VALUE: Visual Analytics driven Linked data Utility Evaluation

Kaustav Bhattacharjee
kb526@njit.edu
New Jersey Institute of Technology
Newark, New Jersey, USA

Aritra Dasgupta
aritra.dasgupta@njit.edu
New Jersey Institute of Technology
Newark, New Jersey, USA

ABSTRACT

The widespread adoption of open datasets across various domains has emphasized the significance of joining and computing their utility. However, the interplay between computation and human interaction is vital for informed decision-making. To address this issue, we first propose a utility metric to calibrate the usefulness of open datasets when joined with other such datasets. Further, we distill this utility metric through a visual analytic framework called VALUE, which empowers the researchers to identify joinable datasets, prioritize them based on their utility, and inspect the joined dataset. This transparent evaluation of the utility of the joined datasets is implemented through a human-in-the-loop approach where the researchers can adapt and refine the selection criteria according to their mental model of utility. Finally, we demonstrate the effectiveness of our approach through a usage scenario using real-world open datasets.

CCS CONCEPTS

• **Human-centered computing → Visualization design and evaluation methods**.

KEYWORDS

open datasets, linking, utility, visual analytics

ACM Reference Format:
Kaustav Bhattacharjee and Aritra Dasgupta. 2023. VALUE: Visual Analytics driven Linked data Utility Evaluation. In *Workshop on Human-In-the-Loop Data Analytics (HILDA '23), June 18, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3597465.3605225

1 INTRODUCTION

The linking of open datasets can create valuable insights for addressing specific problems. For instance, the records of two companies' customers can be combined to identify overlapping records and reveal customers who have patronized both companies. Similarly, the records of police arrests and court proceedings can be merged to extract more comprehensive information about individuals included in both datasets. The open data revolution, founded on the FAIR data principles, has increased the accessibility of such datasets [23]. This growing accessibility can enable researchers to discover new opportunities for joining open datasets to gain deeper insights. However, the open data ecosystem can be considered a forest of datasets, presenting a challenge in leveraging their value through dataset linking. Quantifying the value gained from joining these datasets and selecting dataset pairs with higher utility are complex tasks. Therefore, transparently evaluating the utility of various open dataset combinations has become critically important.

To overcome these challenges associated with joining open datasets, we develop a user-configurable utility metric that expresses the value of pairwise dataset joins based on these datasets' attributes and record space. This metric is then leveraged to develop the VALUE framework and a web-based interactive visual interface, enabling researchers to compare the utility of joinable open datasets and calibrate it. However, manually performing pairwise joins and evaluating their utility can be cumbersome and time-consuming due to the sheer scale and complexity of the combinatorial explosion that arises when dealing with a large number of datasets. For example, with a group of 400 datasets, there can be up to 80,000 potential pairwise combinations, highlighting the need for automating the computing processes to evaluate the utility of these combinations efficiently. While the JOSIE algorithm uses a similar automated approach to identify joinable tables in large data lakes using set similarity techniques, relying solely on automation may overlook valuable insights that can be gained from the user's input and background knowledge, making a human-centric approach necessary [24]. Our approach enables interactive triaging of joinable dataset pairs by human stakeholders (e.g. social science researchers) leveraging the combination of a new utility metric with a visualization interface for distinguishing between the most and the least useful joinable pairs.

In this paper, we first understand the different join scenarios through examples (Section 3). This understanding is then leveraged to contribute the utility metric that can triage the joinable and useful dataset pairs from a large group of datasets (Section 4). Next, we contribute the visual analytic framework VALUE which researchers can use to evaluate the utility of the joined datasets in a transparent manner (Section 5). Finally, we evaluate the algorithm and the VALUE framework through a usage scenario that helps demonstrate their efficacy through real-world datasets (Section 6).

2 RELATED WORK

Evaluating the utility of joined open datasets has been a topic of considerable research for various use cases [3, 7, 22]; however, there is a growing need for developing robust metrics to quantify the usefulness of these joined datasets. Some research works discuss the quality of a dataset based on either the structure of the data or its content and then comment on improving its utility. For example, Ballou et al. first discuss the quality of data based on its completeness and/or consistency [2]. This paper proposes measuring completeness based on the presence of all elements and consistency as uniformity across comparable datasets, followed by

HILDA '23, June 18, 2023, Seattle, WA, USA

ACM ISBN 979-8-4007-0216-7/23/06…$15.00
https://doi.org/10.1145/3597465.3605225

a trade-off analysis between these metrics to achieve the highest possible utility under a budget. However, these metrics alone may not be sufficient to guide the selection of the most useful pair of joinable datasets from a large pool of open datasets.

Several other works have explored the challenge of balancing privacy and utility in datasets. For example, Kenneally and Claffy proposed the Privacy-Sensitive Sharing (PS2) framework to mitigate privacy risks while achieving utility goals when releasing datasets [14]. PS2 consists of components such as authorization, transparency, and access limitations that can help balance the privacy and utility aspects of released datasets. Bhumiratana and Bishop developed an ontology-based framework that enables formal and automatic communication between data collectors and users to ensure privacy-aware sharing of datasets, despite maintaining the utility of these datasets [4]. However, privacy concerns may not always be relevant in evaluating the utility of joined datasets, especially when joining datasets about non-human objects. Moreover, while Noshad et al. proposed the Data Value Metric (DVM) to assess the information content of large datasets for augmentation in specific domains, this approach is limited to evaluating the utility of a single dataset rather than a joined open dataset [16].

Recent research has focused on different approaches for identifying joinable tables in large data lakes. For instance, Zhu et al. developed the JOSIE algorithm, which uses a set similarity search approach with a cost model to enhance performance over large data lakes [24]. However, an entirely automated approach may overlook the nuances of a human-centered approach, which is the focus of our work. Gong et al. developed the Niffler architecture, which finds joinable data tables over pathless table collections without join information [8]. But this approach does not enable the user to triage candidate datasets based on their utility. On the other hand, WarpGate, a semantic join discovery method implemented in Sigma workbooks, first indexes dataset columns and tries to find other datasets with similar columns [5]. However, it provides a score about joinability without a transparent explanation and options for exploration for the reasons behind it, which we attempt to explore through our visual analytic framework. Our work is comparable to the PEXESO framework by Dong et al., which converts the dataset columns into high-dimensional vectors and computes the similarity between these vectors to identify joinable tables [6]. Nevertheless, it does not quantify the utility of joining these datasets, which we attempt to do through the utility metric, which a researcher can transparently evaluate in order to update its components based on their background knowledge and expertise.

3 UNDERSTANDING JOIN SCENARIOS

Understanding the various ways in which two datasets can be joined and the adaptability of a utility metric to different join scenarios is crucial for researchers seeking to gain insights from linking open datasets. Joining can be achieved through intersection, union, master join, or concatenation, each with different implications for the resulting dataset and its utility. The granularity of records, such as individual or aggregated levels, can also impact these join scenarios. In this section, we delve into these different scenarios and how they can influence the utility metric.

age	race	sex	zip
10	Black	M	10012
14	Asian	F	10012
12	White	F	10011

a Dataset D1

age	race	sex	zip	crime
14	Asian	F	10012	larceny
17	Black	M	10013	theft
11	White	M	10021	battery

b Dataset D2

Figure 1: Snapshots of open datasets: (a) Dataset D1 shows the school records while (b) Dataset D2 shows the records of a juvenile criminal activities dataset.

3.1 Intersection join

An Intersection join can be defined as the process of joining two datasets and keeping only those records that have matching values in both datasets for a specific combination of the join key attributes. This is one of the most common types of join encountered, also known as Inner join. Let's see an example of Intersection join.

Suppose we have two datasets, D1 (school records) and D2 (juvenile criminal activity records). A snapshot of D1 and D2 have been shown in Figure 1a and 1b respectively. Joining datasets D1 and D2 based on common attributes age, race, sex, and zip, we observe that there is only 1 common record of age 14, race Asian, gender F and zip 10012 (Figure 2a). We also observe extra information about this individual that this individual has committed larceny. Thus, given the dataset D1, we can follow this process to identify other datasets that can be useful when joined with D1:

- Find datasets that have attributes common with that of D1 (like age, race, gender, and zip)
- Find if the records are similar. Since we need exact matches, we need to find a higher degree of similarity.
- Next, check if there is any other sensitive attribute revealed.

During the analysis of this join scenario, we discovered that record similarity and common attributes play crucial roles in determining the utility metric. It also became apparent that Intersection join is only practical for datasets with similar records, thereby enabling us to recommend pairs of datasets with high utility scores for Intersection join. This also highlights the need to set a defined range for the utility score to classify it as either "high" or "low".

3.2 Master join

Master join can be defined as the process of joining two datasets and keeping the records of either of the datasets and updating values or adding new attributes for those records which have matching values in both datasets, for a specific combination of the join key attributes. It is also known as Left or Right join in the SQL join parlance. This join is mainly useful when we intend to find extra information about the common records between two datasets.

If we perform a Master join on datasets D1 (Figure 1a) and D2 (Figure 1b), we would get an output similar to Figure 2b. Here, all the

a Join Results

age	race	sex	zip	crime
14	Asian	F	10012	larceny

b

age	race	sex	zip	crime
10	Black	M	10012	NA
14	Asian	F	10012	larceny
12	White	F	10011	NA

c

age	race	sex	zip	crime
10	Black	M	10012	NA
14	Asian	F	10012	larceny
12	White	F	10011	NA
17	Black	M	10013	theft
11	White	M	10021	battery

d

age	race	sex	zip	crime
10	Black	M	10012	NA
14	Asian	F	10012	NA
12	White	F	10011	NA
14	Asian	F	10012	larceny
17	Black	M	10013	theft
11	White	M	10021	battery

Figure 2: Results from the Join Scenarios: (a) Intersection join (b) Master join (c) Union join and (d) Concatenation

records from dataset D1 are retained, and the value for the new attribute (i.e., crime) has been updated.

Given dataset D1, the process of finding datasets for Master join is similar to that of Intersection join. Master join is preferred when the datasets have some similar records, and either dataset is selected as the primary one. Though the primacy has to be a user input, considering similarity as an essential constituent of the utility metric, we can say that Master join can be recommended when a pair of datasets have a medium range of utility score.

3.3 Union join

A Union join can be defined as the process of joining two datasets, keeping the records of both datasets and updating values for the common records, for a specific combination of the join key attributes. It is also known as Full join in the SQL join parlance. This join is mainly useful when we intend to keep the records from both datasets but update the values for the common records.

If we perform a Union join on D1 (Figure 1a) and D2 (Figure 1b), we will get an output similar to Figure 2c. Here, all the records from both datasets are retained, and the value for the *crime* attribute has been updated for the common record.

Given dataset D1, the process to find datasets for Union join is also similar to that of the other joins. However, unlike Master joins, Union join does not need a primary dataset since all the records will be retained. During our analysis, we realized that when there is a medium to low similarity between the records of datasets, it could be appropriate to consider a Union join. It is noted here that a Union join can only be performed when datasets have the same granularity. If the granularity is mixed, like having one individual and one aggregated record-level dataset, a Union join wouldn't make sense as it would create a joined dataset with mixed granularity.

3.4 Concatenation

Concatenation can be defined as the process of combining two datasets and keeping all records. Unlike Union joins, no attribute value is updated in this case.

If we perform a concatenation on D1 (Figure 1a) and D2 (Figure 1b), we will get an output similar to Figure 2d. Here, all the records from both datasets are retained as it is.

Given dataset D1, the process of finding datasets for Concatenation is also similar to that of other joins. However, unlike Union join, Concatenation can still be performed if there are some common attributes and no similar records. Thus, a low utility score can indicate a scenario for a Concatenation.

4 CALIBRATING UTILITY

Characterizing the join scenarios helped identify factors that need to be considered for calibrating the eventual utility of the join outcomes. In this section, we first summarize these factors and then we describe the algorithm.

4.1 Key factors impacting utility

Given a dataset D1, we observed that the following factors could be used to quantify the utility of joining it with another dataset:

Shared attributes in a dataset pair: The number of shared attributes between a pair of datasets is one of the important factors for determining the utility of the joined dataset. If two datasets do not share any shared attribute, there is no benefit in joining them through any join.

Degree of similarity between the records of the shared attributes: The degree of similarity can be an indicator of the utility of the joined datasets. We observed that datasets with similar records are useful while performing the joins, while datasets without any similar record can be used for concatenation.

Number of known shared attributes generally used for linking: Through our prior experience, we have observed that certain attributes are commonly employed to join datasets. We start with a list of known attributes like age, gender, race, and location. However, users can update this list based on their background knowledge and expertise. It also serves as a feedback mechanism in our human-in-the-loop approach, thus enabling the user to modify the inputs and transparently evaluate the utility of joining datasets.

While exploring other factors, we hypothesized that the number of exact matches between datasets would determine the join type. However, after conducting some experiments, we found that this hypothesis didn't always hold true. For example, even a single common record between datasets D1 and D2 could lead to a meaningful Intersection join, revealing sensitive information about an individual. Therefore, we decided not to incorporate it as a factor in our algorithm.

4.2 Utility Metric

Algorithm 1 outlines the logic for our proposed utility metric. It is calculated as the weighted sum of three scores: sa_ratio, agl_ratio, and sim_ratio, reflecting the factors we identified as essential in calibrating the utility of joining datasets. Specifically, sa_ratio represents a normalized count of the shared attributes (sa) present while agl_ratio represents a normalized count of the attributes generally used for linking (agl) present in the shared attributes between datasets D_1 and D_2. To ensure consistency, each of these scores has been normalized to return a value between 0 and 1.

sim_ratio quantifies the similarity between the values of the shared attributes of the datasets. If all the values for a shared attribute are numeric, we calculate their cosine similarity using Python's scikit-learn package [17, 20]. However, if the values are categorical, we first generate all possible combinations of string values by selecting each value from records of the categorical attribute

Algorithm 1 Utility Metric Algorithm

Require: Datasets D_1, D_2
Require: User supplied list of attributes generally used for linking (agl)
Require: $cutoffLength \leftarrow 200$
  $f(D_i) \leftarrow$ attributes of D_i
  $sa \leftarrow f(D_1) \cap f(D_2)$
  $sa\_ratio \leftarrow |sa|/\{f(D_1) \cup f(D_2)\}$
  $agl\_ratio \leftarrow (agl \cap sa)/|agl|$
  $simNum, simCat \leftarrow [], []$
  for each $attr$ in sa **do**
    $Z_i \leftarrow dropNA(D_i.attr)$, where $i = 1, 2$ ▷ Keep only values
    if $type(attr)$ = "numeric" **then**
      $Z_i \leftarrow Z_i[: cutoffLength]$, where $i = 1, 2$
      $sim \leftarrow cosineSimilarity(Z_1, Z_2)$
      $AddItem(simNum, sim)$
    else
      $Z_i \leftarrow sort(Z_i, ascending)$, where $i = 1, 2$
      $Z_i \leftarrow Z_i[: cutoffLength]$, where $i = 1, 2$
      $C \leftarrow$ all Z_1-Z_2 combinations with one element from each
      $temp \leftarrow []$
      for each $comb$ in C **do**
        $sim \leftarrow InDelSimilarity(comb[0], comb[1])$
        $AddItem(temp, sim)$
      end for
      $simMean \leftarrow Mean(temp)$
      $AddItem(simCat, simMean)$
    end if
  end for
  $sim\_ratio \leftarrow Average(Mean(simNum), Mean(simCat))$
  $w \leftarrow [20, 20, 60]$ ▷ Weights
  $UtilityScore \leftarrow (w[0] * sa\_ratio) + (w[1] * agl\_ratio) + (w[2] * sim\_ratio)$

of each dataset. Then we calculate the similarity between each combination string using normalized InDel similarity from Python's Levenshtein package [1]. InDel distance is an edit distance between two strings that calculates the number of insert/delete operations required to convert one string to another. The time complexity is $O(m * n)$, where m and n are the number of characters in each string. This distance is then normalized over the maximum possible distance between two strings of size m and n, respectively. The normalized InDel similarity is then calculated as 1 − (normalized InDel distance). Finally, we compute the average of the categorical and numerical attributes' similarities to arrive at sim_ratio.

We use an edit distance-based similarity calculation method for finding the similarity between each record string. This method is preferable over token-based or sequence-based similarity calculations since the order of records does not affect our results significantly. We have considered several candidate algorithms for calculating the similarity between strings, including Levenshtein [15], InDel [11], Jaro-Winkler [13], and Hamming distance [10]. Hamming distance overlays one string over another and finds the number of places where the strings vary. While this method is effective for comparing strings of equal length, it is not well-suited for our purposes since the strings in our datasets can vary in length. Levenshtein distance calculates the number of operations (insert/delete/substitution) required to convert one string to another. Jaro-Winkler distance is similar to Levenshtein, but the substitution operation for close characters is given less weightage than that of far characters. InDel is a similar algorithm, but only insert and delete operations are allowed. We decided to use the InDel algorithm for string similarity calculation over Levenshtein and Jaro-Winkler algorithms. This choice was based on the fact that, in our current context, the substitution of characters may not be a reliable indicator of the level of similarity or difference between two strings of various types.

The final $UtilityScore$ is the weighted sum of these ratios, where more weight is given to the similarity between the attribute records. This score ranges between [0,100], thus making it easier to categorize high and low similarity, implementing the insights gained while characterizing the join scenarios (Section 3). For computational efficiency, we have set a cutoff limit of 200 records for columns while calculating their similarity. Though this does not affect smaller datasets, for larger datasets, we can remove this constraint based on the availability of computational resources.

5 FRAMEWORK FOR TRANSPARENT EVALUATION OF UTILITY

The algorithm for the utility metric can be best evaluated when paired with visual analytic interventions that a researcher can use to explore different open datasets and the utility of joining them. In this section, we first define the tasks for the VALUE framework and then discuss the visual analytic solution required to implement this framework on a web-based interface.

5.1 VALUE framework

The foremost challenge while assessing the utility of joining open datasets is to compare and triage different dataset pairs based on the utility metric. After that, researchers need to update the metric by considering their background knowledge, expertise, and analysis of the joined datasets. Centered around these steps, the tasks of the VALUE framework are as follows:

***T1**: Inspecting utility scores*: The joinable groups of datasets can be further analyzed by ranking each pairwise dataset combination according to their utility score. This task relates to triaging dataset pairs based on their utility score. A dataset pair with a higher utility score will be more useful when joined based on some common attributes than one with a lower utility score. By identifying the most useful dataset pairs, researchers can focus their efforts on those with the highest potential for generating meaningful insights.

***T2**: Incorporating user inputs to utility score*: The utility score of a joined dataset is influenced by the attributes commonly used to link two datasets, and this evaluation can benefit from a human-in-the-loop approach. While we begin with a preliminary list of such attributes, a researcher can supplement this list based on their background knowledge and expertise. This task relates to the modification of the list of attributes generally used for linking, which can affect the utility score and ultimately lead to a meaningful join operation. By involving human expertise and feedback, we can ensure that the list of attributes generally used for linking

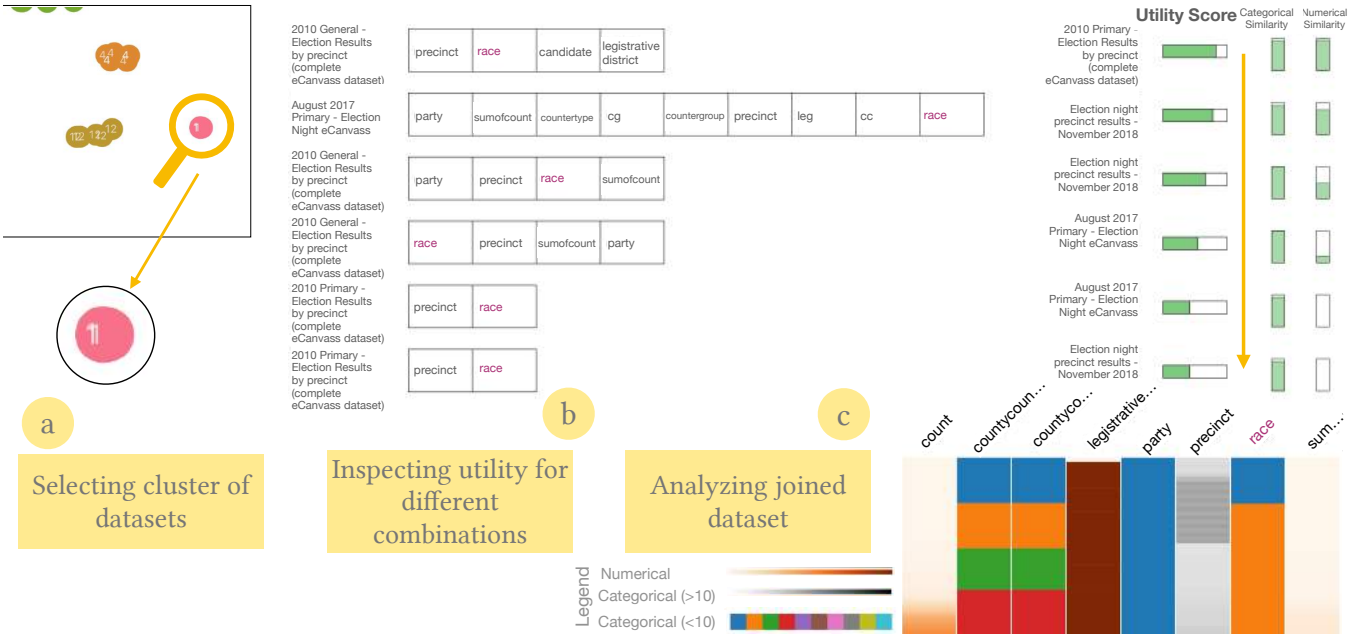

Figure 3: Inspecting utility of joining real world open datasets through the VALUE interface: (a) A researcher selects a cluster of joinable open datasets based on relevant keywords. (b) Then all possible pairwise combinations of datasets are presented for the transparent inspection of the utility scores. Dataset pairs are ranked based on the utility score, and the user-selected attribute (*race*) present in the common attributes is highlighted for each pair. (c) Finally, the researcher can join the most useful pair and analyze the result through color-coded record categories. Numerical attributes are colored through an orange interpolation, while categorical attributes with less than ten categories are assigned distinct colors, and those with more than ten categories are colored through a grey interpolation.

is comprehensive and effective in capturing the most important aspects of the data.

***T3**: Analyzing joined records*: After joining the datasets, a researcher can perform a detailed analysis of the joined records to determine their utility. This task is necessary to extract valuable insights from the linked data and is essential for the success of a human-centered linked data analysis framework.

5.2 Visual analytic solution

The initial objective of the VALUE framework is to identify groups of joinable open datasets, and it is accomplished using two key visual analytic components. The first component is a search box that allows the user to filter datasets based on relevant keywords. The second component is a high-dimensional projection of the datasets based on their similarity in their attribute space (Figure 3a). In order to achieve this, we first transform the dataset attributes into high-dimensional word embedding vectors. These vectors are then projected onto the 2D space using the t-SNE dimensionality reduction algorithm [21]. Then we apply the DBSCAN algorithm to identify and group datasets with similar attributes into clusters [19]. Datasets that belong to the same cluster are color-coded for easy identification. Furthermore, each cluster is ranked based on its intra-cluster distance using the Silhouette coefficient [18], and individual datasets within the cluster are labeled accordingly. This approach allows the researchers to comprehend the relationships between datasets and identify joinable groups.

Once a researcher selects a group of joinable datasets, all possible pairwise combinations of datasets are displayed for further inspection (Figure 3b). Each dataset pair is visually represented using a combination of items, such as the dataset names, rectangular boxes showing the common attributes between these datasets, and their utility score. The utility score is represented using a horizontal green bar where the color green represents the score in a range of 0-100. This abstraction provides a convenient way for the researchers to understand the scores at a glance, but they can also obtain exact score information by hovering over the bar. The choice of the color green is purely for semantic reasons. The similarity between the records of common categorical and numerical attributes is also shown with two vertical green bars. These bars' orientations have been reversed to differentiate them from the main utility score. Thus, this design aids the transparent evaluation of the utility scores **(T1)**. Furthermore, this view also enables the researcher to augment the list of the attributes generally used for joining. If any of these attributes are present in the common attributes, they are highlighted in a distinct color (royal heath) to indicate their significance. This human-in-the-loop approach helps to improve the utility score based on the inputs from the researcher **(T2)**.

As learned during the characterization of join scenarios, we recommend Intersection join for dataset pairs with high utility scores.

To facilitate this, the button for Intersection join is highlighted, but the researcher has the option to choose any other type of join. The joined datasets are visualized through a customized Navio implementation, where each attribute is represented by a stacked bar chart displaying the distribution of different categories for that attribute (Figure 3c) [9]. For a numerical attribute, the records are represented using a sequential scheme of colors. The null values, shown in light pink, help to understand the completeness of the results. This colored categorization of the joined dataset's records helps a researcher understand its composition and analyze them for utility **(T3)**. Users can also download the joined dataset for further investigation. The web-based interface has been developed using a combination of Python and Flask for the backend and Node.js, React.js, and JavaScript for the frontend.

6 USAGE SCENARIO

The performance of the algorithm for utility metric and the VALUE framework can be evaluated in multiple ways. A systematic review by Isenberg et al. observed that Qualitative Result Inspection is one of the most popular evaluation methods for algorithms and visualization interfaces [12]. Hence, in this section, we describe a usage scenario to demonstrate the how the visual analytic interface that embeds the utility metrics can help in distinguishing between the highly usable and the least usable pairwise join outcomes.

Consider a scenario where a researcher at a government laboratory is analyzing local election results obtained from open datasets to gain insights that could inform policy decisions or contribute to a broader understanding of the political landscape in the area. The findings of the study could be crucial for stakeholders such as policymakers, government agencies, or local communities in formulating informed decisions. She began by browsing several county-level open data portals to obtain the necessary data. However, she found it challenging to determine which datasets to combine to form a complete picture of the election results. In search of a solution, she turned to the VALUE interface. After conducting a search on elections, the interface generated several clusters of data related to election results. To make her selection, she carefully analyzed the projection plot and ultimately chose the first cluster (Figure 3a).

This action generated all the possible pairwise dataset combinations from this cluster and ranked them according to their utility score (Figure 3b). The researcher analyzed the dataset pairs and observed that the datasets *2010 General - Election Results by precinct (complete eCanvass dataset)* and *2010 Primary - Election Results by precinct (complete eCanvass dataset)* have a high utility score of 82.77 **(T1)**. These are the datasets for the general and primary election results of 2010 from King County, WA. Since this pair has a high utility score, the interface suggested an Intersection join between the datasets. She also observed that this pair included one attribute (race) that was included in the default list of generally used attributes **(T2)**. Though she did not update this list, she selected all the attributes and performed an intersection join.

On joining these datasets based on all the common attributes, the researcher observed that the joined dataset contains 162, 977 records (Figure 3c). She analyzed these records using the VALUE interface and understood that the joined dataset gives her the combined election results for all the candidates at each precinct, both at the primary and general election levels **(T3)**. She further downloaded the joined dataset and saved it for her research purposes.

Further, the researcher was curious to understand if the utility metric could distinguish between the most useful and the least useful dataset pairs. Hence, she selected the lowest ranked dataset pair: *2010 Primary - Election Results by precinct (complete eCanvass dataset)* and *Election night precinct results - November 2018*. Joining them based on the common attributes ['race', 'precinct'] yielded no record. Thus, the researcher concluded that the utility metric, when used in conjunction with the VALUE framework, can help to find joinable and useful datasets from the open data ecosystem.

7 DISCUSSION

The utility metric can be considered a novel method that can be used to assess the utility of joining open datasets with a human-centric perspective. In our continuous efforts to improve and refine the algorithm behind the utility metric, we aim to unlock even greater insights into the potential of joining open data. We also plan to use the outcomes from the utility metric to train a machine-learning model to classify the usefulness of the joins.

The insights gained from our analysis of the join scenarios represent a crucial foundation for this work. By leveraging the interface for the VALUE framework, we could put some of these lessons into practice, emphasizing the critical role of visual analytic interventions in solving this problem. Although the current interface prototype is designed to work with approximately 400 open datasets, our internal testing has indicated that it can be scaled up significantly. Also, while we did need to implement a cutoff length for larger datasets, we are currently exploring strategies to overcome this limitation, such as increasing our computational resources. Additionally, we are also working on a workflow that can regularly fetch datasets from different sources and integrate them with the VALUE framework, thus enabling us to keep pace with the ever-evolving landscape of open data.

We recognize that there is always room for improvement in the interface components of the VALUE framework, and we are committed to incorporating feedback from a diverse range of users. To this end, we plan to conduct case studies with domain experts and undertake more controlled user studies that will enable us to collect valuable feedback about the interface and the algorithm.

8 CONCLUSION

The utility metric algorithm, presented in this paper, is a first step towards quantifying the utility of joining open datasets. It considers multiple factors like the similarity between records, shared attributes, and a user-supplied list of attributes to develop a score that can help identify the most useful pair of datasets from a group of joinable datasets. The lessons learned during this development also helped develop the VALUE framework, which, when used in conjunction with the web-based interface, helps in the transparent evaluation of the utility score. This human-in-the-loop approach helps researchers, data scientists, and analysts to make more informed decisions and leverage the full potential of open datasets.

REFERENCES

[1] Max Bachmann. 2022. The Levenshtein Python C extension module contains functions for fast computation of Levenshtein distance and string similarity.

https://github.com/maxbachmann/Levenshtein. (Accessed on 03/27/2023).

[2] Donald P. Ballou and Harold L. Pazer. 2003. Modeling completeness versus consistency tradeoffs in information decision contexts. *IEEE Transactions on Knowledge and Data Engineering* 15, 1 (2003), 240–243.

[3] Kaustav Bhattacharjee, Akm Islam, Jaideep Vaidya, and Aritra Dasgupta. 2022. PRIVEE: A Visual Analytic Workflow for Proactive Privacy Risk Inspection of Open Data. In *2022 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE, IEEE, Oklahoma City, OK, USA, 1–11.

[4] Bhume Bhumiratana and Matt Bishop. 2009. Privacy aware data sharing: Balancing the usability and privacy of datasets. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, Corfu, Greece, 1–8.

[5] Tianji Cong, James Gale, Jason Frantz, HV Jagadish, and Çağatay Demiralp. 2023. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses. In *13th Annual Conference on Innovative Data Systems Research (CIDR '23)*. CIDR, Amsterdam, The Netherlands, 1–7.

[6] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, Chania, Greece, 456–467.

[7] Marie Douriez, Harish Doraiswamy, Juliana Freire, and Cláudio T Silva. 2016. Anonymizing NYC Taxi Data: Does It Matter?. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, IEEE, Montréal, Canada, 140–148.

[8] Yue Gong, Zhiru Zhu, Sainyam Galhotra, and Raul Castro Fernandez. 2021. Niffler: A reference architecture and system implementation for view discovery over pathless table collections by example. *arXiv e-prints* NA, NA (2021), arXiv–2106.

[9] John Alexis Guerra Gómez. 2021. Navio | A d3 visualization widget to help summarizing, exploring and navigating large network visualizations. https://navio.dev/. (Accessed on 03/10/2023).

[10] Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal* 29, 2 (1950), 147–160.

[11] Heikki Hyyrö, Yoan J Pinzón, and Ayumi Shinohara. 2005. New bit-parallel indel-distance algorithm. *Lecture notes in computer science* 3503 (2005), 380–390.

[12] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. 2013. A Systematic Review on the Practice of Evaluating Visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2818–2827.

[13] Matthew A Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. Amer. Statist. Assoc.* 84, 406 (1989), 414–420.

[14] Erin Kenneally and Kimberly Claffy. 2009. An internet data sharing framework for balancing privacy and utility. In *Engaging Data: First International Forum on the Application and Management of Personal Electronic Information*. IEEE, Cambridge, MA, USA, 1–8.

[15] VI Levenshtein. 1965. Binary codes with correction for deletions, insertions and substitutions of characters. *Reports of USSR Academy of Sciences, 163.4: 845* 163 (1965), 845–848.

[16] Morteza Noshad, Jerome Choi, Yuming Sun, Alfred Hero III, and Ivo D Dinov. 2021. A data value metric for quantifying information content and utility. *Journal of big Data* 8, 1 (2021), 82.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[18] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[19] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* 42, 3 (2017), 1–21.

[20] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge, Munich, Germany.

[21] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008), 2579–2605.

[22] Theo Vos, Stephen S Lim, Cristiana Abbafati, Kaja M Abbas, Mohammad Abbasi, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Hedayat Abbastabar, Foad Abd-Allah, Ahmed Abdelalim, et al. 2020. Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: a systematic analysis for the Global Burden of Disease Study 2019. *The Lancet* 396, 10258 (2020), 1204–1222.

[23] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9.

[24] 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*. ACM, Amsterdam, The Netherlands, 847–864.