Experiments & Analyses Paper

Analysis of Open Government Datasets From a Data Design and Integration Perspective

Arif Usta
arif.usta@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Chang Liu
c.liu@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Semih Salihoğlu
semih.salihoglu@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

Open governmental data portals (OGDPs) publish large amounts of datasets on many aspects of their countries with the goal of improving transparency and making it easier for journalists, researchers, and the general public to identify societal problems, such as spending waste or health risks. This paper studies the core properties of the datasets in four large OGDPs that publish in English: Canada, Singapore, UK, and US. Our study reveals several important findings, such as the extent of value repetition, lack of key columns, and prevalence of functional dependencies, all indicative of high levels of denormalization in these tables. We also find that overwhelming majority of joinable tables are accidental and offer several other properties of tables that can be used as signals to filter useful joinable tables. We further document the patterns we saw across useful and accidental joinable and unionable table pairs. We hope these findings can guide researchers and developers of data systems on OGDPs about the core properties of these datasets.

1 INTRODUCTION

The launch of open governmental data portals (OGDP), such as data.gov, open.canada.ca, or data.gov.in, has popularized the open data movement of the last decade. These portals publish large numbers of datasets related to a very wide range of topics, such as how countries distribute research funds, how much meat they export, their CO_2 emissions, or daily COVID-19 cases. The overarching vision of OGDPs is to make governments transparent so that journalists, policy analysts, researchers, and the general public can easily monitor how their societies are functioning.

Although the amount of datasets in OGDPs are increasing, achieving this vision requires developing additional data tools and applications over these datasets to discover, understand, link, and integrate them. Excitingly, these are some of the core research problems that interest the database community, and as such OGDPs have become some of the most studied data repositories (aka *data lakes*) [7, 19, 20, 24, 34, 35].

This paper studies the relational structures of the tabular datasets in OGDPs how and when they can be integrated using the common approaches from literature. Understanding these broad questions can inform researchers and developers working on search and data integration systems on these systems. For example, several dataset search and integration systems, such as Toronto Open Dataset Search [36], Auctus [11], and Governor [23], implement algorithms to search and suggest joinable tables. These systems use value-based metrics to make these suggestions, yet it is not known whether when these suggestions lead to useful vs accidental joins. It is therefore important to understand whether and when the join pairs that have high-overlap lead to useful joins? For example, should value-based metrics be complemented with other properties of columns, e.g., data types, to suggest better joinable pairs? These systems further often ignore whether joins grow or not when making their suggestions. How large are the output tables when joining tables with highly overlapping values and is this an important signal for identifying accidental joins? It is also important to know how normalized these tables are. If the published tables are highly denormalized, this can be an indication that the base tables are in fact joins of smaller valuable sub-tables, which can be important sources of information for their users.

In light of this, we pose and answer several concrete questions in this paper, such as: *What is the current size and growth rate of these portals? How sparse or normalized are these datasets? What is the extent of value repetitions? What fraction of the datasets have metadata files and in what formats? Do the common approaches used in literature for identifying tables that can be integrated work in practice, e.g., using value overlap-based approaches to identify joinable tables?* We focus on tabular datasets, which are the most common data formats in OGDPs, and cover four OGDPs that publish in English: Canada [1] (CA), Singapore [2] (SG), UK [3], and USA [4] (US). Our use of 4 portals also allows the properties we study to be compared across portals, which can further inform publishers of these datasets about rooms for improvement and better publishing practices.

In what follows, we list main findings of the analysis:

General Statistics:

- Data sizes: Portals are quite small in size. In March 2022, the largest portal was US with a cumulative size less than 1.9TB in uncompressed format and 433GB in compressed format and the rest of the portals are a fraction of that and this trend is unlikely to change. Therefore academic groups can easily develop systems that index and process entire repositories.
- Null values: Nulls are ubiquitous. Except in SG, half of the columns have null values. In some portals, up to 23% of all columns have at least half of their values as null, and 3% of the columns across the portals are entirely null.
- Metadata files: Metadata files that provide descriptions of columns are common, yet are almost always in unstructured formats that are hard to automatically process (except for SG).

Keys and Normalization Study: We next studied the key columns and how denormalized the tables are. Our findings are:

- Extent of value repetition: Theres is a surprisingly high level of value repetition. For example, in US, the median number of values and unique values across columns is 447 and 30, respectively, with similar levels of repetitions in other portals.

- Lack of keys: A large fraction of tables do not have a single key column. For example, 48% and 36% of the tables in, respectively, SG and CA do not have a key column that identifies tuples. 10% of the tables across all portals do not even have a composite key consisting of 3 or fewer columns. Therefore many records cannot easily be identified through standard key-based techniques to link them with other records.
- Prevalence of FDs: Overwhelming majority of the tables, up to 84% in some portals, have potential non-trivial FDs. This results in data redundancies. More importantly, these tables contain useful sub-tables for users that systems can aim to expose to users through automatic normalization.

Table Integration Analysis: Our first question here was: How useful, i.e. meaningful and interpretable, are the joinable table pairs, which are found using the standard technique of high value overlapping columns [34, 35], and in which cases are they useful? We first analyzed several metrics, such as the sizes of the output joins, and then manually labeled a large sample of joinable table pairs as accidental vs useful. Our main findings are:

- Prevalence of nonkey-nonkey joins: Between 46.6% to 66.4% across portals have at least one other table on a very high value-overlapping column. Yet these columns are overwhelmingly non-key columns and lead to very large output sizes. Hence, they are likely to be accidental or not very useful.
- Common properties of useful joins: Based on manually labeling a large sample of pairs, we observe that value-based joins are likely to be useful when the tables are from the same dataset and are on on key columns that have types other than incremental integers, such as categorial, string, or geospatial. These properties are important signals for identifying useful joinable tables in data integration systems.

We further analyzed a sample of unionable tables based on high schema overlap, which is a common metric used in literature [7, 12, 23]. Here we found overwhelming majority of the unionable pairs are interpretable. However, we also identified and reported several publication patterns where the pairs have same schemas but are accidentally unionable. We made both manually labelled table pairs and associated code to reproduce our analysis public[1]. Our manually labeled table pairs, which are in our repo, can be used as a ground truth benchmark for future research on techniques for suggesting joinable and unionable tables.

Our analysis raise several interesting research questions that we hope can be valuable to researchers who work on OGDPs such as how to automatically find accidental vs real FDs. We hope our survey and findings can be informative for researchers working on OGDPs as well as publishers of these datasets.

2 BACKGROUND

2.1 Overview of Dataset Publishing in OGDPs

Many governments utilize a content management system named CKAN[2], which follows a certain structure for the collection of data to be published. The data published in OGDPs are stored under *datasets*. In other words, an OGDP is a set of datasets $D = \{d_1, d_2, ..., d_n\}$, where d_i is the i^{th} dataset and n is the number of datasets. An example dataset is *NSERC's Awards Data*[3] from Canada's open data portal. Each dataset d_i contains a set of *resource files* $F^i = \{f_1^i, f_2^i, ..., f_m^i\}$, where f_j^i stores the actual data for dataset d_i. Each d_i can have any number of resources possibly in different formats such as html, pdf, csv, and etc. In this paper, we focus on resource files in comma separated values (CSV) format, which is one of the most common formats in OGDPs.

2.2 Experimental Setup

To fetch the CSV files from the ODPs, we first fetch all available metadata of the portal with CKAN REST API of the portal. Then, we use the *format* property of the metadata to identify the CSV files and download them from the *url* provided in the metadata files with an HTTP client. If the HTTP request succeeds with *HTTP Status 200*, we categorized it as *downloadable*. We then process each downloadable file through the following pipeline:

- Type of the downloaded file is detected with *libmagic*[4], which is a pre-built function for ubuntu operating system, to ensure that it is actually a CSV file.
- The header row of the CSV file is determined by the header inference algorithm, which is based on simple yet effective heuristics. We take the first 500 rows to determine the number of columns and pick the first row with no missing value as a header. We randomly picked 100 tables from each portal and evaluated the accuracy of this algorithm, and found it highly accurate: 100% on SG, 93% on CA, 96% on UK, 97% on US.
- After determining the header row of the tabular data for each CSV file, we parse raw data inside the files using *pandas*[5].

If all of the steps above succeed, we categorize the file as *readable* and discarded it otherwise. Table 1 shows the statistics for readable CSV files for each portal. Beside these steps, we performed two additional cleaning steps:

- Several tables had sequences of entirely empty columns at the end of their column lists, which we removed.
- We also observed several very wide tables across the portals (except in SG), many of which had publication errors which lead to repeated periodical columns (e.g., k columns are repeated hundreds of times). Some others were transposed. In general, wide tables have more missing values and many were malformed, so we removed them from our analyses not to skew the results. To do so, we decided to use a cutoff point that would remove a very small fraction of tables from the dataset but would ensure that the very wide malformed tables are removed. We used a cut-off point as 100 columns, which lead to removing 204 (1.4%), 1690 (4.8%), 559 (2.1%) tables for CA, UK and US, respectively.

3 GENERAL CHARACTERISTICS OF OGDPS

In this section we first present statistics about the sizes of these portals. We then analyze the frequency of null values. Finally, we analyze the presence/absence of metadata/dictionary files. Part of our analysis here is done to perform a complete study on the measurable structural properties of these datasets, such as table sizes. However, some properties we analyze can be informative for researchers and even raise interesting research questions.

[1] https://github.com/arifusta/ogdpAnalysis
[2] https://ckan.org/
[3] https://open.canada.ca/data/en/dataset/c1b0f627-8c29-427c-ab73-33968ad9176e

Table 1: Portal size statistics.

	Portal			
	SG	CA	UK	US
total # datasets	1898	30348	51190	335221
avg # tables per dataset	1.82	3.27	5.35	1.51
max # tables per dataset	38	252	326	550
total # tables	2399	36373	78146	46155
total # downloadable tables	2376	14985	35193	26503
total # readable tables	2376	14913	34901	26416
total # columns	12428	352223	1128355	571942
Total size in GiB	1.48	49.78	180.22	1933.89
Total compressed size in GiB	0.26	6.15	36.23	433.69
Size of largest table in GiB	0.13	1.84	1.33	107.49

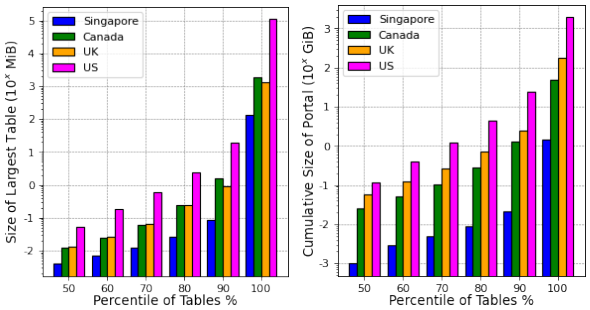

Figure 1: For each percentile in increasing order of size, plots the cut-off table and cumulative portal sizes.

3.1 Portal Sizes

Main statistics on the sizes of OGDPs are provided in Table 1. US is the largest portal in terms of its size, but even its raw format takes less than 1.9TB and only 433GB if we compress the tabular files. The rest of the portals are a fraction of US's portal in size. For example, the second largest portal UK is 180GB/36GB in uncompressed/compressed format. In addition, there is large skew in the table size distributions. In Figure 1, shows the distribution of table sizes. If we ignore the top 10% of the largest tables, then even the US portal reduces to 24GB in uncompressed format and others reduce to smaller than 2.4GB[6]. We also observed that the tabular data in these portals are highly compressible. The high compression rates are already an indication of high repetitions due to potential functional dependencies, which we will be analyzed later. Table 1 presents the compressed sizes of each portal. We see a 1:5 compression ratio on the average across the portals using a standard data compression library *Bandizip*[7].

We further did an analysis of the growth rates of the portals' data sizes. We analyzed the publication dates of the datasets and plotted the size of the portals in their last 5 years. We were able to do this analysis satisfactorily only for UK as other portals seem to have ingested bulk updates on certain dates that give step function-looking curves. Figure 2 shows UK's growth.

[4]https://packages.ubuntu.com/bionic/libmagic-dev
[5]https://pandas.pydata.org/
[6]We inspected the largest table across all corpora, which was a dataset from the US portal with size 108GB on park cleaning records from NYC Open Data
[7]https://en.bandisoft.com/bandizip/

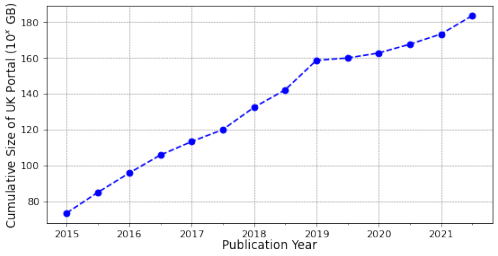

Figure 2: Annual growth of cumulative size of UK portal.

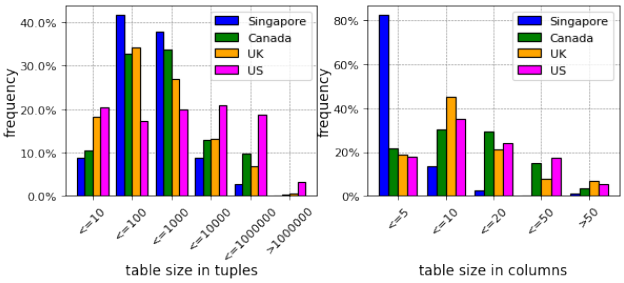

Figure 3: Distribution of table sizes in each portal in terms of number of tuples (left) and columns (right).

Table 2: Table size statistics of OGDPs.

	Portal			
	SG	CA	UK	US
avg # columns per table	5.23	23.55	32.33	21.65
median # columns per table	4	10	9	10
max # columns per table	94	6304	16384	1418
avg # rows per table	4.2K	20.7K	42.8K	518.5K
median # rows table	95	148	86	447
max # rows per table	1.9M	25.4M	10.3M	409.2M

Main Observation: OGDPs are currently fairly small in size and highly compressible. Importantly, academic groups can easily do studies that process entire portals on disk or in memory. As seen in Figure 2, their growth rate seems slow, which indicates that their sizes are likely remain small in the near future as well.
Other observations: We note that although US has the most number of datasets, UK has the most tables. This is due to publication style differences across portals: US publishes primarily 1 table per datasets, whereas other portals often publish multiple tables per dataset (e.g., over 86% of datasets in Canada have multiple tables). Finally, except in SG, there are many tables in these portals than what can be downloaded. For example in Canada, only 41% of all tables are downloadable, though once downloaded almost all are processable, i.e., has automatically parsable headers and values. By fixing these technical problems, some of these portals can potentially make twice as much data available to the public.

3.2 Table Sizes

Table 2 shows the main statistics in terms of column and row sizes. Figure 3 shows the distribution of the number of columns and rows of the tables in these portals. The average number of columns varies between 5 (SG) and 32 (UK). The medians are lower and varies between 4 (SG) and 10 (CA and US). Overall we

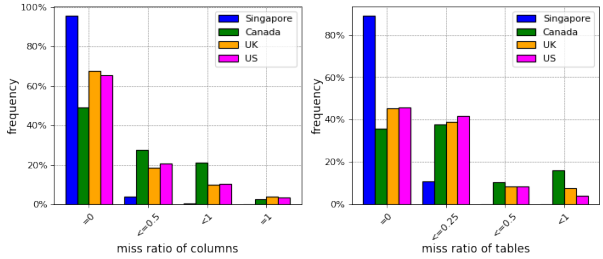

Figure 4: Null value ratios of columns and tables.

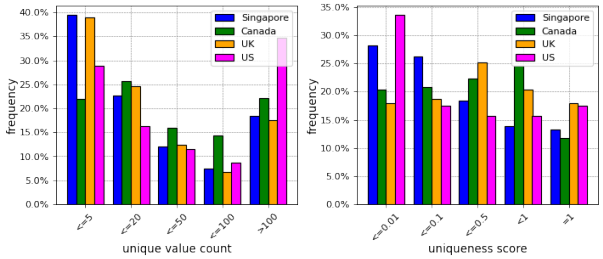

Figure 5: Unique value count and uniqueness score distributions for columns across portals.

Table 3: Distribution of metadata file availability.

Portal	Metadata Presence			
	having			lacking
	structured	unstructured	outside portal	
SG	100%	0%	0%	0%
CA	4%	8%	29%	59%
UK	4%	5%	3%	88%
US	0%	0%	27%	73%

find that the tables in SG consistently have very few columns: more than 80% in SG have 5 columns at most and more than 95% have 10 or fewer columns. More than 95% of tables across all portals have less than or equal to 50 columns. The average number of rows varies between 4.2K (SG) and 518K (US) but the median varies only between 95 (SG) and 447 (US). This indicates that a few very large tables moves the average significantly. Majority of tables across all portals have less than 1000 rows.

3.3 Null Value Analysis

We analyzed the prevalence of null values across tables by searching for empty cells in the CSVs as well as a manual list of popular values that are used for nulls; namely "n/a", "n/d", "nan", "null", "-", and "...". We refer to null ratio in a column/table as the fraction of values that have nulls in that column/table. The distribution of null ratios of columns (left), and average null ratios for tables (right) across portals are shown in Figure 4. Apart from SG, where 95% of the columns have no null values, half of the columns have at least 1 missing value across the portals. 23%, 13% and 13% of all columns and 16%, 7% and 4% of all columns are more than half empty for CA, UK and US, respectively. Interestingly, excluding SG, 3% of the columns are entirely empty across the other portals on average. *This indicates that even some basic cleaning steps are likely not being done before publishing tables in some portals.*

3.4 Metadata/Dictionary File Analysis

Metadata files/data dictionaries, which describe the columns of tables, are important for users to understand the published data. Publishing them in automatically convertible structured formats can allow data integration systems to process and use them in their interfaces automatically. We first sampled 100 datasets from each portal at uniformly random and manually checked whether these datasets have any metadata files and whether these are in a structured, e.g., CSV files, or in unstructured files such as pdf or a separate webpage. If the files are in a separate webpage, we consider them structured if the webpages have a consistent format across the portal. Otherwise we consider them unstructured.

Table 3 shows our results. In SG, every dataset has an associated structured webpage that contains metadata information. For other portals, frequency of metadata presence ranges between 12% (UK) and 41% (CA), although almost all available metadata files are in unstructured formats. Understanding the meaning of the records and values is critical for users to extract value from these datasets. As such, data systems built on top of OGDPs should expose and make it easy for users to access data dictionaries in their interfaces. In light of this, we think research on automatically extracting data dictionaries from the vast unstructured resource files in OGDPs is an important research topic.

4 VALUE REPETITION ANALYSIS

Next, we first provide statistics about unique vs all values in columns. Then we examine the single key columns, i.e., those that have no value repetition, or composite keys in tables. We then look for potential functional dependencies (FDs) and present several properties before and after normalizing the tables.

4.1 Uniqueness and Key Column Analysis

For a column c, let c's *uniqueness score* be $\frac{|set(c)|}{|c|}$, which is the ratio of the number of unique values vs number of values in c, i.e., the row count of the table c belongs to. Figure 5 shows the distributions of number of unique value counts and uniqueness scores of columns. Table 4 provides statistics on unique values and uniqueness scores when grouping columns according to two broad data types: text and numeric.

Main Observation: There is a very high degree of value repetition across all portals, especially in text columns. While the median number of values across columns are 447, 86, and 95, and 148 (recall Table 2), the median number of unique values are only 30, 10, 10, 23 in US, SG, UK, and CA, respectively. 51% and 41% of the columns in US and CA, respectively, have smaller than 0.1 uniqueness score, so their values are on average repeated more than 10 times. Text columns have much more data repetition than numeric columns (see Table 4), e.g., in US, the median number of unique values across text and numeric columns are 14 and 55, respectively.

We next analyze the distributions of key columns. A column c with uniqueness score of 1.0 is a key column. Key columns are desirable as they help identify a table's records. Furthermore, in data integration, joins of two tables on two key columns lead to non-growing joins, which are desirable as they effectively extend these tables with additional columns. For those tables that do not have a key column, we searched for all possible 2-size and 3-size candidate keys. The distribution of the minimum candidate key columns of the tables are depicted in Figure 6.

Table 4: Uniqueness statistics of columns in OGDPs.

	Portal											
	SG			CA			UK			US		
	text	number	all	text	number	all	text	number	all	text	number	all
# columns	7695	4731	12426	77801	111025	189358	254030	131762	385849	179931	131762	362099
avg unique value per column	2.0K	278	1.3K	1.8K	1.5K	1.6K	1.2K	3.2K	1.9K	22K	33K	28K
median unique value per column	5	24	10	15	35	23	8	12	10	14	55	30
max unique value per column	1.9M	72K	1.9M	14M	6.8M	14M	3.9M	4.8M	4.8M	400M	213M	400M
avg uniqueness score per column	0.14	0.56	0.30	0.32	0.42	0.37	0.36	0.53	0.41	0.26	0.44	0.41
median uniqueness score per column	0.02	0.63	0.07	0.10	0.31	0.20	0.18	0.53	0.27	0.03	0.27	0.09

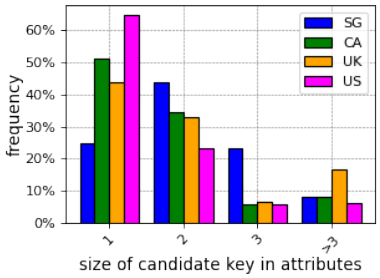

Figure 6: Distribution of minimum candidate key sizes.

Main Observation: A very large number of tables, 58%, 53%, 50%, and 33% in SG, CA, UK, and US, respectively, do not have any single key columns. Therefore data systems, such as search engines that index records, may need to find composite keys to identify majority of the records in some portals. Furthermore, 10% of the tables across all portals do not have a candidate key of size 1, 2, or 3, which indicates the extent of denormalization in these portals.

The extent of value repetitions and the rarity of key columns have an important implication for data integration systems. These systems should differentiate between non-key columns and key columns when suggesting tables to join as non-key columns that contain a lot of repetition can lead to very large join outputs. We will also demonstrate that joins of key columns are significantly more likely to lead to useful joins than joins of non-key columns in Section 5. The extent of these value repetitions, which indicative a high level of denormalization, further the existence of sub-tables in these tables that may themselves be good candidates for integrating with other tables.

4.2 Functional Dependency (FD) Analysis

Next, we analyze the prevalence of non-trivial FDs in OGDPs. Recall that an FD [16] in a table T is an expression $X \rightarrow A$ where $X \subseteq attr(T)$ and $A \in attr(T)$, which informally indicates that a specific set of X values imply the same A values in T. Formally, $X \rightarrow A$ holds iff for any pairs of tuples $t_u, t_v \in T$ if $t_u[X] = t_v[X]$, then $t_u[A] = t_v[A]$. $X \rightarrow A$ is *trivial* if $A \subseteq X$ or if X forms candidate key. It is well known that existence of non-trivial FDs are indicate of poor relation design and lead to value repetitions that can be avoided by decomposing the relation into Boyce Codd normal form (BCNF). In the remainder, LHS and RHS stand for the left- and right-hand side of an FD, respectively.

While all our previous analyses used all datasets in each OGDP, our next analyses on composite keys and FDs require super-linear computations and for these we used tables with $10 \leq t \leq 10000$ tuples and $5 \leq c \leq 20$ columns. To find FDs in tables, we implemented the *FUN* algorithm for finding FDs [28] and limited the algorithm to find FDs whose LHS contain at most 4 attributes. Note that we put an upper bound on number of columns, as the runtime of the *FUN* algorithm [28] grows exponentially with the number of columns (even when limited to finding FDs with 4 attributes). This is the level we observed the *FUN* algorithm to complete at a reasonable time. Although the complexity of *FUN* increases linearly with the number of rows, we put a very high upper bound of 10000 rows to avoid running the algorithm on very large tables, on which the algorithm also took a very long time. The final number of tables along with other statistics from the sample are provided in Table 5. Table 5 shows the percentages of the tables for which we found at least 1 FD across all portals. *Main Observation: Majority of tables in each portal, and overwhelming majority in UK (84.05%) and US (79.86%), have non-trivial FDs. These percentages indicate that most of the table published by OGDPs are not in Boyce Codd normal form, so up to the common normalization standards of relational tables in practice.*

Finally, we note in most of the tables, the FDs have a simple structure where a single attribute on the LHS implies columns on the RHS. Such FDs indicate a direct dependency between two columns in a table. A classic example of such FD is $City \rightarrow Province$, which is prevalent in the Canadian portal. As shown in Table 5 ("tables with a non-trivial FD s.t |LHS|=1" lines), the majority of the tables that have a non-trivial FD has a non-trivial FD in this simple form.

4.3 BCNF Decomposition Analysis

In our next analysis, we decompose tables that have non-trivial FDs into BCNF and study the number of tables generated and impacts of the decomposition on the uniqueness scores of the columns of these tables. When decomposing a table T, we used the textbook BCNF algorithm [16] where we picked one of the remaining non-trivial FDs $X \rightarrow A$ uniformly at random and constructed a table T_1 with attributes $X \cup A$, and another T_2 with $X \cup (attr(T) \setminus A)$. Then we iteratively repeated the process on the latest set of tables until we obtained a set of tables in BCNF.

Figure 7 shows the number of decomposed tables we obtained in each portal. Bars with x-value 1 indicates that the original table was already in BCNF. On average, a table not in BCNF decomposed into 2.42, 3.39, 3.28, and 3.26 tables for SG, CA, UK, and US respectively. These averages are shown in Table 5. Furthermore, there are substantial amount of tables (i.e., more than 40% of the tables across portals excluding SG) that are decomposed into 3 or more sub-tables, with as many as 11 partitions.

We further examined its impact on value repetition of decomposing tables into BCNF. We computed the average uniqueness scores of the columns of tables that were not in BCNF before and after the decomposition. Recall that when using FD $X \rightarrow A$

Table 5: FD and decomposition statistics of the tables after normalizing tables with non-trivial FDs to BNCF.

	Portal			
	SG	CA	UK	US
total # tables	701	7492	18864	9770
total # columns	4142	76976	189930	102118
avg # columns per table	5.91	10.27	10.07	10.45
# tables with a non-trivial FD	381	5500	15855	7802
% of tables with a non-trivial FD	54.35%	73.41%	84.05%	79.86%
# tables with a non-trivial FD s.t |LHS|=1	318	3659	12998	5944
% of tables with a non-trivial FD s.t. |LHS|=1	45.36%	48.83%	68.90%	60.84%
avg # tables after decomposition of tables not in BCNF	2.42	3.39	3.28	3.26
avg # columns in partitions after decomposition of tables not in BCNF	3.34	4.59	4.33	4.66
avg uniqueness score increase for unrepeated columns	2.30x	2.98x	2.49x	2.20x

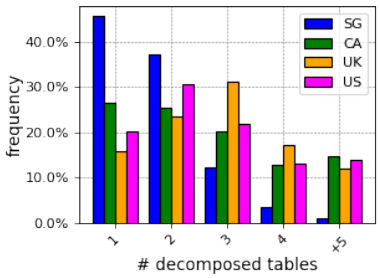

Figure 7: Distribution of the number of decomposed tables after applying normalization to tables. 1 indicates that the original table was already in BCNF.

during each decomposition step, X will be repeated and in both decomposed tables T_1 and T_2 above, X's uniqueness scores are guaranteed to stay same in T_2. In this analysis we focus on the uniqueness scores of non-repeated columns. Table 5 reports the ratio of uniqueness scores of unrepeated columns before and after the decomposition. We see an average uniqueness score increase between 2.20x and 2.98x across the portals.

Our hypothesis, both from having thoroughly studied these tables and observing the prevalence of non-trivial FDs and the number of decomposed tables is that many tables in OGDPs are pre-joined versions of multiple base tables. Governments and institutions publishing on OGDPs tend to publish single tables about a topic instead of databases of tables, so there is likely a tendency to join several tables before publishing. *Therefore data integration and exploration systems over data in OGDPs can automatically decompose these tables and serve the decomposed sub-tables as possible base tables. Such sub-tables may be meaningful and of independent interest to users. An important research question here is how to differentiate between accidental vs real FDs to identify high quality and useful sub-tables that can be useful for users.*

We give two examples out of many others we observed. In SG portal, there is a table[8] storing labour statistics in different industries, and there is a hierarchy between industry values under the attributes $industry\_1, industry\_2, industry\_3$ which exhibit FD. After the decomposition, the table is divided into 4 subtables. One of these subtables is a table of "Industry Hierarchies" that records level 2 industry along with their associated level 1 industry they are under. This is a table that is useful on its own and is not published as a separate table. Similarly, in US portal, a table[9] by City of Chicago stores budget recommendations. There are multiple FDs between the attributes, e.g., $FundCode \rightarrow FundDescription, FundType$, indicating the existence of different entities. After decomposition, we obtain subtables that may be useful on their own but are not published as separate tables, such as about FundCodes and their Descriptions and Department Numbers and their Descriptions.

[8]https://data.gov.sg/dataset/8e06592f-6b3b-4339-8772-797f679197cd

5 JOINABILITY ANALYSIS

Extending existing tables in OGDPs by joining them with other tables in these portals is one of the most widely studied topics in the context of data integration in OGDPs [7, 14, 15, 26, 34, 35]. Our next analyses study several properties of the joinable pairs of tables. Our high-level questions are how frequently pairs of tables with high value overlap columns lead to useful vs accidental joins and what are the common properties of useful vs accidental joins?

5.1 Selection of Joinable Table/Column Pairs

Throughout our analyses, we define *joinable pairs* as quadruplets (t_i, c_k^i, t_j, c_l^j), where (t_i, t_j) are found to be joinable through the pair of columns (c_k^i, c_l^j). Since join is a value-based operation, ultimately successfully integrating tables through a join operation relies on finding pairs of columns with high value overlaps. As done in many prior work [7, 13–15, 21, 23, 35], we used *Jaccard* similarity of the two columns as a metric of joinability. Jaccard similarity of columns (c_k^i, c_l^j) is calculated as $S_{jacc}(c_k^i, c_l^j) = \mid x_k^i \cap x_l^j \mid / \mid x_k^i \cup x_l^j \mid$, where x_k^i refers to sets of unique values for the column c_k in the table t_i. We used all available tables from each portal. We further filtered out joinable pairs based on two criteria:

- *High Jaccard similarity:* Since our overall goal is to analyze useful joinable pairs, we need to pick quadruples only if the columns in it have a high enough Jaccard similarity. As reported in reference [29], when using column similarity metrics (in our case Jaccard similarity), we would expect the higher the metric, the more precision we should have in identifying actual joinable pairs. So we picked pairs only if their

[9]https://catalog.data.gov/dataset/budget-2017-budget-recommendations-appropriations

Table 6: Main statistics of the joinable pairs for each portal.

	Portal			
	SG	CA	UK	US
total # joinable pairs	28770	268103	616956	3786199
total # tables	2376	14707	33359	25857
# joinable tables	1578 (66.4%)	8286 (56.3%)	16157 (48.4%)	14208 (54.9%)
median degree per joinable table	27	17	12	115
max degree per joinable table	169	529	767	3322
total # columns	12428	194022	405093	374400
# joinable columns	1962 (15.8%)	25975 (13.4%)	48221 (11.9%)	66493 (17.8%)
# key joinable columns	410(20.9%)	5311(20.4%)	11722(24.3%)	11918(17.9%)
# non-key joinable columns	1552(79.1%)	20664(79.6%)	36499(75.7%)	54575(82.1%)
median degree per joinable column	17	3	5	4
max degree per joinable column	169	526	433	1606

join columns had objectively very high, over 0.9, Jaccard similarity value.[10]

- *High unique values:* We wanted to avoid analyzing the joinability of columns that have a very small number of values, e.g., columns encoding booleans, as they would perfectly overlap and likely lead to very high expansion rates and false positives. To avoid this, we selected pairs only if their columns had at least 10 unique values, which is the lowest median unique value count across corpuses. We note that similar filtering steps based on unique values has been used in many prior studies on tables from public data sources [6, 10, 22].

5.2 General Characteristics of Joinable Pairs

Table 6 reports the general statistics of the joinable pairs that we analyzed. Between 48.4% (UK) and 66.4% (SG) of the total tables in each portal have at least one other joinable table on some column. In contrast, only between 11.9% (UK) to 17.8% (US) of the columns have another column they are joinable with, of which 17.9% (US) to 24.3% (UK) were key columns. Table 6 reports the "degree" of a joinable table, i.e., the number of other tables a table is found to be joinable with. The median degree varies between 12 (in UK) and 115 (US), while the maximum degrees varied between 169 (SG) and 3322 (US). Similarly, the median "degree" of joinable columns varied between 3 (CA) and 17 (SG) and the max varied between 169 (SG) and 1606 (US). These indicate that there are a large number of tables and columns that have close-to-perfect value overlap with a large number of tables. We manually analyzed some high degree tables and columns and observed three patterns that explain this:

- Tables with the same schema: There are large sets of tables that have the same or almost the same schema, often because these are periodically, e.g., weekly or monthly, published tables. These tables tend to have many columns that have exactly the same domain and tend to be all pairwise joinable.
- Tables in the same dataset: Many datasets have multiple tables storing information about different aspects about an entity, which we refer to as *semi-normalized tables*. The schemas of these tables tend to have common columns with significant value overlaps. These tables can be seen as normalized versions of a larger table yet can still exhibit FDs.
- Common columns: Some columns, such as state or year, exist in many tables and have high joinability degrees.

[10]For our analyses in Section 5.3 on expansion rates, we verified that our choice of this threshold is not very sensitive if we used a lower but still high threshold of 0.7. Specifically, we obtained similar results as those presented in Figure 8, which can be found in our supplementary document in our github repo: https://github.com/arifusta/ogdpAnalysis.

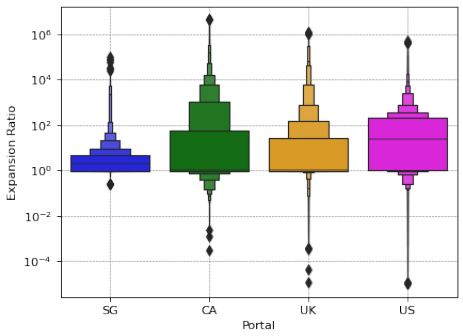

Figure 8: Expansion ratio distribution of joinable pairs.

Anecdote 1: Highest-degree Table

We examined the table with the highest joinability degree, which was 3322, which is published in the *Terrestrial Biodiversity Summary* dataset published by California Department of Fish and Wildlife. The table has 44 total columns. 22 of these join with another column from another table. The highest degree column is an integer *plntendem* with a degree of 941. Not surprisingly, this column has a very low uniqueness score 0.00047 (with only 30 unique values among 63890 rows). Another *county* column stores strings with a degree of 576 and uniqueness score of (0.00091). The table also contains high-degree columns despite being keys, such as *objectid* which stores incremental integers with a degree of 371.

We next analyzed the *expansion ratio* of the joins, which we define as: output size of the join / the size of the larger table.

Expansion ratio distributions for all portals are depicted as letter-value plots in Figure 8[11]. The biggest box in each distribution represents values between the 1st and 3rd quartiles. Vertical line in the biggest box represents median expansion ratios, which we found as 1 for CA and UK, 2 for SG and 24 for US. As shown in the plot, except in SG, very large fractions of joinable pairs grow significantly, often beyond 10. For example in the US, the majority grows beyond 24 and there are at least 25% of the pairs that have an expansion ratio of above 100.

Perhaps the most common motivating case for joins is to extend one table with a new column, without growing the table at all, e.g., to add a new property of an entity in a table as a new column. If the expansion rate of a join is very high it is safe to assume that the joins are accidental. Therefore, this analysis of expansion rates should already give the overall picture that the close-to-perfect value overlaps across pairs seem overwhelmingly accidental. We will confirm this by our analysis in the next section, where we manually annotated large pairs of tables and found only 2 useful pairs with expansion ratios more than 1 (but still very small, at most 1.16x).

5.3 Useful vs Accidental Pairs Analysis

In our next analysis, we sampled a large set of 600 pairs of tables from all of the pairs we used in our previous analyses and manually labeled them as accidental vs useful and studied commonalities across useful and accidental pairs. Note that we could have used an automated-technique, e.g., by analyzing the contents of tables for semantic similarity, to scale the pairs of tables we analyzed. However, ultimately our goal is to get ground truths about the usefulness of joins, for which a manual labeling is needed. We first describe our methodology and then our results.

5.3.1 Sampling Methodology. Our goal was to get a large enough sample from each corpus to be able to observe general patterns about where useful vs accidental pairs appear. We also needed the sample size to be small enough so we could study each pair of tables manually to decide if the join is useful or not, which is a time consuming process. Recall that there are several patterns that lead to some sets of tables having a very large joinability degrees amongst themselves. To avoid seeing the same tables or columns with high degrees many times, we did not sample the pairs uniformly at random. Instead, we first picked a joinable table T_1 uniformly at random. Then, we picked a joinable column c_i^1 of T_1 uniformly random among the set of its joinable columns. This ensures that each table gets an equal chance of being in our sample even if it has a low joinability degree. Finally, we picked a table T_2 uniformly at random that has a column c_j^2 that can join with (T_1, c_i^1). If T_2 has more than 1 such column, we picked the one with the higher value overlap with c_i^1. We also adopted the following rules:

- Removal of pairs of tables with same schema: If T_2 had the same schema as T_1 we removed it from the output. Such tables generally occur in periodically published datasets and lead to useful joinable pairs, e.g., to correlate two columns across two different years. We removed them as a similar analyses can be done through unoining as well and we will cover same-schema pairs under unionability in Section 6.
- Equal size distribution for T_1: After each successful sample, we recorded the size of T_1, and in the end ensured that we sampled equal, 50, samples where T_1's number of rows t_r were between three size ranges: (i) $(10, 100)$; (ii) $[100, 1000)$; and (iii) ≥ 1000. The goal of this bucketing was to study if there was a visible correlation between the usefulness of the joins and the sizes of the tables being joined.
- Equal key/nonkey distribution: We also recorded for each sample (T_1, c_i^1, T_2, c_j^2) for whether the join was between three types of key/non-key combinations: (i) key-key; (ii) key-nonkey; and (iii) nonkey-nonkey and similarly ensured we sampled equal numbers from each category for each size bucket (so roughly 17 samples from each sub-bucket and 50 in total). Similar to the above rule for bucketing pairs based on table sizes, our goal in this bucketing was to study if there was a visible correlation between the usefulness of the joins and the key-nonkey properties of join columns.

We note that we noticed that SG has a specific publication style that many tables, across a very wide range of domains, share the same set of columns such as $\{level\_1, level\_2, year, value\}$ or $\{level\_1, year, value\}$. Not surprisingly, many table pairs sharing the same set of columns are found to be joinable, since they have high value overlaps. During annotation, we observed that with a few exceptions, all of the sampled pairs were in this form and lead to accidental pairs, so we remove SG in the rest of our analysis. This already indicates the limitation of value overlaps as indication of useful joins.

5.3.2 Labeling Methodology. For each sample, we studied its datasets, read the dataset descriptions, the tables and the columns in the pair and labeled the pair using three categories:

- Unrelated Tables and Accidental (U-Acc): These are the clear false positive pairs of tables that come from completely different domains (e.g., crime vs health) and happen to have columns with high value overlaps.
- Related Tables and Accidental (R-Acc): These are pairs that originate from the tables storing same or similar information in a same context (e.g., health), but the join is accidental because the join's output does not have a clear interpretation. Often, this happens because the join is on columns that do not represent the main entities but some other property of these entities. For example, in the popular NSERC research award datasets of Canada [12], different tables, such as Awards, which records the primary investigator (PI) of applications, and Co-Applicants, which records the co-PIs have many columns other than application ID that highly overlap in values, such as Institution and CoAppInstitution. Joining on such columns do not have a clear interpretation.
- Useful: These are the pairs where the output of the table has a clear interpretation.

As many manual human evaluation of datasets, our labeling was done in a best of effort manner, which can be subjective. However, in almost all cases, we found the labels of pairs to be obvious especially when the pairs were labeled as accidental as U-Acc or R-Acc. For pairs we labeled as useful our principle was that whenever in doubt we assumed the join could be useful, so erred on the side of labeling more pairs as useful. Yet we expect

[11] As we mentioned in Section 5.1, we repeated this analysis for all table pairs whose Jaccard similarities were above 0.7 instead of 0.9, which can be found in our supplementary material.

[12] https://open.canada.ca/data/en/dataset/c1b0f627-8c29-427c-ab73-33968ad9176e

Table 7: Distribution of accidental vs useful labels.

Portal	Join Result			
	accidental			useful
	U-Acc	R-Acc	total	
CA	35.95%	50.33%	86.28%	13.72%
UK	31.79%	49.01%	80.80%	19.20%
US	62.67%	24.00%	86.67%	13.33%

Table 8: Distribution of accidental vs useful labels across joinable pairs within inter- and intra-dataset groups.

Portal	Dataset	Join Result			
		accidental			useful
		U-Acc	R-Acc	total	
CA	inter	49.11%	44.64%	93.75%	6.25%
	intra	0.00%	63.41%	63.41%	36.59%
UK	inter	43.64%	40.91%	84.55%	15.45%
	intra	0.00%	70.37%	70.37%	29.27%
US	inter	70.68%	21.05%	91.73%	8.27%
	intra	0.00%	47.06%	47.06%	52.94%

that readers would also find these labelings overwhelmingly obvious. For reference, all of the pairs and our labels are here [13].

5.3.3 Results. Table 7 shows the overall frequencies of the labels we gave across portals.

Key Observation: As we hypothesized, overwhelming majority of the joinable pairs we sampled, whose columns had close-to-perfect value overlaps are accidental, i.e., false positives. The frequency ranges between 80.8% and 86.7% across portals (and 100% in SG). Several prior works, Toronto Open Dataset Search [36], Auctus [11], Governor [23], have studied how to efficiently detect columns with high value-overlaps and used them in systems to suggest joinable pairs. Yet, our results indicate that value overlap alone can be a weak signal of useful joins and such systems need to be more selective in the tables they suggest to users.

There are many pairs that are from unrelated tables, their frequencies in our sample ranges between 31.8% (UK) to 62.7% (US). For example, one of the accidental pairs in the CA portal is from datasets entitled *Lumpfish catch rates* and *Conditional Release - Appeal Decisions*. Systems can eliminate these false positives by checking or predicting the domains of the datasets. However, there are also a large number of false positives (in fact majority in CA and UK) between tables from related domains.

Next we present the frequencies of accidental vs useful pairs with respect to three other properties of the pairs: (i) whether the pairs come from the same or different datasets; (ii) whether the join columns are key vs non-key; and (iii) the data types of the join columns. Each of these properties correlate strongly with whether the pairs are accidental or useful. We also analyzed if the sizes of the tables correlate with whether the pairs are accidental but did not observe a clear correlation there. The table that shows this analysis can be found in the supplementary document in our github repo[14].

[13]https://github.com/arifusta/ogdpAnalysis
[14]https://github.com/arifusta/ogdpAnalysis

Table 9: Distribution of accidental vs useful labels within key column combination groups.

Portal	key column	Join Result			
		accidental			useful
		U-Acc	R-Acc	total	
CA	key-key	31.37%	47.06%	78.43%	21.57%
	key-nonkey	58.82%	23.53%	82.35%	17.65%
	nonkey-nonkey	17.65%	78.43%	96.08%	3.92%
UK	key-key	24.00%	42.00%	66.00%	34.00%
	key-nonkey	47.06%	31.37%	78.43%	21.57%
	nonkey-nonkey	24.00%	74.06%	98.00%	2.00%
US	key-key	66.00%	4.00%	70.00%	30.00%
	key-nonkey	79.59%	14.29%	93.88%	6.12%
	nonkey-nonkey	43.14%	52.94%	96.08%	3.92%

Inter- vs Intra-dataset Pairs: Table 8 presents the frequencies of accidental vs useful pairs when we divide the pairs as inter- and intra-dataset pairs. There is significant discrepancy in frequencies of useful pairs across these groups in all portals: while between only 6.2% and 15.5% of the inter-dataset pairs are useful, this frequency is much higher and between 29.3% and 52.9% in intra-dataset pairs. Note that this does not mean that in absolute numbers there are more useful intra-dataset pairs in all portals because the overwhelming majority of the pairs, 78.2% of all pairs across portals, still come from inter-dataset pairs. However, it shows clearly that overwhelming majority of the pairs with close to perfect value overlaps are accidental if they come from different datasets. We give an example useful inter-dataset pair.

Anecdote 2: Example Inter-dataset Useful Pair

In CA, two tables related to COVID pandemic are published under two different datasets. First is a table[15] that stores information about COVID testing for different age groups. Second is a table[16] about COVID cases. By joining on the `date` columns of the tables, one can correlate vaccination and testing on same dates.

Key- vs Non-key Join Columns: Table 9 shows the frequencies of accidental vs useful pairs across key column combinations. Overall, while only between 2.0% and 4.0% of nonkey-nonkey pairs are useful, this frequency increases to between 18.0% and 27.8% over pairs that have at least 1 key pair. Overwhelming majority of nonkey-nonkey pairs had an expansion ratio more than 1 with a median rate of 6.03x (note that pairs with at least 1 key are guaranteed to have an expansion ratio at most 1). Only 7 of the nonkey-nonkey pairs were useful and only 3 of these had expansion ratio greater than 1. These expansion ratios were still small, between 1.1x and 1.52x only and a result of aggregate columns making the column non-key (explained in Anecdote 3 below). *As we hypothesized in Section 5.2, high expansion ratio of the join and more generally the pairs being between nonkey-nonkey columns are indeed strong signals for accidental pairs.*

[15]https://open.canada.ca/data/en/dataset/ab5f4a2b-7219-4dc7-9e4d-aa4036c5bf36
[16]https://open.canada.ca/data/en/dataset/f4f86e54-872d-43f8-8a86-3892fd3cb5e6

Table 10: Distribution of accidental vs useful labels across joinable pairs within column data type groups.

Portal	column data type	Join Result			
		accidental			useful
		U-Acc	R-Acc	total	
CA	incremental integer	62.5%	33.3%	95.8%	4.2%
	categorical	6.7%	70.0%	76.7%	23.3%
	integer	25.0%	59.4%	84.4%	15.6%
	string	13.3%	66.7%	80.0%	20.0%
	timestamp	40.0%	50.0%	90.0%	10.0%
	geo-spatial	52.9%	29.4%	82.3%	17.7%
UK	incremental integer	80.0%	15.0%	95.0%	5.0%
	categorical	2.9%	64.7%	67.6%	32.4%
	integer	40.0%	40.0%	80.0%	20.0%
	string	0.0%	78.1%	78.1%	21.9%
	timestamp	32.0%	52.0%	84.0%	16.0%
	geo-spatial	25.0%	25.0%	50.0%	50.0%
US	incremental integer	100.0%	0.0%	100.0%	0.0%
	categorical	33.3%	41.7%	75.0%	25.0%
	integer	60.7%	28.6%	89.3%	10.7%
	string	11.1%	66.7%	77.8%	22.2%
	timestamp	56.7%	13.3%	70.0%	30.0%
	geo-spatial	36.4%	54.5%	90.9%	9.1%

Anecdote 3: Nonkey-Nonkey Useful Joins

In one example, the two tables were about the landings of different fish in the two coasts of Canada from two different years. The nonkey column was on the different species of fish but included 4 `Total` and 3 `Other` values, because the single table was divided into multiple sub-tables. The join was labeled as useful because one can imagine correlating two years of statistics about landings (and one would have to ignore the multiple `Total-Total` or `Other-Other` output rows.) This same pattern was true also for the other 2 growing joins between nonkey-nonkey pairs.

Data Type of Join Columns: Table 10 shows the frequencies of accidental vs useful pairs grouped by different data types of the join columns. The table divides integers into incremental vs non-incremental because of the large discrepancy between the frequencies of the labels they lead to. *Our main observation is that joins between columns that store incremental integers are the most common across pairs with high value overlaps. These pairs are also overwhelmingly accidental (between 95% and 100%). This compares sharply with the rest of the data types, where on average across portals between 15.6% and 28.6% of the pairs are useful. The data type that most frequently leads to useful joins are categorical types, such as species storing different fish types.*

Anecdote 4: Accidental Key-Key Pair

In CA portal, there is a joinable pair between tables under datasets *Lumpfish catch rates*[17] and *Conditional Release - Appeal Decisions*[18]. Although both columns in the pair are key columns storing incremental integer values, the resulting join is accidental. Columns storing integer values will cause high value overlap, however the context of the tables are completely irrelevant, which leads to a false positive pair.

Summary of Observations: Joins between tables that are located in the same datasets, between key columns, and on data types other than incremental integers, e.g., categorical, string, or geo-spatial, lead more frequently to useful joins. These properties can be useful signals to filter accidental joins between tables that have columns with high value-overlaps. We believe doing research on identifying accidental vs useful joinable pairs, complementing value-overlap techniques with non value-based techniques and preparing human evaluated benchmarks and/or performing human evaluations of proposed techniques is an important direction for developing useful data integration systems over OGDPs. Our manually labeled pairs can be found in our repo and used as a benchmark with ground truths to evaluate their techniques.

5.3.4 Common Patterns Across Useful and Accidental Pairs. We end this section by summarizing the patterns we have observed both across the useful and accidental pairs. We broadly categorize patterns for useful joinable pairs as follows:

- Joins of two semi-normalized tables under the same datasets: This is the most common pattern resulting in a useful join, where a dataset publishes its information in sets of tables and one can join these tables to construct full records.
- Joins of periodically published tables on key columns: There is a publication style prevalent across all OGPDs, that tables are partitioned into sub-tables spanning over multiple years on same aggregate values. These tables can be joined to correlate some measurements across different sub-tables.
- Joins of tables measuring different statistics on common domain columns: Across all portals, there are common column domains that are present in many tables such as date or state/province. In the case of measuring different statistics for a particular event (e.g., COVID-19), joining these tables on columns of such domains produce a useful result as in Anecdote 2.

We categorize patterns for accidental joinable pairs as follows:

- Joins of unrelated tables on incremental integer or columns of common domains: This is the most prevalent pattern and such columns, e.g., a state/province column or one with an incremental integer domain appear in many unrelated tables, and more importantly from different domains.
- Joins of semi-normalized tables on non-key columns: Although such tables are related, their join on non-key columns lead to uninterpretable and often large tables.
- Join of semi-normalized tables under periodically published datasets on two different time period: This pattern occurs under periodically published datasets where data from each period is further partitioned into sub-tables and tables corresponding to different aspects of the dataset are found joinable for different time periods (e.g., 1990 age statistics with 2020 tax information for those age groups). Yet, the full output records of these joins are not interpretable.

[17] https://open.canada.ca/data/en/dataset/533d694b-b692-4127-bf22-d1e41c0b5bba

[18] https://open.canada.ca/data/en/dataset/1046cd6a-8990-4a02-8e09-025e99a92e91

Table 11: Overall statistics of the unionable tables for each portal.

	Portal			
	SG	CA	UK	US
total # tables	2376	14911	35049	26416
# unionable tables	1447 (61.0%)	9491 (63.7%)	26928 (76.8%)	15093 (57.1%)
median degree per unionable table	2	3	3	2
max degree per unionable table	271	212	1554	2931
# unique schemas	1083 (2.19)	6834 (2.18)	10884 (3.22)	14070 (1.87)
# unionable schemas	154 (14.2%)	1414 (20.7%)	2763 (25.4%)	2747 (19.5%)
unionable schemas with single dataset	47 (30.5%)	706 (49.9%)	1516 (54.9%)	276 (10.0%)

- Joins of "transaction/event tables" that share similar information about different transactions/events: Another pattern is between tables that record a different set of transactions or events that share same property in a common column. For example, in the UK portal, there is a pair of tables storing 2 different sets of events related to Foreign and Commonwealth Office: a table of overseas travel events[19] and a table of meeting events[20], whose join leads to records that do not seem interpretable producing many duplicates.

6 UNIONABILITY ANALYSIS

We next do a brief analysis of sets of tables that are unionable. We consider two tables as unionable if their schemas, i.e., column names and data types, are exactly the same. This is a natural notion of unionability that we expect to be robust and has been used in several prior work [7, 12, 23]. Our goal is to identify common patterns when this is indeed a robust metric and when it may lead to false positives.

Statistics about unionable tables are provided in Table 11. On average across all portals, more than 60% of the tables are found to be unionable to at least 1 other table. Median degree of unionability, i.e., size of unionable sets, among the unionable tables is 2 or 3 and the maximum degree varies between 212 to 2931. Considering the total number of tables, especially SG stands out from others, having a unionable schema shared among more than 10% of all tables. This set in fact contains many false positives due to the common publication style in SG that we discussed in Section 5.3. Recall that, there are two very common schemas in SG: (i) $\{level\_1, level\_2, year, value\}$; and (ii) $\{level\_1, year, value\}$. used for many unrelated datasets. These schemas seem unionable (and joinable) but are in fact false positives .

Table 11 also reports the number of different unionable schemas, i.e., one that is shared by at least 2 tables. For each schema S we also report whether the set of tables that share S are published in a single dataset or across multiple datasets. Overall betwen 14.2% to 25.4% of all schemas are unionable. For CA and UK, 50% of the unionable schemas have their tables under the same dataset, whereas this percentage is 30% and 10% for SG and US, respectively. This difference is primarily due to the differences in the publication styles. Specifically,UK and CA tend to publish periodically published tables under the same dataset compared to SG and US.

Similar to joinability analysis, we manually labeled unionable table pairs as accidental vs useful to identify the common patterns among them. When sampling, we picked a schema S uniformly at random, and for each schema we picked a pair of tables with schema S, again uniformly at random. For each portal, we randomly picked 25 pairs. In total, we labeled 100 unionable table pairs. Unlike the joinability analysis, we found that overwhelming majority of the pairs are indeed useful and leads to interpretable outputs. All of the pairs for CA and UK are found unionable. As expected, this implies that perfect schema overlaps is indeed a robust signal of unionability. Most prevalent patterns for useful unionable pairs are:

- Periodically published tables: Not surprisingly, the most common pattern is tables that are periodically published. As we pinpointed in Section 5, there is a periodic publication style of certain datasets, which constitutes the biggest portion of unionable tables from the annotation sample. We observed that although this behavior is more likely to occur for tables under the same dataset (e.g., under a dataset[21] in CA portal, there are 61 tables each of which store information for a particular year-month), it is also possible under different datasets (e.g., Health Trust Specialist Services Reference Costs in UK portal for 2017-18[22] and 2019-20[23]) published by the same organization.
- Tables partitioned on a non-temporal attribute: Similar to how periodically published tables can be seen as partitioned tables according to a temporal property, many sets of unionable tables are partitioned by other often categorical attributes. A common attribute is state or province of the country (e.g., tax statistics for different provinces under the dataset[24] in CA portal) although many others exist, e.g., statistics about real-estate properties partitioned into tables based on property types[25].

We also observed two patterns for accidental unionable pairs:

[19] https://www.data.gov.uk/dataset/19df755a-c725-4031-a76b-f234ebec9543
[20] https://www.data.gov.uk/dataset/9f1b1f2a-b5d5-447a-bfae-d1c29afb3755
[21] https://open.canada.ca/data/en/dataset/010c8dd5-592b-40b5-b5d4-77a8ed903e42
[22] https://www.data.gov.uk/dataset/7836a8bb-c0c4-4ca4-9bd8-02a62fc0fabd/health-trust-specialist-services-reference-costs-2017-18
[23] https://www.data.gov.uk/dataset/14a3637b-c5be-4ba2-9803-5f03f9ffee05/health-trust-specialist-services-reference-costs-2019-20
[24] https://open.canada.ca/data/en/dataset/21aa2140-3816-4c19-b47a-24fb28fa893d
[25] https://data.gov.sg/dataset/b215274a-8097-4571-bf54-0535e3cb585c

- Standardized schemas in SG: As we mentioned above, the largest set of accidental unionable pairs are unrelated tables that adhere to the two standard schemas that are used in SG.
- Duplicate tables in US: In US, some tables are published multiple times under different datasets.

We note that an important problem for unionability and joinability is ranking of unionable/joinable pairs. During our manual labeling we noticed examples when a particular table can be unioned with many other tables and some of those seem more likely to be useful than others. For example, some tables will be partitioned on two properties, such as a housing dataset[26] in UK portal which is partitioned based on both house type and council. Intuitively one can more easily imagine users wanting to integrate tables having same housing types from different councils or different housing types from the same council compared to integrating tables with both different housing types and councils. Even if multiple tables can be unioned with a target table because they have the same unionability score (e.g., perfect schema overlap), they should still be ranked using other relatedness metrics by data integration systems, which we think is an important research topic to study.

7 RELATED WORK

Closest to our work are studies that analyze several different characteristics of open datasets. Omar et. al. carried out a large scale study [5] in which they explored numerous statistics about the datasets published in Google Dataset Search (GDS) tool [9]. GDS indexes the "metadata" of open datasets from the broad world wide web. Metadata here refers to information about the datasets such as their file formats, licenses, and publishers and not the dictionary files, which we analyzed in this paper. GDS indexes a much broader set of datasets but does not index the contents of these datasets. Similarly, [9, 27] solely focuses on the statistics about the available metadata information across a large set of open data portals, including OGDPs. Finally, in [25], Johann et. al, provided an analysis about general statistics of CSV files from 232 open data portals. These statistics include the number of tables, tuples and columns, and shapes of headers. Although we also report some of the general statistics about the CSV files in these portals, our focus is on the contents of these datasets, specifically the properties that relate to the relational design of these tables and properties of joinable and unionable pairs of tables. This paper presents comprehensive analysis of OGDPs by extending the work in [32].

Finding joinable and unionable tables are two of the most important topics that motivate some of the research done in the database community on OGDPs [7, 14, 24, 34, 35]. Typically when finding a joinable table, it is assumed that a table is given as a query along with possibly a column to find joinable tables. The problem becomes finding column pairs (i.e., joinable columns) from different tables based on a similarity metric to extend the query table. Since join is a value-based operation, ultimately studies on joinability use a value-based metric as we did in this work [24, 34, 35]. Value-based metrics are also used as part of a weighted similarity calculation in the context of finding joinable tables in [7]. There is another study that relaxes the requirement of exact value match to capture semantic variations between the values. For example, Pexeso [14] used semantic embeddings produced by FastText [18] to address equi-join's incapability to capture semantic variations between the values. Irrespective of the value-based metric used, our work highlights that the joinability of tables are often accidental and identifies several patterns of where useful and accidental pairs appear frequently that can inform researchers and developers of data systems on OGDPs.

Several prior work have developed systems that perform data search, discovery and integration using high schema overlaps between tables as a signal for relatedness [7, 12, 23]. In [24, 26], authors employed 3 different metrics to measure similarity between columns, one of which is value overlap. Similarly in [7], q-grams of attribute names and values residing in the columns are two of the 5 proposed metrics to calculate column similarity. In [19], authors exploit the power of pre-trained language models to extract semantically rich high dimensional representations for the columns to calculate pairwise similarity. Evaluating the pros and cons of different unionability metrics is an interesting research direction that is beyond the scope of our work. In this work, we analyzed unionability of tables that share the same schema with the goal of identifying common patterns when this is a robust metric and when it may lead to false positives.

Another line of research is finding related tables given a query table [7, 12, 29, 33]. This literature is not necessarily motivated by data integration [7, 12, 29, 33] but primarily by dataset discovery. Some of the techniques we covered above for table integration have also been used in various tools [8, 11, 15, 23, 30, 36] for dataset discovery in open data lakes.

Although we used FUN [28] as an algorithm to automatically detect functional dependencies in tables, there are many other algorithms one can use. References [17, 31] give overviews of a set of these algorithms. Different algorithms have different performances and scalabilities, any exact algorithm could have been used for the analysis we performed in this paper.

8 CONCLUSION

We studied properties of tabular datasets in 4 OGDPs related to their relational structures. We further studied how and when these datasets can be integrated using the common approaches from the literature, such as solely using value-based techniques. The properties we presented can inform researchers and developers that build data systems over OGDPs about the core properties of the datasets, such as the extent of denormalizations present in these datasets as well as the major shortcomings of value-based metrics for joins and when they are more likely to lead to useful instead of accidental joins. Our observations also raised important research questions that can inform future research, such as how to identify quality sub-tables in these highly denormalized tables, how to automatically extract metadata files, or how to enhance value-based metrics to differentiate between accidental value overlaps across columns vs those that would lead to meaningful table integrations. Our manually labeled table pairs, which are available in our repo[27], can also be used as ground truth dataset in studies that study techniques for suggesting tables to integrate. Finally, our analysis of 4 separate OGDPs allows us to compare the properties we observed across portals, which can inform data publishers about good practices across portals (e.g., US portal is better in publishing datasets with key columns, while SG's datasets all come with a metadata file).

[26] https://www.data.gov.uk/dataset/cf717c64-0c9c-48cb-b454-2e7c97b603e9

[27] https://github.com/arifusta/ogdpAnalysis

REFERENCES

[1] 2023. Canada's Open Government Data Portal. https://open.canada.ca/en/open-data.

[2] 2023. Singapore's Open Government Data Portal. https://data.gov.sg/.

[3] 2023. UK's Open Government Data Portal. https://www.data.gov.uk/.

[4] 2023. USA's Open Government Data Portal. https://data.gov/.

[5] Omar Benjelloun, Shiyu Chen, and Natasha Noy. 2020. Google Dataset Search by the Numbers. In *The Semantic Web – ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, 667–682. https://doi.org/10.1007/978-3-030-62466-8_41

[6] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: Entity linking in web tables. In *International Semantic Web Conference*. Springer, 425–441.

[7] Alex Bogatu, Alvaro AA Fernandes, Norman W Paton, and Nikolaos Konstantinou. 2020. Dataset discovery in data lakes. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 709–720.

[8] Alex Bogatu, Norman W Paton, Mark Douthwaite, and André Freitas. 2022. Voyager: Data Discovery and Integration for Data Science. In *Proceedings 25th International Conference on Extending Database Technology (EDBT 2022)*.

[9] Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a Search Engine for Datasets in an Open Web Ecosystem. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1365–1375. https://doi.org/10.1145/3308558.3313685

[10] Michael J Cafarella, Alon Y Halevy, Yang Zhang, Daisy Zhe Wang, and Eugene Wu. 2008. Uncovering the Relational Web.. In *WebDB*. Citeseer, 1–6.

[11] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A Dataset Search Engine for Data Discovery and Augmentation. *Proc. VLDB Endow.* 14, 12 (jul 2021), 2791–2794. https://doi.org/10.14778/3476311.3476346

[12] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding Related Tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD '12)*. Association for Computing Machinery, New York, NY, USA, 817–828. https://doi.org/10.1145/2213836.2213962

[13] Dong Deng, Albert Kim, Samuel Madden, and Michael Stonebraker. 2017. SilkMoth: An Efficient Method for Finding Related Sets with Maximum Matching Constraints. *Proc. VLDB Endow.* 10, 10 (jun 2017), 1082–1093. https://doi.org/10.14778/3115404.3115413

[14] 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, 456–467.

[15] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[16] Hector Garcia-Molina, Jennifer Widom, and Jeffrey D. Ullman. 1999. *Database System Implementation*. Prentice-Hall, Inc., USA.

[17] Ihab F. Ilyas and Xu Chu. 2019. *Data Cleaning*. Association for Computing Machinery.

[18] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[19] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2022. SANTOS: Relationship-based Semantic Table Union Search. arXiv:cs.DB/2209.13589

[20] Aamod Khatiwada, Roee Shraga, Wolfgang Gatterbauer, and Renée J. Miller. 2022. Integrating Data Lake Tables. *Proc. VLDB Endow.* 16, 4 (dec 2022), 932–945. https://doi.org/10.14778/3574245.3574274

[21] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating matching techniques for dataset discovery. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 468–479.

[22] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th international conference companion on world wide web*. 75–76.

[23] Chang Liu, Arif Usta, Jian Zhao, and Semih Salihoglu. 2023. Governor: Turning Open Government Data Portals into Interactive Databases. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 468–479.

[24] Renée J Miller. 2018. Open data integration. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2130–2139.

[25] Johann Mitlöhner, Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Characteristics of Open Data CSV Files. In *2016 2nd International Conference on Open and Big Data (OBD)*. IEEE, 72–79.

[26] 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 (2018), 813–825.

[27] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Automated Quality Assessment of Metadata across Open Data Portals. *J. Data and Information Quality* 8, 1, Article 2 (oct 2016), 29 pages. https://doi.org/10.1145/2964909

[28] Noel Novelli and Rosine Cicchetti. 2001. Fun: An efficient algorithm for mining functional and embedded dependencies. In *International Conference on Database Theory*. Springer, 189–203.

[29] Masayo Ota, Heiko Müller, Juliana Freire, and Divesh Srivastava. 2020. Data-Driven Domain Discovery for Structured Datasets. *Proc. VLDB Endow.* 13, 7 (mar 2020), 953–967. https://doi.org/10.14778/3384345.3384346

[30] Paul Ouellette, Aidan Sciortino, Fatemeh Nargesian, Bahar Ghadiri Bashardoost, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2021. RONIN: Data Lake Exploration. *Proc. VLDB Endow.* 14, 12 (jul 2021), 2863–2866. https://doi.org/10.14778/3476311.3476364

[31] 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 VLDB Endowment* 8, 10 (2015).

[32] Arif Usta and Semih Salihoglu. 2023. To Join or Not to Join: An Analysis on the Usefulness of Joining Tables in Open Government Data Portals. In *Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23)*. Vancouver, Canada.

[33] 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 (SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1951–1966. https://doi.org/10.1145/3318464.3389726

[34] 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*. 847–864.

[35] Erkang Zhu, Fatemeh Nargesian, Ken Q Pu, and Renée J Miller. 2016. LSH ensemble: Internet-scale domain search. *arXiv preprint arXiv:1603.07410* (2016).

[36] Erkang Zhu, Ken Q. Pu, Fatemeh Nargesian, and Renée J. Miller. 2017. Interactive Navigation of Open Data Linkages. *Proc. VLDB Endow.* 10, 12 (aug 2017), 1837–1840. https://doi.org/10.14778/3137765.3137788