enRichMyData

^ INSID&S Lab

Department of Informatics, Systems and Communication
University of Milan - Bicocca

Semantic Data Enrichment: from Interactive Exploration to Scalable Deployment

Roberto Avogadro *, Flavio De Paoli ^, Dumitru Roman *, Matteo Palmonari ^

Part IV: Semantic Data Enrichment in Practice with Tools

*This work presented in this presentation has received funding from the European Union's Horizon 2020 research and innovation program under grant agreements No 732590 - **EW-Shopp** - and No 732003 – **euBusinessGraph** - and from the European Union's Horizon Europe research and innovation program under grant agreements No 101070284 - **enRichMyData**.*

Outline

- Part II: Semantic Data Enrichment, Applications and Requirements
  - Semantics and KGs for data enrichment
  - The *Link & Extend* enrichment paradigm
    - Interactive exploration and scalability
- Part III: Selected State-of-the-art
  - Data preparation solutions
    - The broader context of data preparation solutions
  - Scalable data pipelines
    - A quick introduction to solutions for scalability
  - Tabular data annotation
    - From heuristic techniques to generative LLMs
- Part IV: Semantic Data Enrichment in Practice with Tools
  - Service-based approach
    - Data model for interoperability
    - Service model for composability
  - Interactive definition of pipelines
    - Exploration with graphical UI
    - Pipeline definition with programmatic UI
  - Pipeline execution at scale
    - Execution with workflow managers (Argo & TAO)
  - Live demos
- Part V: Conclusions and Discussion
  - Wrap-up and take-home messages
  - Discussion

SemTUI framework

- Recall the concepts of transformation, reconciliation and extension
- Issue to build a system
  - Exploration: what data are available to support data preparation
  - Repeatability: how to define sequences of actions (pipelines)
  - Scalability: how effectively apply enrichment pipelines on big datasets
- A service-based approach
  - Define a set of services that can be exploited for data enrichment
    - Prepare pipelines on small data samples -> Browser-based GUI
    - Execute on large tables with Pyhton -> NoteBook
    - Execute on big tables with Docker/workflow manager -> TAO execution
- The SemTUI architecture
  - Create a service model to populate an enrichment ecosystem
  - Create multiple access to services (GUI, NoteBook, batch)

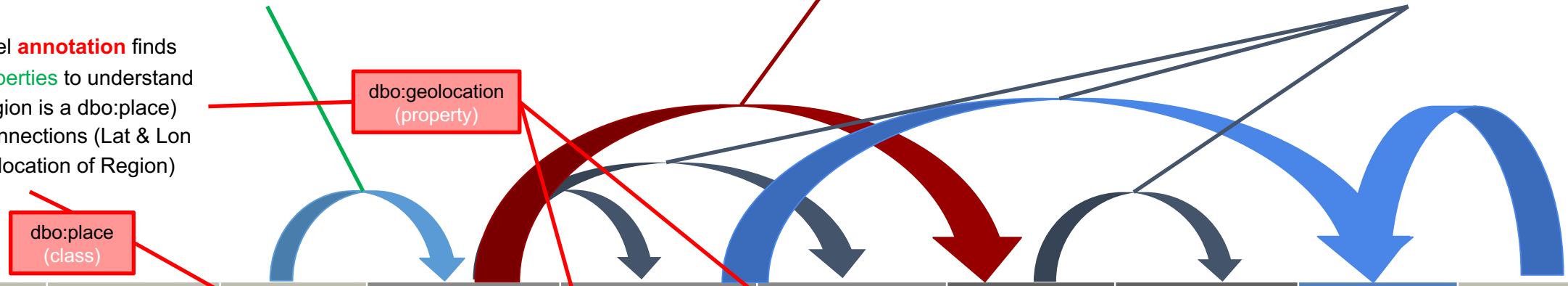
Data linking and Extension

Reconciliation finds a maching between a mention (Bavaria) and an entity (geo:2951839) in a target Knowledge Graph (Geonames) to overcome mismatches between consumer's (light gray table) and third-party data (weather provided by ECMWF) by setting shared identifiers

Knowledge Graphs provide **links** to bridge the gap across systems of identifiers (geo:2951839, wd:Q980 and geo:[48.76,11.42] refer to the same entity)

Identifiers allow **extension** by the addition of data from sources that generated the identifiers (Geonames ID to retrieve Lat & Lon – Wikidata ID to retrieve Area) and from third-party sources (Lat & Lon to retrieve data from ECMWF)

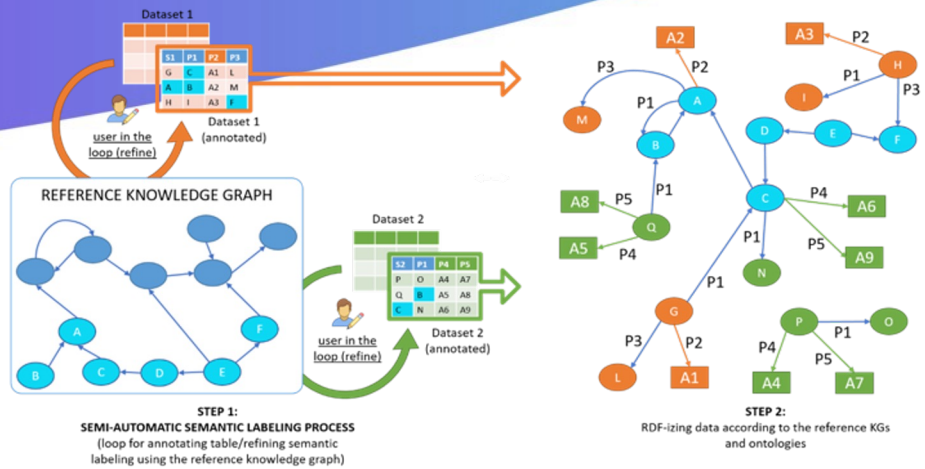
Schema-level **annotation** finds types and properties to understand content (Region is a dbo:place) and create connections (Lat & Lon are dbo:geolocation of Region)



Keyword	#im	City	Region	ID (Geonames)	Latitude (Geonames)	Longitude (Geonames)	ID (Wikidata)	Area (Wikidata)	Temp (ECMWF)	Date
194906	64	Altenburg	Thuringia	2822542	50.98763	12.43684	Q1205	45.6 km^2	18°	11/03/2017
517827	50	Inglostadt	Bavaria	2951839	48.76508	11.42372	Q980	133.35 km^2	17°	12/03/2017
459143	42	Berlin	Berlin	2950157	52.52437	13.41053	Q648102	891.68 km^2	17°	12/03/2017
891139	36	Munich	Bavaria	2951839	48.13743	11.57549	Q980	310.71 km^2	19°	11/03/2017
459143										0/03/2017

Example of data enrichment by composing different individual linking and extension services

Data enrichment



Data enrichment is composed of

- Manipulation
  - The content of cells must be in the right format to ensure interoperability
  - e.g., dates should be in ISO8601 format
- Reconciliation and linking
  - Entities must be uniquely identified with a shared schema to ensure interoperability
  - e.g., locations should be identified in GeoRSS (Geographically Encoded Objects for RSS feeds)
- Extension
  - Source tables are augments by adding new data from external sources
  - e.g., with locations and dates, we can add meteorological information

Data enrichment process

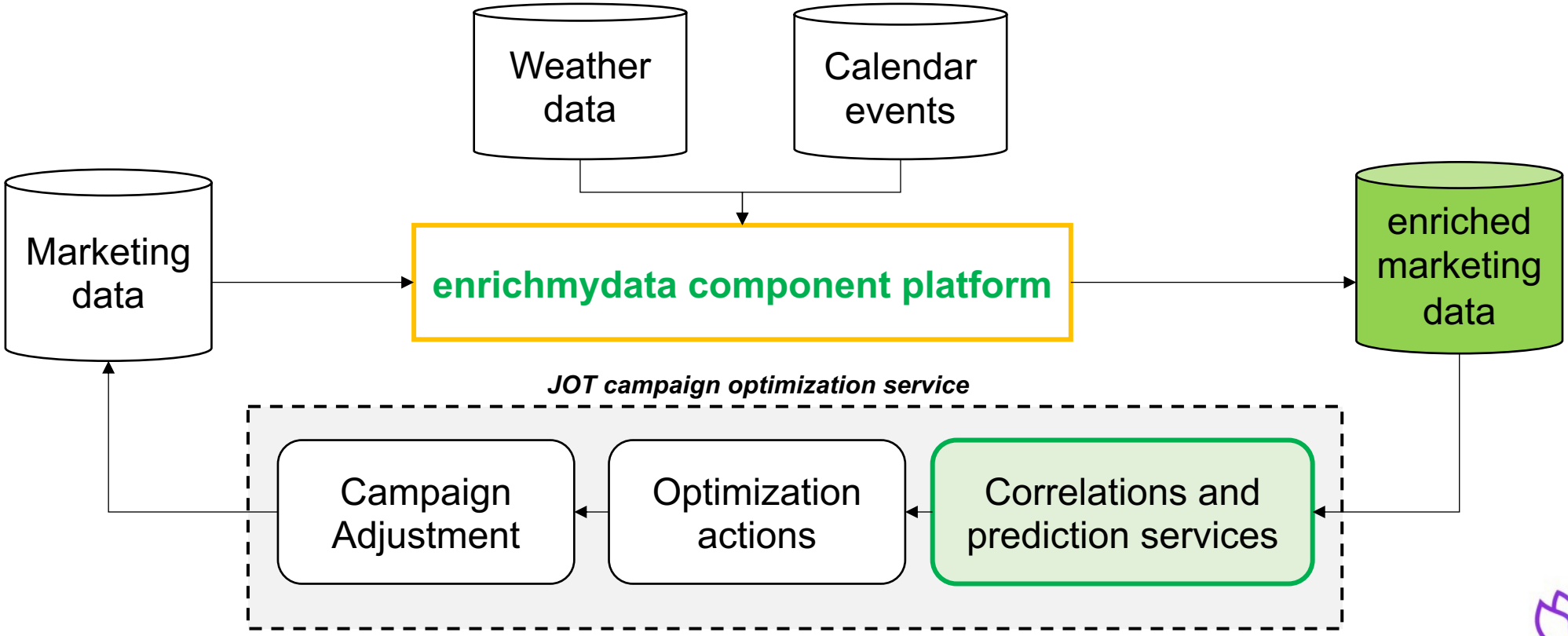
A tabular data enrichment process is modeled and executed as a pipeline

A tabular data enrichment process is built in reverse order

We need to

- Identify the external source for your data
- Identify what information are needed to access that source
- Modify (transform or reconcile) your source dataset

Marketing data Enrichment for smart-bidding optimization



Example of the Enriched Dataset

The dataset is splitted for a better visualization

DateId	CustomerId	CampaignId	AdGroupId	KeywordId	Keyword	QualityScore	Cpc	Impressions	Clicks	Countryid	City	CityId
20221122	9684689623	1990591626	71809765395	25678366	ssd 1tb		1000	3	5	2724	Alcantarilla	9048966
20221122	4403173460	14643997404	126859207853	872122684206	alquiler casa telde	8	20	1	3	2724	Aranjuez	9048987
20221122	4689307439	14790214318	126059172685	297306066693	canal sur andalucia directo	2	30000	9	4	2276	Schopfheim	1004036
20221220	4403173460	14643997374	126859189133	384930025120	callaghan hombre		150000	548	18	2276	Bad Tolz	1004110
20220210	4689307439	14790214318	126059172445	363110218461	mapa parcelario		180000	16	0	2724	Leioa	9049110
20220210	4403173460	14643997326	126859157013	307181814389	bighorn maxxis tyres	1	80000	19	1	2724	Lepe	9049111
20220210	2132505891	14305153810	127715629564	296452675334	south summit		420000	73	3	2840	Seward	1012912
20220210	4689307439	14790214318	126059172445	301552259405	como hacer ali oli		230000	6	2	2724	Torrevieja	9049224
20220210	9684689623	1990591626	71809765395	22778696	juegos juegos de ajedrez	10	440000	12	1	2724	San Isidro	9049193
20220210	4689307439	14790214318	126059172445	299634835438	casa alquiler altafulla		290000	146	2	2724	Oliva	9049152

MARKETING

WEATHER AND EVENTS

State	Event	Type	Latitude	Longitude	MaxTemp	MinTemp	AvgTemp	Sunrise	Sunset	Precipitation	RailSum	SnowSum
North Rhine-Westphalia	BLACK FRIDAY	SHOPPING	37.97174	-1,218,271	16,1	3,5	9,8	7:50	18:30	1	0	1
Saxony	BLACK FRIDAY	SHOPPING	2112345	23	NULL	NULL	NULL	5:26	19:20	2	2	0
Sitka	BLACK FRIDAY	SHOPPING	47.65105	8	9,4	4,3	6,8	6:30	17:23	5	4,3	0,7
Alabama	CHRISTMASS	FAMILY	47.761474	12	14	5,2	9,6	6:01	18:01	0	0	0
Arkansas	VALENTINES DAY	LOVE/FRIENDSHIP	43.333271	-3	17,2	7,4	12,3	5:49	19:40	3,5	3	0,5
Arizona	VALENTINES DAY	LOVE/FRIENDSHIP	37.254503	-7	22,1	14,4	18,2	7:30	17:50	0	0	0
Yolo County	VALENTINES DAY	LOVE/FRIENDSHIP	60.104168	-149	1,1	-5,9	-2,4	6:45	18:10	0,8	0,8	0
Valencian Community	VALENTINES DAY	LOVE/FRIENDSHIP	37.985608	-1	17,4	6,5	12	5:59	19:30	0	0	0
Andalusia	VALENTINES DAY	LOVE/FRIENDSHIP	28.07617	-17	21,8	15,2	18,5	7:01	17:40	0	0	0
Community of Madrid	VALENTINES DAY	LOVE/FRIENDSHIP	38.917975	0	15,1	7,1	11,1	5:08	18:15	0	0	0

OpenMeteo Service

- We are interested in some meteorological parameters to enrich (extend) our source marketing data
  - max temperature, min temperature, precipitation, …
- We find a candidate data source accessible via API that need to be tested with our data
- From the documentation we understand that we need
  - geographic coordinates (lat, lng) of the location of interest
  - dates (yyyy-mm-dd) in ISO 8601 format

  to get historical information on meteorological parameters

OpenMeteo Service

- In the source table
  - Locations are represented by City names and State names
    - There are unique ids from Google, but they cannot serve the purpose
  - Dates are represented by a sequence of figures (yyyymmdd)
- We need
  - Reconcile entities, the locations *(city,state) -> (georss:lat,lng)*
  - Modify literals, the dates *(yyyymmdd) -> (yyyy-mm-dd)*

DateId	CustomerId	CampaignId	AdGroupId	KeywordId	Keyword	QualityScore	Cpc	Impressions	Clicks	Countryid	City	CityId	State
20221122	9684689623	1990591626	71809765395	25678366	ssd 1tb		1000	3	5	2724	Alcantarilla	9048966	North Rhine-Westphalia
20221122	4403173460	14643997404	126859207853	872122684206	alquiler casa telde	8	20	1	3	2724	Aranjuez	9048987	Saxony
20221122	4689307439	14790214318	126059172685	297306066693	canal sur andalucia directo	2	30000	9	4	2276	Schopfheim	1004036	Sitka
20221220	4403173460	14643997374	126859189133	384930025120	callaghan hombre		150000	548	18	2276	Bad Tolz	1004110	Alabama
20220210	4689307439	14790214318	126059172445	363110218461	mapa parcelario		180000	16	0	2724	Leioa	9049110	Arkansas
20220210	4403173460	14643997326	126859157013	307181814389	bighorn maxxis tyres	1	80000	19	1	2724	Lepe	9049111	Arizona
20220210	2132505891	14305153810	127715629564	296452675334	south summit		420000	73	3	2840	Seward	1012912	Yolo County
20220210	4689307439	14790214318	126059172445	301552259405	como hacer ali oli		230000	6	2	2724	Torrevieja	9049224	Valencian Community
20220210	9684689623	1990591626	71809765395	22778696	juegos juegos de ajedrez	10	440000	12	1	2724	San Isidro	9049193	Andalusia
20220210	4689307439	14790214318	126059172445	299634835438	casa alquiler altafulla		290000	146	2	2724	Oliva	9049152	Community of Madrid

The JOT use case

Live demo

A service-based approach

A service-based approach

- Web services can be exploited to support enrichment tasks
  - The approach aims at providing a model to
    - integrate existing services/datasets
    - support the development of new services/datasets
- Mastering semantic-based principles and practices requires advanced skills
  - The approach aims at providing a model to
    - facilitate the use of semantic-based services/datasets
- Engineering data integration requires advanced skills
  - The approach aims at providing a model to
    - facilitate the composition of services / integration of datasets
    - support the definition of pipelines

A service-based approach

- The approach aims at providing a model to
  - integrate existing services/datasets
  - support the development of new services/datasets
- SemTUI provides a unified interface for enrichment services
  - Clients are service agnostic (they can call any service with the same protocol)
  - Clients can be of different nature and scope (exploration, pipeline definition or execution)

  - facilitate the use of semantic-based services/datasets
  - facilitate the composition of services / integration of datasets
  - support the definition of pipelines

A service-based approach

- The approach aims at providing a model to
  - integrate existing services/datasets
  - support the development of new services/datasets
  - facilitate the use of semantic-based services/datasets
- SemTUI provides a unified access to services
  - Semantic details are (often) embedded in the services or in the backend
  - Clients can be of different nature and scope (exploration, pipeline definition or execution)

  - facilitate the composition of services / integration of datasets
  - support the definition of pipelines

A service-based approach

- The approach aims at providing a model to
  - integrate existing services/datasets
  - support the development of new services/datasets
  - facilitate the use of semantic-based services/datasets
  - facilitate the composition of services / integration of datasets
- SemTUI provides a GUI that facilitates exploration and testing
  - The SemTUI backend provides direct and unified access to services
  - Clients can identify the needed services and define the tasks to enrich their data (typically conducted on sample data)
  - support the definition of pipelines

A service-based approach

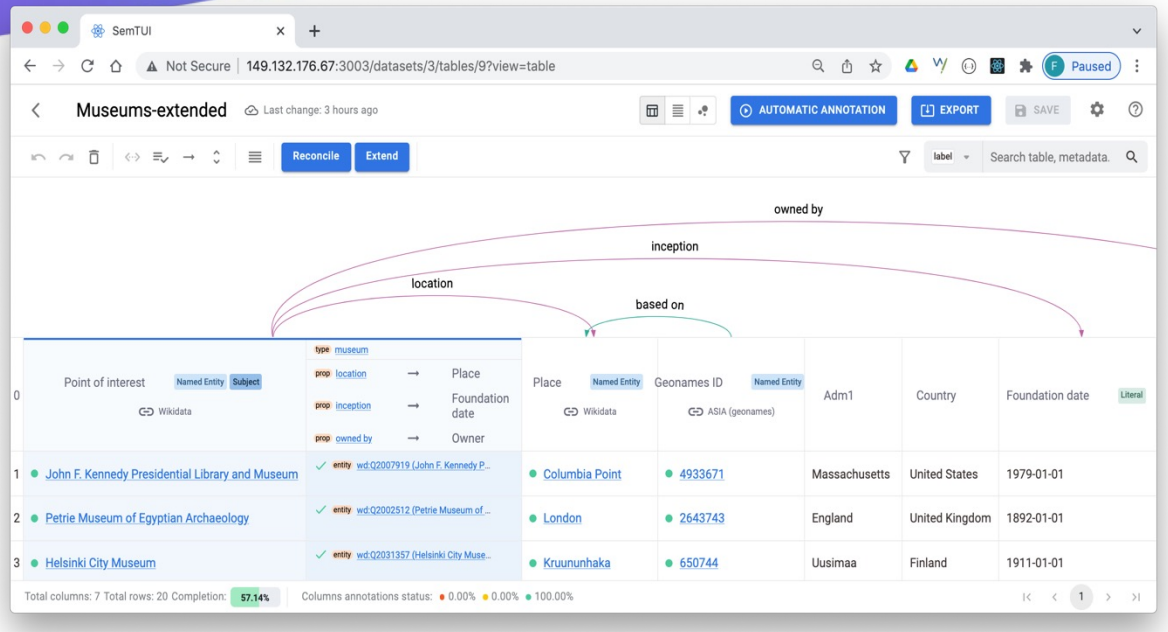
- The approach aims at providing a model to
  - integrate existing services/datasets
  - support the development of new services/datasets
  - facilitate the use of semantic-based services/datasets
  - facilitate the composition of services / integration of datasets
  - support the definition of pipelines
- SemTUI provides a programmatic interface to define and test pipelines
  - The SemTUI backend provides direct and unified access to services
  - Clients can organize a pipelines for their data in a NoteBook through a Python library (typically conducted on large data)

SemTUI framework

SemTUI – Interactive Semantic Enrichment of Tabular Data

- UI accessing external services
  - Complete semantic table annotations
  - Reconciliation/linking services
    - Linking: Wikidata (Alligator)
    - Linking: Wikidata (OpenRefine)
    - Geocoding: coordinates (GeoNames)
    - Geocoding: coordinates (HERE)
    - Linking: Atoka (SpazioDati)

    … based on W3C Specs
  - Extension services
    - Wikidata
    - Weather extension (OpenMeteo)
    - Distances and routes (HERE)
    - Atoka-extension (SpazioDati)
    - …

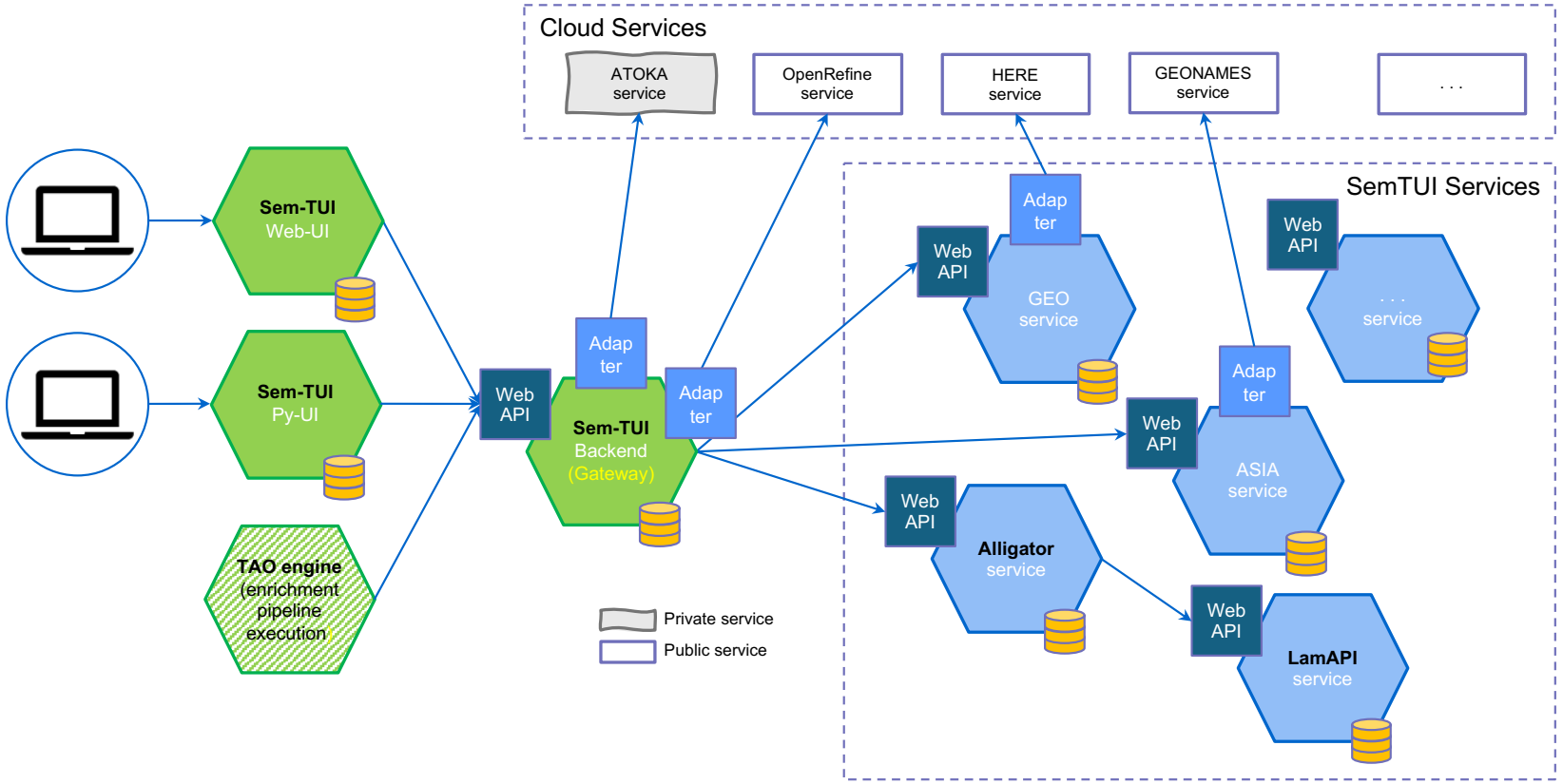


Support to *Linking – Revision – Extension* of tabular data

- Graphical view & revision of annotations
  - Global and specific annotation rendering
  - Single cell editing / annotation revision
  - Column annotation revision

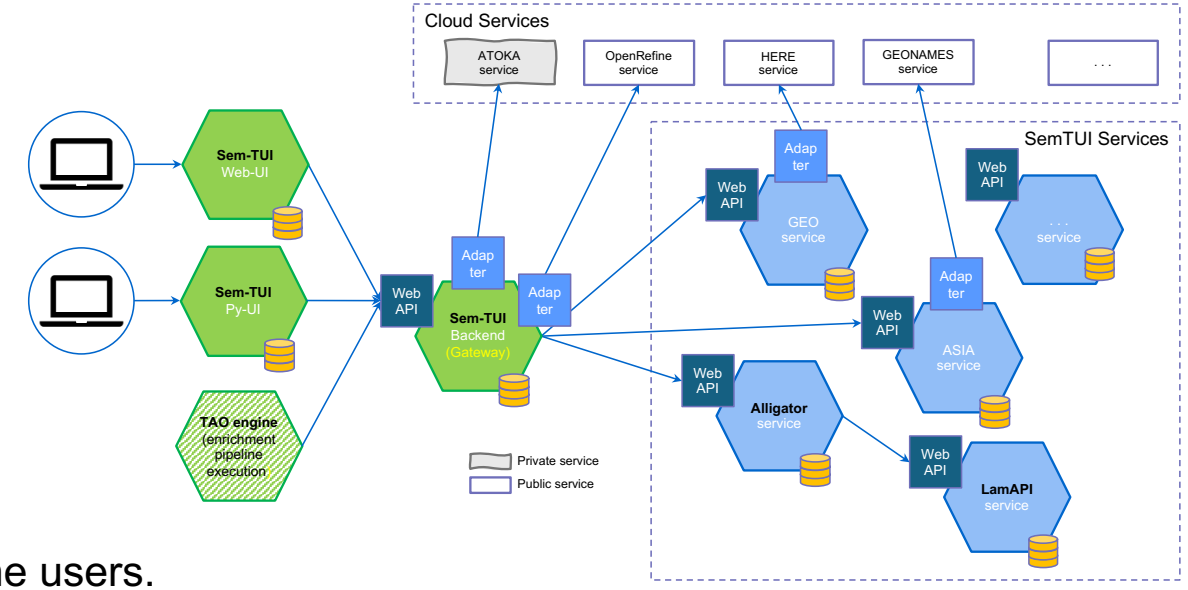
Ripamonti, M., De Paoli, F., & Palmonari, M. (2022). SemTUI: a Framework for the Interactive Semantic Enrichment of Tabular Data. *arXiv preprint arXiv:2203.09521.*

SemTUI architecture



SemTUI architecture

- Classification of Services
  - Self-contained Services
    - They implement functions that do not rely on external APIs.
    - The execution is under the control of the users.
  - Composite Services
    - They implement functions that rely on external APIs.
    - The execution is partially under the control of the users.
  - Proxy Services
    - They act as proxies for external APIs.
    - The execution depends on the service providers.



SemTUI architecture

- Classification of Services
  - Self-contained Services
    - They implement functions that do not rely on external APIs.
    - The execution is under the control of the users.
  - Composite Services
    - They implement functions that rely on external APIs.
    - The execution is partially under the control of the users.
  - Proxy Services
    - They act as proxies for external APIs.
    - The execution depends on the service providers.

Examples

- Content cell modification
  - Transform dates in ISO8601 format
- Extension with annotation properties
  - Create new columns with IDs and names
- Reconciliation with Alligator
  - Rely on a service that provides candidates
- Reconciliation with OpenRefine
  - Requests are sent to external API

Service model

- The backend is the core of the system
  - Provide the proper level of abstraction
  - Decouples client applications from actual services
- Requirements
  - Facilitating the inclusion of existing services
  - Compliancy with standard or shared data models
- Characteristics
  - API based on HTTP
  - Data model based on W3C proposal

Data model: reconciliation

- The reconciliation responses provided by the services are W3C compliant
- Reconciliation Query Responses

Reconciliation Service API

A protocol for data matching on the Web

Draft Community Group Report 11 April 2024

https://reconciliation-api.github.io/specs/draft/

`id`
The identifier of the candidate entity;

`name`
The name of the candidate entity;

`description`
The entity description *MAY* optionally be included;

`type`
The types of the candidate entity;

`score`
An optional numeral indicating how well this candidate entity matches the query: a higher score indicates a better match. If candidates are scored, the reconciliation service *SHOULD* sort candidates in decreasing score order. If `standardizedScore` is set to `true` in the service manifest, this value *MUST* be between 0 and 100 (inclusive);

`features`
An optional array of matching features;

`match`
A boolean matching decision, which indicates whether the service considers this candidate good enough to be chosen as a correct match.

Data model: annotations

- W3C proposal defines the reconciliation data format
- Annotations requires a format to represent an enriched table
  - A table is an array of rows, one of which is the header row (the schema row)

A table

```
[
  {
    th0: {
      tc0
    },
    ...
    {
      tcM
    }
  },
  {
    row1
  },
    ...
  {
    rowN
  }
 ]
```

Data model: annotations

- W3C proposal defines the reconciliation data format
- Annotations requires a format to represent an enriched table
  - A table is an array of rows, one of which is the header row (the schema row)
  - A header cell includes types and properties of the column

A header cell

```
"label": "Point of interest",
  "kind": "entity",
  "role": "subject",
  "metadata": [
    {
      "id": "wd:Q960648",
      "name": "point of interest",
      "type": [
        {
          "id": "wd:Q33506",
          "match": true,
          "name": "museum",
          "score": 100
        }
      ],
      "property": [
        {
          "id": "wd:P276",
          "obj": "Place",
          "match": true,
          "name": "location",
          "score": 100
        }, ...
      ]
    }
  ],
  "context": [ {
      "prefix": "wd:",
      "uri": "https://www.wikidata.org/entity/"
    }
```

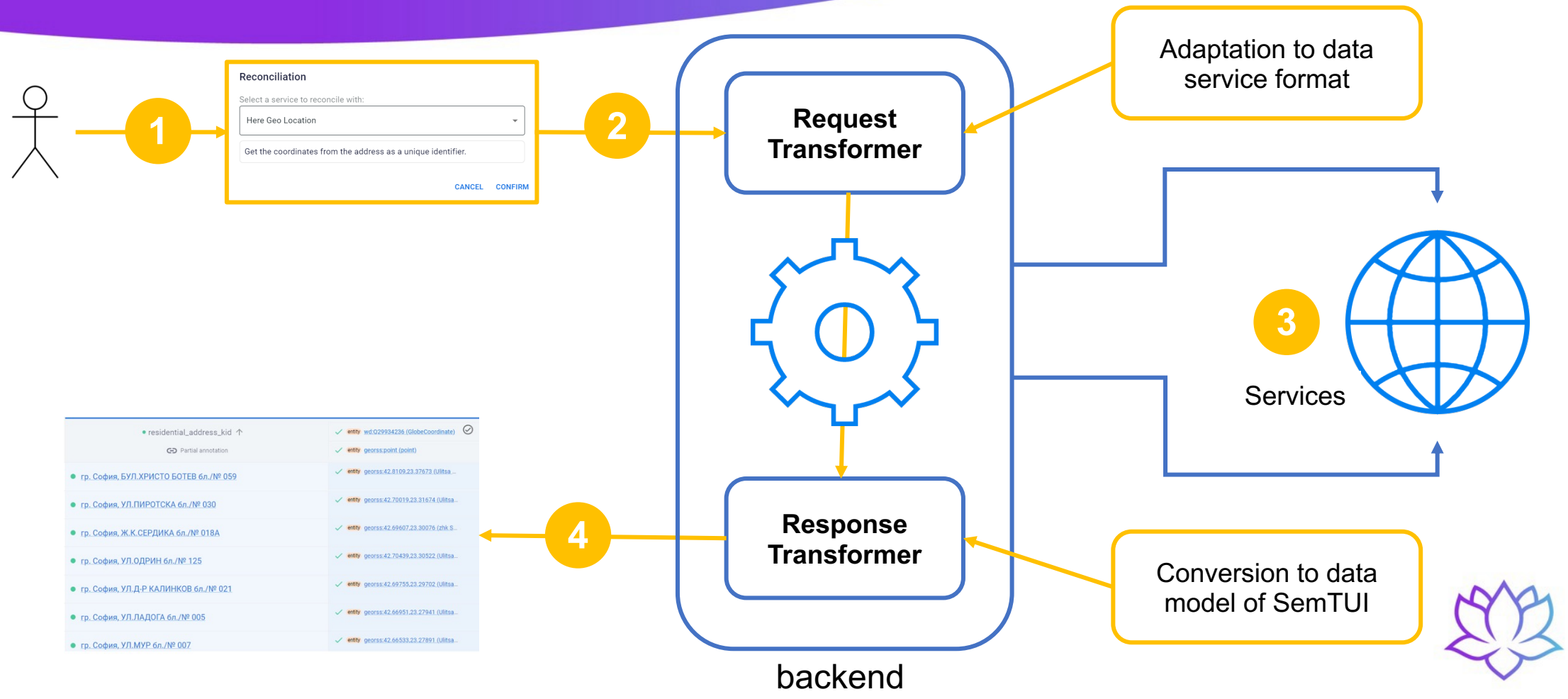
Data model: annotations

- W3C proposal defines the reconciliation data format
- Annotations requires a format to represent an enriched table
  - A table is an array of rows, one of which is the header row (the schema row)
  - A header cell includes types and properties of the column
  - A row is an object composed of table cell that includes reconciliation data

A row of table cells

```
{
  "Point of interest": {
    "label": "John F. Kennedy Presidential Library and Museum",
    "metadata": [
      {
        "name": "John F. Kennedy Presidential Library and Museum",
        "id": "wd:Q2007919",
        "match": true,
        "score": 100
      }
    ]
  },
  "Place": {
    "label": "Columbia Point",
    "metadata": [
      {
        "name": "Columbia Point",
        "id": "wd:Q1112511",
        "match": true,
        "score": 100
      }
    ]
  },
  ...
}
```

Service model



Service model and LMM

- Service design with LLM
  - A precise definition of APIs and data models opens to the use of LLM-based solution to create backend transformers
- Advantage
  - Less skills required
  - Lower the entry barrier

Experiment

- Ask ChatGPT 4.0 to write a Request Transformer to
  - adapt a multiple query to a service that support single requests
  - collects the responses in a single object compliant with the SemTUI data model

LMM query (prompt 1/3)

```
Given:
const prefix = "geo";
const items = {
                "City": {
                  "geo:3117735": [
                    "r0"
                  ],
                  "geo:3128760": [
                    "r1"
                  ]
                }
              }

const access_token = // my username
const endpoint = 'http://api.geonames.org/getJSON?style=full&'

I need to create multiple queries for geoID, which are the parts after the prefix and the
':' (e.g., 3117735 and 3128760) in items with a prototype call:

url = `${endpoint}geonameId=${geoID}&username=${access_token}
(E.g., http://api.geonames.org/getJSON?geonameId=3117735&style=full&username=demo)
```

LMM query (prompt 2/3)

That return a response like this:

```
[
  {
    "geonameId": 3117735,
    "name": "Madrid",
    "lat": "40.4165",
    "lng": "-3.70256",
    "countryCode": "ES",
    "countryName": "Spain",
    // Additional details
  },
  {
    "geonameId": 5128581,
    "name": "New York",
    "lat": "40.71427",
    "lng": "-74.00597",
    "countryCode": "US",
    "countryName": "United States",
    // Additional details
  }
]
```

And collects the responses and creates a structure named response like this

```
{
  "entities": {
    "3117735": {
      "name": "Madrid",
      "lat": "40.4165",
      "lng": "-3.70256",
      "countryCode": "ES",
      "countryName": "Spain",
      // Additional details

    },
    "5128581": {
      "name": "New York",
      "lat": "40.71427",
      "lng": "-74.00597",
      "countryCode": "US",
      "countryName": "United States",
      // Additional details
    }
}
```

LMM query (prompt 3/3)

the code must be included in:

```
export default async (req) => {
...
  return {
    res: response
  };
}
```

LMM response

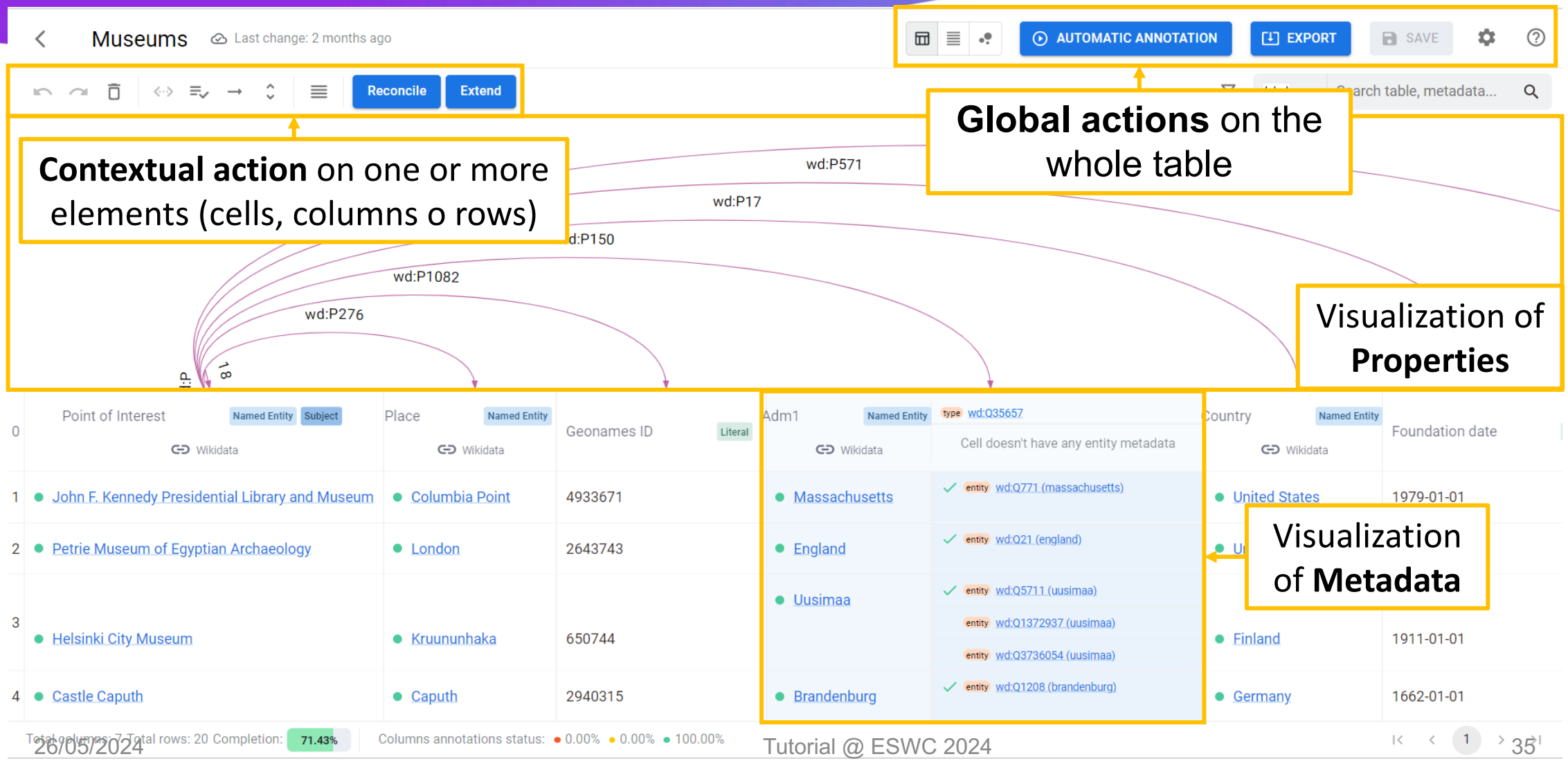
```
export default async (req) => {

  const { items } = req.processed;

  // Function to fetch data for a given GeoNames ID
  const fetchDataForGeoID = async (geoID) => {
    try {
      const url = `${endpoint}geonameId=${geoID}&username=${access_token}`;
      const response = await axios.get(url);
      console.log(`*** response from geonames:
${JSON.stringify(resposnse)}`);
      return response.data;
    } catch (error) {
      console.error(`Error fetching data for GeoNames ID ${geoID}:`, error);
      return null;
    }
  };
```

We copied and pasted the blue code into the Request Transformer, and it worked!

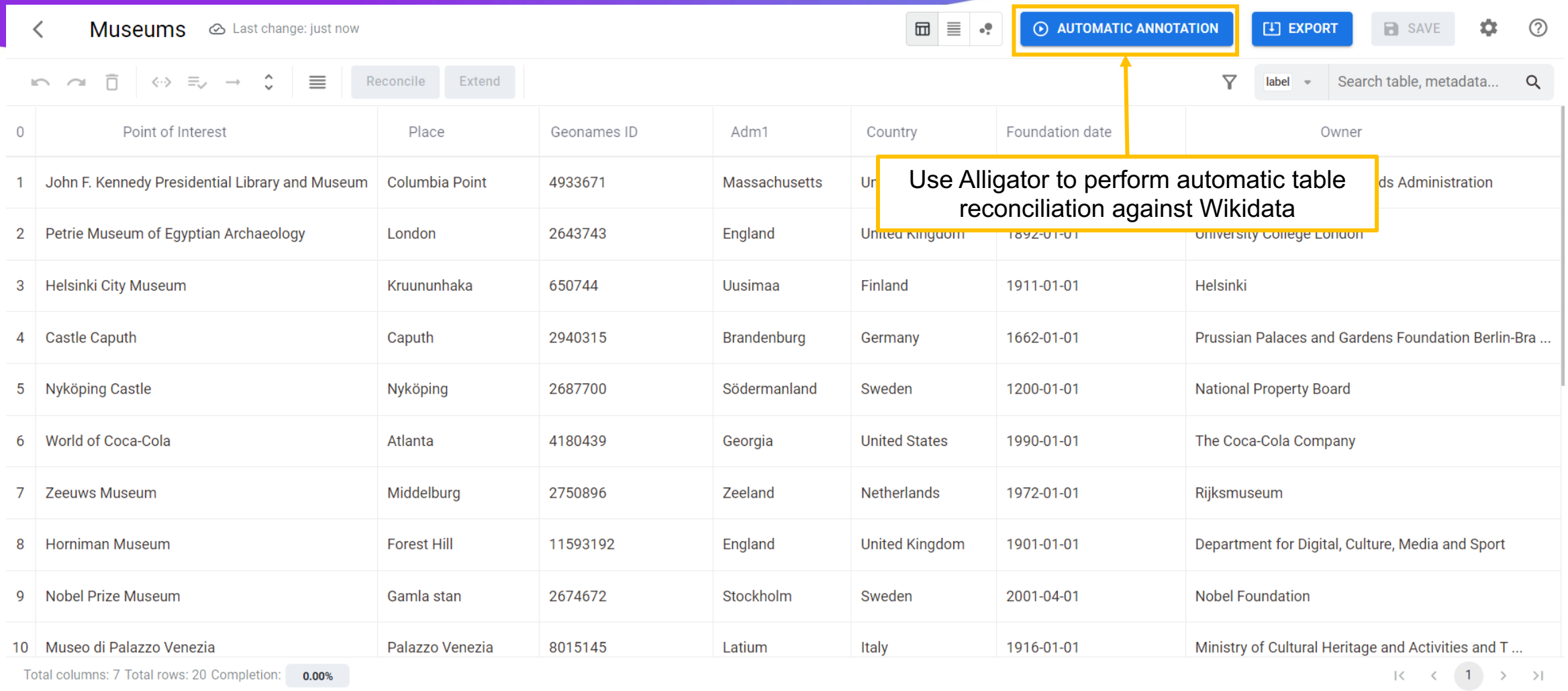
```
// Extract GeoNames IDs from the items object and make API
requests
  const fetchAllGeoData = async (items) => {
    const entities = {};
    const promises = [];

    for (const category in items) {
      for (const geoId in items[category]) {
        const geoID = geoId.split(':')[1];
        promises.push(
            fetchDataForGeoID(geoID).then(data => {
              if (data) {
                entities[geoID] = data;
              }
            })
        ); } }
  await Promise.all(promises);
    return {
      entities
    };
  };

  try {
    const result = await fetchAllGeoData(items);
    return {
      res: result    };
  } catch (error) {
    console.error('Error fetching data:', error);
    return {
      res: {}};
  } };
```

The Graphical User Interface

The Graphical User Interface



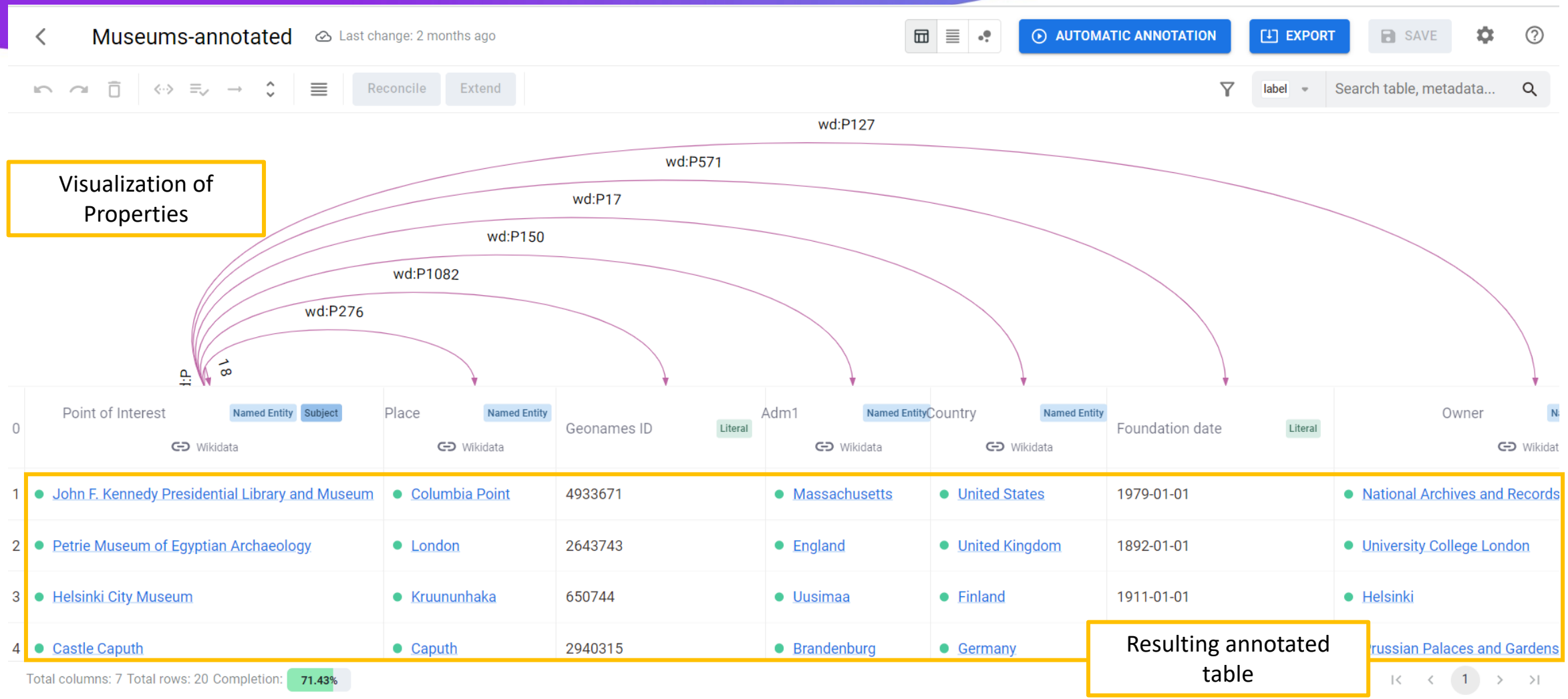
0 - Automatic Table Annotation



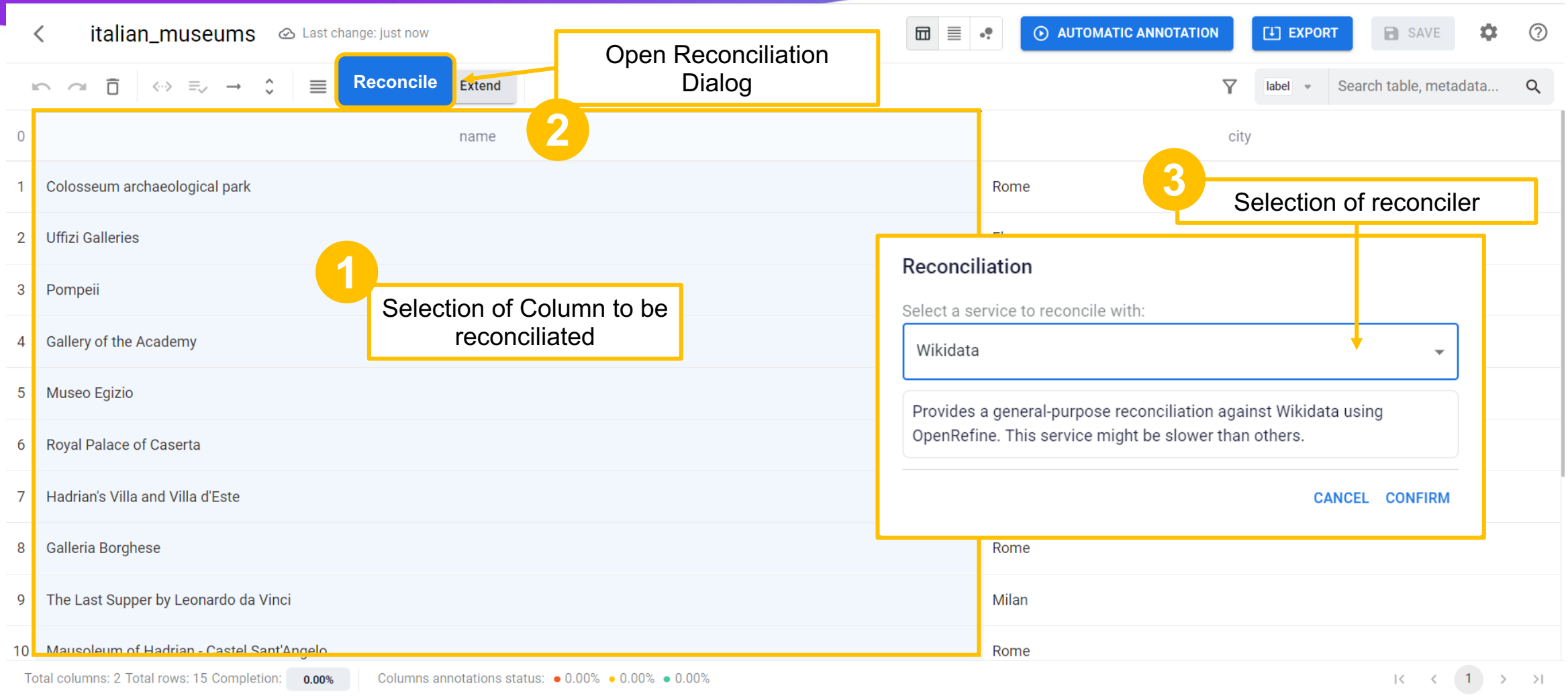
0	Point of Interest	Place	Geonames ID	Adm1	Country	Foundation date	Owner
1	John F. Kennedy Presidential Library and Museum	Columbia Point	4933671	Massachusetts	Ur		ds Administration
2	Petrie Museum of Egyptian Archaeology	London	2643743	England	United Kingdom	1892-01-01	University College London
3	Helsinki City Museum	Kruununhaka	650744	Uusimaa	Finland	1911-01-01	Helsinki
4	Castle Caputh	Caputh	2940315	Brandenburg	Germany	1662-01-01	Prussian Palaces and Gardens Foundation Berlin-Bra ...
5	Nyköping Castle	Nyköping	2687700	Södermanland	Sweden	1200-01-01	National Property Board
6	World of Coca-Cola	Atlanta	4180439	Georgia	United States	1990-01-01	The Coca-Cola Company
7	Zeeuws Museum	Middelburg	2750896	Zeeland	Netherlands	1972-01-01	Rijksmuseum
8	Horniman Museum	Forest Hill	11593192	England	United Kingdom	1901-01-01	Department for Digital, Culture, Media and Sport
9	Nobel Prize Museum	Gamla stan	2674672	Stockholm	Sweden	2001-04-01	Nobel Foundation
10	Museo di Palazzo Venezia	Palazzo Venezia	8015145	Latium	Italy	1916-01-01	Ministry of Cultural Heritage and Activities and T ...

Total columns: 7 Total rows: 20 Completion: 0.00%

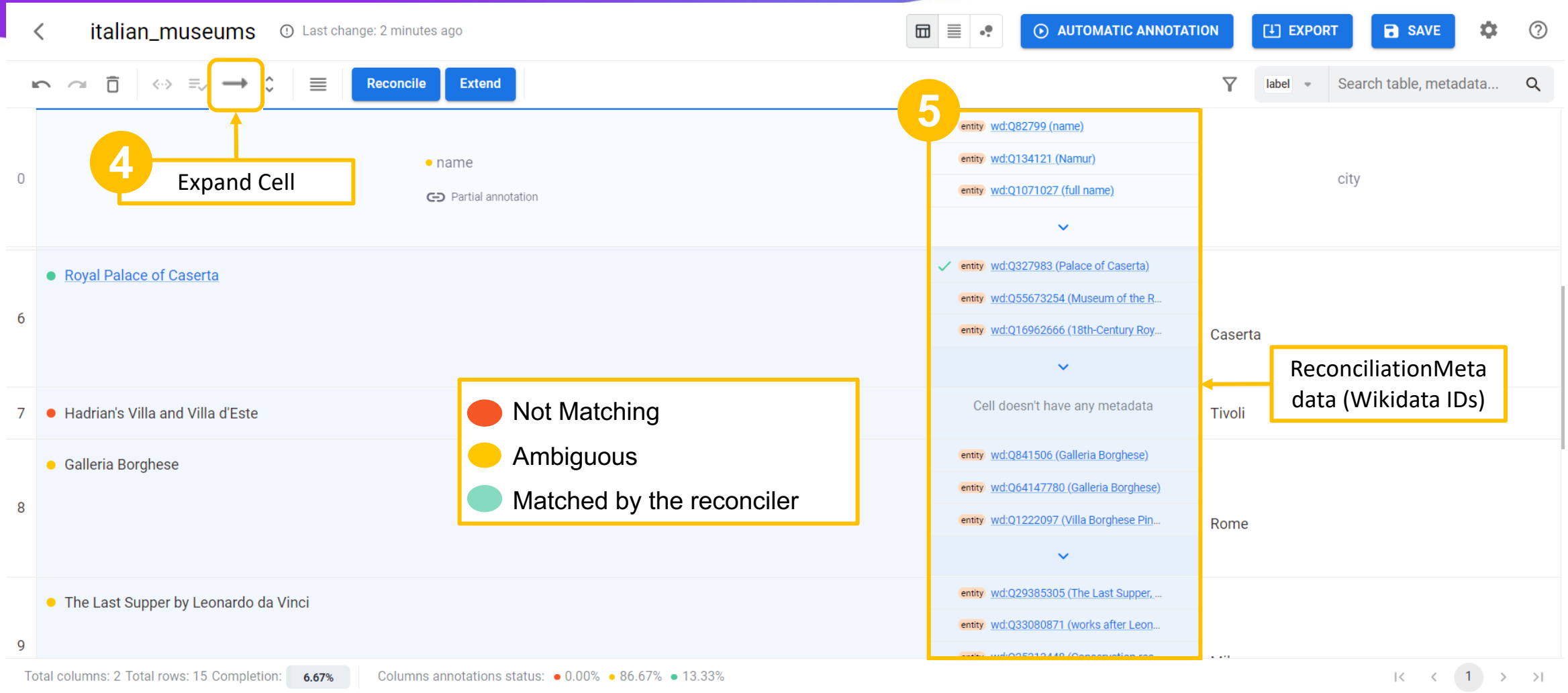
0 - Automatic Table Annotation – Results



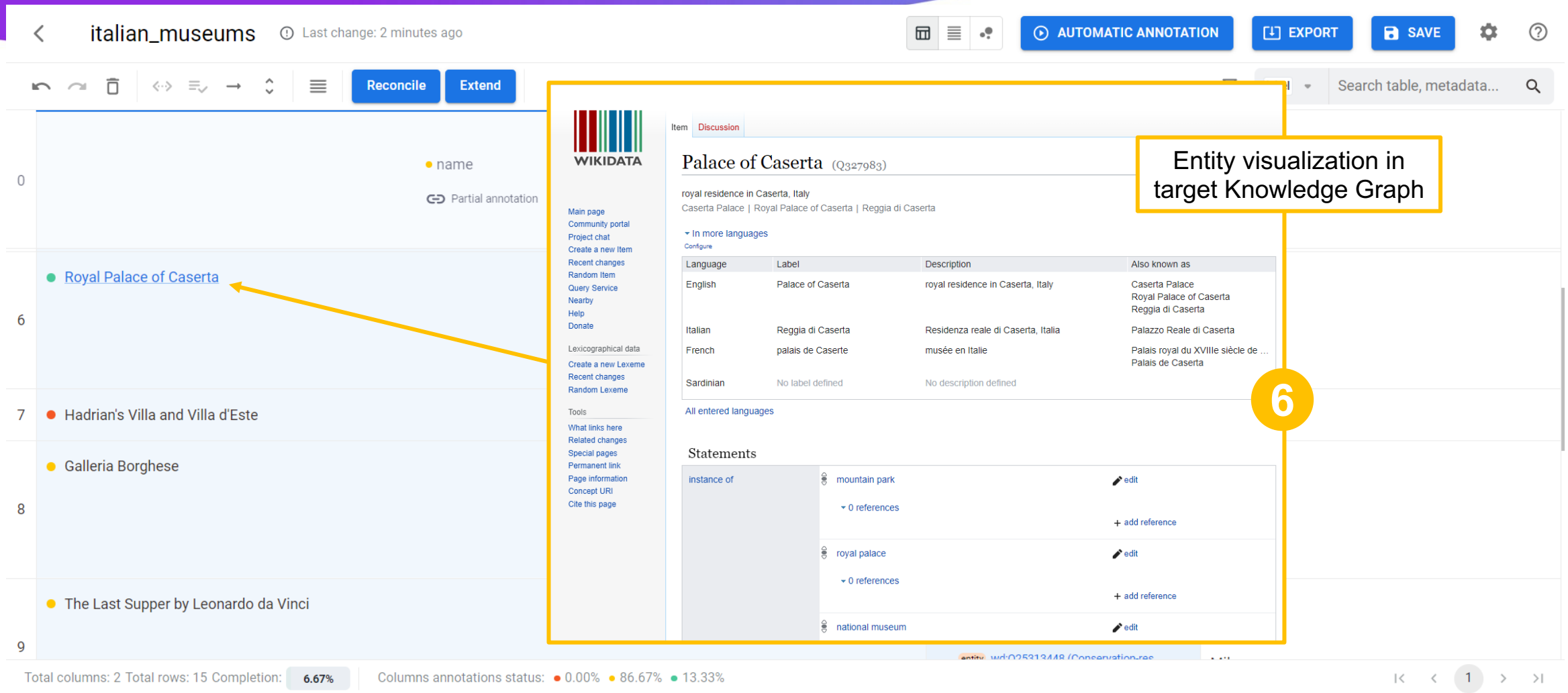
1 - Column annotation



1 - Column annotation – Results

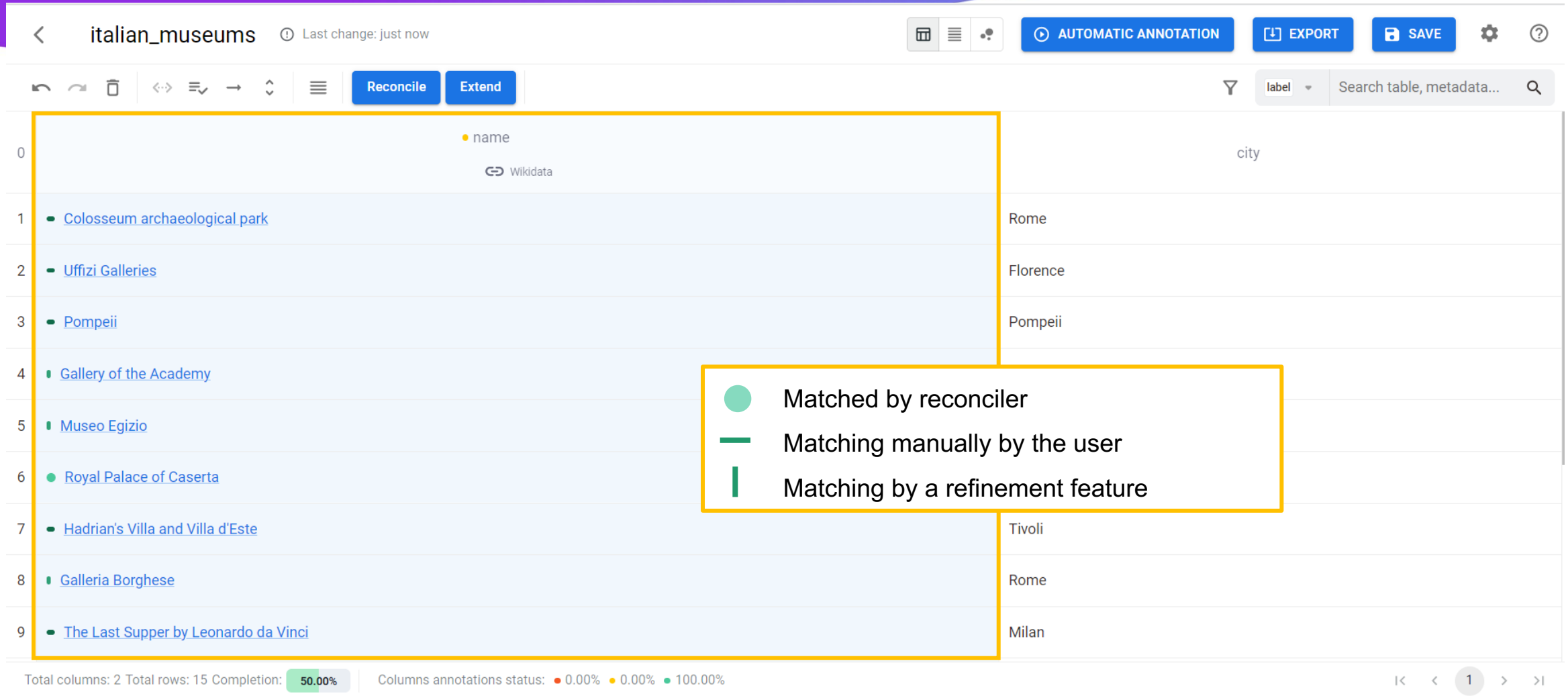


1 - Column annotation – Results

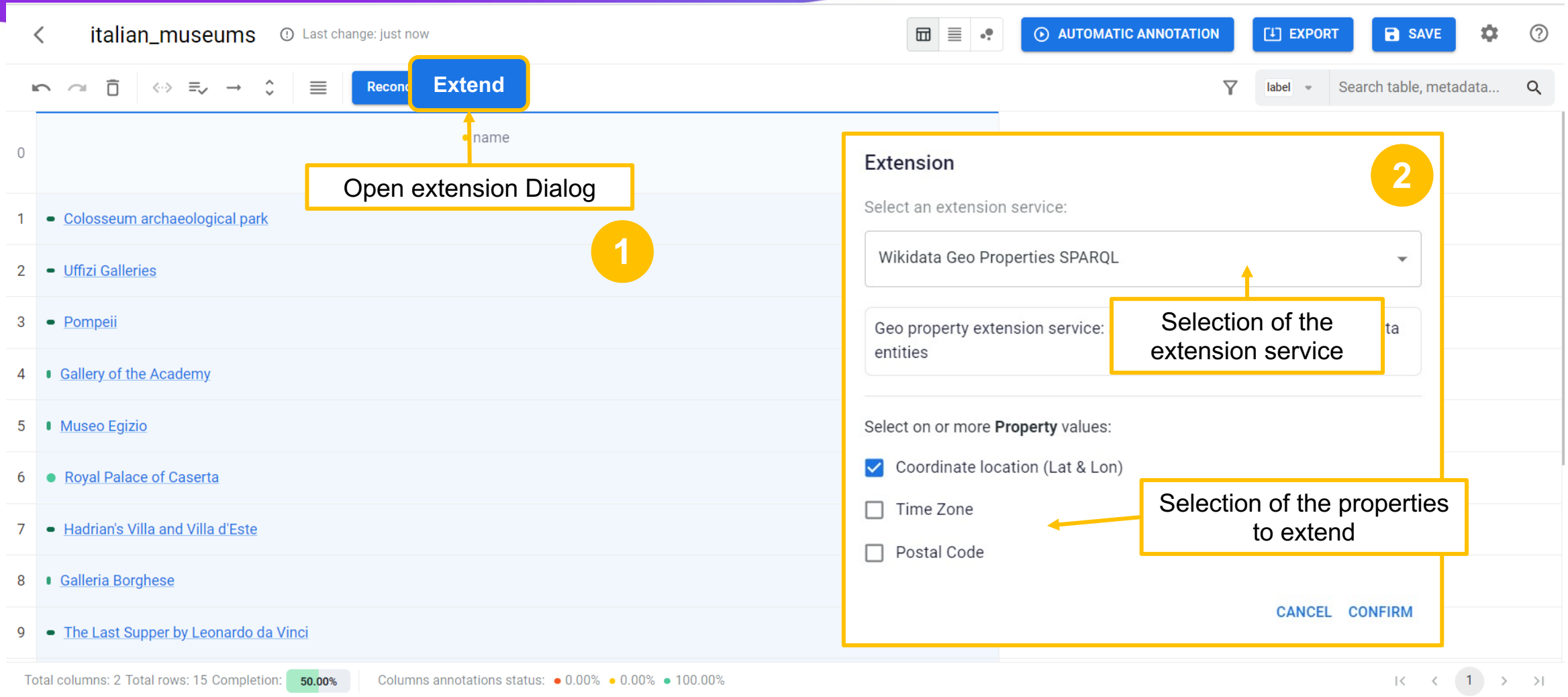


2 - Refine Matching

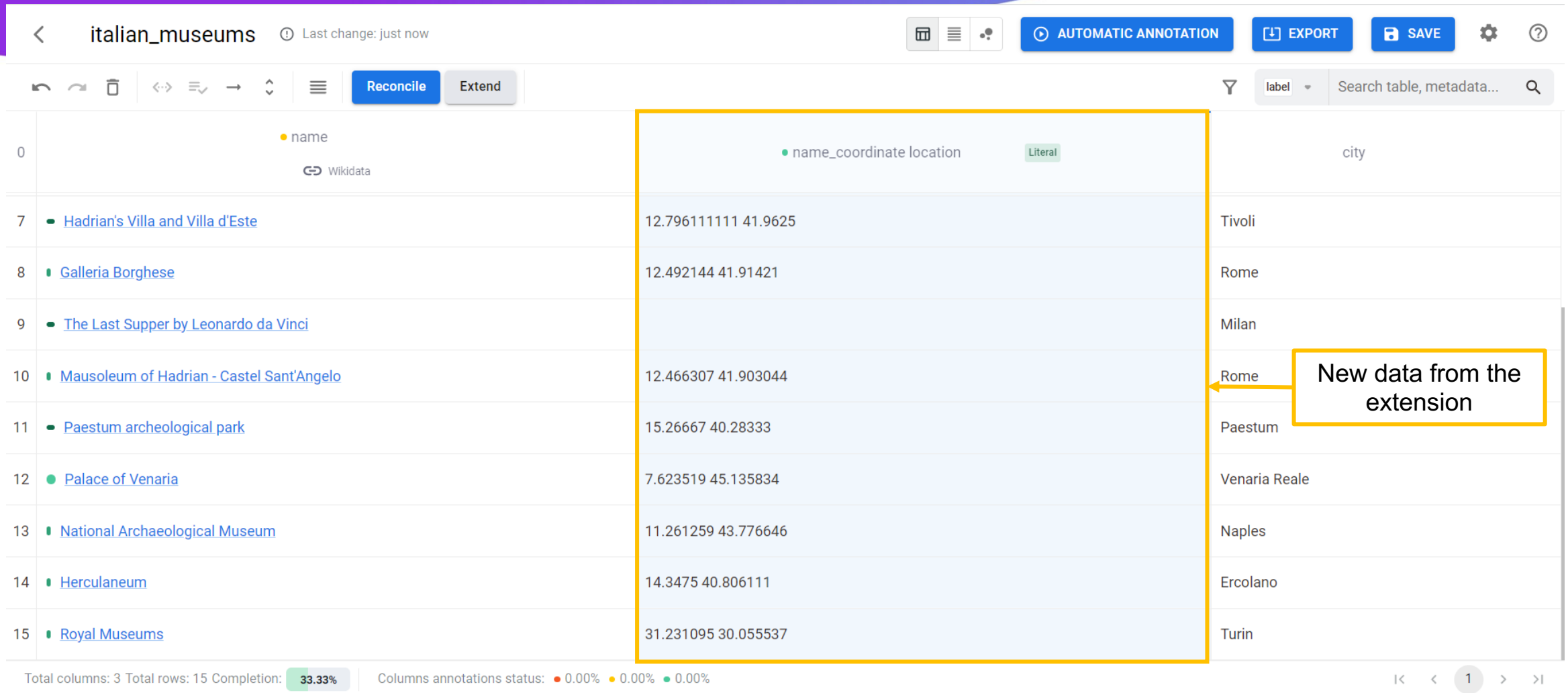
2 - Refine Matching – Results



3 - Extension


italian_museums
Last change: just now
AUTOMATIC ANNOTATION
EXPORT
SAVE
Extend
Open extension Dialog
1
name
Colosseum archaeological park
Uffizi Galleries
Pompeii
Gallery of the Academy
Museo Egizio
Royal Palace of Caserta
Hadrian's Villa and Villa d'Este
Galleria Borghese
The Last Supper by Leonardo da Vinci
Extension
2
Select an extension service:
Wikidata Geo Properties SPARQL
Geo property extension service:
entities
Selection of the extension service
Select on or more Property values:
Coordinate location (Lat & Lon)
Time Zone
Postal Code
Selection of the properties to extend
CANCEL
CONFIRM
Total columns: 2 Total rows: 15 Completion:
50.00%
Columns annotations status:
0.00%
0.00%
100.00%

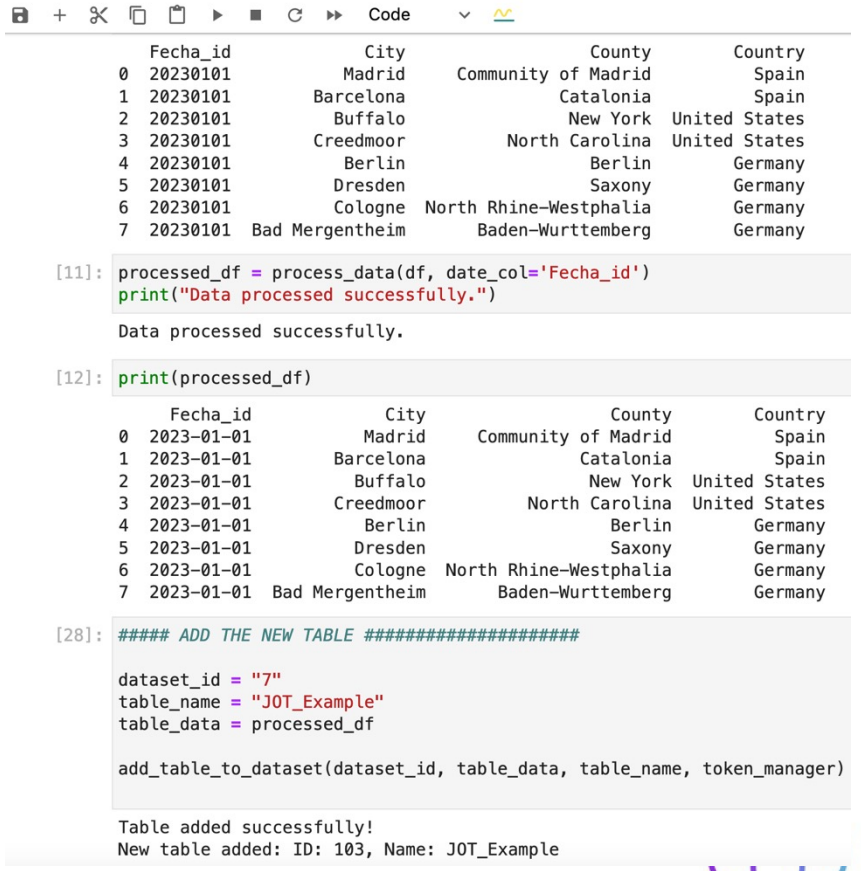
3 - Extension – Results



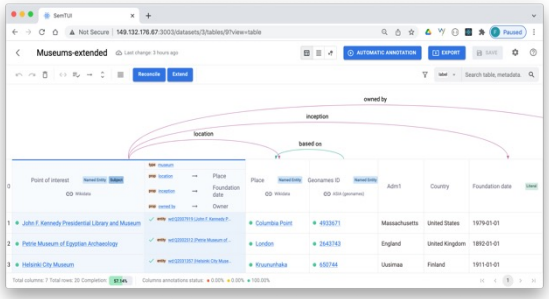
Pipeline definition

Pipeline definition

- Graphical User Interface (GUI):
  - Facilitates exploration and testing of different enrichment services.
- Programmatic User Interface (UI):
  - Enables the definition of pipeline components.
- Notebook Integration:
  - Creates the logic for executing enrichment tasks.
  - Supports execution on large datasets.
  - Allows for defining Docker images for integration with workflow tools.



Pipeline definition



Code

```
   Fecha_id            City                County        Country
0  20230101          Madrid     Community of Madrid          Spain
1  20230101       Barcelona               Catalonia          Spain
2  20230101         Buffalo                New York  United States
3  20230101       Creedmoor          North Carolina  United States
4  20230101          Berlin                  Berlin        Germany
5  20230101         Dresden                  Saxony        Germany
6  20230101         Cologne  North Rhine-Westphalia        Germany
7  20230101  Bad Mergentheim       Baden-Wurttemberg        Germany
```

[11]:
```
processed_df = process_data(df, date_col='Fecha_id')
print("Data processed successfully.")
```

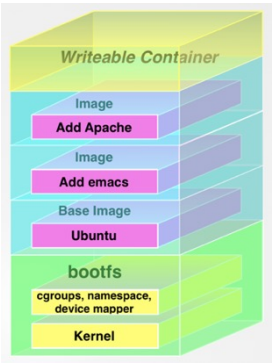
```
Data processed successfully.
```

[12]:
```
print(processed_df)
```

```
   Fecha_id            City                County        Country
0  2023-01-01          Madrid     Community of Madrid          Spain
1  2023-01-01       Barcelona               Catalonia          Spain
2  2023-01-01         Buffalo                New York  United States
3  2023-01-01       Creedmoor          North Carolina  United States
4  2023-01-01          Berlin                  Berlin        Germany
5  2023-01-01         Dresden                  Saxony        Germany
6  2023-01-01         Cologne  North Rhine-Westphalia        Germany
7  2023-01-01  Bad Mergentheim       Baden-Wurttemberg        Germany
```

[28]:
```
##### ADD THE NEW TABLE ####################

dataset_id = "7"
table_name = "JOT_Example"
table_data = processed_df

add_table_to_dataset(dataset_id, table_data, table_name, token_manager)
```

```
Table added successfully!
New table added: ID: 103, Name: JOT_Example
```



Demo live

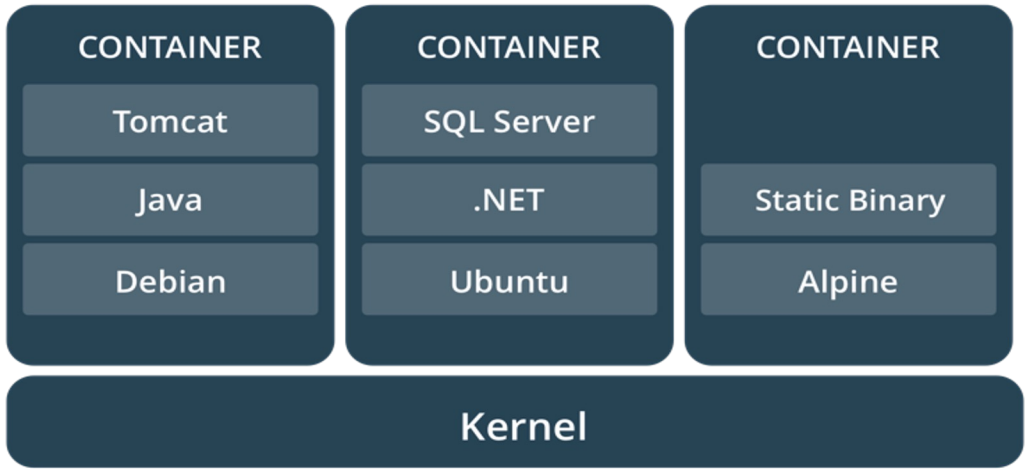
Pipeline execution at scale

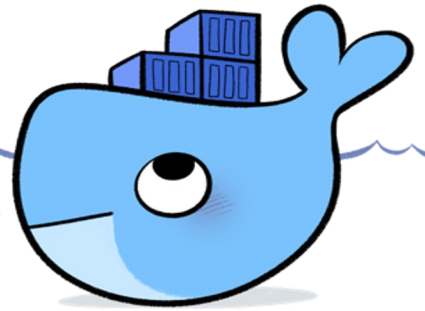
Containers

Containers

- A standard way to package an application and all its dependencies (libs and tools) so that it can be moved between environments and run without changes.
- Containers work by isolating the differences between applications inside the container so that everything outside the container can be standardized.

What's a container?





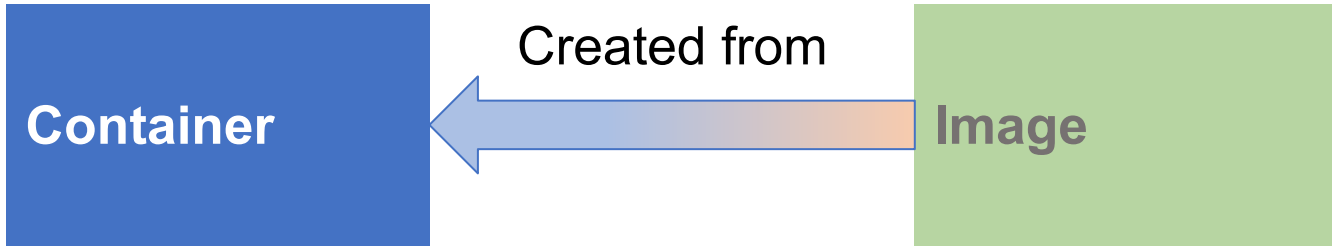
- Standardized packaging for software and dependencies
- Isolate application (security and resource sharing)
- Lightweight: Containers share the same OS kernel
- Works with **Linux** and… Windows Server

Source: https://www.docker.com/what-container

Basics: Containers vs Images

- Images
  - **Read-only templates** used to create containers. They contain the virtual disk with filesystem and configurations
  - Stored in the **Docker Hub** or in a local **Registry.** To be used they must be downloaded
  - **Images can be built upon other images**

- Containers
  - Applications/processes in **isolated execution**
  - Self-contained (no need of libraries outside the container)
  - Based on and linked to one image



Tools for Scaling Data Transformations

1. **Argo Workflows:** An open-source container-native workflow engine for orchestrating parallel jobs on Kubernetes.
   - **Features**:
     - Enables orchestration of diverse computational tasks
     - Facilitates the design and execution of sophisticated data processing workflows
     - Integrates smoothly with Docker and Kubernetes for scalable deployments
     - Offers a comprehensive web interface and a REST API for workflow management
   - **Benefits:**
     - Automates the execution of complex workflows
     - Scales smoothly with Kubernetes, handling large-scale data processing
     - Supports DAG (Directed Acyclic Graph) and step-based workflows
2. **TAO Tool:** Tool Augmentation by User Enhancements and Orchestration (TAO) is a framework initially developed by the European Space Agency and further extended in enRichMyData project
   - **Features:**
     - Allows orchestration of heterogeneous processing components
     - Supports the creation and execution of complex data processing workflows
     - Integrates with Docker and Kubernetes for scalable and flexible deployment
     - Provides both a web interface and a REST API for managing workflows and components
   - **Benefits:**
     - Simplifies the integration and reuse of diverse tools and applications
     - Facilitates scalable and efficient data processing workflows
     - Enhances user control and flexibility in workflow management

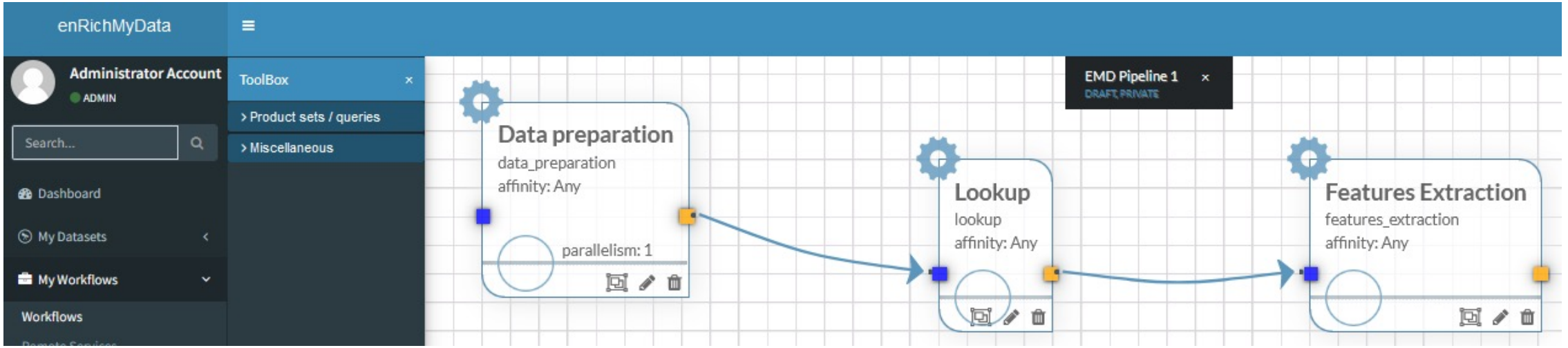
TAO details

Main Features

- **Modular Architecture**: Designed with a modular architecture, TAO allows easy addition and replacement of processing components to meet specific workflow requirements
- **Dynamic Configuration**: Supports dynamic configuration of workflows, enabling on-the-fly adjustments to processing components and parameters without needing to halt or disrupt ongoing processes
- **Extensive Plugin Support**: Features a plugin system that enables the integration of new tools and services into the workflow without extensive customization
- **Security and Compliance Features**: Incorporates security measures and compliance mechanisms to ensure data protection and meet regulatory requirements, especially crucial in space and research data management
- **Advanced Scheduling**: Provides advanced scheduling options that optimize resource utilization and execution timing based on the workload and system capacity

TAO Snapshot

Functionalities: Monitoring, Debugging, Scheduling and Designing



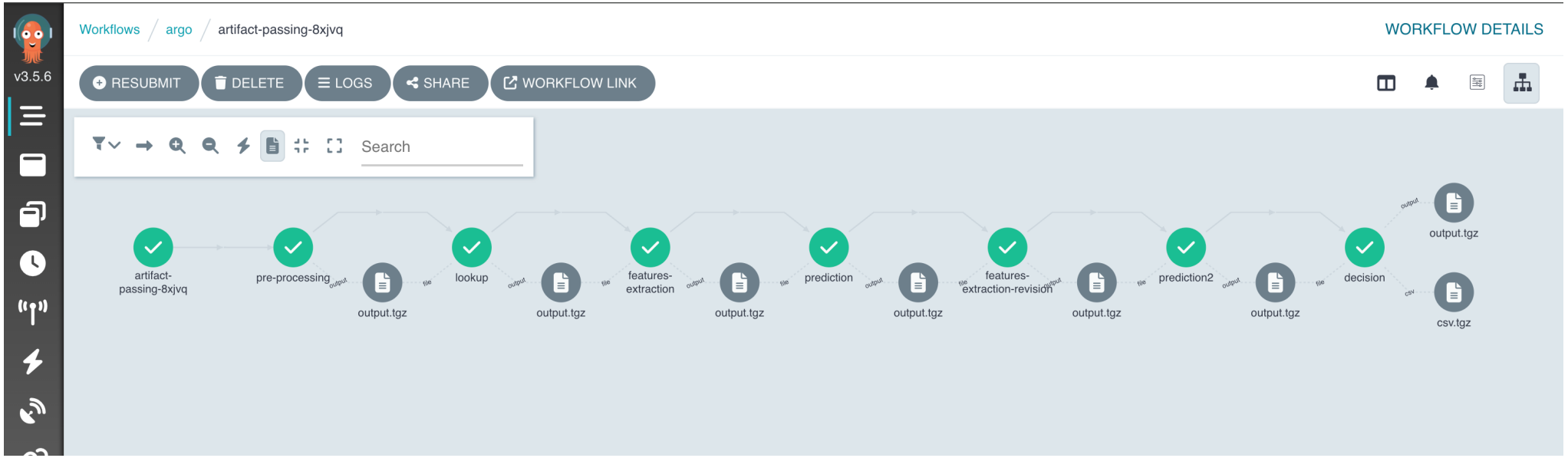
Argo workflows details

Main Features

- **Multi-Step Workflows**: Argo supports the creation of multi-step workflows, allowing users to orchestrate complex sequences of tasks and dependencies within a single workflow
- **Event-Driven Execution**: Enables workflows to be triggered by external events, integrating seamlessly with other Kubernetes resources through webhooks and other event sources
- **Artifact Management**: Provides built-in support for artifacts, allowing the passing of data and files between steps in a workflow
- **Parameterization**: Workflows can be parameterized, enabling the reuse of workflows with different inputs, thereby increasing modularity and flexibility
- **Rich UI Dashboard**: Offers a comprehensive web-based UI that provides real-time monitoring and visualization of workflow execution, facilitating easier debugging and management

Argo Workflows Snapshot

Functionalities: Monitoring and Debugging



Demo live

Thank you! Questions?