An Urban Data Profiler

Daniel D. C. Ribeiro
NYU Center for Urban
Science+Progress
New York, USA
daniel.castellani@nyu.edu

Huy T. Vo
NYU Center for Urban
Science+Progress
New York, USA
huy.vo@nyu.edu

Juliana Freire
NYU School of Engineering
NYU Center for Urban
Science+Progress
New York, USA
juliana.freire@nyu.edu

Cláudio T. Silva
NYU School of Engineering
NYU Center for Urban
Science+Progress
New York, USA
csilva@nyu.edu

ABSTRACT

Large volumes of urban data are being made available through a variety of open portals. Besides promoting transparency, these data can bring benefits to government, science, citizens and industry. It is no longer a fantasy to ask "if you could know anything about a city, what do you want to know" and to ponder what could be done with that information. However, the great number and variety of datasets creates a new challenge: how to find *relevant* datasets. While existing portals provide search interfaces, these are often limited to keyword searches over the limited metadata associated each dataset, for example, attribute names and textual description. In this paper, we present a new tool, UrbanProfiler, that automatically extracts detailed information from datasets. This information includes attribute types, value distributions, and geographical information, which can be used to support complex search queries as well as visualizations that help users explore and obtain insight into the contents of a data collection. Besides describing the tool and its implementation, we present case studies that illustrate how the tool was used to explore a large open urban data repository.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data sharing, Web-based services*

Keywords

Metadata Extraction, Automatic Type Detection, Dataset Analysis

1. INTRODUCTION

About half of humanity lives in urban environments today and that number will grow to 80% by the middle of this century; North America is already 80% in cities, and will rise to 90% by 2050. Cities are thus the loci of resource consumption, of economic activity, and of innovation; they are the cause of our looming sustainability problems but also where those problems must be solved. Our increasing ability to collect, transmit, and store data, coupled with the growing trend towards openness [1, 7, 9, 19, 6, 16, 14], creates a unique opportunity that can benefit government, science, citizens and industry. By integrating and analyzing multiple data sets, city governments can go beyond today's imperfect and often anecdotal understanding of cities to enable better operations and informed planning (see e.g., [5, 7]). Domain scientists can engage in data-driven science and explore longitudinal processes to understand people's behavior [8]; identify causal relationships across datasets, which can in turn, influence policy decisions [3, 18]; or create models and derive predictions that benefit citizens (see e.g., [4]). Putting urban data in the hands of citizens has the potential to improve governance and participation, and in the hands of entrepreneurs and corporations it will lead to new products and services. In short, it is no longer a fantasy to ask "if you could know anything about a city, what do you want to know" and to ponder what could be done with that information.

While in the past, government, policymakers and scientists faced significant constraints in obtaining the data needed for planning and evaluating their policies and practices, currently they are faced with an information overload. The number of open data portals and the volume of data they hold are growing at a fast pace around the world [14, 15, 16, 17] A big challenge, now, is how to discover datasets that are relevant for a given task or information need.

Publishing platforms such as CKAN [2] and Socrata [20], which are widely used for open urban data, provide a simple search interface over the metadata, thus, users are not able to identify datasets based on their content. Besides, there are no standards for attribute names and, often, attributes lack even basic type information [1]. This makes it hard for users to formulate discovery queries.

As a step towards enabling richer queries and helping users identify the datasets they need, we propose a new tool, UrbanProfiler, which automatically extracts detailed information about the contents of the datasets. The goal is to use this information to enable users explore urban data by asking queries over attributes, content, and to filter datasets based on a given time period or a region. The latter is crucial given that a large percentage of urban data contains spatial and temporal information [1]. Furthermore, longitudinal analyses often require multiple datasets that overlap in space and time. Consider, for example, a social scientist, who tries to understand the effects of adding a bike lane to a city neighborhood,

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742135.

would greatly benefit from having all traffic-related datasets that cover the zip codes spanning that neighborhood immediately before and after the policy comes into effect.

UrbanProfiler automatically identifies the types of the columns using RegEx, dictionary lookups and specific rules. The system is extensible and other detectors can be added with minimal effort. It also computes additional statistics over the data values for each column to generate type-specific summaries. Using this information, UrbanProfiler is able to detect missing and (potentially) erroneous values. Thus, besides helping users find datasets, UrbanProfiler can also alert them about data quality issues. Moreover, the visualizations produced by UrbanProfiler assist users in the exploration and comparison of datasets. For example, UrbanProfiler can generate heatmaps to quickly show the spatial coverage of a dataset.

The remainder of the paper is organized as follows. In Section 2 we describe the tool, its architecture and components. We have also used UrbanProfiler over a real repository consisting of over 3,000 datasets for New York City, and present our findings in Section 3. In Section 4, we present scenarios that illustrate the usefulness of the tool. In Section 5, we discuss related work and we close in Section 6 where we outline directions for future work.

2. UrbanProfiler

The UrbanProfiler has two main components, metadata extraction and analysis, which we describe below. It receives as input a list of structured datasets to process and outputs a database with all extracted metadata and a spatial index. The information in the database is used to enable users to search datasets based on their contents and to support visual representations that help in the dataset discovery process.

2.1 Metadata Extraction

For each dataset, UrbanProfiler executes the following steps: download the dataset, retrieve existing metadata (if any), transform data into a canonical format, detect data types, extract column metadata, create spatial index, and save information into a database.

UrbanProfiler starts by downloading the dataset. As files can have different formats, we have *file readers* that load each file to a common structure that is used in the subsequent steps. The most common formats for tabular datasets we found on NYC Open Data [16] are JSON and CSV. By separating data loading from processing enables an extension point within the *file readers*. To analyze a new file format, a new reader can be added to the system. While ingesting a CSV file is simple, JSON files can be more challenging since their structure can be complex and they can include additional data.

Open datasets often come with metadata, e.g., attribute names, author, owner, creation and last update date, category, and a textual description of the contents. When available, UrbanProfiler retrieves this information, as it can help in the later steps of the profiling pipeline. For the current prototype, we implemented a *metadata retriever* that obtains metadata from Socrata portals, other retrievers can be created to support different portals.[1]

For many datasets, we found columns that have GPS information with latitude and longitude together in the same field, or as a composite value with the following components: human readable address, GPS coordinates, City, Borough, and Zip Code. When the column is composite, to simplify the type detection process, we split it into multiple columns, each containing one atomic value.

Table 1: Extracted Metadata

Target	Group	Metadata
Dataset	General	Number of rows, columns, values and missing values
Dataset	Provided	Name, description, author, owner, category, update frequency and time of creation and last update
Dataset	Process	Input file size, total memory used, time (begin & end) and processing time
Column	General (all types)	Number of total, unique and missing values, and most frequent value
Column	Temporal	Minimum (min) and Maximum (max)
Column	Numeric	Min, max, mean and std of values
Column	Spatial	All unique values and Bounding Box
Column	Textual	Min, max, mean and std of text length

We discover the type of the columns in two steps: first we discover all data types in the column; then we select a dominant type to represent this column. The detected values of other types (different than the dominant type) are considered anomalies or missing values (e.g., Null type). By discovering types present in columns, it is possible to support useful search queries. For example, a column named *Notes*, which is defined as *Text*, may contain a combination of rows with spatial values (e.g., borough, city, etc.) and temporal values (e.g., days, months, year, etc.). Knowing all the types in this column allows users to specify queries over time and/or to map the data spatially.

In addition to data types, we also extract metadata about datasets and columns. A summary of the extracted metadata is given in Table 1.

In the last step, we process the spatial columns to create an index that will be used to support spatial queries and to generate visualizations like heatmaps. In our test datasets, GPS coordinates may have been stored either in one or many columns, e.g., one for latitude and one for longitude. In the latter case, we must combine the columns to create a complete GPS coordinate. In the current prototype, we only store the number of records related to a given GPS coordinate and not the full record list. We also have to consider that some datasets have more than one GPS coordinate for the same record. For example, a 311 complaint[2] can include the location of the complainant when the complaint was filed (e.g., her home address) and also the location of the actual problem that is being reported (e.g., an illegally parked car one block away). When this happens, we associate a list of locations to the record. The result of this step is a table-like structure with the dataset id, column id, GPS coordinate and the number of records.

2.2 Automatic Type Detection

To detect the type of the columns, first we compute the percentage of unique values for each type. Besides the types for the values encountered, we also take the name of the column into account to determine the column type. We use the simple types for the column and the detailed types for the values. But all detected types are stored as they can be useful for dataset analysis. We use the simple types to give an overview of the data, while the detailed types provide more information when a deeper analysis is desired. Each detailed type is directly related to a simple type. The list of types that we detect is shown in Table 2.

To illustrate how we detect a specific column type, consider one dataset with a latitude column, which consists of valid values as

[1] The current version of UrbanProfiler searches for metadata using Socrata API. Socrata has an Open Data Portal solution used by more than 20 cities. This is the application behind NYC Open Data, which we used to test UrbanProfiler.

[2] `http://www.nyc.gov/311`

Table 2: Details of the Type Detectors

Simple Type	Detailed Type	Value RegEx	Name RegEx	Dictionary	Rules
Geo	Borough			x	
	Zip-code	x		x	
	Zip-code +4	x			
	GPS	x	x		x
Temporal	Date	x			
	Time	x			
	Date-time	x			
Numeric	Integer	x			
	Double	x			
Textual	Textual				x
Null	Null				x

well as wrong and missing values. In this example, assume that UrbanProfiler found 30% of *Geo-GPS*, 10% of *Text* and the remaining 60% as *Null* (missing). The general type that would accommodate all these values would be *Text*. But if we consider the type that occurs most frequently, the column type would be *Null*. In this case, the column name is actually "*Location*", which indicates the type should be *Geo*.

Currently, UrbanProfiler only considers distinct values during type detection. This helps identify issues in data distribution and reduces the effect of systematic mistakes. For example, imagine a column that has 1,000 records with 90% of them being 11201 (repeated) and 10% being the integer numbers 12,000, 11000 and 20,000. The type detectors will identify the 11201 as *Zip-code* and the others as *Integer*. In this case, if we used the count of values, the column would be classified as *Zip-code*, while using only the count of unique values, this column would be *Integer*.

We create a Type Detector for each type we want to identify. Different detection techniques can be used. In the current prototype, we have used RegEx and dictionaries lookups. The algorithm to detect types has two inputs: the values in the column and an ordered list of Type Detectors. The output is the percentage of values in each type. The order for applying the Type Detector is important because some types are more specific than others. For instance, the set of numeric values is a sub set of Textual values. If we start detecting Textual values, it will match all, and no values would be detected in other types. Thus, we apply the more specific types first. If no specific type is identified for some values, it will be considered as Text. Missing values are associated to the Null type.

It is easy to extend the system to support additional type detectors. UrbanProfiler provides an API that requires the type detection code, the type name, the related simple type, and the position it should be inserted in the Type Detectors list.

To define the simple type of the column, we use rules that consider the percentage of specific type, name of the column and provided type. For example, one of the rules is: if the type provided is location and the column has any spatial data, then the type will be Geo (even if it is not the most detected type). If no specific rule matches we use the most detected type. Only when all values are missing we label the column as *Null*.

2.3 Dataset Analysis

UrbanProfiler also provides tools to help with dataset analysis. The current prototype displays maps, charts and tables. The metadata is organized in four tabs: metadata, columns, charts and map.

To facilitate the analysis of datasets with *Geo* data, we use a map. In the map visualization, it is possible to see just the bounding box of the dataset or a heatmap. The first visualization is useful to compare many datasets, while the second is better to an individual dataset. In the heatmap, UrbanProfiler shows the number of records in each location. Figure 2 shows datasets comparison using the bounding boxes and Figure 3 shows the heatmap visualization.

The metadata tab shows metadata whose target is the dataset itself. This tab displays the general, provided and process metadata groups (Table 1). It also displays the count of columns in each type, the number of records with GPS data and the minimum and maximum latitude and longitude. The columns tab shows an overview of the columns (Figure 4) or details of a single column (Figure 5). Besides, it also shows if the column is a unique key (if any) and if it was extracted from a composite column by UrbanProfiler. To a deeper analysis of the column it shows the most frequent value and the detected type with charts. It also displays specific type metadata, such as the range, mean and STD for *Numeric* columns.

The charts tab provides a quick overview of the dataset. It has charts of number of columns by type and the relation of unique and missing values per column.

UrbanProfiler also provides an overview of its catalog. In the overview page it shows the total number of datasets profiled and if any of them had errors while profiling, a table with a summary and charts for the whole catalog, such as columns types (simple and detailed), datasets and records per category, top dataset authors and so on.

3. PROFILING NYC OPEN DATA

In this section we present some initial and informal analysis of what we could find using UrbanProfiler with datasets from New York City Open Data [3].

We profiled 3,065 datasets (and views) from NYC Open Data – over 57GB of data. We processed more than 66 thousand columns, 275 million records and 6 billion values. The offline profiling took approximately 24 hours, analyzing datasets in parallel using 30 cores.[4] The web interface we built to support search and exploration uses the metadata catalog which contains 1GB of data.

We did not evaluate the application performance, but in the initial tests, we are satisfied with the response time. For most datasets it take less than 3 seconds to show the metadata, charts and visualizations. Rendering map visualizations for big datasets with more than 1 million GPS points takes longer. Improving the performance of these visualizations is a direction we will pursue in future work.

The most frequent type detected was *Textual* (32k), followed by *Numeric-Integer* (10k), *Geo-GPS* (2.8k) and *Geo-Borough* (2.5k). There were 11.5k completely *Null* columns. Figure 3 summarizes the results for the most common types.

Although 53% of the datasets had *Geo* (general location) data, only 26% have *GPS* coordinates. In the next Section we discuss about data granularity to better use the Geo data. Also, almost 40% of the datasets contain *Temporal* attributes.

We also examined the categories of the datasets. The category with most datasets is *Social Services* (31%), followed by *City Government* (13%), *House and Development* (4.6%) and *Public Safety* (4.3%). The other categories had less than 3% each.

The datasets from NYC Open Data already have column types, but we found evidence that UrbanProfiler can improve the detected types in some cases. For columns with provided type *Text*, Urban-

[3] The datasets used in the presented Usage Cases Section were extracted from NYC Open Data and the images are from the actual version of UrbanProfiler.

[4] The computer used has 64 cores Intel Xeon E5-4640 2.40GHz with 1TB RAM.

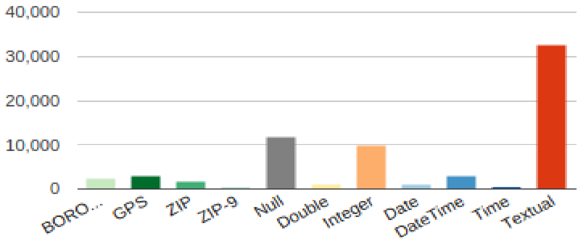


Figure 1: Count of columns by most detected type.

Profiler detected 3,508 as *Geo*, 415 as *Temporal*, 2,605 as *Numeric*, and 8,828 as *Null*.

4. USAGE SCENARIOS

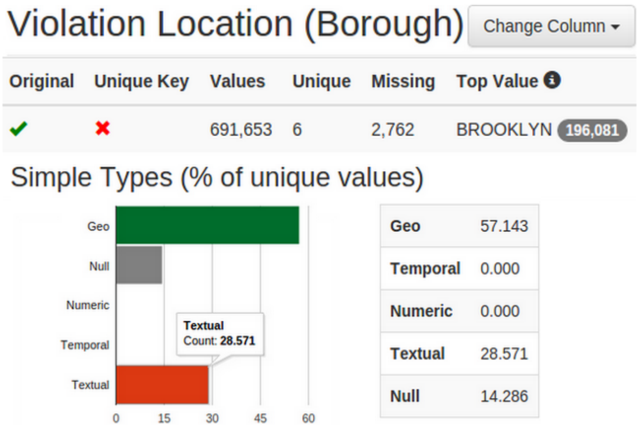
In what follows, we present examples that illustrate how UrbanProfiler can help users quickly find and explore datasets in a portal.

Consider a researcher that wants to analyze the geographic distribution of datasets from NYC Open Data. Using UrbanProfiler, she analyzes the Search Map and notices that some datasets are not limited to the city of New York, and are spread over other cities and states. Figure 2 shows the geographical area covered by the datasets. The system shows the bounding box of all datasets on the same map to facilitate the comparison. In addition, the layered bounding boxes makes the color of the area with more datasets stronger. Using UrbanProfiler, she creates a filter on the map to select only datasets with data outside NYC.

With the map showing the areas of all datasets, she can better understand the limits and distribution of each one. Hovering the mouse over a dataset's id (on the list on the right side), the system highlights its bounding box and shows its name and description. Note that the area covered by the largest dataset is the whole United States. To check if this information is correct or caused by a misplaced record, she clicks on the dataset to inspect its details. The system shows a heatmap of this dataset (Figure 3). The color of the areas go from green to red representing the density of records in the area. Hovering the mouse over an area on the heatmap shows the number of records in that area. By the heatmap, she verifies that there is more than one record outside NYC, some records are in the middle of the country and even on the West Coast. With the help of UrbanProfiler, she concludes that there is real data outside the city of New York and go back on the Search Map to continue her research.

Now consider another researcher interested in government data. Using the UrbanProfiler, she selects only datasets in the *City and Government* category and with data in *Brooklyn*. She finds one dataset particularly interesting and inspects its details.

Initially, she obtains general information about the dataset. In the *Metadata Tab*, UrbanProfiler displays high-level information, such as number of values and missing data, number of columns, detected types and update frequency.[5] To get an overview of the dataset, she checks the *Charts Tab*. In this view, UrbanProfiler presents charts of column types found in the dataset and also the number unique and missing values of each column. To drill down, she clicks on the *Columns Tab* and inspects the columns. Figure 4 shows this tab, with a list of all columns and associated information. There, she sees that the dataset has a unique key (Ticket Number) but it has 11% of missing values. This suggests that she may have to discard some data from this dataset. Continuing the analysis, she discovers that the column *Violation Location (Borough)* has an interesting feature: 57% of the values are Geo-Borough and only 15% are missing. But then, what happened to the rest of the column? She zooms in that column to find out. Figure 5 shows the column details. In this page, it is possible to see that Brooklyn is the most frequent value, and even with most of the values being Geo-Borough, almost 15% are Null and 30% are other textual values that not fit the borough valid values.


Violation Location (Borough)

Original	Unique Key	Values	Unique	Missing	Top Value
✓	✗	691,653	6	2,762	BROOKLYN 196,081

Simple Types (% of unique values)

Geo	57.143
Temporal	0.000
Numeric	0.000
Textual	28.571
Null	14.286

Figure 5: Details of the column Violation Location (Borough).

5. RELATED WORK

Data Near Here (DNH) [11, 12, 13] is a dataset search tool. The general idea of the approach is similar to ours: create a catalog offline and then use it for online search. However, DNH is specifically designed for oceanographic data with a fixed schema, in contrast, UrbanProfiler was designed to support very diverse, urban data. DNH also does not provide an automatic type detection. While DNH has a feature-rich search component, it relies heavily on the dataset schema. This information is often not available for urban data, where datasets often come from multiple sources and in different formats. We also aid analysis by providing information and visualizations specifically for urban data.

Another tool that is more closely related to ours is Profiler [10]. Profiler focuses on data quality assessment, specifically, data anomalies. It uses type inference to identify potential data quality issues in tabular data, and coordinated multi-view visualizations to help analysts to assess anomalies. To identify the type of the column they use RegEx, dictionary lookups, and range constraints. The columns can have values of more than one type, but values that are not from the main type are considered anomalies. In this regard, their tool is similar to ours. However, an important difference between UrbanProfiler and Profiler is that their approach requires validation from a specialist and only works with a single dataset. Even though the tool automates the cleaning process, the choice of which cleaning algorithm to use must be selected manually. In contrast, our tool is completely automated with optional user interventions.

6. CONCLUSION

In this paper we presented UrbanProfiler, a tool that automatically extracts detailed information about the contents of diverse datasets, and creates a catalog that can be used to support rich, discovery queries over the data.

We applied UrbanProfiler to over 3,000 datasets published in the NYC Open Data portal. Our preliminary results are promising, and

[5] The complete list is on Section 2.1

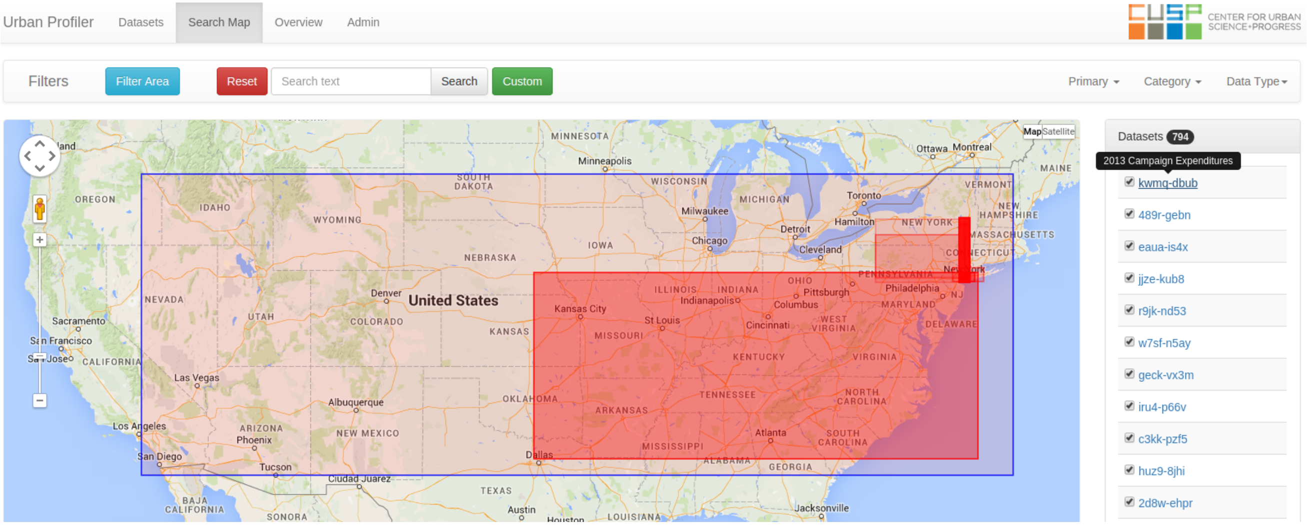


Figure 2: Map showing all datasets. On the right side is a list of the datasets. Highlighted in blue is the area of the selected dataset.

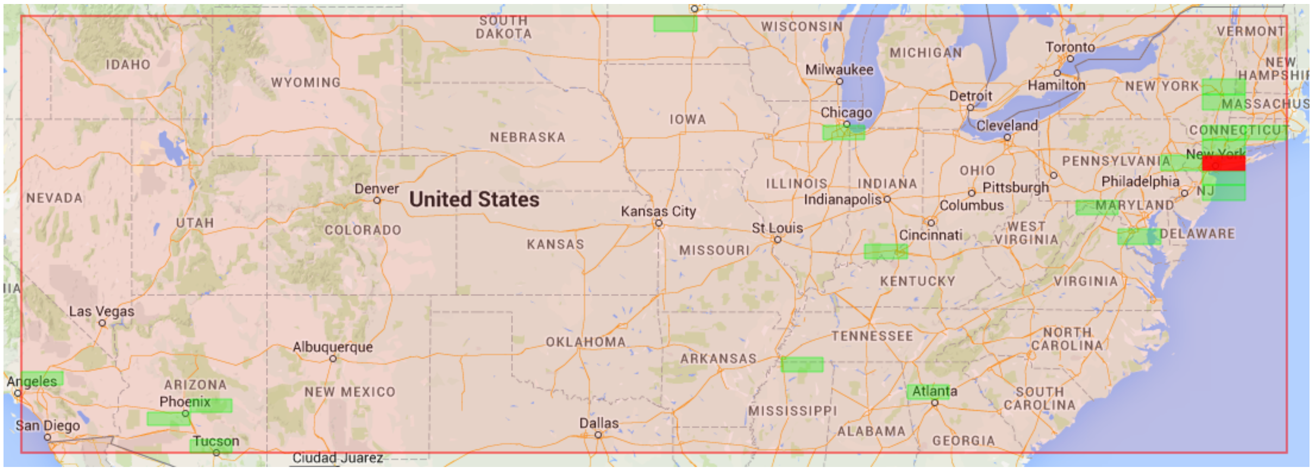
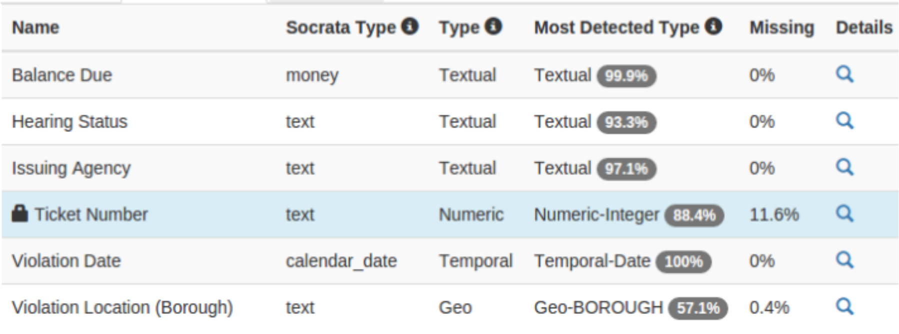


Figure 3: Heatmap of a dataset. The bigger red rectangle is the bounding box and the smaller ones are regions with records.



Name	Socrata Type ℹ	Type ℹ	Most Detected Type ℹ	Missing	Details
Balance Due	money	Textual	Textual 99.9%	0%	🔍
Hearing Status	text	Textual	Textual 93.3%	0%	🔍
Issuing Agency	text	Textual	Textual 97.1%	0%	🔍
🔒 Ticket Number	text	Numeric	Numeric-Integer 88.4%	11.6%	🔍
Violation Date	calendar_date	Temporal	Temporal-Date 100%	0%	🔍
Violation Location (Borough)	text	Geo	Geo-BOROUGH 57.1%	0.4%	🔍

Figure 4: Columns of 2002 dataset. The primary keys is indicated by the blue row with the lock symbol.

indicate that UrbanProfiler provides useful information and helps streamline data discovery and exploration.

UrbanProfiler is still under development and we are working on a number of improvements. Notably, we would like to support different levels of granularity for the spatial index and the use of machine learning to improve automatic type detection.

7. ACKNOWLEDGMENTS

This work was supported in part by a Google Faculty Award, an IBM Faculty Award, the Moore-Sloan Data Science Environment at NYU, the NYU School of Engineering, the NYU Center for Urban Science and Progress, and NSF award CNS-1229185.

8. REFERENCES

[1] L. Barbosa, K. Pham, C. Silva, M. Vieira, and J. Freire. Structured open urban data: Understanding the landscape. *Big Data*, 2(3), 2014.

[2] CKAN. `http://ckan.org`. [Online; accessed 28-May-2014].

[3] I. Ellen, J. Lacoe, and C. Sharygin. Do foreclosures cause crime? *Journal of Urban Economics*, 74:59–70, 2013.

[4] B. Ferris, K. Watkins, and A. Borning. OneBusAway: Results from providing real-time arrival information for public transit. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1807–1816, New York, USA, 2010. ACM.

[5] A. Feuer. The mayor's geek squad. `http://www.nytimes.com/2013/03/24/nyregion/mayor-bloombergs-geek-squad.html?pagewanted=all\&\_r=0`, March 2013.

[6] Freedom of information act (foia), 2014. http://www.foia.gov.

[7] B. Goldstein and L. Dyson. *Beyond Transparency: Open Data and the Future of Civic Innovation*. Code for America Press, San Francisco, USA, 2013.

[8] A. Grossman and A. Sun. MTA swipes show subway trends. http://online.wsj.com, October 2011.

[9] J. Höchtl and P. Reichstädter. Linked open data - a means for public sector information management. In *Electronic Government and the Information Systems Perspective*, volume 6866 of *Lecture Notes in Computer Science*, pages 330–343. Springer, Berlin Heidelberg, 2011.

[10] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: integrated statistical analysis and visualization for data quality assessment. In G. Tortora, S. Levialdi, and M. Tucci, editors, *International Working Conference on Advanced Visual Interfaces, AVI '12, Capri Island, Naples, Italy, May 22-25, 2012, Proceedings*, pages 547–554. ACM, 2012.

[11] D. Maier, V. M. Megler, and K. Tufte. Challenges for dataset search. In S. S. Bhowmick, C. E. Dyreson, C. S. Jensen, M. Lee, A. Muliantara, and B. Thalheim, editors, *Database Systems for Advanced Applications - 19th International Conference, DASFAA 2014, Bali, Indonesia, April 21-24, 2014. Proceedings, Part I*, volume 8421 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2014.

[12] V. M. Megler and D. Maier. When big data leads to lost data. In *Proceedings of the 5th Ph.D. Workshop on Information and Knowledge*, PIKM '12, pages 1–8, New York, NY, USA, 2012. ACM.

[13] V. M. Megler and D. Maier. Data near here: Bringing relevant data closer to scientists. *Computing in Science and Engineering*, 15(3):44–53, 2013.

[14] City of chicago data portal. `https://data.cityofchicago.org`.

[15] Data catalogs a comprehensive list of open data catalogs from around the world. `http://datacatalogs.org`.

[16] Nyc opendata. `https://nycopendata.socrata.com`.

[17] San francisco data. `https://data.sfgov.org`.

[18] A. E. Schwartz, I. G. Ellen, I. Voicu, and M. H. Schill. The external effects of place-based subsidized housing. *Regional Science and Urban Economics*, 36(6):679 – 707, 2006.

[19] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, H. Wendy, and M. Schraefel. Linked open government data: Lessons from data.gov.uk. *IEEE Intelligent Systems*, 27(3):16–24, 2012.

[20] Socrata. `http://www.socrata.com`. [Online; accessed 28-May-2014].