Square	Treemap

than “Table”, and “Intent & Table” exhibited the highest performance, suggesting that both types of features were effective and visualization intents were more effective than tabular features. We conducted a randomized test with Bonferroni correction for the differences between the best model and the other models, and found that all the pairs were significant at $\alpha = 0.01$.

We further investigated the performance of the best model with BiDA. Table 4 presents the performance for each visualization type, while Table 5 provides the terms that received the most attention in the visualization intent for each visualization type. Formally, the

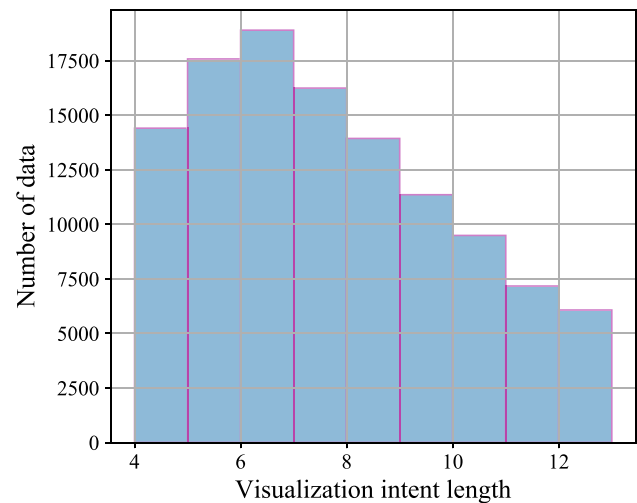


Fig. 8 Length of visualization intents

“Attention” column in Table 5 indicates the average of the attention values for each word in the test set. We found that *multi-polygon* chart exhibited the highest prediction accuracy. The terms receiving the most attention in the *multi-polygon* chart included geographical terms such as *state* and *region*. Since the *multi-polygon* chart was the

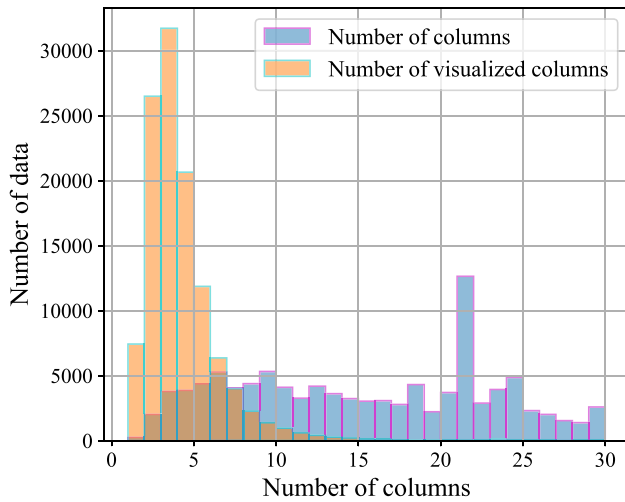


Fig. 9 Number of columns and visualized columns

only geography-related visualization type, prediction was easier than for the other visualization types. The second highest prediction accuracy was achieved by the *Pie* chart. The most attended terms in the *Pie* chart included *pie* and *donut*, which were related to the shape of the visualization. The visualization types that had high prediction accuracy had unique shapes and features that were not found in other visualizations. In contrast, area charts and line charts exhibited lower prediction accuracy, and tended to have low attention to visualization intents. This may be because there were no particularly effective words for predicting these types of visualizations.

Figure 10 illustrates the similarity between a column header and a word in a visualization intent, defined in Eq. 1. The correct visualization type for this example is “line” chart, which was successfully predicted by our model. The user-generated visualization was a line chart comparing the sales of paper and computing products. The figure demonstrates that the depth of each word in the visualization intent changed significantly and that the similarity was not affected by the header, but was greatly affected by the visualization intent. It can be seen that the term *trend* received high attention in the visualization intent. Furthermore, our model also attended to visualized columns, such as *Order Priority* and *Order Date*.

4.3.2 Visualized Column Identification

Table 6 presents the performance of the baseline and proposed models in the task of visualized column prediction. “Table” is a simplified version of our proposed model and uses only the statistical features, that is, $y_j = c_j$. “Single”, “Multi” and “Pairwise” represent the model that does not use statistical features derived from the table. The results

Table 3 Performance of proposed and comparison methods in predicting the appropriate visualization type

Category	Model	Precision	Recall	F1
Baselines [2]	Naive Bayes	0.222	0.139	0.084
	K-Nearest Neighbor	0.468	0.415	0.434
	Logistic Regression	0.313	0.205	0.201
	Random Forest	0.467	0.420	0.438
Ours (without BiDA)	Intent	0.474	0.431	0.444
	Table	0.463	0.362	0.372
	Intent & Table	0.510	0.481	0.489
Ours (with BiDA)	Intent + BiDA	0.491	0.428	0.442
	Table + BiDA	0.454	0.374	0.395
	Intent & Table + BiDA	0.536	0.500	0.512

The bold values indicate which model exhibited the highest performance for each evaluation metric

demonstrate that our BERT-based models significantly outperformed the baseline models, which did not use BERT. In particular, our proposed Pairwise-Column BERT model outperformed the other models, indicating the effectiveness of using this model for predicting visualized columns. In contrast, Multi-Column BERT exhibited lower performance than Pairwise-Column BERT and Single-Column BERT, likely because Multi-Column BERT inputs all column headers and cannot effectively train the correspondence between a visualization intent and a column header. In addition, Pairwise-Column BERT achieved a higher accuracy than Single-Column BERT. This finding is consistent with the higher effectiveness of the pairwise approach than that of the non-pairwise approach in document ranking tasks [17]. The prediction accuracy of Single-Column BERT and Multi-Column BERT was higher when used together with statistical features, whereas the prediction accuracy of Pairwise-Column BERT did not change greatly when statistical features were used. A Tukey honestly significant difference test revealed that the differences for all pairs were statistically significant ($p < 0.01$) except for the pair of Pairwise Column BERT and Pairwise Column BERT with statistical features.

Table 7 presents the performance of Pairwise-Column BERT for each visualization type. “Area” and “Line” charts exhibited relatively high performance. Upon examining examples of these visualization types, we found that these types of charts were likely to have a visualized column representing temporal information. This trend may help predict the correct visualized column. In contrast, “Pie” chart exhibited the lowest performance among the eight visualization types, possibly because there are a wide variety of visualized columns used for “Pie” charts.

Figure 11 presents the performance of Pairwise-Column BERT for different numbers of terms in the visualization

Table 4 Performance of Intent & Table + BiDA for each visualization type

Visualization type	# of examples	Precision	Recall	F1
Area	613	0.515	0.383	0.440
Bar	2761	0.486	0.596	0.535
Circle	2301	0.524	0.488	0.506
Line	1584	0.440	0.506	0.471
Multi-polygon	836	0.614	0.663	0.638
Pie	1365	0.565	0.570	0.568
Shape	1093	0.588	0.415	0.486
Square	966	0.558	0.381	0.453

intents. It can be seen that the prediction accuracy was low for short visualization intents and increased as the length of the visualization intent increased. An exception was observed for 12 terms: the performance significantly decreased compared to that of 11 terms. This may be due to our experimental settings, in which that visualization intents were truncated at 12 words and, as a result, important terms in a visualization intent could not be used for predicting visualized columns. Figure 12 displays the performance of Pairwise-Column BERT for different numbers of columns. When the number of columns increased, the prediction accuracy decreased. This may simply indicate that the difficulty of the task increased as the number of choices increased. The difficulty did not appear to change significantly for more than 15 columns.

Figure 13 illustrates the strength of attention from [CLS] in the output layer of the Pairwise Column BERT model. Figure 13a presents the case in which a visualized column was successfully predicted by our model; this visualization displayed nations participating in the Olympic games. The header word “attended” received high attention, and our model appeared to capture the similarity between the header word “attended” and the visualization intent word “participate”. Figure 13b presents an example where a visualized column was successfully predicted. The ground truth of this example consisted of soccer player statistics. The header word “metric” received relatively high attention, which suggests that our model was able to find a strong relationship between the word “metric” in the header and the word and “stats” in the visualization intent. Figure 13c presents a failure case, where the ground truth consisted of deer-vehicle collision statistics. The highest attention was given to term “id”, although this column was not a visualized column. A possible reason for this failure is that the model could not find appropriate correspondence between the visualization intent words and the column header words and mistakenly selected a column that is frequently used for visualization, (i.e., “id”).

Table 5 Most attended terms in the visualization intent for each visualization type. The “Attention” column is the average of the attention values for each word in the test set

Type	Term	Attention
Area	Area	0.984
	Sheet	0.878
	Chart	0.779
	Time	0.585
	Year	0.345
Bar	Bar	1.449
	Trend	1.086
	Change	1.069
	Area	0.950
	Sheet	0.880
Circle	Bubble	1.530
	Map	1.490
	Box	1.340
	Trend	1.132
	Sheet	0.870
Line	Trend	1.040
	Trends	1.001
	Line	0.996
	Sheet	0.883
	Chart	0.839
Multipolygon	Map	1.594
	State	0.888
	Sheet	0.878
	Region	0.750
	Top	0.681
Pie	Pie	1.962
	Donut	1.464
	Sheet	0.874
	Gender	0.826
	Chart	0.819
Shape	Map	1.649
	Sheet	0.880
	Top	0.698
	Time	0.568
	Year	0.419
Square	Map	1.735
	Table	1.230
	Heat	0.946
	Sheet	0.881
	Top	0.618

4.4 User Study

We conducted a user study and evaluated the proposed models using the visualization intent given by users. Four annotators were recruited to provide visualization intents. Each annotator provided 25 visualization intents, and we collected a total of

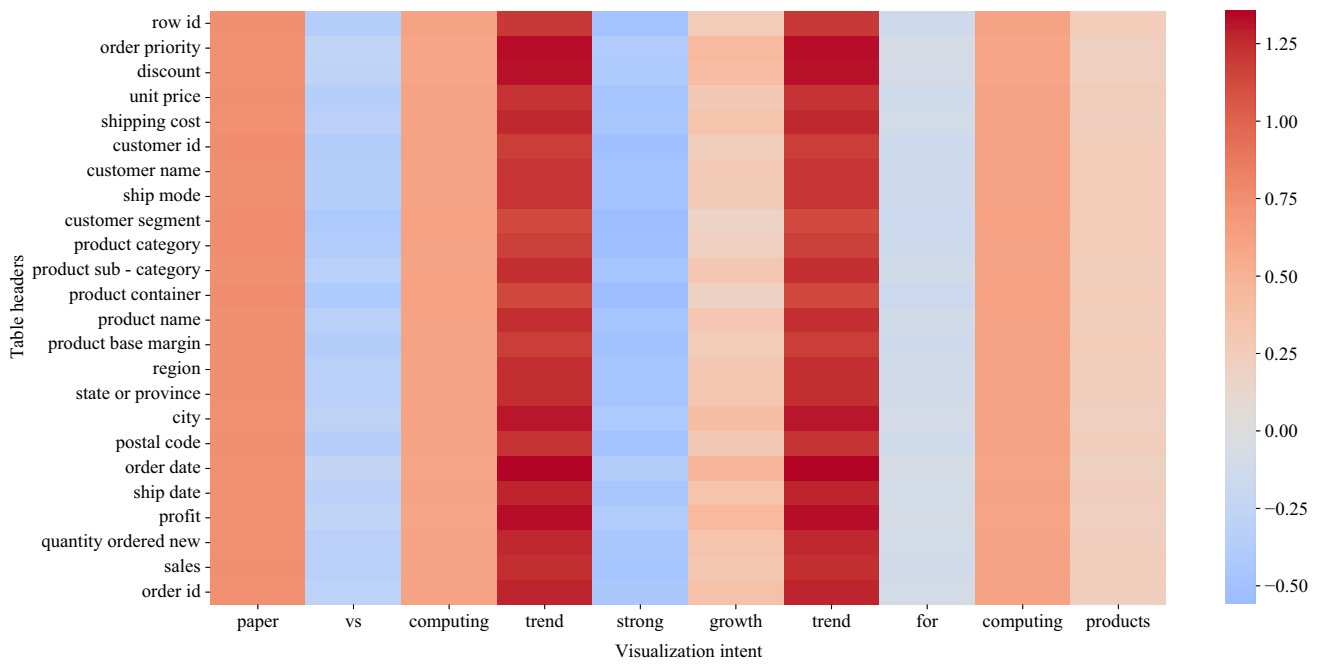


Fig. 10 Visualization of the similarity matrix. The correct visualization type for this example is “line”, which was successfully predicted by our model. The user-generated visualization is a line chart comparing the sales of paper and computing products

Table 6 Performance of proposed and comparison methods for visualized column identification

Category	Model	R-Precision	nDCG@10
Baselines	Random	0.335	0.506
	Word similarity	0.356	0.543
	BM25	0.449	0.622
Ours (without BERT)	Table	0.439	0.592
Ours (with BERT)	Single	0.592	0.719
	Single & Table	0.603	0.734
	Multi	0.531	0.663
	Multi & Table	0.544	0.678
	Pairwise	0.746	0.887
	Pairwise & Table	0.750	0.890

The bold values indicate which model exhibited the highest performance for each evaluation metric

100 visualization intents from all annotators. They were presented with the title of visualization, visualization type, tabular data, and visualized columns, and asked to describe visualization intents assuming that they were users trying to visualize the tabular data in the presented way. We then evaluated our model with the collected real users’ visualization intents, and found that results were similar to our original experiments.

Table 7 Performance of “Pairwise & Table” for each visualization type

Visualization type	# of examples	R-Precision	nDCG@10
Area	613	0.792	0.918
Bar	2761	0.754	0.895
Circle	2301	0.742	0.885
Line	1584	0.780	0.908
Multi-polygon	836	0.758	0.892
Pie	1365	0.715	0.869
Shape	1093	0.742	0.880
Square	966	0.737	0.884

Table 8 presents the results of visualization type prediction with user-generated visualization intents. Our proposed model outperformed the baseline models and was able to perform effective prediction even with the input of user-generated visualization intents.

Table 9 presents the results of visualized column prediction with user-generated visualization intents. It can be seen that our proposed BERT-based models outperformed the

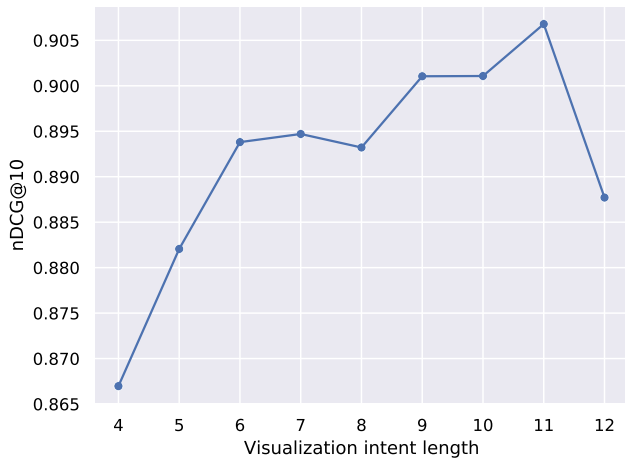


Fig. 11 Performance of “Pairwise & Table” for each length of visualization intents

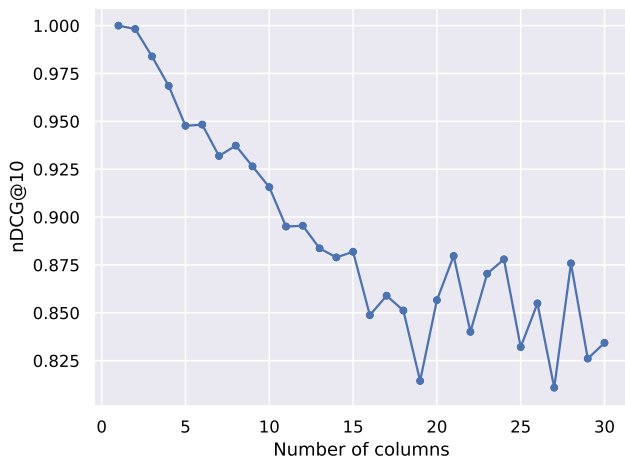


Fig. 12 Performance of “Pairwise & Table” for each number of columns

baseline models and could successfully predict visualized columns. The results thus indicate that our proposed models perform well with user-generated visualization intents.

Table 8 Results of visualization type prediction with user-generated visualization intents

Model	Precision	Recall	F1
Naive Bayes	0.032	0.118	0.039
Logistic Regression	0.200	0.180	0.169
K-Nearest Neighbor	0.363	0.257	0.277
Random Forest	0.329	0.251	0.273
Intent & Table + BiDA	0.382	0.385	0.379

The bold values indicate which model exhibited the highest performance for each evaluation metric

Table 9 Results of visualized column prediction with user-generated visualization intents

Model	R-score	nDCG@10
Random	0.323	0.540
Word Similarity	0.357	0.570
BM25	0.465	0.650
Single BERT + Table	0.619	0.736
Multi BERT + Table	0.515	0.651
Pairwise BERT + Table	0.760	0.902

The bold values indicate which model exhibited the highest performance for each evaluation metric

5 Conclusions

In this paper, we proposed a visualization recommender system for tabular data given a visualization intent. The proposed method predicts the most suitable visualization type and visualized columns based on statistical features extracted from the tabular data, as well as semantic features derived from the visualization intent. To predict the appropriate visualization type, we proposed a BiDA model that identifies important table columns using the visualization intent, and important parts of the intent using the table headers. To identify visualized columns,

[CLS] nations number of nations which have participated in each summer olympic games [SEP] number of nations attended [SEP] host nation [SEP]

(a) The correct visualized column was successfully predicted by our model. This visualization shows nations which participate Olympic.

[CLS] arsenal player stats back app ##area ##rance ##s graph [SEP] metric [SEP] rank [SEP]

(b) This example was successfully predicted by our model. This visualization shows soccer player statistics.

[CLS] deer collision by location and month [SEP] id [SEP] n ##f [SEP]

(c) This example was failed to predict visualized columns. This visualization shows about deer collision.

Fig. 13 Visualization of the BERT attention

we employed BERT to encode both visualization intents and table columns, and estimate which columns are the most likely to be used for visualization. Since there was no available dataset for this task, we created a new dataset consisting of over 100 K tables and their appropriate visualizations. The experiments revealed that our proposed methods accurately estimated suitable visualization types and visualized columns. Future work will include prediction of more detailed settings for data visualization such as layouts and styles.

Author Contributions Atsuki Maruta conceived the study, collected data, implemented the experiments, analyzed the results, and wrote the paper under the supervision of Makoto P. Kato. Makoto P. Kato provided overall research guidance to Atsuki Maruta and made contributions, especially in the conception of the idea and editing of the paper. Both authors contributed significantly to this paper.

Funding This work was supported by Japan Society for the Promotion of Science KAKENHI Grant Numbers JP18H03243 and JP18H03244 and Japan Science and Technology Agency Precursory Research for Embryonic Science and Technology Grant Number JPMJPR1853, Japan.

Data Availability All data used in this paper were collected from a public Web service.

Declarations

Ethics Approval Not applicable.

Consent for Publication Not applicable.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Dibia V, Demiralp Ç (2019) Data2vis: automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Comput Graph Appl* 39(5):33–46
- Hu K, Bakker MA, Li S, Kraska T, Hidalgo C (2019) Vizml: a machine learning approach to visualization recommendation. In: *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–12
- Luo Y, Qin X, Tang N, Li G (2018) Deepeye: towards automatic data visualization. In: *2018 IEEE 34th international conference on data engineering (ICDE)*, pp 101–112
- Moritz D, Wang C, Nelson GL, Lin H, Smith AM, Howe B, Heer J (2018) Formalizing visualization design knowledge as constraints: actionable and extensible models in draco. *IEEE Trans Vis Comput Graph* 25(1):438–448
- Liu C, Han Y, Jiang R, Yuan X (2021) Advisor: automatic visualization answer for natural-language question on tabular data. In: *2021 IEEE 14th Pacific visualization symposium (PacificVis)*. IEEE, pp 11–20
- Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*
- Wongsuphasawat K, Moritz D, Anand A, Mackinlay J, Howe B, Heer J (2015) Voyager: exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans Vis Comput Graph* 22(1):649–658
- Mackinlay J (1986) Automating the design of graphical presentations of relational information. *ACM Trans Graph* 5(2):110–141
- Roth SF, Kolojejchick J, Mattis J, Goldstein J (1994) Interactive graphic design using automatic presentation knowledge. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, pp 112–117
- Stolte C, Tang D, Hanrahan P (2002) Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans Vis Comput Graph* 8(1):52–65
- Mackinlay J, Hanrahan P, Stolte C (2007) Show me: automatic presentation for visual analysis. *IEEE Trans Vis Comput Graph* 13(6):1137–1144
- Srinivasan A, Stasko J (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*, pp 55–59
- Shen L, Shen E, Luo Y, Yang X, Hu X, Zhang X, Tai Z, Wang J (2021) Towards natural language interfaces for data visualization: a survey. *arXiv preprint arXiv:2109.03506*
- Luo Y, Tang N, Li G, Tang J, Chai C, Qin X (2021) Natural language to visualization by neural machine translation. *IEEE Trans Vis Comput Graph* 28(1):217–226
- Seo M, Kembhavi A, Farhadi A, Hajishirzi H (2016) Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*
- Hwang W, Yim J, Park S, Seo M (2019) A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*
- Nogueira R, Yang W, Cho K, Lin J (2019) Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*
- Pennington J, Socher R, Manning C.D (2014) Glove: global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp 1532–1543
- Kingma D.P, Ba J (2014) Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*