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 [2, 21]. 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 [9, 11, 20]. 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 not uncommon 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 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.

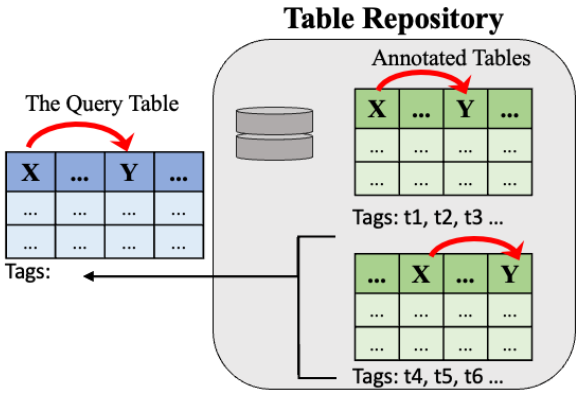

Figure 1: Descriptive tag augmentation in a table repository. Tags of the already annotated tables that are related to the query table are suggested for the new incoming query table.

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 tag suggestions 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) how to visualize already annotated tables in a table repository to help annotators understand its content and characteristics? (2) how to visualize which existing tables are origins of the potential tags and related to the incoming query table? (3) how to provide an efficient pair-wise view for comparing the recommended related tables and 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 [2, 3, 18]. 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 [8], and the matching methods usually leverage a cross-domain knowledge base or ontology. Ramnandan et al. [17] 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 [4, 19]. I have two suggestion methods in the tag-recommending system I am currently working on. The first one uses column name matching rules 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 [22]. 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. [13] 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. [23] 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 Interactive Visualization Applications

Galhotra and Khurana propose an automated data explanation system to identify a concept for each column in a table [7] and provide a user interface to visualize how the output results are obtained [6]. However, the interface's visualization views are very preliminary and do not support analysis as their intention is to present how the explanation algorithm work under the table in an informative manner. TimeLineCurator [5] is an interactive tool for human users to author visual timelines from unstructured text. The authoring is through an underlying data processing pipeline and a general user interface with different functional areas for a control panel and visualization views for various components in the visual timeline authoring tasks. VizCommender [15] provides content-based recommendations for visualizations based on text similarity. It allows comparative judgments by supporting pairwise comparisons between visualizations. VizSnippets [16] create visual inspectors and use them for developing and evaluating building blocks in compressing visualization bundles into previews. Similar concepts can be adopted to tackle analysis questions in this project. CorGIE [10] is an interactive interface that explains the representation learning of a graph neural network by visualizing nodes in different data spaces and their correspondence between spaces.

The papers mentioned above provide many insights that could guide this design study project to implement an interactive interface with table visualization views facilitating descriptive tag augmentation. In addition to these papers, I will also further explore other related interactive interfaces that use visualization techniques to facilitate similar tasks.

3 DATA & TASK ABSTRACTION

I propose designing an interactive annotation tool that visualizes tables in a table repository to help a human user determine which descriptive tags should be augmented to a new query table.

The dataset is from a previous project I worked on automatically suggesting potential tags. I retrieved the CSV files with a table header of column names from the Open Data Site of the City of Surrey [14]. Data were collected with minimal human intervention and were not further gauged or refined to emulate that data in online table repositories is usually not well-formed.

tables	tags	columns	rows	attribute pairs
160	104	1,907	1,443,378	4,074

Table 1: Dataset statistics.

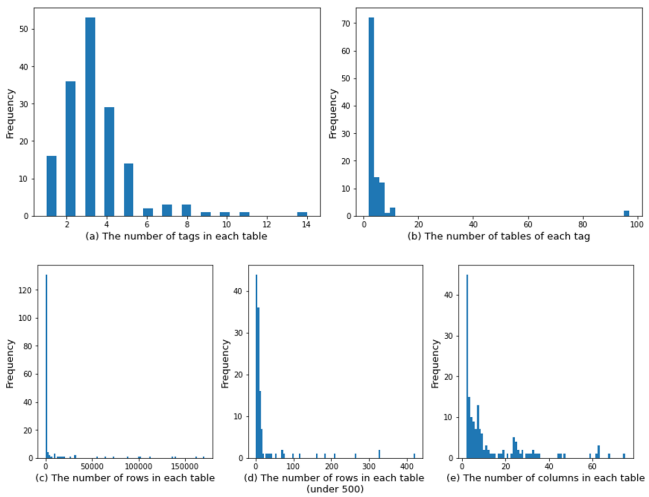

Figure 2: Histograms of table statistics.

The descriptive tags of each CSV file were scraped from the open data site. Each CSV file is represented as a two-dimensional table with a header of column names and a list of items stored in the table. Since the tag suggestion algorithm looks for related tables by examining whether they share the same correlated attribute pairs, I also annotated the correlated attributes in the tables. The correlatedness here is defined by having a functional dependency between two attributes (i.e., each value in one column is uniquely associated with a value in another column).

The table dataset has 160 tables, 1,907 columns, and 1,443,378 rows. There are 104 distinct tags in the dataset. However, a tag can be labelled for more than one table, and the total tag count is 469. Across all the tables, there exist 4,074 pairs of correlated attribute pairs. Figure 2 shows the histograms of various table statistics. Most tables in the dataset are small in size, as around two-thirds of the tables, or 104 to be precise, have less than 100 rows and 20 columns. Each table has at least 1 annotated tag, and over half of the tables have 5 tags or fewer. On the other hand, most tags have less than 12 but at least 2 tables. Therefore, the dataset is balanced regarding table and tag association.

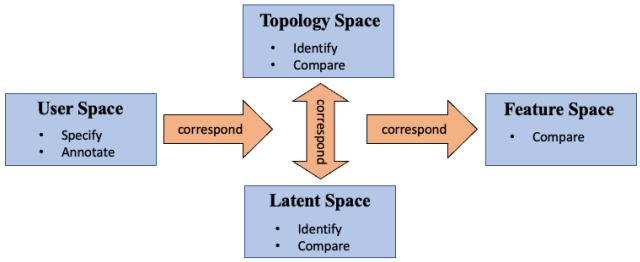

Figure 3: Data and task abstraction.

The data and task abstraction are based on the paradigm in CorGIE [10], which visualizes data items in three different spaces and supports corresponding items between different spaces. A table can be abstracted as a single item with explicit and implicit properties despite holding data in a two-dimensional structure. The explicit properties are the information stored or required in the table repository, such as its table name, column names, annotated descriptive tags, and cell values. Their values and distribution can be visualized in a feature space. The implicit properties are the inherent information of a table that is implicitly extracted by the tag suggestion algorithm developed in my previous project. There are two types of implicit properties: the first one is the representation of a table learned by a deep learning model in a high-dimensional latent space. And the second is the links between related tables that share a common correlated attribute pair in their table headers in a topology space.

I propose another component, the user space, to provide general interactive control for the users. The **specify** task allows users to input a new query table or select a related table recommended by the underlying algorithm that is already in the table repository as focus data items. Additionally, users can also **specify** a recommended potential tag as the focus tag to select all the tables already annotated with that tag in the table repository. The **correspond** task supports visualizing a focus item across different data spaces. However, the correspondence is not always bi-directional. Currently, I propose corresponding items from the user space to all other spaces, between the topology space and latent space, and from all other spaces to the feature space. The feature space is dedicated to pair-wise table comparison with tables selected in other spaces.

The **identify** task in the topology space and latent space allows users to select a table they find interesting in either data space. This task is different from the previously mentioned **specify** task in the user space, as the options in the **identify** task depends on what users **specify** in the user space. For example, if users specify a table in the user space, the **identify** task will be finding tables that are close to the query table in the latent space or in the topology space by having a direct connection with the query table. The **compare** task in three data spaces is to understand the relatedness between the selected table and the query table by providing a pair-wise comparison and to determine whether the tags from the selected table can be applied to describe the query table. All the tasks mentioned above lead to the final **annotate** task: users decide which tags should be augmented to the query table.

4 SOLUTION

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. D3.js [1] is a popular choice for web-based visualizations as it is a JavaScript library that supports dynamic and interactive visualization in web standards. In this project, D3.js and JavaScript will be used to create front-end visualizations and provide the general graphical user interface, while Python will be used for underlying data processing. However, the exact implementation choice is subject to change as the project progresses.

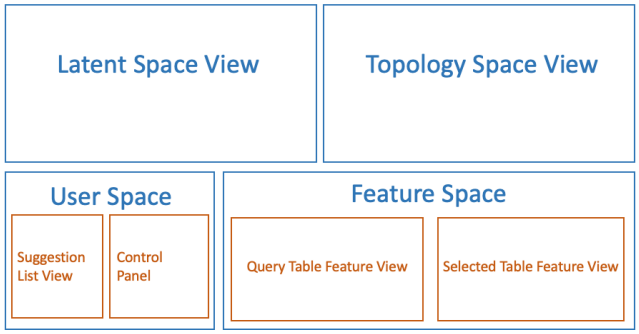

Figure 4: The general interface.

General interface and the user space: Figure 4 depicts the proposed general interface for the iterative descriptive tag annotation tool for table repositories, which comprises four components for the spaces mentioned in the data and task abstraction. The size of each component will be proportional to the need for corresponding visualization views. The user space provides general interactive control for the users. 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 and related tables that are already annotated in the repository will be listed in the

suggestion list view. Users can select a recommended related table as a focus item or a potential tag to include all the tables already annotated with that tag in the repository. The selected table(s) will be visualized in other data spaces with the query table to help users understand how the suggestions are related to the query table and whether the recommended tags can be applied to augment the query table.

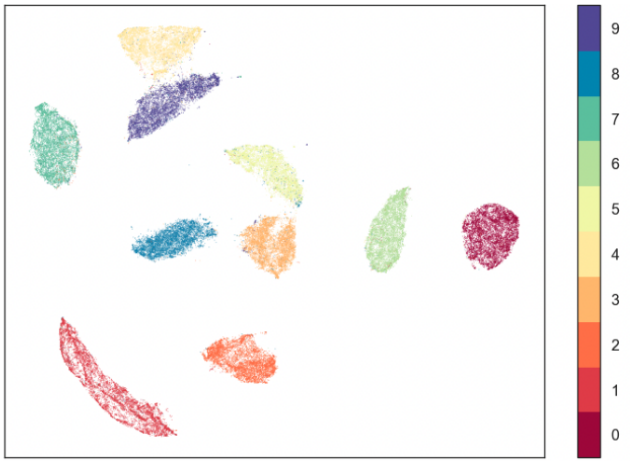

Figure 5: An example of colour coding in UMAP [12].

Latent space view: The underlying tag suggestion algorithm uses a natural language processing model to abstract a table in a high-dimensional latent space for predicting each label's probability. Visualizing the distribution of table representations in the latent space will help users understand semantic relatedness between tables. If the location of the query table is close to an annotated table or a cluster of annotated tables, that means the tables hold similar values. Therefore, it is more likely for these tables to share the same descriptive tags. Similar to CorGIE [10], I propose to use UMAP [12] to project table representations from the latent space into nodes in a 2-dimensional scatter plot.

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 5. However, a table can be annotated with more than one tag in the table repository. Currently, I cannot find a way to colour 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.

Users should be able to directly select a node on the scatter plot to switch to a different **focus table** or **focus tag** to allow for comparing other nodes' distribution with the query table. The newly specified table(s) should also be reflected in visualization views in other data spaces and vice versa.

Topology space view: There are two properties that I propose to visualize in the topology space: the cluster of tables that share a common descriptive tag and the connection between tables that share the same correlated attribute pair of column names, as used in the underlying tag suggestion algorithm.

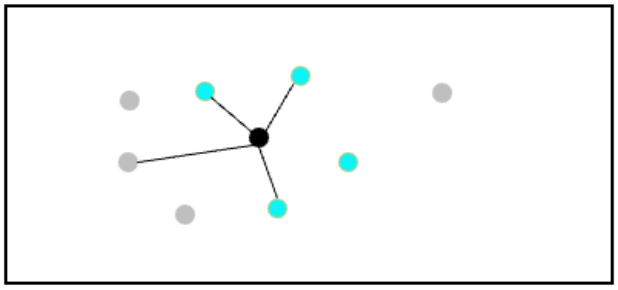

Figure 6: A potential graph view for the topology space. The black node is the query table. Tables with the focus tag are colour coded, while others are in gray.

Figure 6 shows a potential topological graph view that visualizes tables as nodes. With the same colour-coding strategy in the latent space, tables belonging to a focus tag will be coded with a highly saturated colour to make them stand out from other tables in gray. The connections of having the same correlated attribute pair can be visualized as links between nodes. 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. Otherwise, users should be able to select a table linked to the query table, inside or outside the coloured cluster, and switch to a different focus tag from that table.

Feature space view: The feature space is dedicated to the pair-wise comparison between a focus table and the query table. The focus table can only be selected from other spaces and correspondingly visualized in this space. With the pair-wise comparison, users can determine whether the focus table is similar to the query table and which existing tags can be applied to augment the query table. The minimum is to provide a tabular view for each table side-by-side so that users can manually examine the column names and cell values in the two tables. I am still looking for other visualization paradigms that allow pair-wise comparison. I am thinking 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.

5 MILESTONES

Task	Hours	Due date
Continue literature review	10	Oct. 25
Learn d3 and front-end tools	20	Oct. 31
Implement the general interface	10	Nov. 4
Design visualization views	30	Nov. 11
Prepare for peer reviews	5	Nov. 15
Prepare for post-update meeting	5	Nov. 15
Refine the design	30	Dec. 9
Prepare for presentation	10	Dec. 11
Finish the report	15	Dec. 15

6 DISCUSSION & FUTURE WORK

Intentionally left empty.

7 CONCLUSION

Intentionally left empty.

REFERENCES

[1] Mike Bostock. 2021. *D3.js - Data-Driven Documentations.* https://d3js.org/

[2] 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

[3] 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.

[4] 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

[5] 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

[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] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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

[13] 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

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

[15] 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

[16] Michael Oppermann and Tamara Munzner. 2022. VizSnippets: Compressing Visualization Bundles Into Representative Previews for Browsing Visualization Collections. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 747–757.

[17] 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.

[18] 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

[19] 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

[20] 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

[21] 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

[22] 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

[23] 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