TableRepoViz: Visualizing Tabular Data Repositories for Facilitating Descriptive Tag Augmentation

Jianhao Cao
Department of Computer Science
University of British Columbia
jhcao@cs.ubc.ca

ABSTRACT

Intentionally left empty.

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, 29]. They make good data sources for data science tasks because of their categorical structure and richness in data content. Storing and maintaining multiple tabular datasets in a centralized data lake, or a table repository, can provide a single access point that allows data practitioners to query data on a certain topic or search for similar datasets.

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 [12, 16, 28]. 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 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 am currently working 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 1). The overall technical details are omitted in this project proposal, 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. Human judgment may be the most accurate method to decide whether the suggested tags should be applied to the query table.

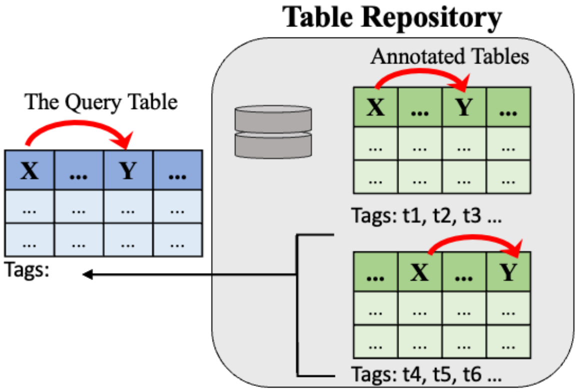


Figure 1: Descriptive tag augmentation in a table repository. Tags of the already annotated tables that are related to the query table are suggested for the new incoming query table.

I propose a design study project to implement an interactive application that visualizes the descriptive tag augmentation scenario in a table repository. The intention of the interactive augmentation application is to provide a interactive visualization interface that assists human annotators in examining the tag suggestions and their origins and comparing recommended related tables with the query table. Ideally, an annotator can use the interactive interface to judge which potential tags can be applied to the query table after browsing the visualization views of the related tables in the table repository containing the suggested tags and comparing them with the query table. There are three questions to be answered in this project: (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 descriptive of the query table? The detailed data and task abstraction is addressed in a later section.

2 RELATED WORK

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, 2, 24]. 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 [11], and the matching methods usually leverage a cross-domain knowledge base or ontology. Ramnandan et al. [23] 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 [3, 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 [30]. 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. [18] 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. [31] 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 [25] are visualization systems that offer many tabular data visualization solutions for analyzing their patterns. Furmanova et al. [5] 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 [13, 14]. 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 [7, 8] emphasizes attribute value comparison rather than the high-level semantic similarity between tables. Regarding other data types, VizCommender [20] 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 [10] and provide a user interface to visualize how the output results are obtained [6]. However, this application only visualizes the table explanation in an informative manner as their intention is to present how the explanation algorithm work 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 [4], 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 [15] 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 repertory in these three spaces. This design can help users understand how the descriptive tag recommendation algorithm works and determine whether the tag suggestions are truly descriptive and related to the query table.

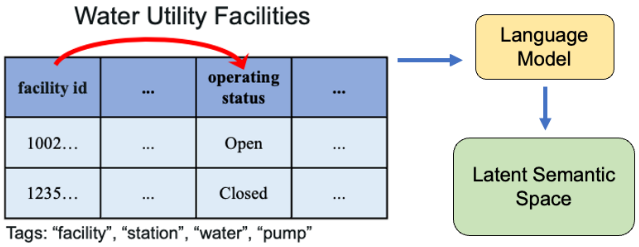
3 DATA & TASK ABSTRACTION

3.1 Background

I propose designing an interactive annotation tool, TableRepoViz, to facilitate descriptive tag augmentation for a new query table when it is being added to a table repository. I worked on another project about automated descriptive tag augmentation for new tables being added to a table repository. 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.

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 [21] 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** → **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, the interactive augmentation tool 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 provide hints and help users decide which recommended tags should be applied to the new query table. The tool's utility and the quality of the tag suggestions from the automated recommendation methods can be examined in a user study if time permits.



Water Utility Facilities

facility id	...	operating status	...
1002…	...	Open	...
1235…	...	Closed	...

Tags: “facility”, “station”, “water”, “pump”

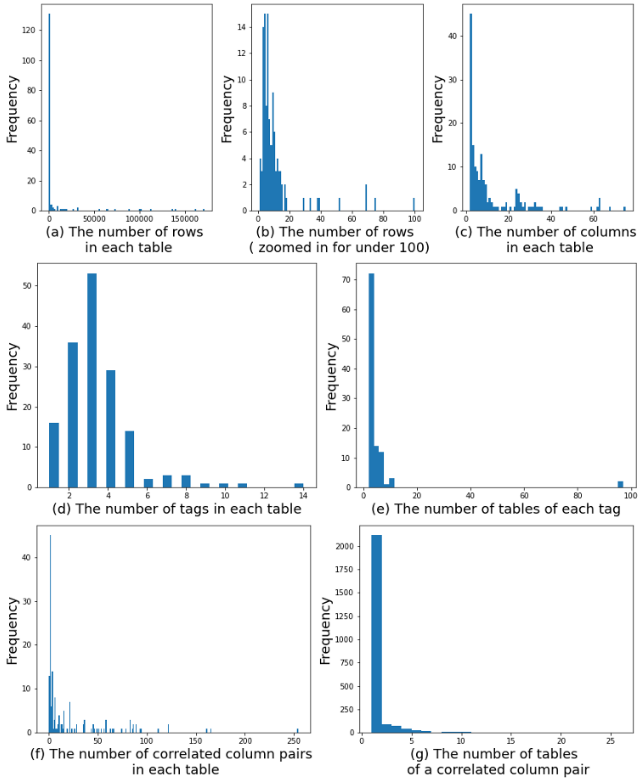
Figure 2: A table and its properties.

3.2 Data

The tabular dataset is from my previous project on automated descriptive tag augmentation. I retrieved the CSV files that come with a table header and column names from the Open Data Site of the City of Surrey [19]. 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 2 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 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. As for the per-table statistics, Figure 3 shows the histograms of various table statistics in the table repository. 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.



(a) The number of rows in each table
(b) The number of rows (zoomed in for under 100)
(c) The number of columns in each table
(d) The number of tags in each table
(e) The number of tables of each tag
(f) The number of correlated column pairs in each table
(g) The number of tables of a correlated column pair

Figure 3: Histograms of table statistics.

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

At a high level, TableRepoViz is intended to facilitate the descriptive tag augmentation process when adding a new table into a table repository with visualizations. Specifically, users can 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 how recommended tags and the query table are related or view 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 descriptive of 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 potential 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 descriptive of the query table and should be assigned. Visualizing how the recommendation algorithm work 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 separately visualize them to explain how the potential tags and their origin tables are related to the query table.

Examine 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 truly descriptive of the input query. Users can visualize the query table to view the data in the query table to inspect whether it is consistent with the recommended tag or have a pair-wise comparison between the query table and another table annotated with the same tag to examine if they share similar semantics.

These tasks can be abstracted in different solution spaces as the visualization paradigm in CorGIE [15], 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 4.

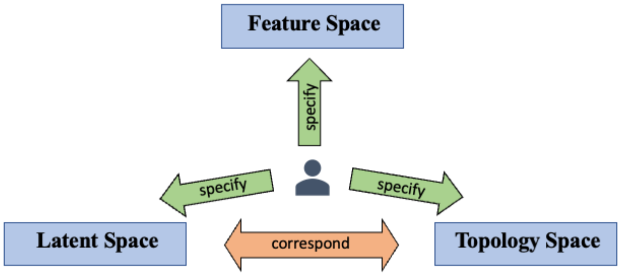
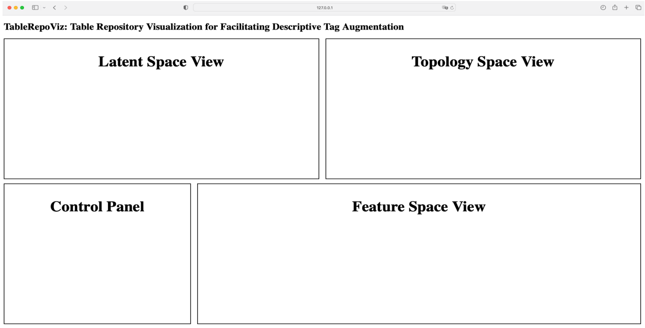


Figure 4: The three data spaces for task abstraction.

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 space. Visualizing the proximity between the query table and other tables with the specified 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 specified tag in the topology space can help users understand how this part of the recommendation algorithm works. The correspondence between the latent space and the topology space allows visualizing the query table and a focus tag, with the tables labelled with that tag, across two data spaces to provide a holistic view of the recommendation algorithm. The feature space 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 the specified tag in a pair-wise manner.

4 SOLUTION

General interface and the control panel: Figure 5 depicts the layout and the general interface of the interactive descriptive tag annotation application for table repositories. It comprises a control panel and three visualization views for the three spaces mentioned in the task abstraction, respectively. The size of each component is proportional to the need for its corresponding visualization view. The control panel provides interactive support for the users. Figure 7 shows the control flow in action. In the control panel, 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 4 and shows the origin of a recommended tag in each data space view.



127.0.0.1

TableRepoViz: Table Repository Visualization for Facilitating Descriptive Tag Augmentation

Figure 5: The general interface.

Latent 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 in 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 and alike high-level semantic meaning. Therefore, it is more likely for these tables to share the same descriptive tags. Similar to CorGIE [15], I propose to use UMAP [17] to project table representations from the latent space to nodes on a 2-dimensional scatter plot.

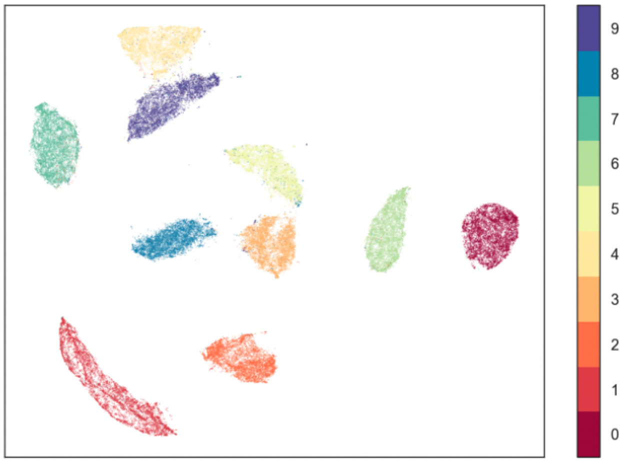
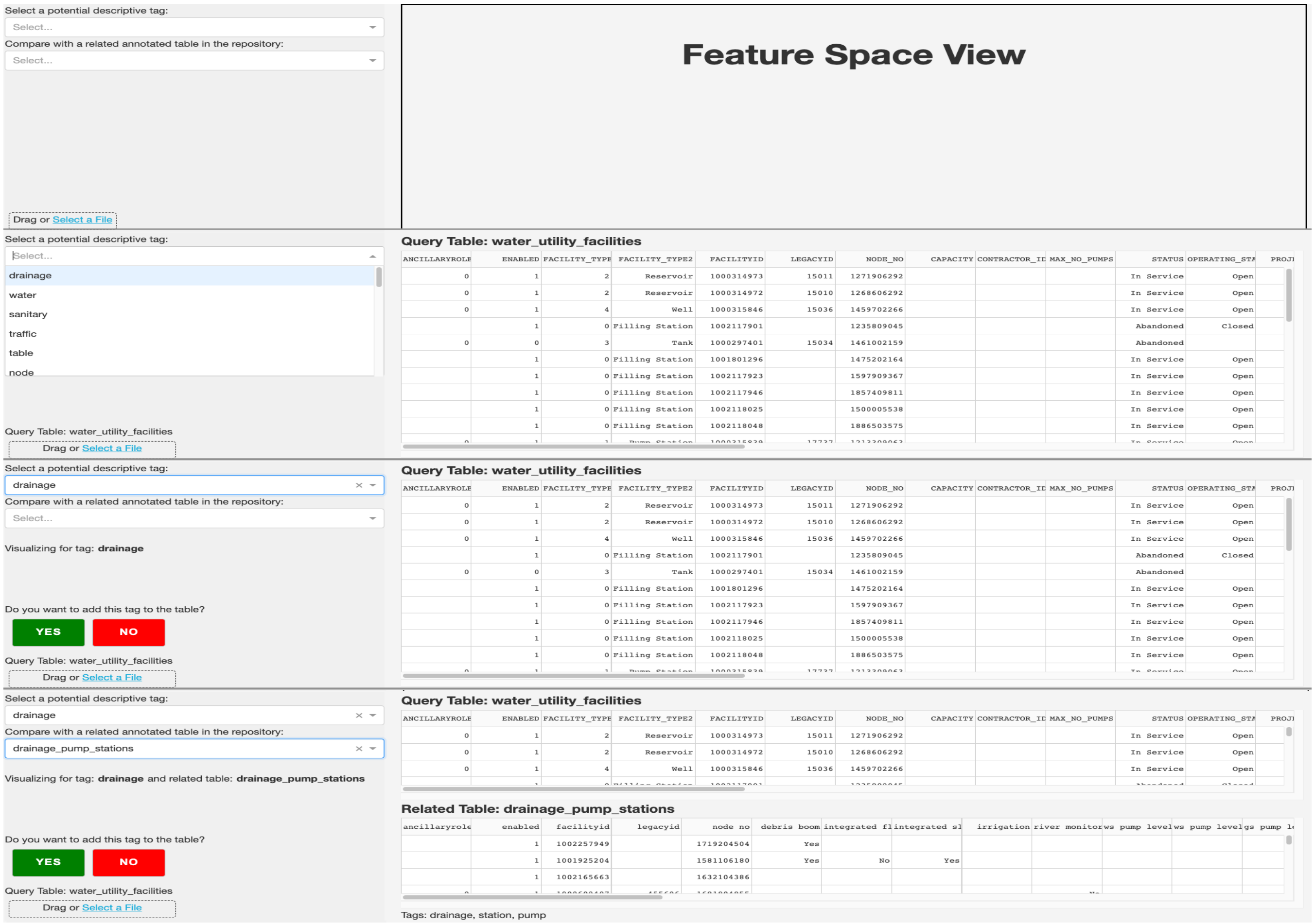


Figure 6: An example of colour coding in UMAP [17].

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 6. However, a table can be annotated with more than one tag in the table repository. Currently, I cannot find a way to code a node with multiple tags. As a compromise, I introduce the concept of a **focus tag** to colour code only one tag at a time. If a 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 a highly saturated colour in the scatter plot. The user can better compare these nodes' distribution with the query table's location in such a design. The other nodes will be coded in gray or simply omitted from the scatter plot. If the user selects a related table in the control panel, the related table will be highlighted in the latent space view for comparison.

Ideally, users should be able to directly select a node on the scatter plot to switch to a different **focus table** or **focus tag** to compare other nodes' distribution with the query table. The newly specified table(s) should be reflected in visualization views in other data spaces and vice versa. However, I am unsure about the feasibility of supporting interaction with UMAP; this design approach will be revisited later if time permits.



Select a potential descriptive tag:

Select...

Compare with a related annotated table in the repository:

Select...

Drag or Select a File

Feature Space View

Select a potential descriptive tag:

Select...

drainage
water
sanitary
traffic
table
node

Query Table: water_utility_facilities

Drag or Select a File

Query Table: water_utility_facilities

ANCILLARYROLE	ENABLED	FACILITY_TYPE	FACILITY_TYPE2	FACILITYID	LEGACYID	NODE_NO	CAPACITY	CONTRACTOR_ID	MAX_NO_PUMPS	STATUS	OPERATING_STA	PROJ
0	1	2	Reservoir	1000314973	15011	1271906292				In Service	Open	
0	1	2	Reservoir	1000314972	15010	1268606292				In Service	Open	
0	1	4	Well	1000315846	15036	1459702266				In Service	Open	
	1	0	Filling Station	1002117901		1235809045				Abandoned	Closed	
0	0	3	Tank	1000297401	15034	1461002159				Abandoned		
	1	0	Filling Station	1001801296		1475202164				In Service	Open	
	1	0	Filling Station	1002117923		1597909367				In Service	Open	
	1	0	Filling Station	1002117946		1857409811				In Service	Open	
	1	0	Filling Station	1002118025		1500005538				In Service	Open	
	1	0	Filling Station	1002118048		1886503575				In Service	Open	

Select a potential descriptive tag:

drainage ×

Compare with a related annotated table in the repository:

Select...

Visualizing for tag: **drainage**

Do you want to add this tag to the table?

Query Table: water_utility_facilities

Drag or Select a File

Query Table: water_utility_facilities

ANCILLARYROLE	ENABLED	FACILITY_TYPE	FACILITY_TYPE2	FACILITYID	LEGACYID	NODE_NO	CAPACITY	CONTRACTOR_ID	MAX_NO_PUMPS	STATUS	OPERATING_STA	PROJ
0	1	2	Reservoir	1000314973	15011	1271906292				In Service	Open	
0	1	2	Reservoir	1000314972	15010	1268606292				In Service	Open	
0	1	4	Well	1000315846	15036	1459702266				In Service	Open	
	1	0	Filling Station	1002117901		1235809045				Abandoned	Closed	
0	0	3	Tank	1000297401	15034	1461002159				Abandoned		
	1	0	Filling Station	1001801296		1475202164				In Service	Open	
	1	0	Filling Station	1002117923		1597909367				In Service	Open	
	1	0	Filling Station	1002117946		1857409811				In Service	Open	
	1	0	Filling Station	1002118025		1500005538				In Service	Open	
	1	0	Filling Station	1002118048		1886503575				In Service	Open	

Select a potential descriptive tag:

drainage ×

Compare with a related annotated table in the repository:

drainage_pump_stations ×

Visualizing for tag: **drainage** and related table: **drainage_pump_stations**

Do you want to add this tag to the table?

Query Table: water_utility_facilities

Drag or Select a File

Query Table: water_utility_facilities

ANCILLARYROLE	ENABLED	FACILITY_TYPE	FACILITY_TYPE2	FACILITYID	LEGACYID	NODE_NO	CAPACITY	CONTRACTOR_ID	MAX_NO_PUMPS	STATUS	OPERATING_STA	PROJ
0	1	2	Reservoir	1000314973	15011	1271906292				In Service	Open	
0	1	2	Reservoir	1000314972	15010	1268606292				In Service	Open	
0	1	4	Well	1000315846	15036	1459702266				In Service	Open	

Related Table: drainage_pump_stations

ancillaryrole	enabled	facilityid	legacyid	node no	debris boom	integrated fl	integrated sl	irrigation	river monitor	ws pump level	ws pump level	gs pump le
	1	1002257949		1719204504	Yes							
	1	1001925204		1581106180	Yes	No	Yes					
	1	1002165663		1632104386								

Tags: drainage, station, pump

Figure 7: The two bottom views in action.

Topology space view: The other part of the recommendation algorithm uses the implicit connection between tables with the same intra-table relationships, as described in the data abstraction section. Such connections between tables that have the same correlated column pairs can be visualized as links in a network, with each table being a node. The visualized connectivity in the network is expected to help explain why the matching rules used in the recommendation algorithm suggest a tag.

Figure 8 shows a potential topological graph view solution. With the same colour-coding strategy in the latent space, tables belonging to a selected tag will be coded with a highly saturated colour to make them stand out from other tables in gray. The tables with the same correlated column pairs are connected with a link. The width of the link will be adjusted to reflect the number of the same correlated column pairs between the two connected tables. To ask the analysis task, the users can inspect the connectivity of the query table with respect to a **focus** tag. If the query table has significantly more links with colour-coded tables than the other tables in gray, 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 the network to provide visual cues to the user.

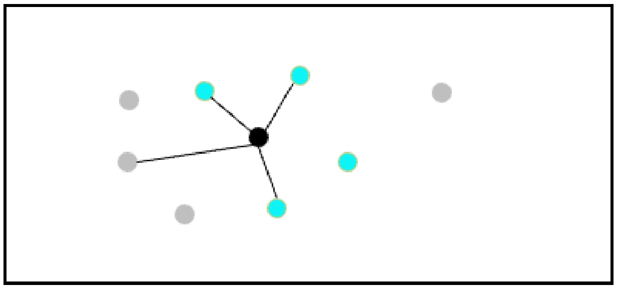


Figure 8: A potential graph view for the topology space. The black node is the query table. Tables with the focus tag are colour coded, while others are in gray.

Feature space view: The feature 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 descriptive of the query table. Figure 7 shows the feature space view under different user actions. When users upload a CSV file as a query table, the feature space will present the detailed tabular representation of the query table so that users can look over its column names and cell values. The feature space view will be partitioned into juxtaposed views for pair-wise comparison between the query table and a related table selected from the control panel. With the pair-wise comparison, users can determine whether the selected table is similar to the query table and whether the recommended known tag can be applied to augment the query table. The tabular view is the only solution I can think of for the feature space. I thought about visualizing data distributions in each table, but it is not easy because the values in a table are a mix of categorical, ordered, and text data in different columns. I look forward to feedback about other visualization paradigms that support high-level semantic understanding or pair-wise comparison.

5 IMPLEMENTATION

In order to create 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 [22], 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. The general interface is implemented with Dash's HTML components module for the web page layout and the core components module for control flow. The current table visualization in the feature space uses Dash's visualization component. The latent space view will be implemented with UMAP [17] to project table semantics into a 2-dimensional scatter plot. In terms of the implementation choice for the topology space, there are two different options that I am currently considering. The first one is Visdcc, which is an extended Dash components module that supports network visualization, and the second one is PyVis [9], an interactive network visualization Python package. Python scripts are used for underlying data processing in the web application.

6 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	working	Nov. 20
Prepare for peer reviews	5		Nov. 15
Prepare for post-update meeting	5		Nov. 20
Refine the design	30		Dec. 7
Prepare for presentation	10		Dec. 11
Finish the report	15		Dec. 15

7 DISCUSSION & FUTURE WORK

Intentionally left empty.

8 CONCLUSION

Intentionally left empty.

REFERENCES

[1] Michael Cafarella, Alon Halevy, Hongrae Lee, Jayant Madhavan, Cong Yu, Daisy Zhe Wang, and Eugene Wu. 2018. Ten Years of Webtables. *Proceedings of the VLDB Endowment* 11, 12 (Aug. 2018), 2140–2149. https://doi.org/10.14778/3229863.3240492

[2] Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz, and Matteo Palmonari. 2020. Tough Tables: Carefully Evaluating Entity Linking for Tabular Data. In *The Semantic Web – ISWC 2020*, Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal (Eds.). Springer International Publishing, Cham, Switzerland, 328–343.

[3] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proceedings of the VLDB Endowment* 14, 3 (Nov. 2020), 307–319. https://doi.org/10.14778/3430915.3430921

[4] Johanna Fulda, Matthew Brehmer, and Tamara Munzner. 2016. TimeLineCurator: Interactive Authoring of Visual Timelines from Unstructured Text. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 300–309. https://doi.org/10.1109/TVCG.2015.2467531

[5] Katarina Furmanova, Samuel Gratzl, Holger Stitz, Thomas Zichner, Miroslava Jaresova, Alexander Lex, and Marc Streit. 2020. Taggle: Combining overview and details in tabular data visualizations. *Information Visualization* 19, 2 (2020), 114–136. https://doi.org/10.1177/1473871619878085 arXiv:https://doi.org/10.1177/1473871619878085

[6] Sainyam Galhotra and Udayan Khurana. 2022. Automated Relational Data Explanation Using External Semantic Knowledge. *Proc. VLDB Endow.* 15, 12 (aug 2022), 3562–3565. https://doi.org/10.14778/3554821.3554844

[7] Samuel Gratzl, Nils Gehlenborg, Alexander Lex, Hanspeter Pfister, and Marc Streit. 2014. Domino: Extracting, Comparing, and Manipulating Subsets Across Multiple Tabular Datasets. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2023–2032. https://doi.org/10.1109/TVCG.2014.2346260

[8] Reem Hourieh. 2016. *Comparative Visualization of Large Tabular Data.* Master's thesis. Johannes Kepler University Linz, Linz, Austria.

[9] West Health Institute. 2018. *Interactive network visualizations — pyvis 0.1.3.1 documentation.* https://pyvis.readthedocs.io/en/latest/

[10] Udayan Khurana and Sainyam Galhotra. 2021. Semantic Concept Annotation for Tabular Data. In *Proceedings of the 30th ACM International Conference on Information Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 844–853. https://doi.org/10.1145/3459637.3482295

[11] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and Searching Web Tables Using Entities, Types and Relationships. *Proceedings of the VLDB Endowment* 3, 1–2 (Sept. 2010), 1338–1347. https://doi.org/10.14778/1920841.1921005

[12] Fang Liu, Clement Yu, Weiyi Meng, and Abdur Chowdhury. 2006. Effective Keyword Search in Relational Databases. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data* (Chicago, IL, USA) *(SIGMOD '06)*. Association for Computing Machinery, New York, NY, USA, 563–574. https://doi.org/10.1145/1142473.1142536

[13] Zhicheng Liu, Shamkant B. Navathe, and John T. Stasko. 2011. Network-based visual analysis of tabular data. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 41–50. https://doi.org/10.1109/VAST.2011.6102440

[14] Zhicheng Liu, Shamkant B Navathe, and John T Stasko. 2014. Ploceus: Modeling, visualizing, and analyzing tabular data as networks. *Information Visualization* 13, 1 (2014), 59–89. https://doi.org/10.1177/1473871613488591 arXiv:https://doi.org/10.1177/1473871613488591

[15] Zipeng Liu, Yang Wang, Jürgen Bernard, and Tamara Munzner. 2022. Visualizing Graph Neural Networks With CorGIE: Corresponding a Graph to Its Embedding. *IEEE Transactions on Visualization and Computer Graphics* 28, 6 (2022), 2500–2516. https://doi.org/10.1109/TVCG.2022.3148197

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

[17] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://doi.org/10.48550/ARXIV.1802.03426

[18] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proceedings of the VLDB Endowment* 11, 7 (March 2018), 813–825. https://doi.org/10.14778/3192965.3192973

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

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

[21] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener, and Felix Naumann. 2015. Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms. *Proceedings of the VLDB Endowment* 8, 10 (June 2015), 1082–1093. https://doi.org/10.14778/2794367.2794377

[22] Plotly. 2022. *Dash Documentation User Guide*. https://dash.plotly.com/

[23] S.K. Ramnandan, Amol Mittal, Craig A. Knoblock, and Pedro Szekely. 2015. Assigning Semantic Labels to Data Sources. In *The Semantic Web. Latest Advances and New Domains*, Fabien Gandon, Marta Sabou, Harald Sack, Claudia d'Amato, Philippe Cudré-Mauroux, and Antoine Zimmermann (Eds.). Springer International Publishing, Cham, Switzerland, 403–417.

[24] Dominique Ritze and Christian Bizer. 2017. Matching Web Tables To DBpedia - A Feature Utility Study. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017*, Volker Markl, Salvatore Orlando, Bernhard Mitschang, Periklis Andritsos, Kai-Uwe Sattler, and Sebastian Breß (Eds.). OpenProceedings.org, 210–221. https://doi.org/10.5441/002/edbt.2017.20

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

[26] Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (jan 2002), 52–65. https://doi.org/10.1109/2945.981851

[27] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating Columns with Pre-Trained Language Models. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1493–1503. https://doi.org/10.1145/3514221.3517906

[28] Partha Pratim Talukdar, Zachary G. Ives, and Fernando Pereira. 2010. Automatically Incorporating New Sources in Keyword Search-Based Data Integration. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data* (Indianapolis, Indiana, USA) *(SIGMOD '10)*. Association for Computing Machinery, New York, NY, USA, 387–398. https://doi.org/10.1145/1807167.1807211

[29] Shuo Zhang and Krisztian Balog. 2020. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.* 11, 2, Article 13 (Jan. 2020), 35 pages. https://doi.org/10.1145/3372117

[30] Yi Zhang and Zachary G. Ives. 2020. Finding Related Tables in Data Lakes for Interactive Data Science. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1951–1966. https://doi.org/10.1145/3318464.3389726

[31] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 847–864. https://doi.org/10.1145/3299869.3300065