TableRepoViz: Visualizing Tabular Data Repositories for Facilitating Descriptive Tag Augmentation

Jianhao Cao

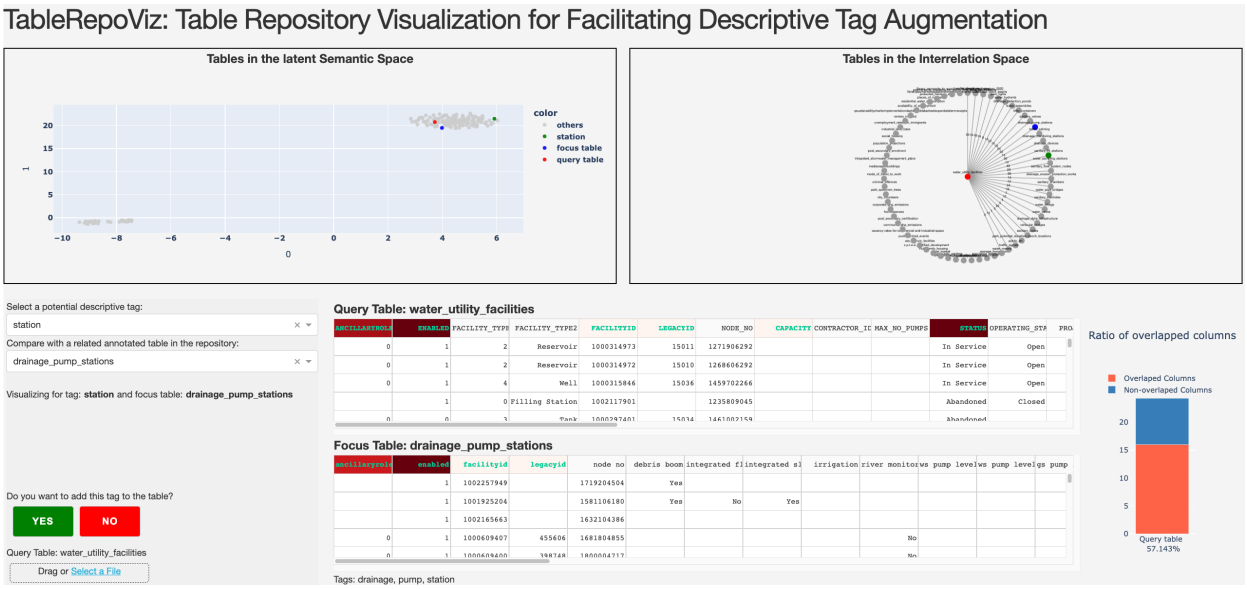

Fig. 1. The main layout and interface of TableRepoViz featuring the control panel and detailed visualization views for tables in the latent semantic space, interrelation space, and attribute space. The control panel on the bottom left allows users to upload a query table and visualize potential tags and related tables in the table repository with respect to the query table. The visualization views for table spaces are intended to help users understand how the underlying tag recommendation algorithm works and assist humans in making tag augmentation decisions for the query table.

Abstract—Many online tabular datasets are maintained in centralized repositories and annotated with descriptive tags. These tags are helpful for data practitioners to search and understand tables. However, manually annotating descriptive tags for new tables added to a large repository is expensive and may be inconsistent. I previously worked on using the table repository's existing tags to automatically recommend tags for new query tables. In this design study project, I propose TableRepoViz, an interactive visualization tool to explain how the recommendations are obtained and help a human examine whether a recommended tag is truly suitable for the new table. TableRepoViz explains different components in the recommendation algorithm with separate views and visualizes the relations between the query table and other tables in the repository to assist humans in deciding whether the recommendation is applicable to the query table. To present the functionality of TableRepoViz, I demonstrate how to use TableRepoViz to track recommendation origins and perform a contextualized comparison between a query table and already annotated tables in the repository.

Index Terms—Visualization for table repositories, tag augmentation, recommendation explainer

1 Introduction

With the current surge of machine learning and data science, practitioners in these fields have a better appreciation and increased eagerness for data. Nowadays, many organizations and agencies publish large open datasets on the Internet for public information accessibility. Online open data comes in different forms, and tabular data is a prevalent type of open data. These open tabular datasets are "database-like" web tables, constituting a popular research topic in the information retrieval community [1, 30]. They make good sources for data science tasks because of their categorical structure and richness in data content. Maintaining multiple tabular datasets in a centralized data lake, or a table repository, can provide a single access point that allows users to query data on a certain topic or search for similar datasets.

• *Jianhao Cao is with Department of Computer Science, The University of British Columbia. E-mail: jhcao@cs.ubc.ca.*

However, a major issue with accessing these open tables is that they are not always easy to interpret, especially if the table is large and has a complex schema. Ideally, a tabular dataset is annotated by the dataset creator or the table repository administrator when it is first added to a table repository. The annotator should label metadata, such as descriptive tags, for the table to provide straightforward and concise information about the table's content and characteristics for better table comprehension. These descriptive tags, usually as nouns or short noun phrases, do not only serve as subject hints for users to understand what is in the dataset but also can be used as table keywords to search and link different datasets for downstream data integration or information retrieval tasks [13, 17, 29]. But in reality, manually annotating descriptive tags is expensive and time-consuming and may not be consistent over time, even if labelled by the same annotator. In addition, the annotated metadata is not guaranteed to be comprehensive, as its quality generally depends on the annotator. The possible factors that could influence the quality of table annotation include the annotators' understanding of the data, their domain knowledge of the

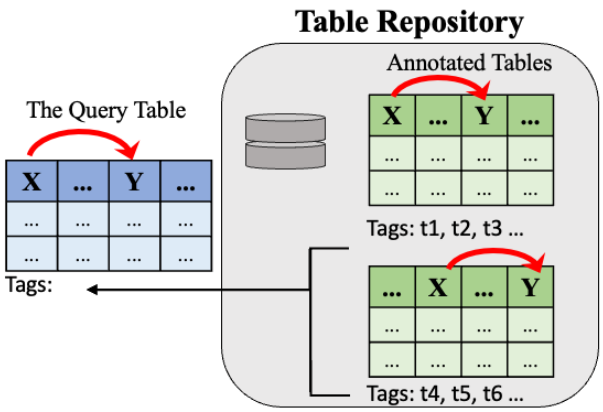

Fig. 2. Descriptive tag augmentation in a table repository. The current method leverages existing tags in the table repository to make tag recommendations. Tags of the already annotated tables that are related to the query table are recommended for the new incoming query table.

table's topic, their expressibility, and their target satisfaction level of tag completeness. It is common to come across open tabular datasets with not-well-annotated or non-existent descriptive tags.

In the case of a table repository, the existing tags of the already annotated tables could be leveraged to suggest potential tags for a new table in a consistent manner. I previously worked on a project about automatically suggesting descriptive tags for an incoming query table when it is added to a table repository. I used matching rules and a deep learning model to suggest potential tags for the query table by finding the relevant tags that are already in the table repository (see Figure 2). The overall technical details are omitted in this project report, but certain aspects will be addressed when needed to justify design choices. After a rudimentary case study, I found the potential tags cannot be simply applied to the query table because of two reasons. First, some of the tag suggestions and the query table are under a common broader topic but have some nuances, making the tags unsuitable. Second, the tagging algorithm usually yields an overwhelming number of potential tags; it could confuse users about which ones are actually important if all tag suggestions are adopted for the query table. With this in mind, the recommendation algorithm can facilitate descriptive tag augmentation by yielding potential tags for a query table. Human judgment may be the most accurate method to examine whether a recommended tag should be applied to the query table.

I propose TableRepoViz, an interactive visualization application, to visualize and facilitate the descriptive tag augmentation scenario in a table repository. The intention of developing TableRepoViz is to provide an interactive visualization interface that can respond to queries from human annotators and assist them in examining the recommended tags and comparing the related tables in the table repository with the query table. TableRepoViz visualizes a query table with the table repository in the latent semantic space, table interrelation space, and table attribute space to explain how already annotated tables are related to the query table and how the recommended tags are obtained. Ideally, an annotator should be able to determine which potential tags can be applied to the query table with TableRepoViz after browsing the views that visualize the related tables in the table repository containing the recommended tags and comparing them with the query table. TableRepoViz will help users answer the following questions in a tag augmentation task: (1) which annotated tables in the repository are the origins of the recommended potential tags? (2) why does the underlying algorithm recommend these tags for the query table? (3) Is a recommended tag truly suitable for the query table? To illustrate the functionality of TableRepoViz, I demonstrate how TableRepoViz helps to answer the three abovementioned questions with usage scenarios on tabular datasets collected from the Open Data Site of the City of Surrey [2] in British Columbia, Canada.

2 Related Work

TableRepoViz and its underlying tag recommendation algorithm relate to previous research on table annotation and table search. To ideate the visualization component of TableRepoViz, I also compare current visualization practices for tabular data to what is required in TableRepoViz and discuss the related work for interactive visualization applications.

2.1 Table Annotation

Table annotation is a well-studied topic in data management. It consists of multiple sub-tasks, such as semantic annotation and entity annotation that map table cells to entities, columns to classes, and inter-column relations to properties [1, 3, 25]. The exact solution depends on what kind of metadata is inquired in the annotation task and what other information is available along with the tabular data. Lexical matching on available table metadata can be used to search and annotate web [12], and the matching methods usually leverage a cross-domain knowledge base or ontology. Ramnandan et al. [24] use the statistical similarity of column values to assign semantic labels and integrate heterogeneous data sources. Recently there has been a new trend of using deep learning models to implicitly learn semantic representations for tabular dataset [5,27]. I have two suggestion methods in the tag-recommending system I am currently working on. The first one uses matching rules on correlated columns to find tables related to the incoming query table and suggest their existing tags. The second approach is to train a deep learning NLP model to predict the probability of existing tags for an input table. These two methods will be reflected and visualized in this proposed interactive application for assisting a human annotator in the task of selective descriptive tag augmentation.

2.2 Table Search

Table search is a problem in data integration for finding tables that are related to a query table within massive data repositories. It is helpful for augmenting the training dataset in a machine learning problem with data from related tables for interactive data science tasks [31]. The goal of table search is to find tables that are contextually related, contain similar data content, or have structural connections to the query table. Table search methods usually involve a pair-wise comparison to find tables with similar contexts. Nargesian et al. [19] propose a data-driven approach to find unionable tables in the sense that the tables share attributes from the same domain to grow a table vertically. Zhu et al. [32] apply column heading overlap similarity search to find tables that are joinable to the query table to enrich each row. As a data augmentation approach, the related tables discovered in the table search process share similar characteristics with the query table. Ideally, the related tables would provide additional information about the query table, and their descriptive tags can be adopted for the query table. In this project, I propose to visualize the relatedness between the annotated tables in the table repository and the incoming query table to show where the tag suggestions are from.

2.3 Visualizing Tabular Data

Visualizing tabular data for presentation and exploration tasks is well-researched. Depending on the visualization purposes and data types, there are various idioms for visualizing tabular data. Polaris [26] and Tableau [28] are visualization systems that offer many tabular data visualization solutions for analyzing their patterns. Furmanova et al. [7] have reviewed tabular data visualization literature with three categories of tabular data visualization techniques: (1) overview techniques, which present high-level summary and connections across attributes, (2) projection techniques, which reduce tabular data into a lower dimension, and (3) tabular techniques, which encodes a cell value while retaining its position in the table. These techniques can be combined to visualize tabular data with a coordinated multi-view setup for presenting data in different aspects for various analysis tasks.

Apart from visualizing a single table, presenting tables in a tabular dataset as a network can help users understand how a table is related to another. Not all tabular datasets have an explicit network architecture, but the connections between tables can be found if assigned edge semantics that are tailored to accommodate specific tasks. There are

previous researches about modelling and visual analyzing tabular data as networks [14, 15]. In the case of this project, finding the tables in the table repository that are related to a new query table and visualizing their connections will help users understand what tags were previously annotated for similar tables and guide the annotation process for the incoming query table. The users may still need to manually perform a pair-wise examination between the query table and a related table to compare their high-level semantics and determine whether the recommended tags from the connected table are truly deceptive of the query table. However, the current work on comparative visualization of tabular data [9, 10] emphasizes attribute value comparison rather than the high-level semantic similarity between tables. TACO [20] offers different levels of information when visualizing changes in a table over time. However, its focus is still on presenting cell value differences with either a detailed or aggregated view rather than reflecting the context in a column or the whole table. Regarding other data types, VizCommender [21] supports pair-wise comparisons between visualizations in a repository to provide content-based recommendations based on text similarity. The same strategy can be applied to tabular data by comparing the text similarity between tables with either column names or their high-level semantic representations.

2.4 Interactive Visualization Applications

Interactive visualization applications can help users understand tabular data. Galhotra and Khurana propose an automated data explanation system to identify a concept for each column in a table [11] and provide a user interface to visualize how the output results are obtained [8]. However, this application only visualizes the table explanation in an informative manner, as the authors intend to present how the explanation algorithm works under the table. In the case of an interactive application for facilitating descriptive tag augmentation, the visualization idioms and views must support analysis and search tasks in the table repository so that the users can decide which recommended tags can be applied to the query table. TimeLineCurator [6], as an interactive authoring tool example, is designed for human users to author visual timelines from unstructured text. The authoring is through an underlying data processing pipeline and a multi-view user interface; the multiple coordinated views are dedicated to different functionalities required in the timeline authoring task. In order to provide a solution to a complicated interactive problem, it is advantageous to practice a modular design strategy by separating the overall task into different sub-tasks and managing the sub-tasks in independent but interlinked components in a general user interface. For example, CorGIE [16] is an interactive explanation tool that helps users understand a graph's characteristics and node representations learned by a graph neural network (GNN). It visualizes a graph in three data spaces: the latent space for visualizing the node embedding learned by the GNN, the topology space for visualizing the connectivity of the input graph, and the feature space for visualizing attributes of the graph nodes. Each node in the graph can be visualized in all three spaces, and the node's representation in one space can correspond to its counterparts in other spaces. Similarly, if a table is considered a single data item, it is possible to visualize a new query table with the table repository in these three spaces. This design can help users understand how the tag recommendation algorithm works and determine whether the tag suggestions are truly suitable for and related to the query table.

3 Data & Task Abstraction

To provide a comprehensive review of the visualization requirements in the design of TableRepoViz, I first present the background information about the underlying tag recommendation algorithm that TableVizRepo will be used to visualize. The data and design-specific tasks are described later in this section.

3.1 Background

The intention of designing TableRepoViz is to facilitate descriptive tag augmentation for a new query table when it is being added to a table repository. I previously worked on automating the descriptive tag augmentation process. However, the tag recommendation algorithm may generate an overwhelming number of potential tags, and some tag suggestions may seem relevant to the query table but have some nuances. It requires human users to make the final judgment about which potential tags are important and accurate enough to be augmented to the query table. This course project is a visualization design study and an extended component for the aforementioned work to help human users facilitate the decision-making process in descriptive tag augmentation.

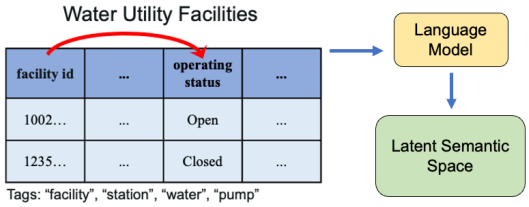

Fig. 3. A table and its explicit and implicit properties.

For an incoming query table that is being added to a table repository and requires tag annotation, the recommendation algorithm will use matching rules and a deep learning model to provide a list of potential tags for the query table from the tags that are already in the table repository. The matching rules rely on correlated column pairs to find already annotated tables that are related to the query table and then directly recommend their tags to the query table. A correlate column pair is defined by having a functional dependency [22] between the two columns in a single table (i.e., a functional dependency $\mathbf{X} \rightarrow \mathbf{Y}$ means each value in column $\mathbf{X}$ is uniquely associated with a value in column $\mathbf{Y}$). If two tables share the same functional dependency with identical column names, they are more likely to have similar semantic meanings and thus can be described with the same tags. For example, when the query table and an already annotated table in the table repository have the same correlated column pair **facility id** $\rightarrow$ **operating status**, it is reasonable to assume they both contain maintenance information, and the query table can be annotated with tags from that already annotated table if the tags are about maintenance. Another tag recommendation method uses a deep learning language model to abstract each table's semantic representation into a high dimension and then trains the model to predict each existing tag's probability for the table as a multi-label classification task. The model can be applied to any incoming query tables to recommend the existing tags in the table repository that have a high probability from the model output.

In this design study project, TableRepoViz will present visualization views that help users understand how annotation suggestions and their origins in the table repository are related to the query table. The final goal is to make TableRepoViz provide hints and help users decide which recommended tags should be applied to the new query table so that it can be used to examine the quality of tag recommendations.

3.2 Data

The tabular dataset representing the table repository in this project is from my previous work on automated descriptive tag augmentation. I retrieved the CSV files with a table header and column names from the Open Data Site of the City of Surrey [2] in British Columbia, Canada. The descriptive tags of each CSV file were also scraped from the open data site, and I eliminated the descriptive tags associated with only one table to rule out random tags after the retrieval. No other changes were made to the descriptive tags or column headings to improve the quality and readability of annotations in the dataset; the intention is to preserve and reflect the existing annotations in a table repository.

Each CSV file is in the form of a two-dimensional table, and such a table is abstracted as a single data item with explicit and implicit properties. Figure 3 shows the properties of a single table, or the attributes of a single data item. The explicit properties are the information stored in the table repository, such as its table name, column names, annotated descriptive tags, and cell values. The implicit properties are the inherent information of a table. I searched for the correlated column pairs and labelled them in each table, as they are used in the

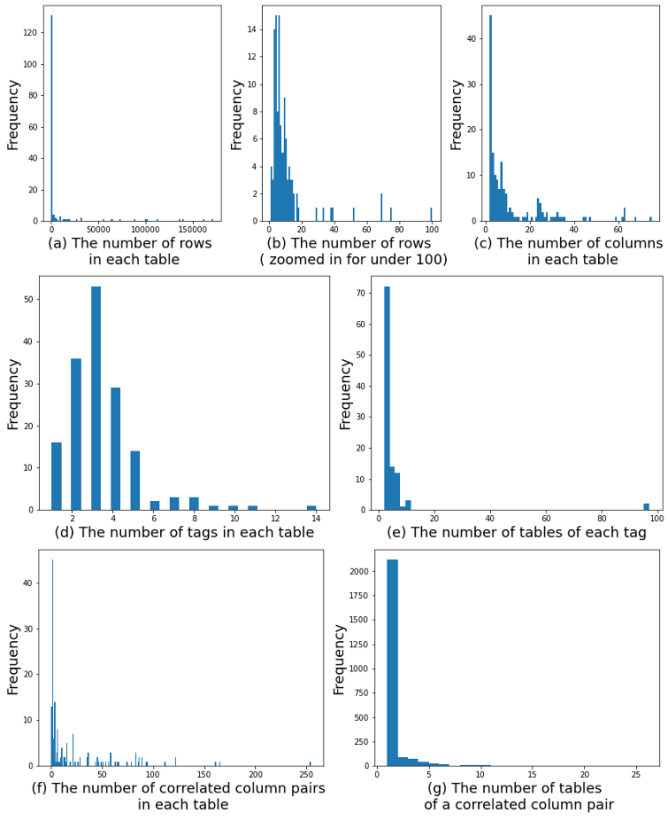

Fig. 4. Histograms of table statistics. (a), (b) and (c) show the size of every table in the dataset. (d) and (e) illustrate the number of tag and table correspondence. (f) and (e) presents how many tables and correlated column pairs correspond to each other.

underlying tag recommendation algorithm to find tables that contain the same intra-table relationships suggested by the correlated column pairs. In other words, the tables that share the same correlated column pairs are implicitly connected because they have similar intra-table relationships, and such connections can be abstracted as links between tables. Also, the deep learning language model in the tag recommendation algorithm learns the semantic representation of a table and projects it into a high-dimensional latent space.

Holistically, the 160 tables have 1,907 columns and 1,443,378 rows in total, which is about the normal size of a table repository. Figure 4 shows the histograms of various table statistics at the per-table level. Most tables in the dataset are small in size, as around two-thirds of the tables have less than 100 rows and 20 columns. There are 104 distinct tags in the dataset, but the total tag count is 469 across all the tables, as the same tag can be labelled for more than one table. Tables usually do not have many tags: over half of the tables have 5 tags or fewer, and the highest number of tags in a table is 14. On the other hand, most tags are labelled for less than 12 but at least 2 tables.

Regarding correlated columns in each table, 13 tables have no such column pairs. About 60% of the tables have less than 10 correlated column pairs, and the highest number of correlated column pairs in a single table is 252. There are 2439 pairs of correlated columns across all 160 tables. 2118 pairs appear in only one table, but they can provide recommendation hints for a new query table if the column pairs also have a functional dependency in the incoming query table. Most of the rest of the correlated column pairs appear in less than 10 tables.

3.3 Tasks

Since TableRepoViz is intended to facilitate the descriptive tag augmentation process when adding a new table into a table repository with visualizations, users will be able to input a query table and obtain a list of potential tags from the existing tags in the table repository that are recommended by the underlying algorithm. With TableRepoViz, users can explore through visualizations to understand how the recommended tags are related to the query table and inspect individual tables in detail to make the final decision on whether to annotate the query table with a recommended tag. TableRepoViz will help users to answer the three following questions:

- which annotated tables in the repository are the origins of the recommended potential tags?
- why does the underlying algorithm recommend these tags for the query table?
- Is a recommended tag truly suitable for the query table?

Show the origin table of a recommended tag. The recommended tags are from the already annotated tables in the table repository. A tag is a categorical attribute of a table that is abstracted as a single data item. The overarching annotation task is equivalent to labelling a new data item with known attributes that were previously annotated for other data items. Although the underlying recommendation algorithm automatically finds potential tags, understanding their origins and knowing which tables are previously annotated with these tags can help users have a contextualized understanding of the annotations inside the table repository. If the user is interested in a tag, TableRepoViz will help to look up the existing tables in the table repository labelled with that tag.

Explain why the algorithm recommends a tag. Simply presenting the potential tags and their origins is not convincing; it does not help users decide whether these tags are suitable for the query table and should be assigned. Visualizing how the recommendation algorithm works under the table can explain why the recommendation tags are chosen. There are two different methods used in the recommendation algorithm; a deep learning language model that learns table semantics in a high-level latent space and a matching-rule method that relies on connections between tables with the same correlated column pairs. Since these two methods work independently, TableRepoViz will visualize them separately to explain how the potential tags and their origin tables are related to the query table.

Help to analyze the validity of a recommended tag. Grasping the source of a recommended tag is only the premise of the final annotation task; users still need to decide whether a recommended tag is valid for the input query. Validity means two things: accuracy and consistency. TableRepoViz should help users not only verify if the tag is accurate in describing the query table but also analyze whether the combination of the tag and the query table is consistent with past annotations in the repository by visualizing if they share similar semantics or content.

These tasks can be abstracted in different solution spaces as the visualization paradigm in CorGIE [16], which visualizes data items in three different data spaces and supports corresponding items across different spaces. Users can specify a recommended tag and view tables with the same tag in the repository together with the query table as data items in the three data spaces in Figure 5.

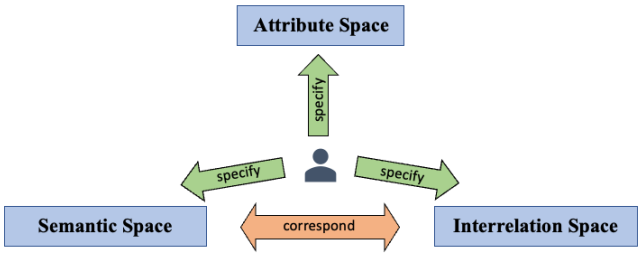

Fig. 5. The three data spaces for task abstraction.

The deep learning language model in the recommendation algorithm extracts table semantics for predicting the probability of a tag when given an input query table. The internal semantic representation of

tables learned by the language model can be projected into a high-level latent semantic space. Visualizing the proximity between the query table and other tables with a specific tag can help users understand how the language model works under the table and evaluate the semantic similarity between these tables. The other method in the recommendation algorithm uses the implicit links between tables with the same correlated column pairs to find annotated tables that have similar intra-table relationships with the query table. Visualizing the connectivity between the query table and the tables that have the specific tag in the table interrelation space can help users understand how this part of the recommendation algorithm works. The correspondence between the semantic and interrelation spaces allows visualizing the query table and a specific tag, with the tables labelled with that tag, across two data spaces to provide a holistic view of the recommendation algorithm. The table attribute space that visualizes data inside tables is for the final examination of tag validity if users need to inspect the query table in detail or compare the data between the query table and another table labelled with a specific tag in a pair-wise manner.

4 Solution

I propose to design an all-in-one webpage for TableRepoViz with separate areas for different visualization views, which is similar to CorGIE [16]. The visualization idioms for the three data spaces are described in detail in this section.

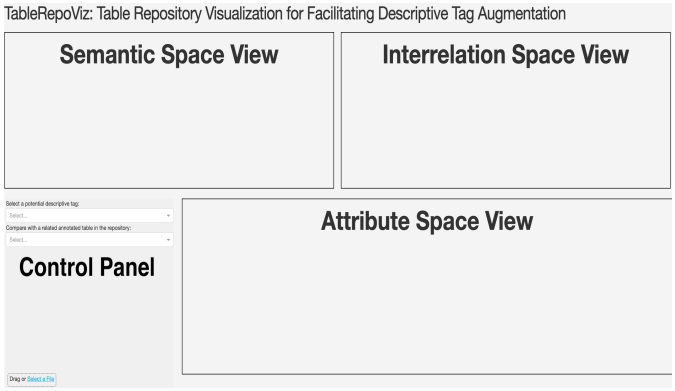

Fig. 6. The interface of TableRepoViz. The views for semantic space and interrelation space are aligned at the top of the interface to facilitate corresponding visualizations between these two spaces. The control panel is located at the bottom left. The attribute space view takes a wide area on the bottom right to accommodate tables with many columns.

General interface and the control panel: Figure 6 depicts the webpage layout and the general interface of TableRepoViz, which comprises a control panel and three visualization views for the data spaces mentioned in the task abstraction. The size and location of each component area are proportional to the need for its corresponding view. The control panel provides interactive support for the users. Figure 7 depicts the control panel where users can input a CSV file as an incoming query table that is about to be added to the table repository. A list of recommended potential tags will be available for users to inspect. In case users want to examine a potential tag, the tables with the same tag in the repository that are related to the query table will be found by the recommendation algorithm and presented to users. These related tables are the origin of the recommended tag. Users can select a related table as a focus item or a potential tag to include all the related tables. The selected items will be visualized in data space views with the query table to help users understand how the tag suggestions are related to the query table and decide whether a recommended tag should be applied to augment the query table. This interaction corresponds to the "specify" process in Figure 5 and shows the origin of a recommended tag in each data space view. A text prompt will be shown to remind users about which focus tag and tables they are visualizing, as in Figure 7. The users can select whether to apply the focus tag to the query table after viewing visualization views in the three data spaces.

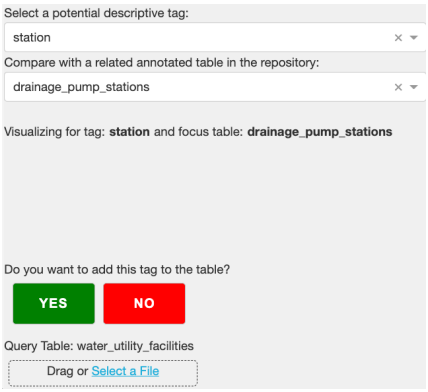

Fig. 7. The control panel. After uploading a query table, users can select a potential tag and a related table they would like to visualize in TableRepoViz from drop-down menus. The users will be asked if they want to annotate the query table with the selected tag.

Semantic space view: The underlying tag recommendation algorithm uses a natural language processing model to abstract a table in a high-dimensional latent space for predicting the probability of each known tag into the repository. Visualizing the distribution of table representations in the latent space will help users understand semantic relatedness between tables, which explains this part of the algorithm recommends a tag. If the location of the query table in the latent space is close to an annotated table or a cluster of annotated tables, that means these tables hold similar values or share similar high-level semantic meaning. Therefore, it is more likely for these tables to share the same descriptive tags. Similar to CorGIE [16], I use UMAP [18] to project table representations from the high-level latent semantic space to nodes on a 2-dimensional scatter plot.

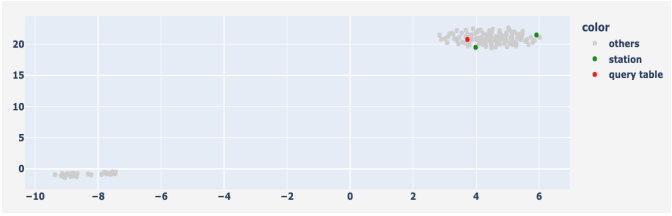

Fig. 8. The table semantics on a 2D space projected by UMAP. Each node represents a table and is colour coded.

Since each descriptive tag can be considered a label, descriptive tag augmentation is inherently a multi-label classification task. In addition to proximity, colour coding can provide visual cues to the nodes with the same label as shown in the UMAP example in Figure 8. Since a user will only examine one tag at a time, I introduce the concept of a **focus tag** and colour code the focus tag in the scatter plot. If the user is interested in a particular potential tag recommended for the query table, all the tables labelled with that tag in the repository will be coded with green in the scatter plot. The user can compare these nodes' distribution with the query table's location in such a visualization idiom. The other nodes will be coded in gray. If the user selects a related table in the control panel, the related table will be highlighted in blue in the latent semantic space view for comparison with the query table.

Interrelation space view: The other part of the recommendation algorithm uses the implicit connection between tables that share similar intra-table relationships, as described in the data abstraction section, to find related tables to infer potential tags. Such implicit connections between tables with the same correlated column pairs can be visualized as links in a network, with each table being a node. The visualized connectivity between nodes in the network is expected to help explain how the matching rules in the recommendation algorithm find a potential tag from the tables related to the query table.

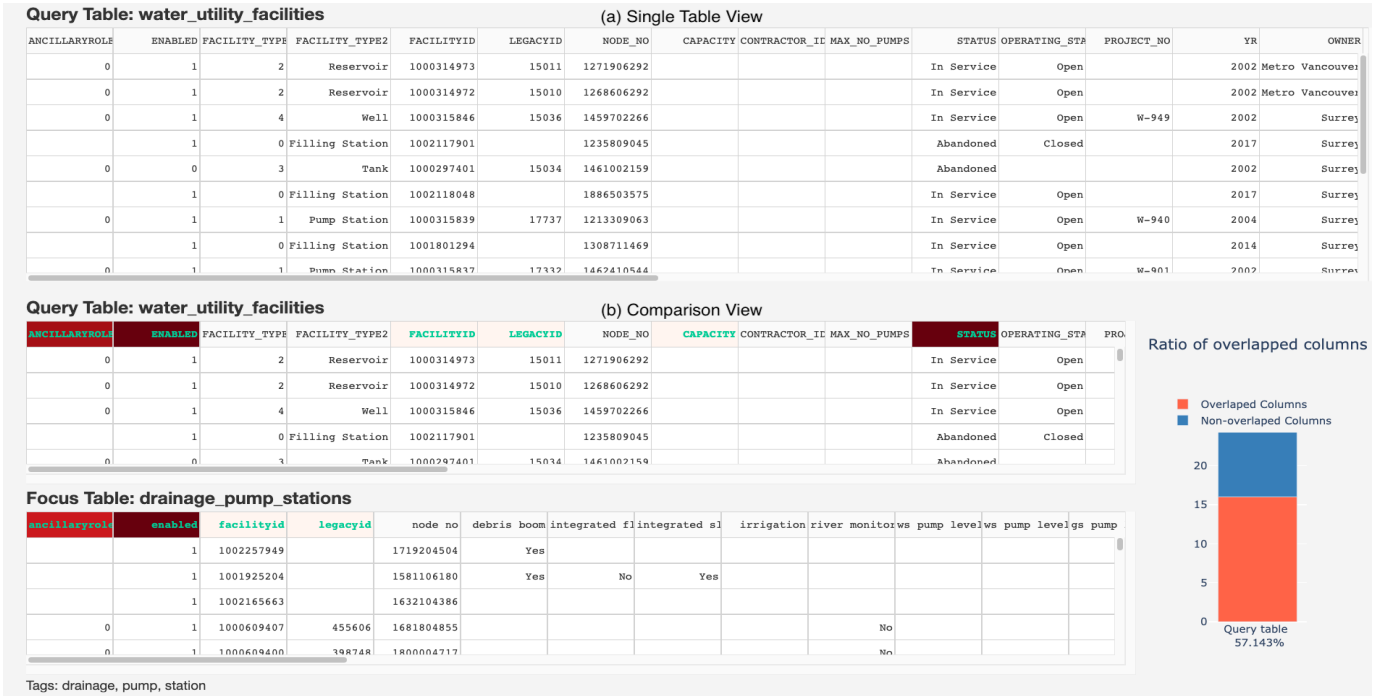

Fig. 9. The attribute space views: (a) the single table view shows the tabular representation of the query table. (b) the comparison view juxtaposes two tables and provides visual cues on column headers, with a bar chart showing the ratio of overlapped columns in the query table.

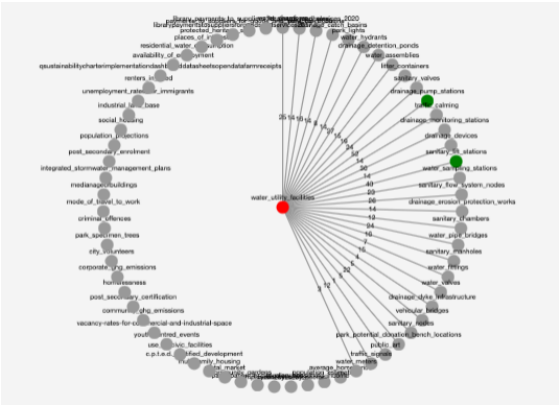

Fig. 10. A network graph for the interrelation space. The red node in the centre is the query table. Tables with the focus tag are colour coded, while others are in gray. The weighted edges depict how many correlated column pairs are shared by the query table and other tables in the repository.

Figure 10 shows a network graph for the query table and the table repository in the table interrelation space. With the same colour-coding strategy in the latent space, tables belonging to a focus tag will be coded in green to make them stand out from other tables in gray. The tables with the same correlated column pairs are connected with a link. The edge weight represents how many common correlated column pairs exist in the two connected tables to help users know to what extent these two tables share similar intra-table relationships. To perform the analysis task, the users can inspect the connectivity of the query table with respect to a focus tag. If the query table has higher weight values on edges with colour-coded tables than other tables, it has a higher chance of belonging to that tag as well. When a related table is selected in the control panel, it will be highlighted in blue in the network graph to correspond to the latent semantic view, as in Figure 5, and provide visual cues to the user.

Attribute space view: The attribute space is dedicated to presenting the feature of the query table to help users examine the validity of a recommended tag. The users need to understand the content of the query table to decide whether a recommended tag is truly suitable for the query table. Figure 9 shows the attribute space view under different user actions. When users upload a CSV file as a query table, the attribute space will present the detailed tabular representation of the query table so that users can look over its column names and cell values. There are few visualization elements in the single table view, as this is the first view the users will see after uploading a CSV file. They may have yet to have a clear exploration plan in mind. This tabular representation showing all cell values allows users to explore the contents of the query table in an unrestricted manner.

For the tables that are the origin of potential tags for the query table, users can select one as a focus table and compare it with the query table. The attribute space will be partitioned into juxtaposed views for pair-wise comparison. With the pair-wise comparison, users can better understand whether the data in the focus table is similar to the query table and whether the tag is valid for the query table. Since the table interrelation space already reflects the intra-table relationships with correlated column pairs, I choose to focus on individual columns in the attribute space.

A column is a cluster of similar entities or values, and it represents a high-level topic in the table. Therefore, column names are a convenient entry point to understand the main thrust of a table. I search for the columns that exist in both the query table and the focus table and plot a bar chart to illustrate the ratio of overlapped columns to all columns in the query table. The overlapped columns are highlighted in the column header of each table. Like a heat map, I add visual cues on the column name cells to encode the percentage of values in the column that also appear in the other table with a colourmap of different shades of red. With these visualization idioms, users can easily perceive how much data the two tables share and infer to what extent the topics in the query table are covered in the focus table. This design will help users determine whether the focus table is similar to the query table and whether the tag can be applied to the query table to describe its content because of the overlapped column names and values.

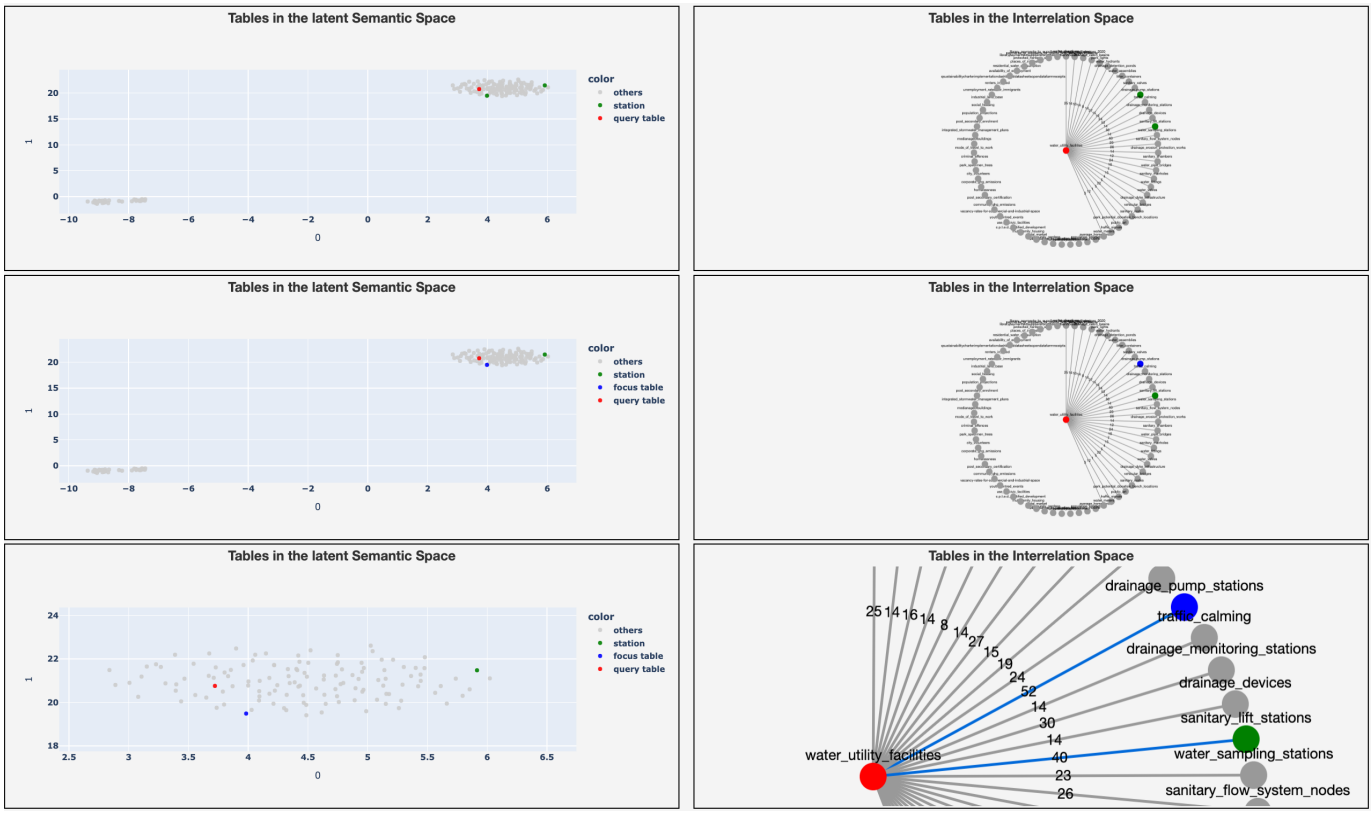

Fig. 11. The semantic and interrelation space views in action. TableRepoViz supports visualizing the query table and the table repository in the semantic and interrelation spaces with respect to a potential tag and a related table. Both graphs allow the user to zoom in to inspect details.

5 Implementation

In order to make TableRepoViz an easy-to-deploy interactive descriptive tag augmentation tool for table repositories, the viable design choice is to implement a web-based visualization application. I used Dash [23], which is a Python framework built on top of Plotly.js and React.js, to build a full-stack web application that supports interactive visualization. Dash takes the responsibility of maintaining the server and responding to client requests from the browser. The web interface was implemented with Dash's HTML components module for the layout of the webpage and the core components module for control flow. Dash has built-in visualization components that support visualization idioms for tabular data and its analysis plots in the attribute space; I hand-picked the visualization encoding designs and the UI elements, such as buttons and drop-down menus. The latent semantic space view was implemented with UMAP [18] to project table semantics into a 2-dimensional scatter plot. Regarding the table interrelation space, I used Cytoscape [4] to visualize the network layout with nodes and edges representing the table repository. All the visualized views were converted to Dash's interactive graphs and passed to the server component for rendering. I wrote Python for the web application's underlying data processing and analysis.

6 Results

I present two usage scenarios demonstrating how TableRepoViz answers the three questions described in the task abstraction. The first two tasks are grouped into a single usage scenario that uses TableRepoViz to visualize the underlying algorithm and recommendation origins. The second usage scenario is to help users examine the validity of a recommended tag.

6.1 Usage Scenario: Grasp recommended tags

Consider a data administrator in charge of maintaining a table repository whose job is annotating descriptive tags to tables newly uploaded to the repository. The user wants to be consistent in the tag augmentation process by labelling tables that have similar content with the same descriptive tags. The tag recommendation algorithm yields a list of already annotated tables related to the new query table and recommends some of their existing tags. TableRepoViz will help the user grasp where the recommended tags are from and how they are related to the query table with visualizations.

After uploading the query table in the control panel (see Figure 7), a list of potential tags that are recommended for the query table will appear in a drop-down menu. The semantic and interrelation space views allow the user to see two graphs depicting the relationship between the query table and other tables in the repository with respect to a potential tag (see Figure 11). The user may be interested in a specific potential tag, say "station," and select it as the focus tag. The tables labelled with the tag "station" are coloured green in the two views. The scatter plot in the semantic space view helps the user understand the similarity between the table semantic representations, which are extracted by a deep learning language model, by comparing their proximity visualized on the plot. Regarding similar intra-table relationships, the network graph in the interrelation space view illustrates the relatedness between two tables with the number of correlated column pairs shared by two tables as edge weight. These pieces of information can help users understand the similarity and relatedness between the query table and the focus tag, as well as the related tables with that tag. To inspect details, the user can zoom in to magnify a local area in either view. In this case, a table labelled with the tag "station" is close to the query table in the semantic space, and it also has the highest edge weight in the interrelation space. The user can select this table as a focus table. TableRepoViz will visualize it more saliently in the space views for better comparison. Moreover, the focus table can be used in the next usage scenario, where the user examines if applying the potential tag to the query table is consistent with past annotations. The graphs for the semantic and interrelation space views can be rendered within 0.5 seconds for changes made in the control panel.

6.2 Usage Scenario: Verify a potential tag

Understanding the legitimacy of tag recommendations is the premise of the final annotation tags. The user still needs to inspect the content in the query table to verify if the recommended tag is truly suitable for the query table. To continue and further validate the inference from the previous usage Scenario, the user may want to have a closer look at the data in the query table. In the attribute space view (see Figure 9), TableRepoViz presents a tabular representation for the query table without visual cues for table elements. Therefore, the user can explore the column names and cell values in an unrestricted manner and obtain a general understanding of the content in the query table.

Apart from accuracy, consistency is another important factor in tag augmentation tasks. TableRepoViz provides a comparison view that juxtaposes the query table and a user-selected table so that the user can compare whether the two tables cover similar topics and content. A bar chart shows the ratio of column names that appear in both tables to the total number of columns in the query table. The user can find the overlapped column names in light green in the table header. The percentage of common values in each overlapped column is visualized with a colormap of different shades of red on the column names cells. The rendering of the comparison view may take a significant amount of time if the table size is large, as it requires comparing cell values between two tables. After viewing these two visual cues, the user can understand to what extent these two tables cover similar topics. This cognitive thinking helps the user answer the final question: whether the tag accurately describes the query table and is consistent with past annotations. The user can click the yes or no button in the control panel to confirm their decision.

7 Milestones

Task	Expeced hours	Acutal hours	Finish date
Continue literature review	10	15	Oct. 25
Learn the front-end and vis tools	20	28	Oct. 31
Implement the general interface	10	18	Nov. 10
Design visualization views	30	32	Nov. 18
Prepare for peer reviews	5	4.5	Nov. 15
Prepare for post-update meeting	5	3.5	Nov. 20
Refine the design	30	21	Dec. 6
Prepare for presentation	10	6	Dec. 12
Finish the report	15	19	Dec. 15
Total	135	147	-

8 Discussion

As an iterative application, TableRepoViz is a successful and useful design study prototype, as it accomplishes the design objective of facilitating descriptive tag augmentation. However, I encountered design limitations attributed to technical factors and the nature of the data and tasks in the tag augmentation problem. Due to time constraints, the design limitations could not be resolved in this design study project and are left for future work.

I believe TableRepoViz is easy for novice users to use, as interactions with the interface are simple and straightforward. Users only need to manipulate the drop-down menus to inform TableRepoViz to generate visualizations. To the best of my knowledge, TableRepoViz is the first visualization tool for facilitating tag augmentation. With this in mind, its target users may have yet to become accustomed to interactive data explainers. To accommodate novice users who do not have experience with analysis through visualizations, TableRepoViz is designed with a simple interface and intuitive visualization idioms. The design borrows the notion of data spaces from CorGIE [16] and follows a straightforward modular design strategy with separate data space views. Each data space view is dedicated to solving one problem. The semantic space view visualizes the similarity between tables within the context of the table repository, while the interrelation space view depicts the relatedness between tables. The visualizations in these two views correspond to each other, and together, they explain why the underlying algorithm recommends a potential tag. The attribute space view allows users to inspect data in the query table and verify if a tag is suitable for the query table. I intentionally attempted to minimize the number of visualization idioms and coding in each view by presenting only the essential ones. This design choice prevents scenarios where excessive visualizations overwhelm and confuse the users.

I encountered design limitations attributed to technical factors and the nature of the data and tasks in the tag augmentation problem. Although user interactions with TableRepoViz's interface are generally simple and straightforward, they bring other disadvantages. In the current design, almost all user interactions are confined to the control panel, with the exception of zooming in and out in data space views. In other words, the visualization views lack support for interactivity. For example, if the user is interested in a table in the semantic space or the interrelation space, it is not possible to directly select that table in the graph view. Instead, they need select that table in the control panel from the drop-down menu, which is inconvenient. Currently, the correspondence relationship is only between the semantic and interrelation spaces. Ideally, future work should implement interactivity that supports visualization correspondence across all three data spaces. If the user clicks a table representation in the semantic space, the table should be considered a focus table and visualized in the other two space views accordingly.

The design of the comparison view in the attribute space only focuses on the columnar similarity between tables by looking for common column names across two tables and overlapped cell values in those columns. This is partly because the tag recommendation algorithm prioritizes column-wise inference to deduce high-level concepts covered in the table. However, common entity reference is another criterion to measure table content similarity. It would be interesting future work to support visualizing whether rows from two tables refer to the same entity and explore other visualization practices for tabular data.

Another limitation is that TableRepoViz is not likely to be scalable. TableRepoViz has already shown a slow response time for rendering large tables in the comparison view of the attribute space, as it needs to compare cell values between the query table and the focus table to help visualize the percentage of shared content on column names. It may also fail to visualize a large table repository, as visualizing a large number of tables could lead to overwhelming visual clutter in data spaces and cognitive overload for the users. In future work, it would be helpful to apply data reduction idioms to circumvent the clutter problem and reduce the computation costs.

To assess the functionality and usability of TableRepoViz, conducting user studies is essential in future work. The user studies can also gather user feedback about visualization idioms used in the data space views. It is an important part of the iterative design cycle that can help improve TableRepoViz.

9 Conclusion

In this work, I present TableRepoViz, an interactive visual explainer tool for helping users make tag augmentation decisions when adding a new table into a table repository. TableRepoViz uses the notion of data spaces to visualize how the recommendation algorithm works in the context of a table repository. It also helps users find the origin of a recommended tag in the repository and analyze if it is suitable for the query table. The semantic space view visualizes the similarity between tables with their semantic representations, and the interrelation space view illustrates the tables in the repository are related to the query table. These two spaces correspond to each other to provide a holistic overview of the recommendation algorithm. The attribute space view assists users in verifying if the tag is accurate in describing the new table and consistent with previous annotations in the repository.

As a rudimentary design study for visual facilitation in descriptive tag augmentation tasks, TableRepoViz fulfills all functional requirements through interactive visualizations with a simple and straightforward user interface. I believe this project can inspire and guide future work on the same topic or other data explainers that compare tables.

ACKNOWLEDGMENTS

The author wish to thank Tamara Munzner for her guidance and my peers for their valuable feedback on this project.

REFERENCES

[1] M. Cafarella, A. Halevy, H. Lee, J. Madhavan, C. Yu, D. Z. Wang, and E. Wu. Ten years of webtables. *Proceedings of the VLDB Endowment*, 11(12):2140–2149, Aug. 2018. doi: 10.14778/3229863.3240492

[2] City of Surrey. Datasets - City of Surrey Open Data Catalogue, 2022. https://data.surrey.ca/dataset.

[3] V. Cutrona, F. Bianchi, E. Jiménez-Ruiz, and M. Palmonari. Tough tables: Carefully evaluating entity linking for tabular data. In J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, eds., *The Semantic Web – ISWC 2020*, pp. 328–343. Springer International Publishing, Cham, Switzerland, 2020.

[4] Cytoscape Consortium. Cytoscape An Open Source Platform for Complex Network Analysis and Visualization, 2018. https://cytoscape.org/.

[5] X. Deng, H. Sun, A. Lees, Y. Wu, and C. Yu. TURL: Table understanding through representation learning. *Proceedings of the VLDB Endowment*, 14(3):307–319, Nov. 2020. doi: 10.14778/3430915.3430921

[6] J. Fulda, M. Brehmer, and T. Munzner. TimeLineCurator: Interactive Authoring of Visual Timelines from Unstructured Text. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):300–309, Jan. 2016. doi: 10.1109/TVCG.2015.2467531

[7] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, A. Lex, and M. Streit. Taggle: Combining overview and details in tabular data visualizations. *Information Visualization*, 19(2):114–136, Apr. 2020. doi: 10.1177/1473871619878085

[8] S. Galhotra and U. Khurana. Automated relational data explanation using external semantic knowledge. *Proceedings of the VLDB Endowment*, 15(12):3562–3565, Aug. 2022. doi: 10.14778/3554821.3554844

[9] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2023–2032, Dec. 2014. doi: 10.1109/TVCG.2014.2346260

[10] R. Hourieh. Comparative visualization of large tabular data. Master's thesis, Johannes Kepler University Linz, Linz, Austria, Mar. 2016.

[11] U. Khurana and S. Galhotra. Semantic concept annotation for tabular data. In *Proceedings of the ACM International Conference on Information Knowledge Management (CIKM)*, p. 844–853. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3459637.3482295

[12] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1–2):1338–1347, Sept. 2010. doi: 10.14778/1920841.1921005

[13] F. Liu, C. Yu, W. Meng, and A. Chowdhury. Effective keyword search in relational databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, p. 563–574. Association for Computing Machinery, New York, NY, USA, 2006. doi: 10.1145/1142473.1142536

[14] Z. Liu, S. B. Navathe, and J. T. Stasko. Network-based visual analysis of tabular data. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 41–50, 2011. doi: 10.1109/VAST.2011.6102440

[15] Z. Liu, S. B. Navathe, and J. T. Stasko. Ploceus: Modeling, visualizing, and analyzing tabular data as networks. *Information Visualization*, 13(1):59–89, 2014. doi: 10.1177/1473871613488591

[16] Z. Liu, Y. Wang, J. Bernard, and T. Munzner. Visualizing Graph Neural Networks With CorGIE: Corresponding a Graph to Its Embedding. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2500–2516, 2022. doi: 10.1109/TVCG.2022.3148197

[17] V. Mala and D. K. Lobiyal. Semantic and keyword based web techniques in information retrieval. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 23–26, 2016. doi: 10.1109/CCAA.2016.7813724

[18] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. doi: 10.48550/ARXIV.1802.03426

[19] F. Nargesian, E. Zhu, K. Q. Pu, and R. J. Miller. Table union search on open data. *Proceedings of the VLDB Endowment*, 11(7):813–825, Mar. 2018. doi: 10.14778/3192965.3192973

[20] C. Niederer, H. Stitz, R. Hourieh, F. Grassinger, W. Aigner, and M. Streit. Taco: Visualizing changes in tables over time. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):677–686, 2018. doi: 10.1109/TVCG.2017.2745298

[21] M. Oppermann, R. Kincaid, and T. Munzner. VizCommender: Computing Text-Based Similarity in Visualization Repositories for Content-Based Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):495–505, 2021. doi: 10.1109/TVCG.2020.3030387

[22] T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J.-P. Rudolph, M. Schönberg, J. Zwiener, and F. Naumann. Functional dependency discovery: An experimental evaluation of seven algorithms. *Proceedings of the VLDB Endowment*, 8(10):1082–1093, June 2015. doi: 10.14778/2794367.2794377

[23] Plotly. Dash Documentation User Guide, 2022. https://dash.plotly.com/.

[24] S. Ramnandan, A. Mittal, C. A. Knoblock, and P. Szekely. Assigning semantic labels to data sources. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, eds., *The Semantic Web. Latest Advances and New Domains*, pp. 403–417. Springer International Publishing, Cham, Switzerland, 2015.

[25] D. Ritze and C. Bizer. Matching web tables to dbpedia - A feature utility study. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pp. 210–221. OpenProceedings.org, Mar. 2017. doi: 10.5441/002/edbt.2017.20

[26] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, Jan. 2002. doi: 10.1109/2945.981851

[27] Y. Suhara, J. Li, Y. Li, D. Zhang, c. Demiralp, C. Chen, and W.-C. Tan. Annotating columns with pre-trained language models. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, p. 1493–1503. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3514221.3517906

[28] Tableau Software. Business Intelligence and Analytics Software, 2003. https://www.tableau.com/.

[29] P. P. Talukdar, Z. G. Ives, and F. Pereira. Automatically incorporating new sources in keyword search-based data integration. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, p. 387–398. Association for Computing Machinery, New York, NY, USA, 2010. doi: 10.1145/1807167.1807211

[30] S. Zhang and K. Balog. Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(2), Jan. 2020. doi: 10.1145/3372117

[31] Y. Zhang and Z. G. Ives. Finding related tables in data lakes for interactive data science. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, p. 1951–1966. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3318464.3389726

[32] E. Zhu, D. Deng, F. Nargesian, and R. J. Miller. Josie: Overlap set similarity search for finding joinable tables in data lakes. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, p. 847–864. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3299869.3300065