# Metadata Extraction

Members: Abhishek Nadgeri, Salmaan Tariq, Omar Ejjeh, Lei Wang, Amit Mudgal, David Abdelmalek

Final Presentation, 20.07.2021

# Table of Contents

# Problem Statement

# Problem Statement

- Age of big data
- Massive proliferation of accessible datasets
- Want to put this data to use

- Problem: How to find relevant datasets?

# Inspiration

# Inspiration

- Main inspiration is data.europa portal:

# data.europa.eu

The official portal for European data

**82** Catalogues

**36** Countries

**1 359 566** Datasets

## Search datasets

Agriculture, Fisheries, Forestry & Foods

Economy & Finance

Education, Culture & Sport

Energy

Environment

Government & Public Sector

Health

International Issues

Justice, Legal System & Public Safety

Population & Society

Regions & Cities

Science & Technology

Transport

Search datasets

Search

> DATA CATALOGUES

> ALL DATASETS

> EU INSTITUTIONS DATASETS

# Datasets

Filter by location

london

Search


Datasets | Catalogues | Editorial Content

Last Modified | Relevance | More

3 721 datasets found

## London Underground Performance Reports

Transport For Londons's key London Underground performance measures (since May-11). The key measures of underground performance contained in the Excel spreadsheet are: Total operated kilometres, Total number of lost customer hours (LCH) (all causes), Average excess journey time, and Percentage of scheduled operated. More indicators are available from th…

HTML

London Datastore

## London Business Survey 2014 - London as a place to do business

### Settings

Operator AND OR

### Data scope

EU data

# Inspiration

- Allows searching for metadata with keywords, timeframes and dataquality constraints
- Problem: most of the metadata is not very informative:

https://data.europa.eu/api/hub/repo/distributions/1d7f0ddd-a...

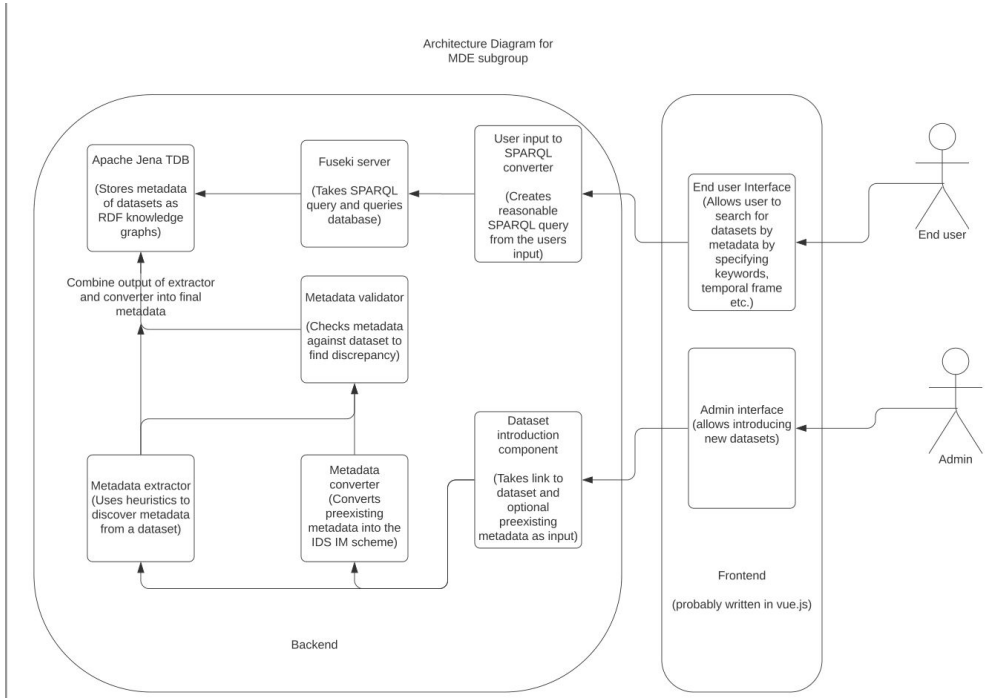
```
<https://europeandataportal.eu/set/distribution/1d7f0ddd-a689-4880-9979-310e68befdb3>
        a       <http://www.w3.org/ns/dcat#Distribution> ;
        <http://purl.org/dc/terms/description>
                ""@en ;
        <http://purl.org/dc/terms/format>
                <http://publications.europa.eu/resource/authority/file-type/CSV> ;
        <http://purl.org/dc/terms/identifier>
                "88c02d2c-30ac-4fb2-9495-7465c75cb0a3" ;
        <http://purl.org/dc/terms/license>
                <http://reference.data.gov.uk/id/open-government-licence> ;
        <http://purl.org/dc/terms/title>
                "tomt-homes-audit-data.csv\n"@sv-t-en-t0-mtec , "empty-homes-audit-data.csv"@en , "prÃ¡zdny domov-audit-data.csv\n"@sk-t-en-t0-mtec , "prÃ¡zdnÃ©-homes-audit-
data.csv\n"@cs-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@nl-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@hr-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@de-t-en-t0-mtec ,
"empty-homes-audit-data.csv\n"@fr-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@it-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@hu-t-en-t0-mtec , "empty-homes-audit-
data.csv\n"@pt-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@bg-t-en-t0-mtec , "empty-homes-audit-data.csv\n"@ro-t-en-t0-mtec , "folamh-baile-iniÃºchadh-data.csv\n"@ga-t-en-t0-
mtec , "ulÃ¸nnede audit-data.csv\n"@da-t-en-t0-mtec , "tukÅ¡s mÄ©joklis-revÃ«zija-data.csv\n"@lv-t-en-t0-mtec , "empy-homes-audit-data.csv\n"@es-t-en-t0-mtec , "resultater i
forbindelse med Ã¥ komme inn i lÃ¸pet av Ã¥tte Ã¥r.\n"@no-t-en-t0-mtec , "prazni dom-revizija-podatki.csv\n"@sl-t-en-t0-mtec , "homes-audit-data.csv empty-homes-audit-
data.csv\n"@mt-t-en-t0-mtec , "â€žW empty-homes-audit-data.csvâ€œ\n"@lt-t-en-t0-mtec , "empty-homes-audit-datacsv\n"@el-t-en-t0-mtec , "tÃ¼hi-kodu-audit-data.csv\n"@et-t-en-t0-
mtec , "puste miejsce zamieszkania-dane z audytu.csv\n"@pl-t-en-t0-mtec , "tyhjÃ¤-homes-audit-data.csv\n"@fi-t-en-t0-mtec ;
        <http://www.w3.org/ns/dcat#accessURL>
                <https://data.london.gov.uk/download/london-empty-homes-audit/2baf1040-c5c2-473e-8079-48f8a316874d/empty-homes-audit-data.csv> .

<http://reference.data.gov.uk/id/open-government-licence>
        a       <http://purl.org/dc/terms/LicenseDocument> ;
        <http://purl.org/dc/terms/identifier>
                "uk-ogl" ;
        <http://purl.org/dc/terms/title>
                "UK Open Government Licence (OGL)" .
```

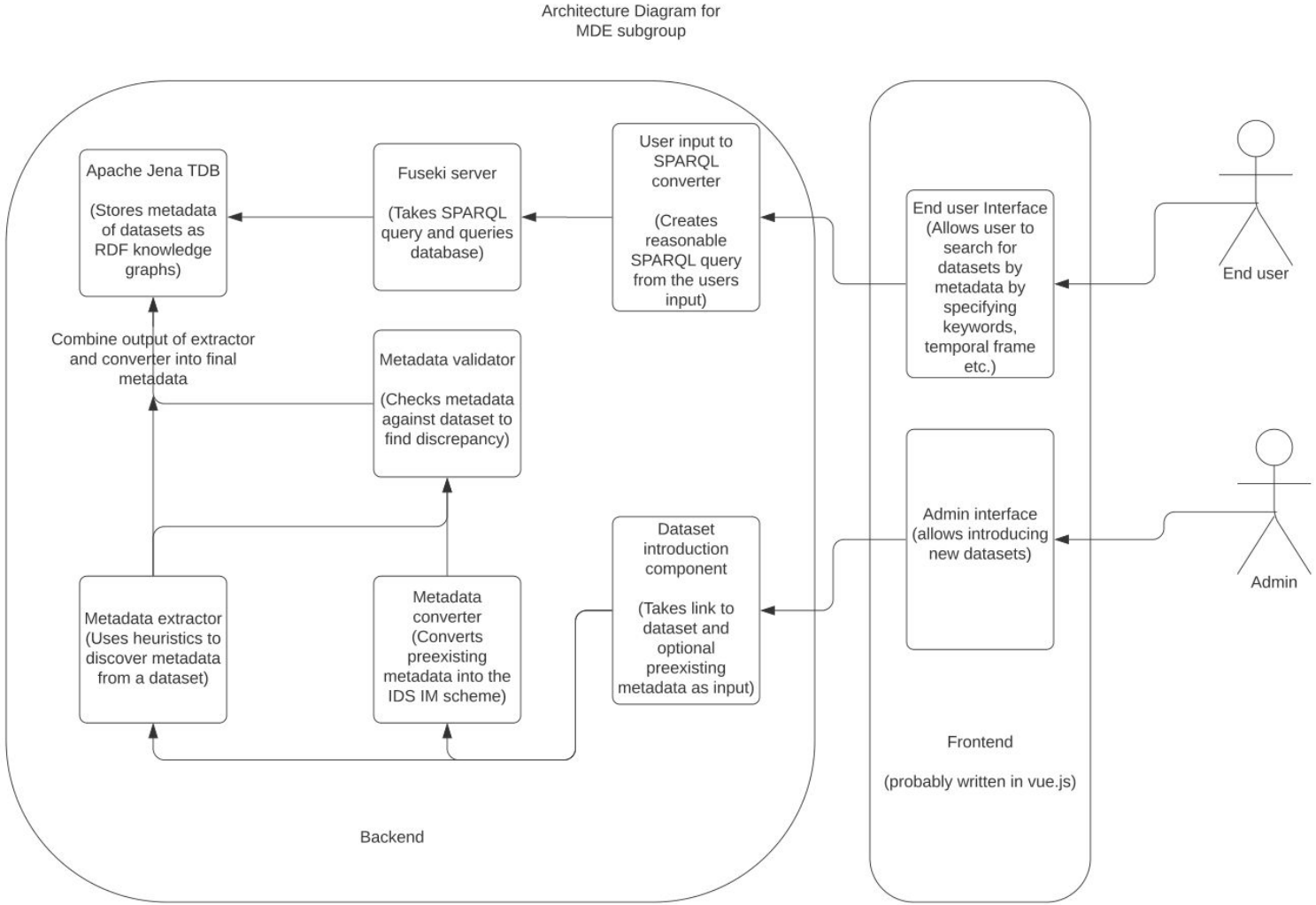
# Approach

# Requirements

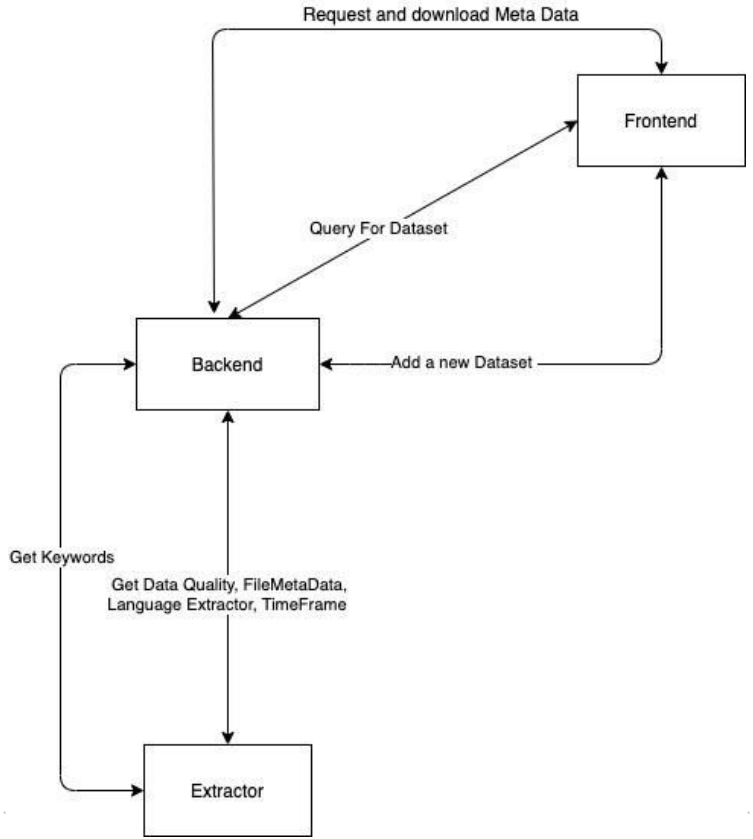
- Must allow **automatic** extraction of semantically **meaningful** metadata
- I.e. metadata that describes the **contents** of the files
- Must store this metadata in a **standardized** and **machine-readable** format for automatic discovery
- Must provide way to **link** to new data
- Must provide way to **query** datasets based on metadata constraints

# Technical Overview: Original architecture design




Architecture Diagram for
MDE subgroup
Apache Jena TDB
(Stores metadata of datasets as RDF knowledge graphs)
Fuseki server
(Takes SPARQL query and queries database)
User input to SPARQL converter
(Creates reasonable SPARQL query from the users input)
End user Interface
(Allows user to search for datasets by metadata by specifying keywords, temporal frame etc.)
End user
Combine output of extractor and converter into final metadata
Metadata validator
(Checks metadata against dataset to find discrepancy)
Admin interface
(allows introducing new datasets)
Admin
Dataset introduction component
(Takes link to dataset and optional preexisting metadata as input)
Metadata extractor
(Uses heuristics to discover metadata from a dataset)
Metadata converter
(Converts preexisting metadata into the IDS IM scheme)
Frontend
(probably written in vue.js)
Backend

# Technical Overview: Current Architecture



# Technical Overview

- Three major components: Frontend, Backend, Extractor
- Modularize design: components can run on different servers
- Frontend allows users to **ingest** and **query for** data
- Extractor **analyzes** new datasets for metadata concerns
- Backend manages **database** and processes user requests

- Internally: Uses IDS IM as ontology to store metadata as knowledge graphs

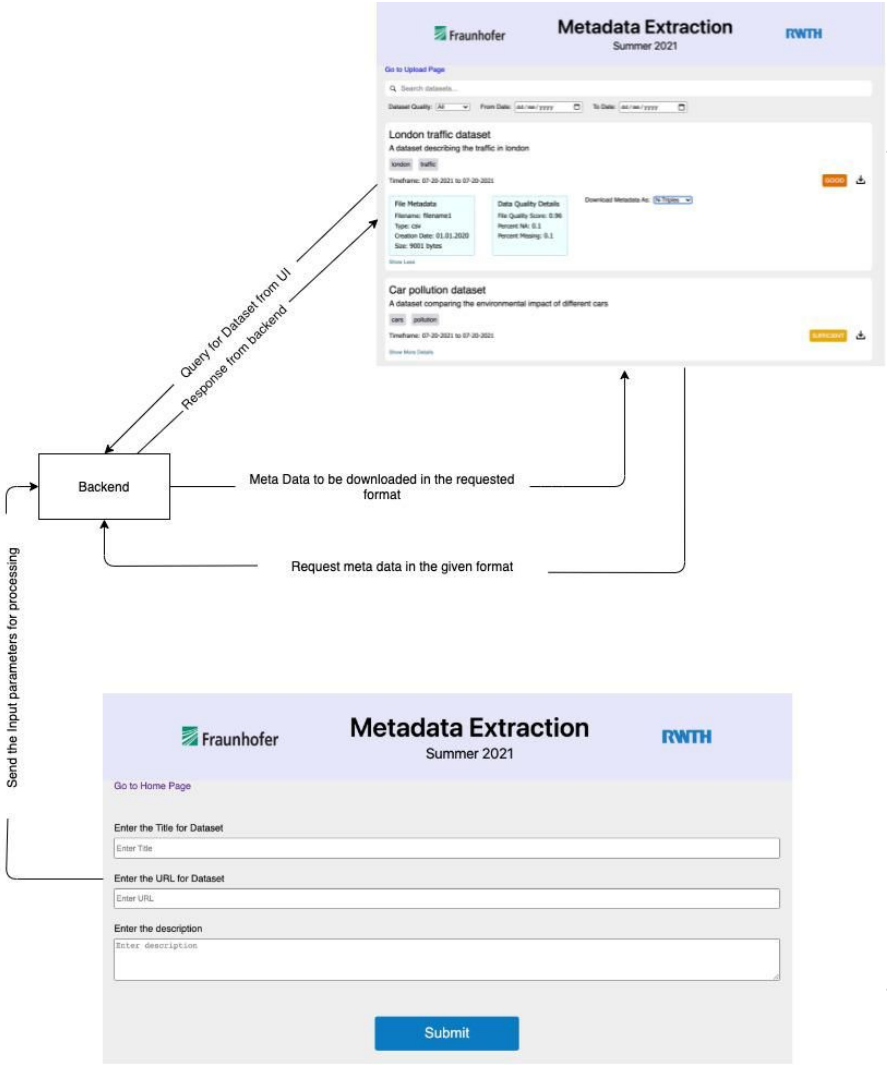
- Data: Focus on CSV files for easy analysis

# Technical Overview

- Note: Only support **automatic** extraction of metadata
- Do not support direct upload of RDF files
- Simplifies design
- Ensures that metadata always conforms to our schema

# Modules

## Frontend


Fraunhofer
Metadata Extraction
Summer 2021
RWTH
Go to Upload Page
London traffic dataset
A dataset describing the traffic in london
Car pollution dataset
A dataset comparing the environmental impact of different cars
Query for Dataset from UI
Response from backend
Backend
Meta Data to be downloaded in the requested format
Request meta data in the given format
Send the Input parameters for processing
Fraunhofer
Metadata Extraction
Summer 2021
RWTH
Go to Home Page
Enter the Title for Dataset
Enter Title
Enter the URL for Dataset
Enter URL
Enter the description
Enter description
Submit

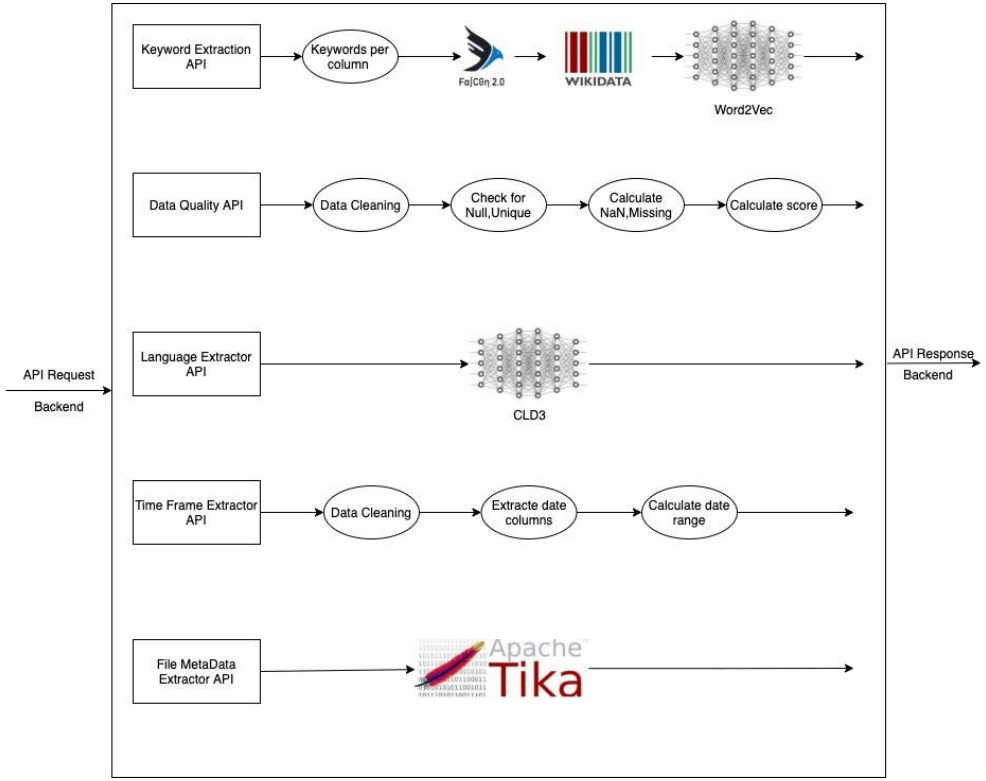
# Module: Frontend

- The front end is the face of the application built using  React framework.
- We have adopted the data centric approach while designing the front end.
- It consists of 2 basic pages  -
    - For querying the dataset.
    - For adding a new dataset.
- The user can query using dataquality, keywords, timeframe.
- The user can download the dataset from the interface.
- The user can also download the meta-data from the interface.

# Modules

## Extractor

# Extractor: General Architecture



# Extractor: Keyword Extraction

- Goal: The goal is to automatically discover the datasets, which can be later then used search for the dataset in an unsupervised manner.
- Motivation: As the data is structured, each column in the dataset is self contained. We use this inductive bias to our advantage.
- Approach:
  - For each column in the dataset get unique values.
  - Filter the unique values by the Frequency.
  - Link the unique values with a knowledge base.
  - Get the metadata of the associated knowledge base.
  - Filter the meta data from the  knowledgebase to extract the final keywords.

# Extractor: Keyword Extraction

- Goal: The goal is to automatically discover the datasets, which can be later then used search for the dataset in an unsupervised manner.
- Motivation: As the data is structured, each column in the dataset is self contained. We use this inductive bias to our advantage.
- Approach:
  - For each column in the dataset get unique values.
  - Filter the unique values by the Frequency.
  - Link the unique values with a knowledge base.
  - Extract the metadata of the associated knowledge base.
  - Filter the meta data from the  knowledgebase to extract the final keywords.

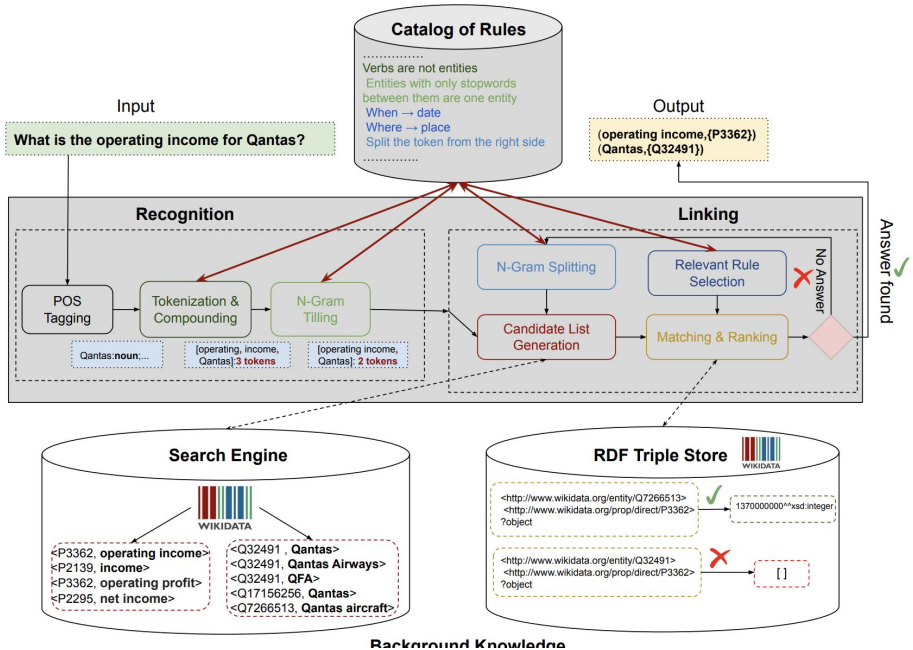
# Keyword: Filter values by frequency

- The data contains other values example data time, numerical etc.
- The column of interest id area_name.
- We dont need all the values for the column, only the most frequent.
- After the processing we will have values of area. Eg. City of London, Redbridge, Southwark etc

| date | area_name | area_code | retail_and_recreation_percent_change_from_baseline | grocery_and_pharmacy_percent_change_from_baseline |
|---|---|---|---|---|
| 2020-02-15 | City of London | E09000001 | -5 | -9 |
| 2020-02-16 | City of London | E09000001 | -1 | -21 |
| 2020-02-17 | City of London | E09000001 | -3 | -2 |
| 2020-02-18 | City of London | E09000001 | -2 | -2 |
| 2020-02-19 | City of London | E09000001 | -7 | -4 |
| 2020-02-20 | City of London | E09000001 | -7 | -7 |
| 2020-02-21 | City of London | E09000001 | -3 | -2 |
| 2020-02-22 | City of London | E09000001 | 4 | 5 |
| 2020-02-23 | City of London | E09000001 | 8 | 1 |
| 2020-02-24 | City of London | E09000001 | -7 | -12 |
| 2020-02-25 | City of London | E09000001 | -1 | 1 |
| 2020-02-26 | City of London | E09000001 | 0 | -3 |
| 2020-02-27 | City of London | E09000001 | 0 | -4 |
| 2020-02-28 | City of London | E09000001 | -2 | -6 |
| 2020-02-29 | City of London | E09000001 | 11 | 3 |
| 2020-03-01 | City of London | E09000001 | 22 | 14 |
| 2020-03-02 | City of London | E09000001 | -2 | 0 |
| 2020-03-03 | City of London | E09000001 | -5 | -1 |
| 2020-03-04 | City of London | E09000001 | -6 | -5 |
| 2020-03-05 | City of London | E09000001 | -14 | -17 |
| 2020-03-06 | City of London | E09000001 | -4 | 2 |
| 2020-03-07 | City of London | E09000001 | 2 | 4 |

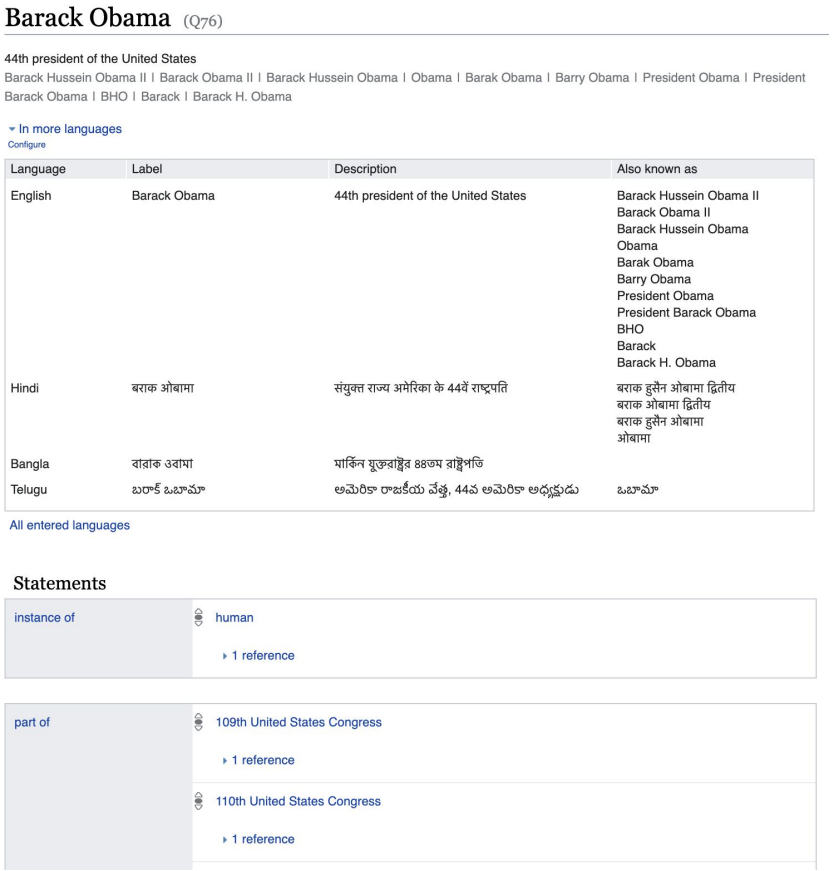
# Keyword: Link to KnowledgeBase

- To extract meta data, they need to be linked to a knowledgebase.
- To Achieve this we use the tool **Falcon** (Falcon 2.0: An Entity and Relation Linking Tool over Wikidata)



# Keyword: Extract the metadata

- Wikidata is rich source of structured knowledge base.
- Every entity has property like instance of, part of etc.
- For now we only extract the instance of part.



**Barack Obama** (Q76)

44th president of the United States

Barack Hussein Obama II | Barack Obama II | Barack Hussein Obama | Obama | Barak Obama | Barry Obama | President Obama | President Barack Obama | BHO | Barack | Barack H. Obama

In more languages

Configure

| Language | Label | Description | Also known as |
|---|---|---|---|
| English | Barack Obama | 44th president of the United States | Barack Hussein Obama II<br>Barack Obama II<br>Barack Hussein Obama<br>Obama<br>Barak Obama<br>Barry Obama<br>President Obama<br>President Barack Obama<br>BHO<br>Barack<br>Barack H. Obama |
| Hindi | बराक ओबामा | संयुक्त राज्य अमेरिका के 44वें राष्ट्रपति | बराक हुसैन ओबामा द्वितीय<br>बराक ओबामा द्वितीय<br>बराक हुसैन ओबामा<br>ओबामा |
| Bangla | বারাক ওবামা | মার্কিন যুক্তরাষ্ট্রের ৪৪তম রাষ্ট্রপতি | |
| Telugu | బరాక్ ఒబామా | అమెరికా రాజకీయ వేత్త, 44వ అమెరికా అధ్యక్షుడు | ఒబామా |

All entered languages

**Statements**

| | |
|---|---|
| instance of | human<br>1 reference |
| part of | 109th United States Congress<br>1 reference<br>110th United States Congress<br>1 reference |

# Keyword: But all is not good

- Here the value City of London gets mapped to a city in USA.
- But the city gets mapped to the correct city.
- Now, here if we have simple heuristics it may not help.
  - It depends on the accuracy of entity linker.
  - For example all the cities in the dataset may be linked to USA and only may be linked to UK.
- Question - How to solve this ?

**London** (Q3061911)

city in Kentucky, United States
London, Kentucky

edit

In more languages
Configure

| Language | Label | Description | Also known as |
|---|---|---|---|
| English | London | city in Kentucky, United States | London, Kentucky |
| Hindi | No label defined | No description defined | |
| Bangla | No label defined | No description defined | |
| Telugu | No label defined | No description defined | |

All entered languages

**Redbridge** (Q7305499)

suburb of Southampton, in Hampshire, England
Redbridge, Southampton

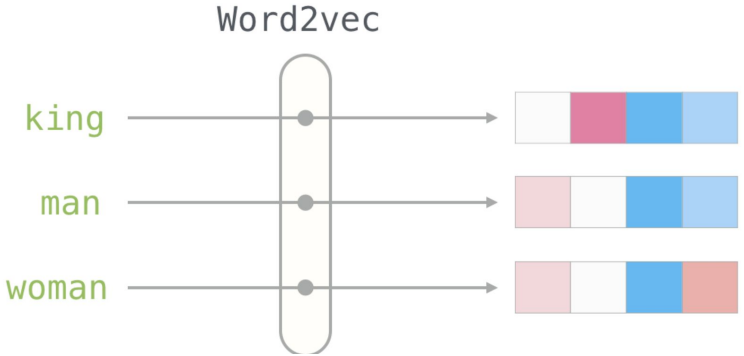
edit

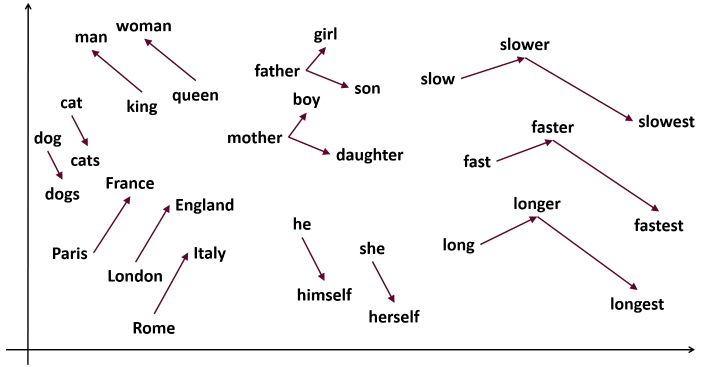
In more languages
Configure

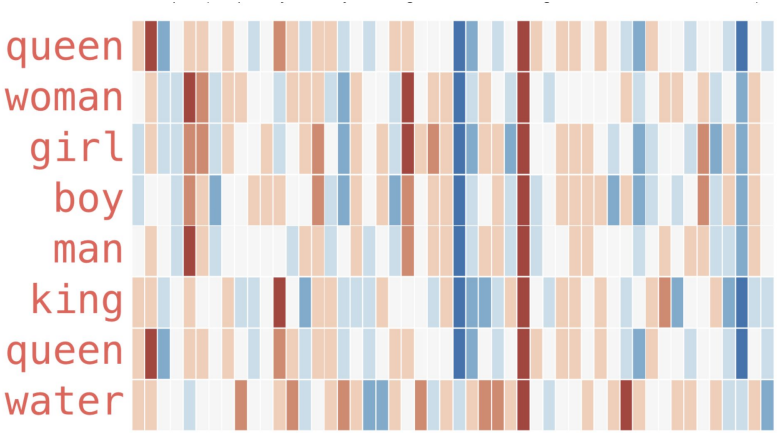
| Language | Label | Description | Also known as |
|---|---|---|---|
| English | Redbridge | suburb of Southampton, in Hampshire, England | Redbridge, Southampton |
| Hindi | No label defined | No description defined | |
| Bangla | No label defined | No description defined | |
| Telugu | No label defined | No description defined | |

All entered languages

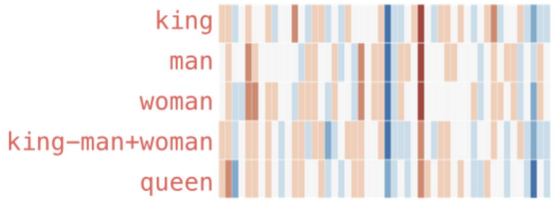
# Keyword: Enter Deeplearning



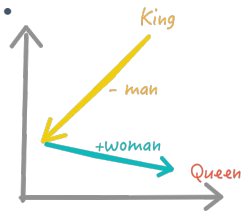

king − man + woman ~= queen



# Keyword: Filter Metadata

- We use word2vec model as it is trained on Wikipedia, and our knowledge base is based on wikipedia.
- for each column:
    - Iterate through all the keys and all the metadata and convert it into the respective embedding.
    - For each of meta embedding:
        - Take the dot product of each key embedding with the metadata obtained.
        - Sum the scores obtained from the dot product and that forms the metadata
    - Select the metadata with the maximum score.

- Time Complexity: O(C*N^2)



a) Learns Analogy

b) Similar Words have same angles

# Extractor: Language Detection

- **Goal:** Is to detect the main language used by selecting the language that has high probability happening inside the dataset.

- **How:**
  - **Parse content:** use Apache Tika parser to parse the text.
  - **Use language detection library:** use the CLD3 library on text to predict the probability of all languages found in the dataset.
  - **Select lang :** choose the language that has highest probability in dataset.


```
{
"data": "{
  'language': 'en',
  'language_probability': 0.9999997
        }",
"success": true
}
```

# Extractor: Time frame

- **Goal:** is to extract time frame from the input file to be able to choose between data sources depend on start and end dates.

- **How:**
  - **Parse content:** use Apache Tika parser which extracts text and metadata.
  - **Read as pandas dataframe:** from date column select min date to be start date and maximum date to be end date.

```
{
"data": "{
  'start_date': '2020-02-15T00:00:00',
  'end_date': '2020-03-15T00:00:00'
        }",
"success": true
}
```

# Extractor: File Metadata

- **Goal:** is to extracts more metadata to be added in the info model.

- **How:**
  - **Get metadata file:** use Apache Tika metadata extractor to get creatation_date of the input file.
  - **Use OS library:**
    - File name
    - File size
    - File type


```
{
"data": "{
  'file_name': 'google_activity_by_London_Borough.csv',
  'file_type': 'text/csv',
  'creation_date': '2020-01-15T00:00:00',
  'file_size': '908714'
        }",
"success": true
}
```

# Extractor: Data Quality

- **Goal:** is to extracts more metadata to be added in the info model.

- **How:**
  - **Get metadata file:** use Apache Tika metadata extractor to get creatation_date of the input file.
  - **Use OS library:**
    - File name
    - File size
    - File type


```
{
  "data": {
    "file_quality": "good",
    "file_quality_score": 79.02,
    "percentNA": 1.24,
    "percentage_missing": 0.0
  },
  "success": true
}
```

# Data Quality = Fitness For Use

- Data quality is relative
- High-quality data is required for trusted decisions
- Bad data quality is a big loss
  - Gartner.com states that organizations **lose $13.3 Million yearly** average on poor data
  - Kissmetrics states businesses lose up to **20 percent** of their **revenue** because of bad data.
  - CrowdFlower states data scientists spend **60 percent** of their **time** cleaning and organizing data.
- Data quality measurement can help quality to be improved

# Data Quality Dimensions

- Completeness

A data set with too many holes are not going to be able to answer questions. We are checking the presence of values in every column and calculate a weighted average with 80% weightage to this factor.

- Uniqueness

A high uniqueness score assures minimized duplicates or overlaps, building trust in data and analysis. We are measuring data uniqueness in all columns and giving a weightage of 20%.

Finally, data quality is calculated having a positive impact on completeness and uniqueness while the weight percent of NaN and empty records is recorded negative.

# Duckdq: Embedabble Data Quality Validation

DuckDQ is a python library that provides a fluent API for definition & execution of:

- Data quality checks for ML pipelines, built upon the estimator/transformer paradigm of scikit-learn.
- Used for "unit tests for data". It excels on small-to medium-sized datasets.
- Data quality checks on:
  - pandas dataframes
  - CSV files
  - parquet files
  - database tables from relational database systems

# Output

- **Goal:** To return data quality with different parameters

- **How:**
  - Pre processing/cleaning data before passing to duckdq library
  - Executing relative functions for the described dimensions

➔ **Excellent** if weighted average is >= 85.
➔ **Good** if weighted average is between 55 and 85.
➔ **Sufficient** if weighted average is between 30 and 55.
➔ **Bad** if weighted average is < 30.


```
{
  "data": {
    "file_quality": "good",
    "file_quality_score": 79.02,
    "percentNA": 1.24,
    "percentage_missing": 0.0
  },
  "success": true
}
```

# Modules

## Backend

# Module: Backend

- Written in Java 11 and Spring Boot 2.5.1
- Communicates with frontend and extractors to provide functionality
- Internally manages Fuseki/TDB database to store metadata in the IDS IM
- Exposes 3 APIs:
    - /query
    - /submit
    - /download

# Module: Backend (/query)

- /query API is called with 4 parameters:
  - Keywords
  - Dataquality
  - Timeframestart
  - Timeframeend

- Queries Fuseki and returns all datasets that match these constraints
- Keywords allow querying for data based on the **contents** of the dataset

# Module: Backend (/submit)

- /submit API is called with 3 parameters:
  - title
  - description
  - url

- Backend sends the URL of the dataset  to the extractor for processing
- Returned metadata concerns are cast into a knowledge graph using the IDS IM

```
@prefix ids: <https://w3id.org/idsa/core/> .
@prefix idsc: <https://w3id.org/idsa/code/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix mdeprops: <https://www.mde.com/customproperties/> .
@prefix dqv:      <http://www.w3.org/ns/dqv#> .

@prefix conn2: <https://aastat.gov.de/connector/conn2/> .
@prefix data2: <https://aastat.gov.de/connector/conn2/data2/> .
@prefix part1: <https://im.internationaldataspaces.org/participant/part1> .

data2:
    a ids:TextResource ;
    ids:title "Data about the lebanese economy"@en ;
    ids:description "This dataset describes the lebanese economy between the years 1990 and 2005."@en;
    ids:keyword "London borough" ;
    ids:temporalCoverage [
        a ids:Interval ;
        ids:begin [
            a ids:Instant ;
            ids:dateTime "1990-02-15T00:00:00.000+02:00"^^xsd:dateTimeStamp ;
        ];
        ids:end   [
            a ids:Instant ;
            ids:dateTime "2005-05-15T00:00:00.000+02:00"^^xsd:dateTimeStamp ;
        ] ;
    ] ;
    ids:language idsc:EN ;
    ids:representation [
        a ids:TextRepresentation ;
        ids:mediaType <https://www.iana.org/assignments/media-types/text/csv> ;
        ids:instance data2:activity_csv ;
    ] ;
```

```
        dqv:hasQualityMeasurement  [ dqv:isMeasurementOf  mdeprops:qualityScore ;
                                     dqv:value            "1.0"
                                   ] ;
        dqv:hasQualityMeasurement  [ dqv:isMeasurementOf  mdeprops:percentMissing ;
                                     dqv:value            "0.0"
                                   ] ;
        dqv:hasQualityMeasurement  [ dqv:isMeasurementOf  mdeprops:dataquality ;
                                     dqv:value            "good"
                                   ] ;
        dqv:hasQualityMeasurement  [ dqv:isMeasurementOf  mdeprops:percentNA ;
                                     dqv:value            "0.0"
                                   ] ;



    ids:resourceEndpoint [
        a ids:ConnectorEndpoint ;
        ids:endpointArtifact data2:activity_csv ;
        ids:accessURL <https://tmpfiles.org/dl/62995/lebanon_economy.csv> ;
    ] ;
        .


data2:activity_csv
    a ids:Artifact ;
    ids:byteSize "492817"^^xsd:integer ;
    ids:fileName "lebanon_economy.csv" ;
    ids:creationDate "2021-06-15T12:00:00.000+02:00"^^xsd:dateTimeStamp ;
    .
```

# Module: Backend (/download)

- /download API is called with 2 parameters:
  - Url
  - type

- We use the access URL of the materialization of the dataset to identify it
- The matching knowledge graph is returned as an RDF file in the specified format
- Supports TTL, JSON-LD and all common RDF formats

# Conclusion

# Where to go from here?

- Possible improvements:
  - Performance
    - Keyword extractor performance is O(#cols*#entries^2).
    - Keyword extractor uses Falcon API and Wikidata API -> source of latency.
    - Uses Apache Tika for extracting file metadata. Current implementation spins up a new server for every call.
  - More data quality measurements can be added
  - Keyword extractor works only for english datasets right now, can be expanded
  - Supports only CSV right now -> expand to other data

# Where to go from here?

- Possible improvements:
    - Stretch goal we didn’t get to do: Metadata validation.
    - Check our extracted metadata against preexisting metadata

# Thank you

Questions?
Slide theme credit - Isaiah Mulang