Auctus: A Dataset Search Engine for Data Discovery and Augmentation

Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati and Juliana Freire
New York University
{s.castelo,remi.rampin,aecio.santos,aline.bessa,fchirigati,juliana.freire}@nyu.edu

ABSTRACT

The large volumes of structured data currently available, from Web tables to open-data portals and enterprise data, open up new opportunities for progress in answering many important scientific, societal, and business questions. However, finding relevant data is difficult. While search engines have addressed this problem for Web documents, there are many new challenges involved in supporting the discovery of structured data. We demonstrate how the Auctus dataset search engine addresses some of these challenges. We describe the system architecture and how users can explore datasets through a rich set of queries. We also present case studies which show how Auctus supports data augmentation to improve machine learning models as well as to enrich analytics.

1 INTRODUCTION

With the push towards transparency and openness, scientists, governments, and companies have been increasingly publishing structured datasets on the Web. Google Dataset Search alone indexes over 30 million datasets [3]. The availability of these data creates opportunities to answer many important scientific, societal, and business questions.

The Need for Data Discovery. While data are abundant, finding *relevant data* is difficult. Data are spread over a large number of sites and repositories. Recognizing this challenge, a number of approaches have been proposed to *organize and index data collections* [6], from domain-specific repositories such as NYCOpenData, which collects datasets from the various NYC agencies [18], general data portal infrastructure and data lakes [10, 22], to Google Dataset Search, which indexes a wide range of datasets published on the Web [17]. While these present a significant step towards simplifying data discovery, they have an important limitation: they only support simple, keyword-based search queries over published dataset metadata. This greatly limits a user's ability to express information needs. In addition, published metadata is often incomplete, and in
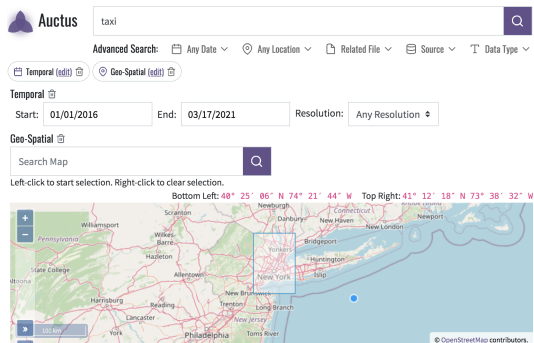
Figure 1: Searching for datasets that mention *"taxi"* and contain records within the NYC area for the 2016-2021 period.

some cases it can be inconsistent with the actual data. Thus, relying solely on the metadata also limits the discoverability of datasets.

Our Approach. Auctus is an *open-source dataset search engine that was designed to support data discovery and augmentation.* It supports a rich set of discovery queries: in addition to keyword-based search, users can specify spatial and temporal queries, data integration queries (i.e., searching for datasets that can be concatenated to or joined with a query dataset), and they can also pose complex queries that combine multiple constraints, as shown in Figure 1. These queries are enabled in part by a data profiler [9] that we developed to extract useful information from the actual datasets. This includes not only summaries (or sketches) of column contents, but also their data types. In particular, it detects columns that contain spatial and temporal information. This information is then used to construct indices that support efficient query evaluation. Users can explore large dataset collections through an easy-to-use interface that guides them in the process of specifying complex queries. To help users identify relevant datasets, Auctus displays snippets that summarize the contents of datasets.

Auctus has been developed in the context of the DARPA D3M program [8], and it is currently being used in production and to support research of different groups in the project [7, 15, 20]. In this demo paper, we give an overview of the architecture and features of Auctus (Section 2), and discuss a few use cases that we will present during our demonstration (Section 3). Demo visitors will be able to interact directly with Auctus to query over 19,000 datasets.

Related Work. Our work is related to methods for data profiling [1], and for the discovery of joinable [5] and unionable [16] tables — which we use in our system. Moreover, Auctus is related to multiple data discovery systems [14, 27]. In particular, it is related to JUNEAU [26], a system that supports search capabilities in data science environments such as Jupyter notebooks. While our work shares some features and goals with these systems, Auctus provides a unique search interface that combines the familiar user
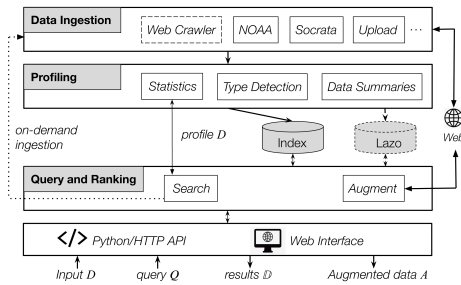
Figure 2: Overview of Auctus' architecture.

interface (UI) of Web search engines with visual metaphors of modern data analytics tools such as Tableau [23]. Auctus' UI allows users not only to perform expressive queries that include multiple constraints, but also allows users to make sense of the tables returned as search results. Further, it supports the materialization of joins and unions into CSV and D3M file formats, and communication through a REST API that allows for integration with external interactive data analytics systems [4, 20, 21].

2 THE AUCTUS SYSTEM

2.1 Auctus Architecture

The high-level architecture of Auctus is depicted in Figure 2. In what follows, we describe its key components.

Data Ingestion. Auctus makes use of plugins to retrieve datasets from repositories using their APIs. This makes the system extensible and able to ingest data from many different sources. Currently, it supports Socrata [22] (a platform for open government data), Zenodo [25] (open-access data repository), the World Bank Open Data [24] (datasets on global development), among others. Auctus also allows users to upload their own datasets.

Profiling. Once datasets are ingested, Auctus profiles them to infer relevant metadata necessary to support discovery queries as well as to construct dataset summaries for result presentation. The profiler performs different tasks, including: *type detection*, i.e., it detects whether columns correspond to categorical, numerical, spatial, or temporal attributes; type-dependent *statistics computation* (e.g., frequency of values, mean, and variance for numerical values), and *data summarization* (see below).

Storing Data and Data Summaries. Since Auctus was designed to serve as a dataset discovery system, it stores dataset summaries instead of the full datasets. These summaries are concise and sufficient to create the indices required to answer all queries supported by the system. Currently, Auctus creates summaries for categorical, numerical, spatial, and temporal attributes. The data summaries generated by Auctus are represented by the ranges of their corresponding attributes. During the search phase, Auctus uses these summaries to estimate the size of the intersection between two attributes, concluding whether a join is feasible. To estimate join intersections well, Auctus captures fine-grained ranges for data attributes by using the clustering algorithm k-means. This strategy produces the desired results while also being efficient. Auctus also stores provenance information to enable the retrieval of the datasets. This allows the system to perform augmentations, and

users to download the datasets. Auctus can also cache datasets for efficiency purposes.

Indices. After the metadata are generated, including data summaries, they are indexed in an Elasticsearch [13] server. Numerical and temporal summaries are indexed using range data types, and spatial summaries are indexed using geo-shape data types. To support joinable dataset search, we use Lazo [5], which is a method for set-overlap search based on MinHash sketches and locality-sensitive hashing (LSH), to build an index for categorical attributes.

Querying and Ranking. Auctus supports queries that combine multiple constraints including keywords, temporal, spatial, data type, and data source. The system also supports *data integration queries*: Given an input dataset D_Q, Auctus allows the user to search for datasets that can be concatenated to or joined with D_Q. The dataset D_Q must first be uploaded using a provided API or selected from the set of indexed datasets. The system will then generate a dataset profile which is used to probe those indices that support join and union. Finally, the lists of matching datasets from different indices are merged, ranked, and returned as search results.
Join Search. To find other datasets that can be joined with D, Auctus first searches, for each attribute a of D, which other attributes, in the index that corresponds to a's data type (e.g., temporal attributes searched in the Elasticsearch range index, or categorical attributes probed against the Lazo index), have summaries intersecting the summary of a. Every dataset that has at least one intersecting attribute is a potential join result.
Union Search. To find datasets that can be concatenated with D, the indices are searched for any dataset that has attributes with the same data types present in D, as well as similar names. Name similarity is computed with the *fuzzy query* in Elasticsearch. Results from union searches may match only a subset of D's attributes.
Ranking. The results of join and union searches, after being filtered based on query Q, are ranked and returned as $\mathbb{D}$. Join results are ranked based on the intersection of the summaries; union results are ranked based on the Levenshtein similarity between the names of the matching attributes.

Augmentation. Besides providing search capabilities for data augmentation, Auctus also performs the actual augmentation. Users can choose a dataset $R \in \mathbb{D}$, and Auctus will materialize it (using the provenance annotated in the metadata) and perform the join or union operation with D, returning the new, augmented dataset A. If multiple attribute pairs match between R and D for a join operation, users can choose which pair(s) they want for the join. For temporal and spatial joins, the attributes are translated into the same resolution before the augmentation.

2.2 Auctus User Interface

Auctus provides an easy-to-use interface where users can query for datasets, explore search results including data exploration and augmentation options, explore ingested datasets, and upload new datasets.

Data Discovery Queries. Users can query indices by specifying keywords and constraints using various filters (see Figure 3(A)).
Temporal and Spatial Search. Users search for datasets by specifying a date range (Figure 1) – datasets containing a temporal column that overlaps with that range will be retrieved. They can refine the
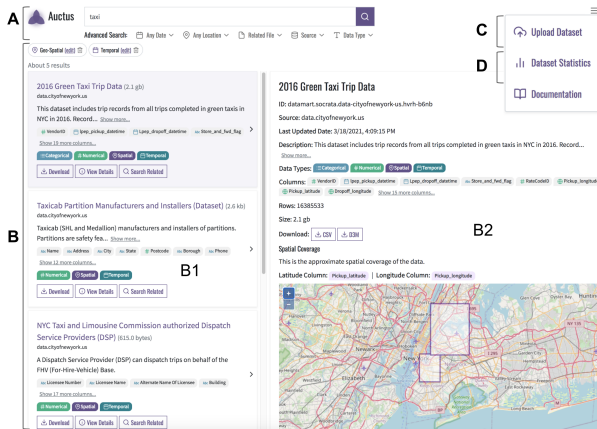
Figure 3: Components of `Auctus`' user interface: keyword and filter-based search box (A); search results (B); dataset-snippets (B1); dataset summary (B2); dataset upload (C); dataset collection statistics (D).

search by specifying a temporal resolution (e.g., year). To perform a spatial search, users can either draw a bounding box around a geographical area on the map (Figure 1), or specify an administrative area, which `Auctus` translates into a polygon. Datasets containing spatial attributes that overlap with the search polygon are returned. *Source filter.* The source filter allows users to restrict the data sources of interest, and only results from the selected sources are retrieved. *Data Type filter.* The data type filter allows users to search for datasets based on the types of their attributes, e.g., categorical, numerical, spatial and temporal, which were inferred by the profiler. *Data Integration Queries.* The related file filter allows users to find datasets that can augment a given input dataset (see Figure 4(a)(A)). The user can upload the input dataset or select a dataset from a set of search results. Result snippets for data integration queries include an "augment options" button, that allows users to request and customize the augmentation to be performed by `Auctus`.

Result Presentation and Exploration. Unlike Web documents which can be summarized using short text snippets, datasets have many different facets that the user must consider to determine their relevance. Thus, an important challenge for Auctus is how to present search results. Figure 3 shows the results for the query specified in Figure 1. On the left, there is a list of search results displayed as snippets (Figure 3(B1)). Users can select a dataset to inspect its details (Figure 3(B2)), which include description, source, attribute names and types, as well as a summary of the dataset's contents. For spatial datasets, a visualization showing the geographical extent covered by the dataset is displayed. The summary also includes a sample of the dataset records and statistics about its columns (Figure 4(b)). The tabs above the dataset sample allow for the visualization of these statistics in different levels of detail. For example, *Detail View* shows the columns' value distributions. *Augmentation.* If the user provides an input file through the Related File filter, the snippets display a button "Augment Options" under each result on the left. If the user clicks on this button, the augmentation interface will be shown (see Figure 4(a)(B)). If the result is a union, users can select matched pairs of columns, and the columns from the result (right) will be appended to the matching

columns of the input data (left). If the result is a join (an example is shown in Figure 4(a)(B)), users can select the columns to be matched at the top of the augmentation screen. Those are the "join keys" that determine which records from the input data should be matched with records from the selected result. Under this area, there is an interface to include columns from the selected result into the augmented table. Users can grab the desired columns from the "available columns" area, drag them to the "included after merge" box, and drop them over the aggregation function they wish to use for that column.

Uploading Data and Curating Metadata. Users can add new datasets to `Auctus`. After loading a data file, `Auctus` automatically infers its data types with our datamart-profiler library [9]. As any method for type inference, our profiler is not fool-proof and can derive incorrect results. To address that, `Auctus` enables users to correct data types manually, and to provide additional annotations for the columns. `Auctus` also displays a dataset sample so that the user can verify if the detected data types are correct, and check the uploaded data. Additionally, `Auctus` provides support for custom metadata fields (e.g., data source or grid size). Since these fields can vary widely for different applications, we defined a flexible configuration schema that allows users to customize them for different deployments. The upload page is shown in Figure 4(c).

2.3   Scalability and APIs

`Auctus` has been implemented with scalability in mind: the search engine is entirely containerized using Docker [11]. Each data discovery plugin is an independent container, allowing multiple plugins to be executed in parallel. `Auctus` can also spin up as many profiling and query containers as required in response to load. Our system can be accessed via a Web UI [2] or programmatically via Python and REST APIs. So far, we have indexed over 19,000 datasets.

3   USE CASES

Visitors will be able to see how `Auctus` is used for different use cases (presented below), and interact with it through its interface.

Bicycle Usage Prediction. Predicting the number of daily bicycle trips is an important step towards implementing better policies for this means of transportation in NYC. Consider that an expert from the NYC department of transportation decides to build a model using the East River Bicycle Trips data. This department installed automated counters in all East River bridges, which provide the number of bikes crossing them on a daily basis. This number can be used as a proxy for the overall bicycle usage in NYC. Besides daily bicycle counts, the data also contains maximum daily temperatures for NYC. Unfortunately, her initial dataset only covers bicycle usage for April 2020, which is not very informative. To take into account a larger period of time, she uses `Auctus` to find compatible data for more months of 2020, which can then be concatenated to her original input data within our system. When using a random forest regressor to predict the number of bicycle trips using daily temperatures as a model feature, she obtains an R^2 score close to 0.25. She is relying on a single feature to make her predictions, so her model is probably underfitting. To improve it, she uses `Auctus` again to find and augment her current input data with

Figure 4: (a) A data integration query in [A], and the augmentation interface in [B]. (b) Data Summary views. (c) Upload page.

weather information, including daily rainfall levels. The R^2 score then increases to approximately 0.56, which represents a substantial improvement. A video demonstrating this use case is available at http://bit.ly/auctus-demo.

Conflict Forecasting. While analyzing conflict forecasting problems, researchers often use predictive modeling to guide policy-making decisions, and to assess and compare theories of conflict [12]. In this context, it is crucial to identify new data sources, merge those data, and evaluate the contribution of different features. Furthermore, most conflicts materialize as events, and if they occur in heterogenous spatio-temporal levels and need to be analyzed in tandem, data integration can be challenging. Consider that a researcher is studying conflict events in Africa. She uses the grid [19] dataset for Africa with conflict events aggregated into quarterly counts to predict state-based conflicts. Since protest events data can be very useful, and since her grid dataset does not have any measure of protests, she decides to use Auctus to discover and join this type of dataset to her input data. She uses "protest" as a keyword, and also uploads the grid dataset as an input query in Auctus. After running the query, the highest ranked result is the Social Conflict Analysis Database (SCAD). To verify its suitability, she explores the dataset through the "View Detail" tab. After inspecting the Summary Data view, she realizes that SCAD contains information not only on protests, but also on other destabilization events. She can also quickly see that it captures events in Africa because of the map visualizations, and also finds spatio-temporal matches. She then checks the possible augmentation options moving in to the augmentation interface. As shown in in Figure 4(a)(B), Auctus automatically detects joinable columns. She then selects the temporal and spatial levels of her interest, and uses aggregation functions to count the number of destabilizing events and to sum the number of fatalities. Next, she presses the augmentation button and all the events in SCAD become aggregated into the grid dataset. The augmented dataset can be used to explore new research questions, such as whether the number of destabilizing events at time t is a predictor for state-based violence at time $t+1$.

ACKNOWLEDGMENTS

REFERENCES

[1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. 2015. Profiling relational data: a survey. *The VLDB Journal* 24, 4 (2015), 557–581.
[2] Auctus Dataset Search Engine. 2021. https://auctus.vida-nyu.org/.
[3] Omar Benjelloun, Shiyu Chen, and Natasha Noy (Eds.). 2020. *Google Dataset Search by the Numbers.* https://arxiv.org/abs/2006.06894
[4] Aline Bessa, Sonia Castelo, Rémi Rampin, Aécio Santos, Mike Shoemate, Vito D'Orazio, and Juliana Freire. 2021. An Ecosystem of Applications for Modeling Political Violence. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD/PODS '21)*. Association for Computing Machinery, 2384–2388.
[5] R. Castro Fernandez, J. Min, D. Nava, and S. Madden. 2019. Lazo: A Cardinality-Based Method for Coupled Estimation of Jaccard Similarity and Containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 1190–1201.
[6] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis Daniel Ibáñez-Gonzalez, Emilia Kacprzak, and Paul T. Groth. 2019. Dataset Search: A Survey. *CoRR* abs/1901.00735 (2019).
[7] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David Karger. 2020. ARDA: Automatic Relational Data Augmentation for Machine Learning. *Proc. VLDB Endow.* 13, 9 (May 2020), 1373–1387.
[8] Data-Driven Discovery of Models (D3M). 2019. https://www.darpa.mil/program/data-driven-discovery-of-models.
[9] Datamart Profiler Library. 2021. https://pypi.org/project/datamart-profiler/.
[10] Dataverse. 2021. https://dataverse.harvard.edu/.
[11] Docker. 2021. https://www.docker.com/.
[12] Vito D'Orazio. 2020. Conflict forecasting and prediction. In *Oxford Research Encyclopedia of International Studies.*
[13] Elasticsearch. 2021. https://www.elastic.co/.
[14] R. Castro Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker. 2018. Aurum: A Data Discovery System. In *ICDE '18*. 1001–1012.
[15] Sungsoo Ray Hong, Sonia Castelo, Vito D'Orazio, Christopher Benthune, Aecio Santos, Scott Langevin, David Jonker, Enrico Bertini, and Juliana Freire. 2020. Towards Evaluating Exploratory Model Building Process with AutoML Systems. *arXiv preprint arXiv:2009.00449* (2020).
[16] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proc. VLDB Endow.* 11, 7 (2018), 813–825.
[17] Natasha Noy, Matthew Burgess, and Dan Brickley. 2019. Google Dataset Search: Building a Search Engine for Datasets in an Open Web Ecosystem. In *TheWebConf '19*.
[18] NYC Open Data. 2021. https://nycopendata.socrata.com.
[19] PRIO Grid. 2021. https://www.prio.org/Data/PRIO-GRID/.
[20] Aecio Santos, Sonia Castelo, Cristian Felix, Jorge Piazentin Ono, Bowen Yu, Sungsoo Hong, Claudio T. Silva, Enrico Bertini, and Juliana Freire. 2019. Visus: An interactive system for automatic machine learning model building and curation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA 2019*. Association for Computing Machinery.
[21] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing data science through interactive curation of ML pipelines. In *SIGMOD'19*. 1171–1188.
[22] Socrata. 2021. https://socrata.com/.
[23] Tableau. 2021. https://www.tableau.com/.
[24] World Bank Open Data. 2021. https://data.worldbank.org/.
[25] Zenodo. 2021. https://zenodo.org/.
[26] Yi Zhang and Zachary G Ives. 2019. Juneau: data lake management for Jupyter. *Proc. VLDB Endow.* 12, 12 (2019).
[27] Erkang Zhu, Ken Q Pu, Fatemeh Nargesian, and Renée J Miller. 2017. Interactive navigation of open data linkages. *Proc. VLDB Endow.* 10, 12 (2017), 1837–1840.