Enabling Data Enrichment Pipelines for AI-driven Business Products and Services

HORIZON-CL4-2021-DATA-01-03

D1.1: Technical and market requirements

Work Package 1

Type of document:	Report
Dissemination level:	Public
Lead beneficiary:	UB
Authors:	Matteo Palmonari and Flavio De Paoli (editors), all other technical partners contributors
Version:	1.0
Due Date of document:	30.06.2023
Delivery Date of document:	30.06.2023

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Document History

Version	Date	Contributor	Comments
0.1	31.01.2023	Matteo Palmonari (UB)	Structure and ToC
0.2	02.03.2023	Matteo Palmonari (UB)	Formatted Draft + Guidelines
0.3	03.04.2023	All	Draft of SoTA
0.4	30.05.2023	Flavio De Paoli, All	Definition of requirements tables and preliminary requirements definition
0.5	13.06.2023	Flavio De Paoli, Matteo Palmonari, All	Revised state of the art and complete draft of requirements
0.6	14.06.2023	Flavio De Paoli, Matteo Palmonari	First complete draft
0.7	20.06.2023	Flavio De Paoli, Matteo Palmonari	Revised draft shared to reviewers
0.8	27.06.2023	Flavio De Paoli, Matteo Palmonari, All	Revised document after review
0.9	29.06.2023	Brian Elvesæter	Quality assurance and formatting and layout
1.0	30.06.2023	Till Christopher Lech	Final QA

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Executive summary

The enRichMyData project aims to provide a novel paradigm and a toolbox for building rich, high-quality, and valuable datasets to feed Big Data Analytics and AI applications. Consequently, in this deliverable we identify the requirements that tools in the toolbox must satisfy to provide data enrichment solutions useful in several application domains, but also innovative in the context of the vast and competitive landscape of data management solutions.

A methodology for the elicitation of the requirement has been specified, which consists in consolidating into a requirement description templates requirements derived from two main sources: 1) the requirements collected for the enRichMyData business cases and discussed in D4.1; 2) the analysis of research and technology solutions existing today for data enrichment. The business case requirements reflect the result of a requirement elicitation process based on tight interactions between the EMD tool providers and the business case partners and structured interviews. In the SoTA analysis, we define precisely the many different aspects of data enrichment addressed in the project, and discuss which aspects are less covered and more challenging.

The SoTA analysis is organized into two main sections addressing the data modalities considered in enRichMyData: semi-structured data, e.g., tables, JSON records, RDF, and relational databases, and textual data. For semi-structured data the following data enrichment aspects are considered: *model transformations, data wrapping and retrieval*, which are related to the WrappR and ResourcR tools; *data discovery*, related to the DiscoverR tools; *data cleaning*, related to the CleanR tools; *data linking*, related to the LinkR tools. For textual data, the following data enrichment aspects are considered: *information extraction*, related to the StructureR tools; *text classification*, related to the ClassifieR tools. Some aspects concern features of data enrichment processes that do not depend on a specific data model; we therefore consider these additional aspects: *scaling, replication and adaptation*, which are related to the ScalR and ReusR tools; support for *streaming data* processing, related to StreamR tools; assessment and improvement of *energy consumed* by the algorithms, related to GreenR tools.

For each of these aspects: 1) we report about the scope of the feature covered in the section and present some explicative examples; 2) we discuss the status of research in fields that are relevant for the considered feature; 3) we discuss existing tools that are relevant for the considered feature, including, of course, tools provided by consortium partners.

In the presentation of the actual requirements, we consider two main types of requirements: requirements specific to tool collections, and system-wide requirements. For each requirement, we provide: an identifier, a title, a description, the sources of the requirement (SoTA and/or business case requirements), related enRichMyData tools, means of verification, dependencies and priority (must-have, should-have or could-have). As a result, for each tool collection, we collect the following numbers of requirements. For s*emi-structured data:* WrappR (3), ResourcR (5), DiscoverR (4), CleanR (8), LinkR (6); *textual data*: StructureR (12) and ClassifieR (3); *other relevant aspects*: ScalR (13), ReusR (2), StreamR (7), GreenR (3). System-wide requirements are organized into: *functional requirements*, further split into security and privacy requirements (6), resource administration (3) and workflows (3); requirements for the system interfaces, further split into user interfaces (4), and interfaces with external systems (3).

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Contents

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

List of Tables

List of figures

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

1 Project introduction

enRichMyData provides a novel paradigm for building rich, high-quality and valuable datasets to feed Big Data Analytics and AI applications. It aims at facilitating the specification and scalable execution of data enrichment pipelines, with a focus on supporting various data enrichment operations such as discovery, understanding, selection, cleaning, transformation, integration of Big Data from a variety of sources. **enRichMyData** makes this paradigm easily accessible to a wide range of large and small organizations that encounter difficulties in delivering suitable data to feed their data analytics solutions, due to the lack of specific tools and expertise to support cost-effective and energy-efficient management of data enrichment pipelines.

enRichMyData aims to deliver a toolbox consisting of software tools and infrastructure services for setting up, deploying, executing and managing data enrichment pipelines with the following objectives:

- **Objective O1:** Improve data discovery and profiling featuring search on data, ontologies, and semantic data profiles, to identify data that are potentially valuable for data enrichment.
- **Objective O2:** Improve wrapping of data sources in different formats so they can be securely accessed as virtual semantic graphs are used more easily for data enrichment.
- **Objective O3:** Simplify cleaning, linking (to reference resources), and extension of semi-structured data, featuring approaches that enable users to specify such operations visually.
- **Objective O4:** Simplify annotation and classification of textual data, featuring entity and concept extraction, feature extraction (via embeddings), and classification with predefined and custom classifiers.
- **Objective O5:** Support the management of data enrichment pipelines, including creation and operation of data, linking and extension of services, a framework for deployment and execution of pipelines at large scale, and reuse and extension of existing pipelines to deliver a hub of data and services for data enrichment.
- **Objective O6:** Support data streaming in data enrichment pipelines, featuring support for setting up appropriate endpoints and ensuring high throughput pipeline execution.
- **Objective O7:** Monitor and reduce energy consumption for executing data enrichment pipelines by using models to estimate and track their carbon footprint.

enRichMyData validates its plan through a strong selection of complementary business cases offered by commercially focused organizations targeting the development of novel products in a wide range of domains, including *Marketing data Enrichment for smart-bidding optimization, AI-based Welding Analytics, Smart Maintenance of Medical Imaging Systems, European Register of Entities from Known Actions, Global Innovation Ecosystems Knowledge Graph and Mineral Processing Optimization*.

The consortium consists of 13 partners from 11 countries. With a mixture of R&D/technology and business case providers, enRichMyData gathers five large companies, three SMEs, two research institutes, and three universities.

1.1 Conceptual Architecture and Software Components

The project's data enrichment solutions require interactive exploration and evaluation, and new ideas and concepts need to be tested. The requirements will be refined in multiple steps based on the outcome of initial versions of the technological components. A standard waterfall or V-model project life cycle is not suitable, so the project will use an iterative Spiral life cycle (Boehm) model based on User-Centred Design techniques and an agile development approach.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

The Spiral life cycle model begins with initial requirement gathering, followed by rapid proposition of new basic ideas and assumptions, design and development of an initial tool version implementing them, and finally test and validation to produce enhancements for the next iteration. Test results provide feedback and a new set of requirements, and a new design and development is performed. This process continues until a complete set of requirements that meet all functional and non-functional requirements has been implemented.

The iterative approach is particularly suitable for this project, as business case providers can check correspondence with expected results in the early stages of the project. The iterative development approach will include both big iterations concluded by deliverables and small iterations that provide constant collaborations with stakeholders. The project will also pay specific attention to developing non-discriminatory and unbiased solutions both from technical and social perspectives.

The project will deliver its capabilities in a set of interoperable tools and services that will form the enRichMyData Toolbox, which will handle complex data enrichment scenarios. The Toolbox's conceptual architecture is cantered around the data enrichment pipeline that receives input data to be enriched and data to enrich with, generating enriched data. The enrichment process is supported by a set of functional capabilities related to supporting the design of pipelines and non-functional capabilities related to supporting the effective and efficient deployment and execution of pipelines. A better understanding of the mentioned process in described in Figure 1.

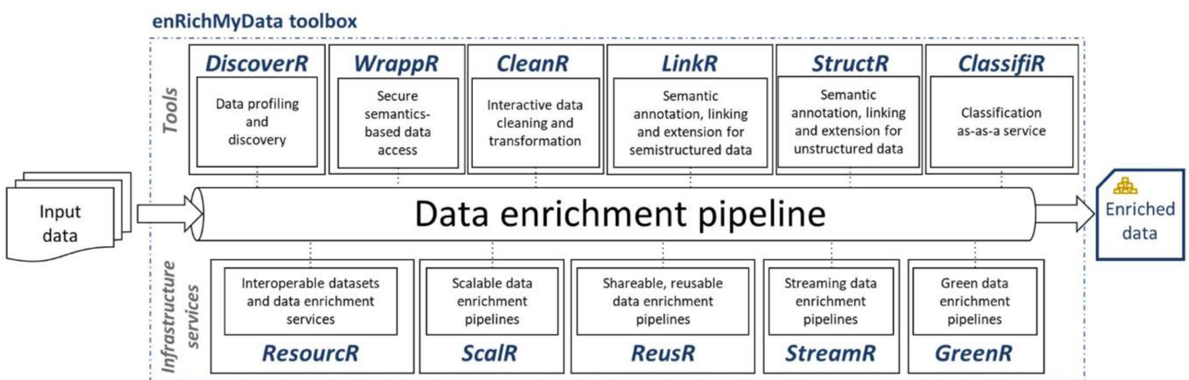


Figure 1: enRichMyData conceptual architecture for managing data enrichment pipelines

1.2 Research Challenges

The field of data processing features several tools, such as Trifacta Wrangler, KNIME, and Talend, that can be used for data enrichment tasks like data cleaning and support for streaming data. They offer good support for tasks such as customizable data cleaning of different formats of data and, in most cases, support streaming data. However, these tools pose several limitations, including:

- Lack of holistic support for the entire data enrichment lifecycle, from the identification and assessment of useful data for enrichment, to the creation, validation, deployment, execution, and reuse of data enrichment pipelines, especially when enrichment operations include critical and uncertain tasks such as data linking, classification, etc.
- Steep learning curve for performing data enrichment tasks: Data enrichment tools should be targeted not only to expert engineers of AI solutions, but also to developers, data scientists and, where possible, domain users. (C3) Limited explicit support for candidate reference datasets used for enrichment, which has been acknowledged by the need of making data FAIR. These limits have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

an impact on the identification and assessment of useful data and on enrichment operations involving data linking.

- Lack of Machine Learning (ML) with Humans-In-The-Loop (HITL) support to combine the power of automation while addressing (assess, tune, correct) uncertainty in data enrichment activities/steps (a challenge acknowledged both in the data management and machine learning communities).
- Limited support for scalable execution of data enrichment pipelines, repeatability of existing processes by setting reusable enrichment pipelines, and sustainability by efficient use of resources that reduces the carbon footprint.

To address the identified challenges, enRichMyData plans to go beyond existing approaches by delivering a toolbox–an integrated set of tools that support the full lifecycle of data enrichment pipelines featuring:

- Data discovery tool (using browsing or full-text search over dataset descriptions/metadata) for reference datasets/enrichment services providing insights using rich data profiles (manually entered or computed).
- Secure data access system using a virtual semantic layer that ensures interoperability and secure access to multiple types of data sources and enables semantics-based data enrichment using the data exposed through it.
- Novel data manipulation and cleaning functionalities for enrichment-related operations on semi-structured data that can be seamlessly integrated within a data enrichment pipeline.
- Tool for semantic enrichment of structured and semi-structured data (such as CSV, Excel, XML, JSON, etc.) by cleaning, transforming and linking data to reference datasets and ontologies.
- Tool for configurable single-endpoint structured data extraction from multi-lingual unstructured text including semantic annotation and extension with vectorized representation.
- Configurable single-endpoint service for classification of documents into standard taxonomies or with custom labels provided by the user.
- Supporting infrastructure to enable the creation of linking services for reference datasets from data providers to be exposed for internal or external enrichment services as well as access mechanisms for search and query.
- A dedicated infrastructure for practical and scalable execution of custom data enrichment pipelines both for large-scale datasets and with reference Big Data knowledge graphs.
- Supporting infrastructure for editing and reusing entire data enrichment pipelines across use cases.
- Support for streaming datasets in data pipelines, with a variety of transport and middleware options.
- A green dashboard to calculate the environmental impact of data enrichment pipeline deployments and support for adjusting computational overheads with regards to their carbon footprint.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

2 Deliverable overview

2.1 Deliverable purpose, scope and context

This document presents the requirements collection and elicitation process that was carried out in the first phase of the enRichMyData project, which aims to provide a novel paradigm and a toolbox for building rich, high-quality and valuable datasets to feed Big Data Analytics and AI applications. The overall purpose of the project is to identify the requirements that need to be satisfied to make the data enrichment solutions developed in the project *useful* in a variety of application domains, but also *innovative*, in the vast landscape of data management solutions that exist today. Development of the project innovations will be therefore driven by two main factors:

- The analysis of business case requirements, which is covered in the deliverable D4.1; these requirements resulted from close interactions between the EMD tool providers and the business case partners and structured interviews
- The analysis of state-of-the-art (SoTA) solutions for data enrichment; this analysis encompasses scientific literature, research and community-driven tools, but also technologies available in the market; the goal of this analysis is to define more precisely the solutions that covers the many different aspects of data enrichment, to understand which aspects are well covered and which provide challenges, and, eventually, identify gaps between research and and tools available in the market to better drive the innovation objectives of the project.

Since data enrichment is a very broad concept, we need to further specify the scope of this deliverable and of the target innovation objectives.

We consider data enrichment techniques and technologies that apply to different data modalities. In the reminder of this document, we refer to structured and semi-structured data as one broad data modality that can be interpreted as, or converted to, some tabular format. For simplicity and by referring to the lower bound in terms of structure, in the rest of the document we use the term (*semi-)structured data* when referring to structured or semi-structured data. While we consider graph-data models as semi-structured data models, transformations across graph-based and tabular data format can be defined and exploited; when this difference is relevant it will be discussed more specifically. Therefore, in the project, we target the enrichment of data that comes in two main data modalities: (*semi-)structured data*, and *textual data*. These modalities cover, all together, a large variety of use cases, which are well represented by the EMD business cases. In fact, we can frame the *main target users* of the EMD data enrichment solutions as those organizations whose data assets consist in large amount of (semi-)structured data, which are complemented by textual data. On the one hand, (semi-)structured data may have textual data fields that need to be processed as such for the enrichment of tabular data, e.g., processed to identify structured information or classified. On the other hand, unstructured data may be enriched in such a way to be better linked to the (semi-)structured core data asset of a company. Consequently, the scope of the deliverable is restricted to data enrichment solutions that can be applied to these two main data modalities.

A main asset we plan to integrate in the EMD toolbox is the exploitation of semantic technologies and knowledge graphs as means to facilitate several aspects of data enrichment, e.g., to access data, to improve the quality, to link to reference shared systems of identifiers and vocabulary, to structure textual content, and so on. Therefore, in the analysis of SoTA and requirements, we will dedicate particular attention to solutions that consider semantic technologies and knowledge graphs.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Overall, this document will provide a clear and comprehensive overview of the project's vision, goals, challenges, and solutions.

2.2 Target audience

The target audience of this deliverable is a reader who is familiar with basic concepts related to data management and machine learning.

2.3 Deliverable structure

The deliverable is organized in three main parts. The first short part presents the methodology used to define the requirements in more detail (Section 3), the second part covers SoTA (Section 4), while the third part covers the actual requirements (Sections 5-6). At the end we draw some conclusions (Section 7) and include bibliographic references grouped by related SoTA section (Section 8).

In Section 4, related to SoTA, we cover different aspects of data enrichment, grouped in three main categories, each one considered in a specific sub-section:

- Aspects related to the enrichment of structured and semi-structured data (Section 4.1)
  - Model transformations, data wrapping and indexing
  - Data discovery
  - Data cleaning
  - Data linking
- Aspects related to the enrichment of unstructured data, focusing on textual data (Section 4.2)
  - Information extraction
  - Classification
- Other aspects of data enrichment (Section 4.3)
  - Scaling, replication and adaptation
  - Streaming data
  - Energy efficiency assessment and improvement

For each aspect, we discuss 1) its scope and the main computational tasks it covers, 2) research work, 3) remarkable tools, i.e., prototypes and community-driven projects or tools available in the market, including tools provided by EMD partners, and 4) the main technical challenges that can lead to technological innovation.

The part related to **requirements** consists in two sections. In Section 5**,** we list requirements specific for tools of the EMD toolbox. In Section 6 we list system-wide requirements.

2.4 Applicable documents

The deliverable contains references to the following documents:

- [D4.1] D4.1 - Business Case Specification
- [DoA] Description of Action

2.5 Updates of this deliverable

Updates of the deliverable may be provided upon major changes to the requirements.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

2.6 Definitions and interpretations

- **Data enrichment**: the process of taking one input data source *S* and returning an augmented data source S', where S' contain additional information with respect to *S* or consists in a transformed representation of *S,* which is deemed to be useful for some downstream application.
- **Knowledge graph**: a data source structured in a graph-based data model, consisting of entities, entity types (also referred to as categories, or classes), relationships between entities, and entity properties. Also abbreviated as "KG".
- **RDF knowledge graph:** a knowledge graph based on the Resource Description Framework[1] (the RDF data model), accessed using related languages, e.g., the SPARQL query language[2], and organized using vocabularies or ontologies specified using semantic web languages such as SKOS[3], RDFS[4] and OWL2[5].

[1] https://www.w3.org/TR/rdf11-concepts/

[2] https://www.w3.org/TR/sparql11-query/

[3] https://www.w3.org/2004/02/skos/

[4] https://www.w3.org/TR/rdf12-schema/

[5] https://www.w3.org/TR/owl2-overview/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

3 Methodology for Requirements Definition

3.1 Requirements Elicitation

As previously indicated, the requirements for the project were collected from two main sources:

1. An analysis of the current scientific literature (referred hereinafter as SoTA). Extensive research has been carried out to understand the current problems that modern data-context have to face. This allowed us to identify ideas, and solutions to be implemented in enRichMyData toolbox to advance the current state-of-the-art.
2. Interviews and User Stories: We conducted semi-structured interviews with market stakeholders, including business case partners. The interviews have been AS-IS interviews, aiming the understanding the working context of each business case partner and their specific needs. The outcomes of interviews led to user stories, which describe a feature from the perspective of a user, their goals, and the benefits/results. User stories were focused on specific datasets and software components to elicit feature requests and explicit requirements from the business case partners.

3.2 Requirements Analysis and Specification

During the requirements elicitation phase, we used a process based on the work of Nuse [2000]. First, we identified key findings from research, including needs, pain points, technical setup, best practices, research challenges, and technological constraints. Then, we linked each key finding to a specific functionality, feature, need, risk, or issue. Finally, for each of these elements, we derived a requirement. Based on this process, we identified three main categories of requirements:

- Business case requirements, referring to each specific business case. These requirements are reported in D4.1.
- Collection-specific requirements, referring to requirements that each collection of tools in the enRichMyData toolbox should address in terms of usefulness and innovativeness. These requirements are reported in Section 5.
- System-wide requirements, which aim to guarantee acceptable levels of service, quality, and usability of the enRichMyData toolbox as a whole. These requirements are reported in Section 6.

Collection-wide and system-wide requirements are formalized using the following template.

Table 1: Technical requirement form template

Requirement ID	ID of the requirement
Requirement Short Title	Name of the requirement
Requirement Description	Description of the requirement
Source	SoTA (State of the Art) / Business Case #number
Relevant enRichMyData tools	List of tool names
Means of verification	Synthetic datasets / Validation with a Business Case / Collaboration between team members etc.
Depends on	Fill in if this requirement depends on the achievement of other requirements
Priority	Could have / Should have / Must have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

3.3 Requirements Validation

The validation process aims to ensure that the identified requirements are in line with the description of the ecosystem that needs to be implemented. This process involves iterative reviews of each requirement by both the component owners and business case partners. The reviews serve three purposes:

- To identify any technical errors or ambiguities in the requirements.
- To determine whether the requirement aligns with the project's goals and does not contradict other requirements.
- To verify that the requirement can be objectively measured and validated.

3.4 Requirements Evolution

This step concerns the process used to manage changes in the requirements that were collected. We have adopted a spiral life cycle (Boehm) model, based on User-Centred Design (UCD) techniques and an agile development approach, as we believe that this iterative evolution process best adapts to the needs of this project.

Any proposed changes will be evaluated in terms of the existing requirements and pipeline architecture so that the costs and benefits of making the change can be assessed and the change can be either approved or rejected. This will ensure that the requirements remain relevant and aligned with the overall project goals, while also allowing for flexibility in response to changing needs.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4 State-of-the-Art

4.1 Organization of State-of-the-art

A broad range of research fields and related technologies are related to data enrichment. In this SoTA we focus in particular on:

- aspects of data enrichment that are covered by the collections of tools that are part of the EMD toolbox, which is comprehensive in terms of covered aspects and selective on the input data considered.
- **semi-structured data**, and **textual data,** which are covered respectively in Subsections 4.2 and 4.3. Some aspects of data enrichment, concern features of the processes that do not depend on a specific data model, such as scalability, replicability and adaptability of data enrichment pipelines, support for streaming data processing, and support for the assessment and improvement of energy consumed; all these aspects are covered in Subsection 4.4.

Each sub-subsection within these three subsections addresses a feature of data enrichment that is covered by one or more collection of tools in the EMD toolbox, and is organized as follows:

- **Scope and example:** it defines the scope of the feature covered in the section and presents some examples to explain the supported feature; examples of solutions may be based on the EMD tools.
- **Research:** it presents recent research advancements in research fields that are relevant for the considered feature.
- **Tools and market:** it focuses on tools, therefore presenting the technological landscape that is relevant for the considered feature; it also contains a specific subsection that covers the tools from the consortium.
- **Research & technical challenges:** it draws conclusions from the previous subsections to identify the most promising assets (functionalities that is good to have) and challenges (functionalities not sufficiently covered by other products in the market) to cover the considered feature with innovative and helpful solutions.

4.2 Approaches to support enrichment of (semi-)structured data

4.2.1 Model transformations, data wrapping and retrieval (WrappR, ResourcR)

4.2.1.1 Scope and example

A critical task to effectively support data enrichment consists in making resources available for processes that are usually required in data enrichment pipelines. There are two main processes that play a key role in data enrichment:

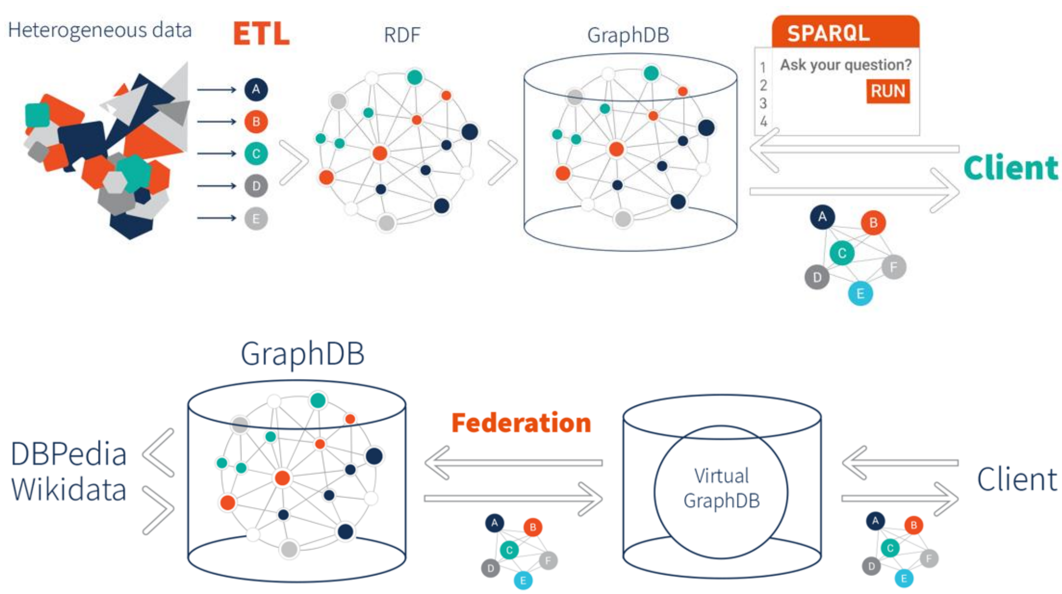
- **Querying a data source using a structured query language**, which is useful to fetch data upon a precise information need; query-based access is supported by data transformation and data wrapping solutions as further explained below.
- **Retrieving resources and related information using keywords**, which is useful in two main application scenarios as further explained below: 1) supporting users in their explorative search, where they can search for data items, e.g., entities, without having knowledge about the data source schema (required for querying); 2) supporting algorithms, especially linking algorithms (see Section 5.2.3), by searching for data items, e.g., entities, that match some input label (the keywords); the retrieved resources are usually considered as candidates to be further evaluated by a more sophisticated algorithm.

Most of the remainder of this section is organized by treating these two types of solutions separately. However, some tools of the consortium like GraphDB offer both wrapping and retrieval functionalities in an integrated solution.

4.2.1.1.1 Data transformation and data wrapping

With respect to query-based data access, two main approaches can be used. One is applying *data transformations* that migrate and *materialize* data into a different data source, changing their data model and/or target schema to meet application requirements. A second approach is *data virtualization;* in this case, data are not migrated into a different data source but are made accessible using data models and/or schemas that may differ from the original one and considered more useful for downstream usage. Both approaches support structured queries over the original data sources. The first family of approaches have the benefit that the materialized data can optimally be queried and integrated but have the downside that data proliferates and is more difficult to keep different systems coordinated, especially for applications of scale. The virtualization approach has the benefit of only having one source of truth. The downside is that very often query performance on virtualized datasets is limited and requires careful engineering and indexing to be performant. These are the primary features covered by tools of the **WrappR** collection.

A large variety of industry-level solutions exist to transform data, and a few exist that support data virtualization. We will discuss solutions that make data accessible using graph-data models and semantic web technologies. These are expected to be particularly useful to support interoperability. We will focus on such technologies as they are the most versatile approach for data for which we do not know the schema *a priori*. While in the EMD project we will consider the business cases, we do not want to produce *ad hoc* solutions for them but rather to be as generic as possible and this means using schema less graph-based technologies such as RDF.



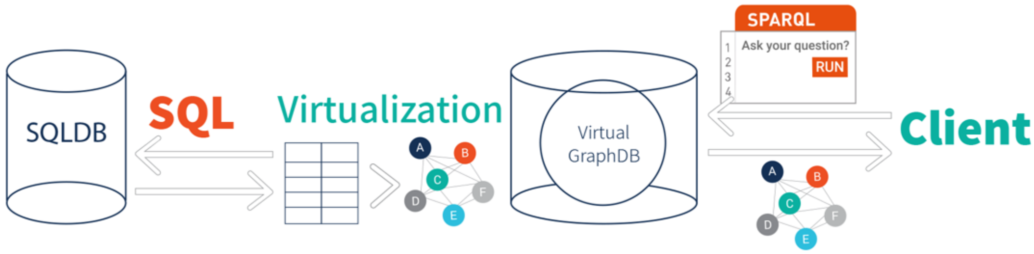


Figure 2: Comparison between the ETL approach with data materialization, data federation and data virtualization

4.2.1.1.2 Retrieval

With respect to keyword-based retrieval, we focus on solutions that are targeted to support algorithms, especially linking, as they are more relevant in the context of data enrichment. These solutions are considered as those relevant for the **ResourcR** tools because they are targeted to prepare the resources that will be used for downstream data enrichment.

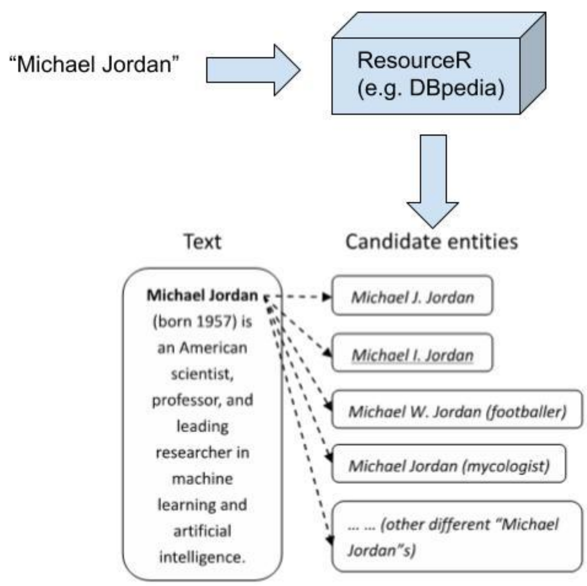


Figure 3: An illustration for the entity linking task. The named entity mention detected from the text is in bold face; the correct mapping entity is underlined (modified version of a figure by [Shen et al. 2015])

4.2.1.2 Research

4.2.1.2.1 Data transformation and data wrapping

The literature on data transformation and data wrapping is large and the main topics can be summarized as follows. Most of these considerations are addressed in the solutions we plan to use in the EMD project.

- **Query optimization:** Efficient query processing and optimization are crucial in data virtualization environments. Researchers work on developing algorithms for cost-based optimization, query rewriting, and adaptive query processing [Ali et al 2022].
- **Mapping languages and tools:** Researchers develop languages and tools for defining mappings between ontologies and data sources, enabling seamless integration and access. This area includes research on mapping languages, such as R2RML, and mapping generation tools [Hert et al 2011, Calvanese et al 2016, Sahoo et al 2009].

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **Semantic query translation:** This research area focuses on translating high-level, ontology-based queries into low-level queries executable over data sources. This includes the development of algorithms and techniques for query rewriting, decomposition, and optimization [Michel et al 2014, Spanos et al 2012, Haw et al 2017].
- **Scalability and performance:** Researchers work on improving the scalability and performance of data virtualization and ontology-based data access systems, addressing challenges like large-scale data integration, distributed processing, and real-time data access [Lanti et al 2016].
- **Access control and privacy:** Ensuring secure and privacy-preserving data access is a critical concern in data virtualization and ontology-based data access. Research in this area focuses on techniques such as access control policies, data anonymization, and secure data sharing. [Ambritta et al 2016].

4.2.1.2.2 Retrieval

When it comes to retrieval from (semi-)structured data, including knowledge graphs (KGs), two commonly used approaches are Information Retrieval (IR) and dense retrieval based on embeddings.

A recent survey conducted by [Hambarde et al. in 2023] discusses the current state-of-the-art models, including methods based on terms, semantic retrieval, and neural approaches. KGs play a crucial role in this context because entity linking approaches heavily rely on structured data resources, such as KGs, to perform effective linking.

Let's take a closer look at each approach:

1. **Information Retrieval (IR):** Information Retrieval is a traditional approach for data retrieval that focuses on matching query terms with the textual representation of data. In this approach, documents or entities are represented as bags of words or terms. IR techniques use various algorithms such as vector space models, boolean models, or probabilistic models to rank the documents or entities based on their relevance to a given query. In the context of datasets or KGs, IR-based approaches typically involve indexing the textual information associated with the data, such as entity names, descriptions, or attributes. Queries are formulated using keywords or structured queries, and the retrieval process involves matching these queries against the indexed data. The retrieved results are ranked based on their relevance scores, which are usually computed using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or BM25 (Best Matching 25). Information retrieval (IR) techniques have reached a stage of being well-established and standardized. However, the challenge lies in optimizing the indexing and querying processes [Azad et al 2019]. One approach that addresses this issue is LamAPI [Avogadro et al 2022], which focuses on studying the impact of various configurations. LamAPI aims to improve the efficiency and effectiveness of indexing and querying by analysing and experimenting with different setups.
2. **Dense Retrieval based on Embeddings:** Dense retrieval is a relatively recent approach that leverages the power of dense vector embeddings for data retrieval. In this approach, each document or entity in the dataset is encoded into a dense vector representation (embedding) in a high-dimensional space. These embeddings are learned using techniques like word2vec, GloVe, or transformer-based models like BERT [Tenney et al 2019] or GPT [Floridi et al 2020]. The retrieval process in dense retrieval involves encoding the query into an embedding as well. Then, the similarity between the query embedding and the document/entity embeddings is computed using metrics like cosine similarity or dot product. The top-k most similar documents or entities are retrieved based on their embedding similarities. Dense retrieval has gained popularity due to its

ability to capture semantic relationships and similarities between data elements. It can handle more nuanced queries and is less reliant on exact keyword matches. Dense retrieval models can be trained using large-scale datasets and can be fine-tuned or optimized for specific retrieval tasks. In summary, Dense retrieval is currently a highly studied area of research. It has been extensively utilized to support text linking algorithms such as BLINK [De Cao et al 2022] and GENRE [Li et al 2020].

Both IR and dense retrieval approaches have their strengths and weaknesses, and the choice of approach depends on the specific requirements of the retrieval task, the characteristics of the dataset or KG, and the available resources. It is worth noting that these approaches are not mutually exclusive, and hybrid approaches that combine the benefits of both IR and dense retrieval techniques are also being explored in the research community.

4.2.1.3 Tools and market

4.2.1.3.1 Data transformation and data wrapping

4.2.1.3.1.1 Research and community

The following notable community and commercial tools are available on the market:

- **ONTOP**[6] is an ontology-based data access system that enables querying of relational databases as if they were RDF triple stores. It provides a mapping language to connect database tables to concepts in an ontology and uses SPARQL queries to retrieve the data.
- **MASTRO**[7] is a knowledge management system that allows users to create and manage ontologies and knowledge bases. It features a visual editor for ontology modelling, an ontology reasoning engine, and an API for integrating with other systems.
- **Morph** is a data integration and transformation tool that allows users to map and transform data from various sources into RDF format. It supports a range of input formats and provides a mapping language for defining the transformations.
- **OptiqueVQS**[8] is a visual query formulation tool for expressing information needs in terms of queries over ontologies. OptiqueVQS is composed of an interface and a navigation graph extracted from the underlying ontologies. The interface components are populated and driven according to the information in the navigation graph that automatically generates mappings between data sources and an ontology-based query language that supports complex queries.
- **SparqlMap**[9] A SPARQL to SQL rewriter based on R2RML specification. It can be used in both extracting RDF from a relational database and rewriting SPARQL queries into SQL.
- **AKSW Sparqlify**[10] is a scalable SPARQL-SQL rewriter.

[6] https://github.com/ontop/ontop

[7] http://obdm.obdasystems.com/mastro/

[8] https://sws.ifi.uio.no/project/optique-vqs/

[9] https://github.com/tomatophantastico/sparqlmap

[10] https://github.com/SmartDataAnalytics/Sparqlify

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.2.1.3.1.2 Commercial offerings

Notable data virtualization commercial solutions, as identified by Garthner in their 2023 Data Virtualization Tools Review[11] are:

- **Denodo Platform**[12] is a data virtualization solution that allows organizations to access, combine, and manage data from multiple sources in real-time. It offers a wide range of connectors to various data sources, a unified data catalog, and a self-service interface for data discovery and exploration.
- **Data Services and Informatica Data Virtualization Solutions by Informatica**[13] are data virtualization tools that enable users to access, integrate, and manage data from disparate sources in real-time. They offer a wide range of connectors, data profiling and quality tools, and support for cloud and on-premises deployments.
- **SAP HANA**[14] is a high-performance, in-memory database platform that enables organizations to process large amounts of data in real-time. It offers a range of analytics tools, machine learning capabilities, and support for a variety of deployment options, including cloud, on-premises, and hybrid.
- **TIBCO Data Virtualization**[15] is a data integration and management tool that allows users to access and manage data from multiple sources in real-time. It offers a wide range of connectors, data profiling and quality tools, and support for cloud and on-premises deployments. It also provides a unified view of data for reporting and analytics purposes.

4.2.1.3.1.3 Retrieval

A wide range of solutions are available in most data management frameworks to facilitate search over structured data, employing the Information Retrieval (IR) paradigm. Many of these solutions leverage external search libraries and frameworks, including but not limited to **Apache Lucene**[16] and the **Elasticsearch platform**[17]. These technologies are very well-known and have reached a high level of maturity and efficiency, making them popular choices for indexing data and enabling effective search capabilities. While there are other notable search libraries and frameworks available, we have omitted them due to the extensive list of options in this domain.

A few solutions are emerging as reference technologies to support efficient dense retrieval. These technologies are designed to handle high-dimensional vector data efficiently, by enabling fast and accurate retrieval of similar items from large-scale datasets, especially k-nearest neighbor search operations. Extension to handle dense vectors and support search on top of them are being introduced in search and database technologies. For example, dense vectors are supported now in Elasticsearch[18] and

---

[11] https://www.gartner.com/reviews/market/data-virtualization

[12] https://www.denodo.com/en/denodo-platform

[13] https://www.informatica.com/

[14] https://en.wikipedia.org/wiki/SAP_HANA

[15] https://www.tibco.com/products/data-virtualization

[16] https://lucene.apache.org/

[17] https://www.elastic.co/

[18] https://www.elastic.co/guide/en/elasticsearch/reference/current/dense-vector.html

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

PGVector extends PostGres with support for vector representations[19]. In the following we mention a few of the most popular applications as of today. A larger number of such solutions are likely to populate the market in the future due to the relevance of dense vectors in most AI-based applications.

- **FAISS**[20], which stands for "Facebook AI Similarity Search." FAISS is a widely used library for efficient similarity search and clustering of dense vectors. It offers state-of-the-art performance for tasks such as nearest neighbour search, approximate k-nearest neighbour search, and similarity search over large-scale vector datasets.
- **Milvus**[21] is an open-source vector database that can manage large vector datasets and supports multiple vector search indexes and built-in filtering.
- **Weaviate**[22] is an open-source vector database. It supports the storage of data objects and vector embeddings from arbitrary ML-models and promises to scale seamlessly into billions of data objects.
- **Qdrant**[23] is a vector database & vector similarity search engine. It deploys as an API service providing search for the nearest high-dimensional vectors. With Qdrant, embeddings or neural network encoders can be turned into full-fledged applications for matching, searching, recommending.
- **Vespa**[24] supports the co-location of vectors, metadata, and content on the same item on the same node, to make sure that inference can achieve scalable performance; it scales this across nodes to handle a large amount of data and traffic.

4.2.1.3.2 Tools from the consortium

4.2.1.3.2.1 Data transformation, wrapping and retrieval

4.2.1.3.2.1.1 Ontotext GraphDB

GraphDB is an enterprise-scale semantic repository that fully supports all relevant W3C semantic web standards: RDF and all its serializations, SPARQL (Query, Update, Federation, and its related protocols including Graph Store Protocol), RDF-star, reasoning (RDFS, RDFS-plus, OWL-Horst, OWL RL, OWL QL, and custom).

In addition, GraphDB implements several features/connectors for working with non-RDF data (hybrid storage strategies):

- SQL Access over JDBC[25]: uses saved SPARQL queries that are mapped to relational tables, to allow the consumption of relational views over semantic. Uses Apache Calcite and JDBC to expose the relational schema, so that traditional analytics tools like PowerBI and Tableau can discover the relational structures and help the user make joins for analytic purposes.

[19] https://github.com/pgvector/pgvector

[20] https://faiss.ai/

[21] https://milvus.io/

[22] https://weaviate.io/

[23] https://qdrant.tech/

[24] https://vespa.ai/

[25] https://graphdb.ontotext.com/documentation/10.0/sql-access-over-jdbc.html

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- Relational Virtualization[26] (Ontology-Based Data Access, OBDA) to access relational data in-situ (NoETL) by translating SPARQL queries to SQL through an R2RML or OBDA mapping. SQL results are translated back to SPARQL results and can be used together with real RDF data through federation. GraphDB uses the leading open OBDA implementation: ONTOP. Our OBDA experience includes:
- POC implementations for a major manufacturer and a major insurance company.
- Production implementation to virtualize the Microsoft Academic Graph and OpenAlex for "Science KG" projects.
- MongoDB integration[27]: large amounts of JSON (semi-structured) data can be kept in Mongo and queried using SPARQL. Then the results are interpreted as JSON-LD and can be joined to real RDF data. Usually, we also store key JSON fields redundantly as real RDF in order to speed up joins and retrieval.
- Connectors[28] to Lucene[29], Solr[30] and Elasticsearch[31] for implementing full-text and faceted search. The connectors use a powerful language for specifying which parts of RDF data should be synchronized to the full-text engines and how they should be indexed. Then synchronization happens automatically and incrementally as RDF data is changed. Queries to the full-text engines can be run from SPARQL and joined to results from real RDF data, and also directly from the search engine's API. The Elastic stack enables further enterprise analytics through tools like Kibana.
- Kafka integration for Enterprise Data Integration scenarios:
    - The Kafka source connector[32] (output from GraphDB) provides a means to synchronize changes to the RDF model to any downstream system via the Kafka framework. Changes are manifested as JSON messages. This enables easy processing of RDF updates in any external system and covers a variety of use cases where a reliable synchronization mechanism is needed.
    - The Kafka sink connector[33] (input to GraphDB) allows capture of Kafka events to GraphDB. It supports smart updates with SPARQL templates, thus reducing the amount of code needed for raw event data transformation.
- GeoSPARQL support[34] for indexing and querying of GIS data, fully compliant with the OGC GeoSPARQL 1.0 specification (and we are considering GeoSPARQL 1.1)

[26] https://graphdb.ontotext.com/documentation/10.0/virtualization.html
[27] https://graphdb.ontotext.com/documentation/10.0/mongodb-graphdb-connector.html
[28] https://graphdb.ontotext.com/documentation/10.0/connectors.html
[29] https://graphdb.ontotext.com/documentation/10.0/lucene-graphdb-connector.html
[30] https://graphdb.ontotext.com/documentation/10.0/solr-graphdb-connector.html
[31] https://graphdb.ontotext.com/documentation/10.0/elasticsearch-graphdb-connector.htmlю
[32] https://graphdb.ontotext.com/documentation/10.0/kafka-graphdb-connector.html
[33] https://graphdb.ontotext.com/documentation/10.0/kafka-sink-connector.html
[34] https://graphdb.ontotext.com/documentation/10.0/geosparql-support.html

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.2.1.3.2.1.2 Ontotext Platform and GraphQL Federation

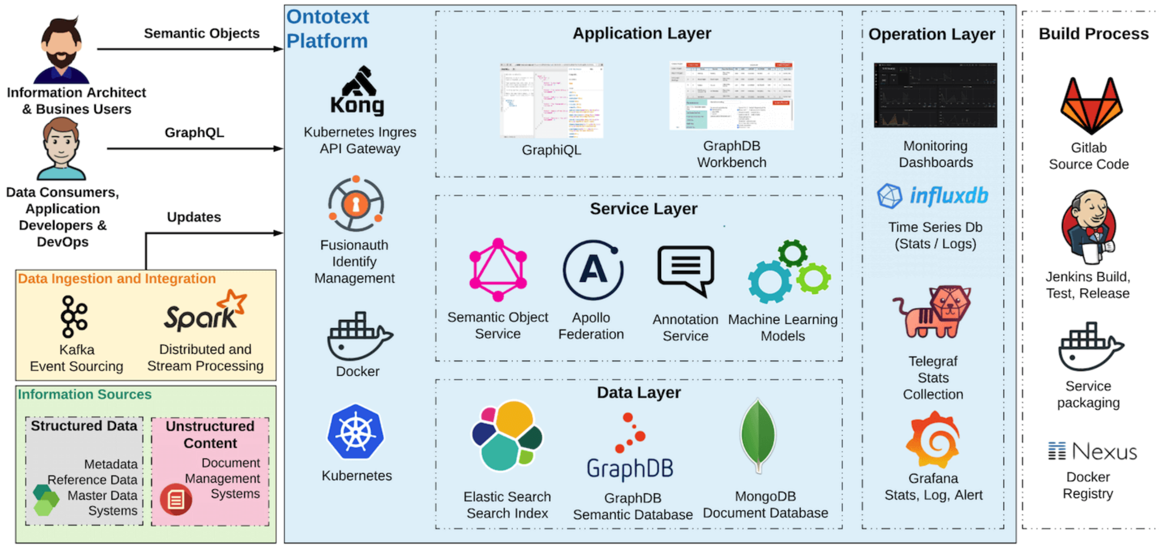


Figure 4: Ontotext Platform and GraphQL Federation

The Ontotext Platform is a collection of value-added services. Two of them (**Semantic Objects and Semantic Search**) enable data wrapping and GraphQL access:

- GraphQL[35] access to RDF data through Semantic Objects[36]. This creates a "bounded context" view over a selected part of the RDF data and exposes a GraphQL schema for easy creation of GraphQL queries. The queries are transpiled (dynamically translated) to SPARQL using optimizations such as correlated subquery (lateral join), selecting query strategy using database statistics, and selecting only required object fields and sub-objects. Query results are translated to JSON that conforms to the structure of the query. This GraphDB add-on follows the GraphQL Federation specification and allows to federate the RDF data on GraphQL level with other systems and data sources.
- Easier Elastic integration through Semantic Search[37]. After creating a Semantic Objects model with a few Elastic annotations, this component creates Elastic indexes automatically and allows querying of Elastic and RDF data from GraphQL.

GraphQL is used to build "facades" between various storages.

- You can federate data between GraphQL endpoints using two mechanisms, Apollo Federation and GraphQL Mesh

4.2.1.3.2.1.3 Apollo Federation

Apollo Federation[38] provides a mechanism to combine multiple GraphQL endpoints and schemas into a single aggregate endpoint and composite schema. The basic principles of GraphQL federation are as follows:

---

35 https://graphql.org/

36 https://platform.ontotext.com/semantic-objects/

37 https://platform.ontotext.com/semantic-search/

38 https://www.apollographql.com/docs/federation/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- Collects the schemas from multiple endpoints.
- Combines the schemas into a single composite schema.
- Serves the single composite schema to the user completely hiding the inner dependencies.
- When a query is performed, the Federation Gateway calls each of the endpoints in a specific order and combines their responses.

The approach allows you to combine schemas to provide a single unified view over your data. In turn, this helps increase the modularity of your architecture. Federation works on queries only. You cannot use mutations directly on the federated schema, but you can still perform mutations on the constituent services.

It is recommended to use concern-based separation when combining multiple schemas in Apollo. In a computing example, you can keep information on all users and their operating systems in one schema. The information on users and their hardware purchases can be another schema, etc.

The GraphQL add-on (Platform) ships with a dedicated federation server, called the Apollo Federation Gateway. The gateway acts as a router. The schemas of all constituent GraphQL services are provided to the router. There must be no conflicts between the schemas. The process is called schema composition.

The Platform provides an extended Apollo GraphQL federation gateway[39], that provides some notable extensions:

- RBAC participation
- transaction/log aggregation and correlation
- health, good to go status checks and monitoring
- declarative federation configuration

Apollo Federation is enabled by toggling the graphql.federation.enabled: "true" configuration line. By default, the Platform seamlessly integrates between the Semantic Objects Service and the Search Service. If you want to add further schemas, you will need to define them, using the @external[40] directive and extended types[41]. The Platform always uses the "id" field - which corresponds to the RDF subject - as a key[42].

4.2.1.3.2.1.4 GraphQL Mesh[43]

A network to build GraphQL Gateways on top of REST, gRPC and other services. Mesh started with the intent of providing a WrappR service that does not require much coding and maintenance and does not strongly depend on GraphQL. Mesh works like Apollo, providing transformations on the queries and output schema and compiling them down to queries for the individual services. It also ships together with a server that provides the federation logic.

---

[39] https://github.com/apollographql/apollo-server/tree/master/packages/apollo-gateway

[40] https://www.apollographql.com/docs/federation/federated-types/federated-directives/#referencing-external-fields

[41] https://www.apollographql.com/docs/federation/v1/entities#extending-entities

[42] https://www.apollographql.com/docs/federation/federated-types/federated-directives/#key

[43] https://the-guild.dev/graphql/mesh/docs

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Unlike Apollo, Mesh allows you to directly perform transformations to alter your federated schema. You can also add custom GraphQL types and resolvers and employ different caching strategies. Mesh is also a superset on Apollo, you can perform Apollo Federation using the Mesh tools.

Mesh aims to support data providers out of the box. It currently supports the following sources:

- GraphQL
- Open API
- JSON Schema
- SQL - PostgreSQL, SQLite, MySQL
- gRPC
- SOAP
- MongoDB with Mongoose
- OData
- Apache Thrift
- Neo4J

While Mesh allows you greater flexibility compared to Apollo, this comes at the cost of more complex resolvers and transformations.

4.2.1.3.2.1.5 LamAPI by UNIMIB

LamAPI: (short for Label Matching API) is a versatile tool for IR-based entity retrieval, which allows to perform entity lookups from knowledge graphs, and also retrieve related data such as entity relations and literal values. This makes it easy to get a comprehensive understanding of the data in a knowledge graph. LamAPI also includes an additional service for datatype identification, which uses a combination of regular expressions and spaCy to accurately identify the data types. LamAPI is also augmented with type-based filtering features. The tool supports string-based retrieval but also hard and soft filters [Cutrona et al 2021] based on an input entity type (i.e., rdf:type for DBpedia and Property:P31 for Wikidata). Hard type filters remove nonmatching results, while soft type filters promote or demote results when an exact match is not feasible. These filters are useful to support either EL in texts (e.g., by exploiting entity types returned by a classifier [Raiman et al 2018, Onoe et al 2020]), or in tables (e.g., by exploiting a known column type (rdf:type) to filter out irrelevant entities).

4.2.1.4 Research & technical challenges

4.2.1.4.1 Data transformation and data wrapping

Here are some research and technical challenges related to data wrapping, which will be addressed in the EMD project.

- **Wrapping heterogeneous sources.** Increasingly specialized solutions are appearing on the market. Besides the well-known relational databases, now document stores (such as Mongo DB) and even highly specialized stores such as the vector databases that we see appear as a result of the success of large language models.
- **Proper query** due to the need to produce an optimized set of queries that can seamlessly access and integrate data from multiple, heterogeneous sources. It is complex due to the need to address differences in data models, formats, semantics, and query languages, and to ensure efficiency, accuracy, and consistency in the delivery of unified data views.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.2.1.4.2 Retrieval

Here are some research and technical challenges related to data retrieval from datasets or Knowledge Graphs (KGs):

1. **Indexing a new data source:** The challenge here lies in developing an index for a new dataset while minimizing the need for manual operations. Efficient indexing is crucial for quick and accurate retrieval of data. Finding automated methods to generate indexes for new datasets would reduce the burden of manual labour and improve the overall efficiency of data retrieval processes.
2. **Dense indexing using embeddings:** This challenge pertains to constructing an indexing mechanism specifically designed for structured data such as tabular data. Embeddings, which represent data points in a lower-dimensional vector space, can be leveraged to encode the structural and semantic relationships within the data. Developing effective techniques to utilize embeddings for dense indexing would enhance the search capabilities and enable more efficient retrieval of structured data.

4.2.2 Data discovery (DiscoverR)

4.2.2.1 Scope and example

With data discovery we refer to functionalities that support a user who is enriching an input dataset to search relevant external data sources and/or select specific features of the data to fetch for the enrichment task. Data discovery techniques have also been considered as relevant to support business data exchange in data marketplaces [Abbas et al. 2021]. Data from external data sources can be available via download, dedicated or open connections, or REST APIs. Therefore, datasets and APIs are relevant for data discovery applications. Supporting these search and selection tasks implies additional support for getting insights into the content of the potential data source to assess its usefulness for the enrichment process. The main objectives of data discovery functionalities are:

- Find data sources that are relevant for a data enrichment task at hand.
- Provide insights on the content of a potentially relevant data source.
- Support an early assessment of the usefulness of the data source.
- Hhelp understand the structure of the data in the data sources to make it possible to fetch data from it.

It is usually assumed in order to make it possible to search and inspect potential data sources, these data sources must be catalogued, described, profiled and, in general, pre-processed. We can group data discovery functionalities into three main categories:

- Web-scale dataset or API cataloguing and search
- Deep schema insights
- Knowledge discovery

We also clarify that knowledge discovery does not refer here to data mining and machine learning techniques proposed to uncover domain-specific patterns from data, which are also frequently referred to as Knowledge Discovery and Data Mining methodologies (see, for example, the KDD conferences).

4.2.2.1.1 Web-scale dataset and API cataloguing and search

Cataloguing and dataset search solutions provide access to datasets descriptions, which usually include names, natural language descriptions and metadata. Several applications supporting data catalogs in specific organizations exists; since many of them have functionalities that try to extract information that go beyond explicitly metadata inserted by the users, providing, to some extent, some deep schema insights, they are covered in the latter category. However, solutions to crawl and index dataset descriptions available online to provide a web-scale dataset search engine have been proposed. An example of search interface over a data catalogue is given in Figure 5, while an example of dataset search engine interface is given in Figure 6. The typical search paradigm is based on faceted search, that includes keyword-based search over names, descriptions and metadata, as well as filters based on metadata. Metadata can be defined using agreed upon vocabularies such as the Data Catalog Vocabulary (DCAT)[44], which is the official W3C recommendation for data cataloguing, or Schema.org[45], a widely used vocabulary. Metadata can also be inferred using classifier and machine learning approaches. Finally, since data can be available via API, we will briefly cover the status of API search paradigms.

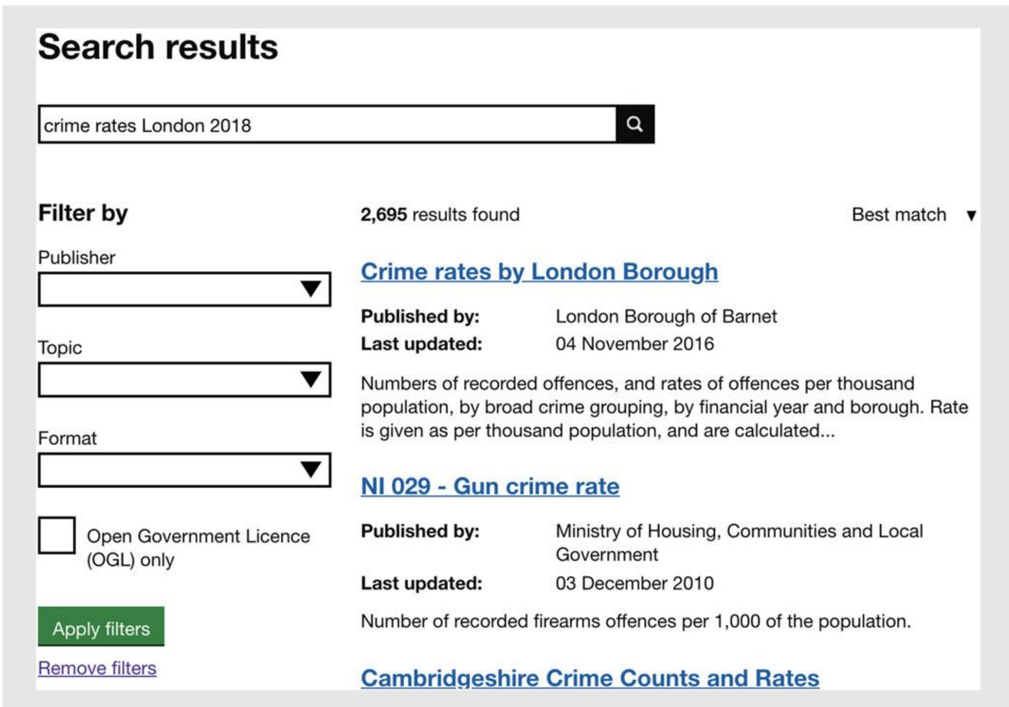


Figure 5: Dataset search engine for the UK government's open data portal, data.gov.uk. Form inputs create statements to query the underlying data; original image and caption from [Chapman & al. 2020].

[44] https://www.w3.org/TR/vocab-dcat-3/

[45] https://schema.org/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

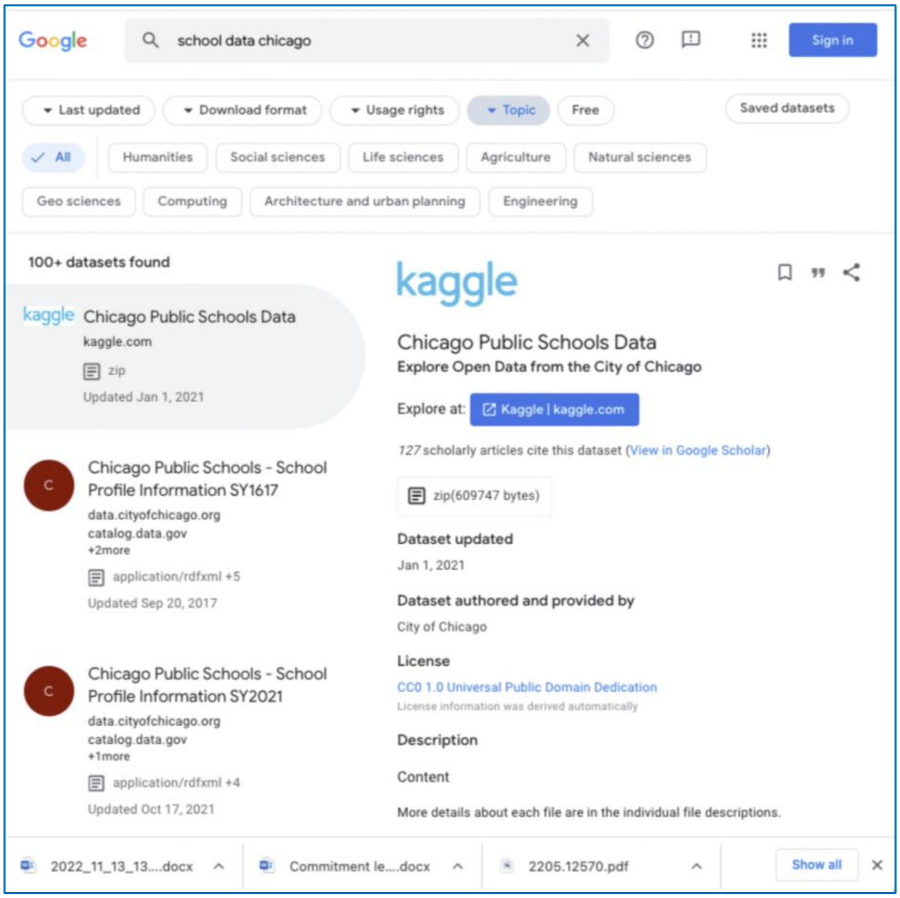


Figure 6: Interface of Google's Dataset Search with results for the query "school data chicago". The interface presents results of datasets matching the query and several filters to filter data based on native and inferred metadata.

4.2.2.1.2 Deep schema insights

With deep schema insights we refer to solutions that allow to get information about the content of a dataset by looking beyond what is specified by their name and their explicit description and metadata. Deep schema insights refer to functionalities that help users inspect the content of data sources, in terms of both structure and quality. Examples of pieces of information provided to users are: types of formats and data in data collections, structure and hidden relationships, value distributions for specific attributes (e.g., min/max/avg values, most frequent values), semantic tags describing the data content, data usage and stewardship. These pieces of information are typically accessible with a visual interface and through different dashboards. Metadata are relevant for deep schema insights solutions, thus establishing some continuity with dataset search solutions, and in fact several solutions that provide deep schema insights are solutions that support data catalogs. However, being that these data catalog applications developed for organizations managing a variety of data sources within their organization system, more advanced solutions are made available in terms of both automatic metadata extraction, and data quality insights. Commercial applications providing deep schema insights can be categorized as dedicated applications for data cataloguing (focusing on metadata and tags), dedicated applications for data profiling (focusing on statistics and hidden relationships), or applications that come as components of other data management platforms, i.e., database management systems, or data integration platforms. Metadata extraction and management, and statistics calculation, are features provided to some extent in different applications that provide deep schema insights; however, some of the most advanced research approaches may not have been included in commercial applications yet. Features to reconstruct the relationships across data sources are developed especially for NoSQL data sources, which do not have a pre-defined and prescriptive schema associated with them. Among NoSQL data sources we include tabular databases (e.g., Cassandra, Big Table, HBase), key-value databases (e.g., Redis), document databases (e.g., MongDB), and, especially relevant for enRichMyData where we intend to use graph-based knowledge representation to support data enrichment, graph databases (e.g., Neo4J and all RDF triple stores). In most of the above databases, data can be stored or exchanged in JSON format. Researchers have investigated especially solutions to tackle harder problems, like finding hidden relationships across data sources using AI methods, which are included in some of the most advanced commercial applications originating as spin-offs of research groups (e.g., Trifacta[46]). Applications to reconstruct the schema of JSON files have been proposed. However, commercial solutions for data profiling and schema reconstruction focus mostly on relational data, other tabular data formats and JSON files. Based on the available knowledge, it seems that most of dedicated data profiling applications do not provide profiling methods explicitly addressing graph-based data, which have a peculiar structure compared to other NoSQL data sources. Commercial graph databases therefore provide some profiling functionality. However, if we consider large knowledge graphs that use a rich vocabulary and complex ontologies (e.g., hundreds of types and thousands of properties), naïf solutions may not be enough. For this reason, a large body of research has addressed the problem of profiling RDF knowledge graphs in such a way to help users (and machines) understand their content, quality and structure, by exploiting schema pattern extraction, statistics, summarization or a combination of these features.

As an example, Figure 7 depicts the search interface of ABSTAT, one of the EMD tools that provide similar features. A pattern in ABSTAT is a triple of the form <class, property, class> and is intuitively interpreted as describing that there are instances of the first (subject) class, e.g., dbo:Politician, which are linked to instances of the second (object) class, e.g., dbo:Country, with the specified property, e.g., dbo:country. Most of the reviewed research work will focus on solutions to profile RDF data and support access to these profiles. Finally, there are two main functionalities to support access to collections of schema-level descriptions of RDF data:

- **Browsing**, which supports exploration of content by means of summaries (e.g., statistics), filters and hyperlink navigation.
- **Search**, which supports full-text search over the vocabulary used in the datasets.

Finally, since data can be available via API, we will briefly cover the status of API profiling solutions.

[46] https://www.trifacta.com/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

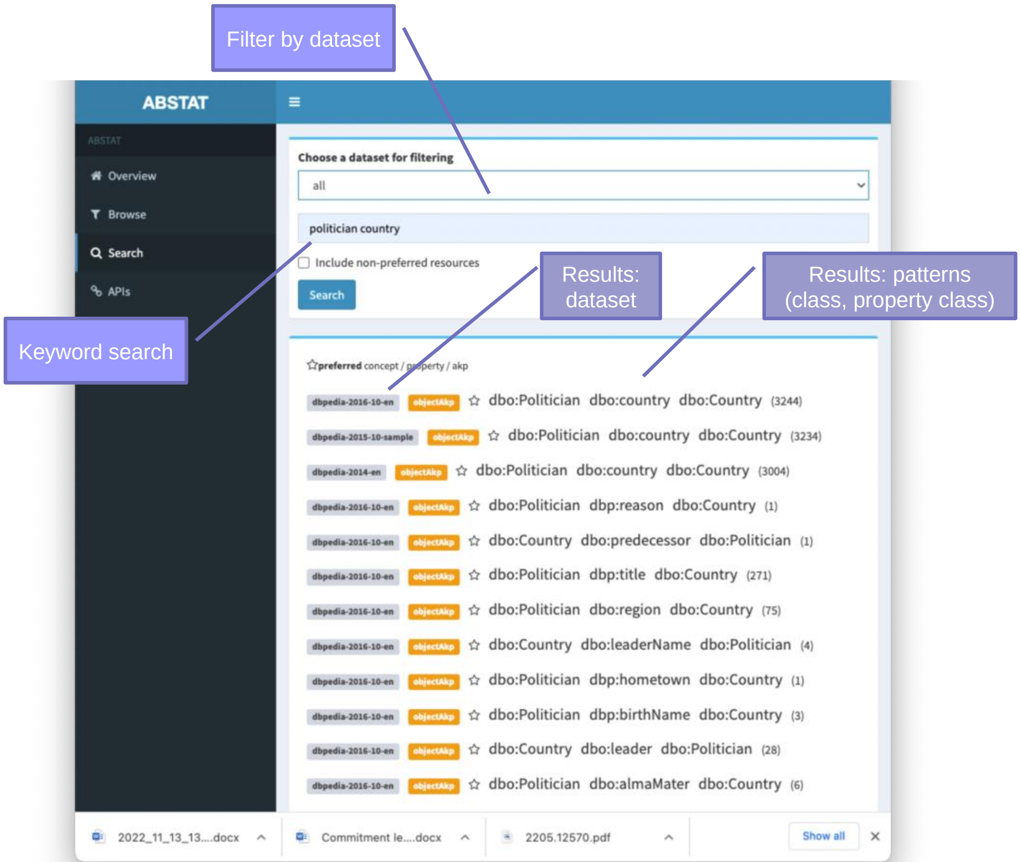


Figure 7: Search interface of the RDF data profiling tool ABSTAT, with results for a search for "politician country". The interface list schema patterns found in different datasets expressed as triples (class, property, class); users can filter the results by dataset.

Finally, we observe that in all datasets that can be indexed, some schema insights could be obtained by issuing (schema-agnostic) *full-text searches over the data*; however, this solution is impractical if applied to a collection of datasets: the retrieved results may be too many and too specific, thus making it difficult to aggregate the results or preventing the user from understanding the coverage of the datasets. Also, full-text search over semi-structured datasets is covered, in our classification, by data wrapping and indexing, discussed in Section 4.2.1. Hereby we only notice that some solutions that provide also cataloguing functionalities for open data, e.g., CKAN[47], can provide this functionality when users drill down into a specific dataset.

4.2.2.1.3 Knowledge discovery

- With knowledge discovery, we refer to features of data enrichment applications that help find relevant data from external data sources that can be fetched to enrich an input dataset. Knowledge discovery features are therefore expected to be available for a user who is working on a specific dataset. We do not cover classification mechanism for structured data, which are reviewed in a recent survey [Borisov et al. 2022].

[47] https://ckan.org/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- With respect to the input, we focus especially on the enrichment of tabular data, because it is the input format that is more frequent in industry applications, and because it could serve as a common interchange format; in particular, JSON data and RDF data can be transformed into tabular data (e.g., in RDF, as views built via SPARQL query results). For other data formats such as relational databases and (native) RDF data, we will include some pointers. Finally, with respect to the external data sources, fetching data requires interoperable solutions that can be based either on direct connections with the target external data sources (e.g., SPARQL queries via SPARQL endpoints) or connections mediated via API (e.g., REST interactions with the API). Therefore, we consider relevant both kinds of solutions and solutions that allow to find available data in a known API. By focusing on tables as input datasets, our interpretation of knowledge discovery extends the interpretation of *tabular search* proposed in [Chapman & al. 2020], by also considering data extension services available via APIs.

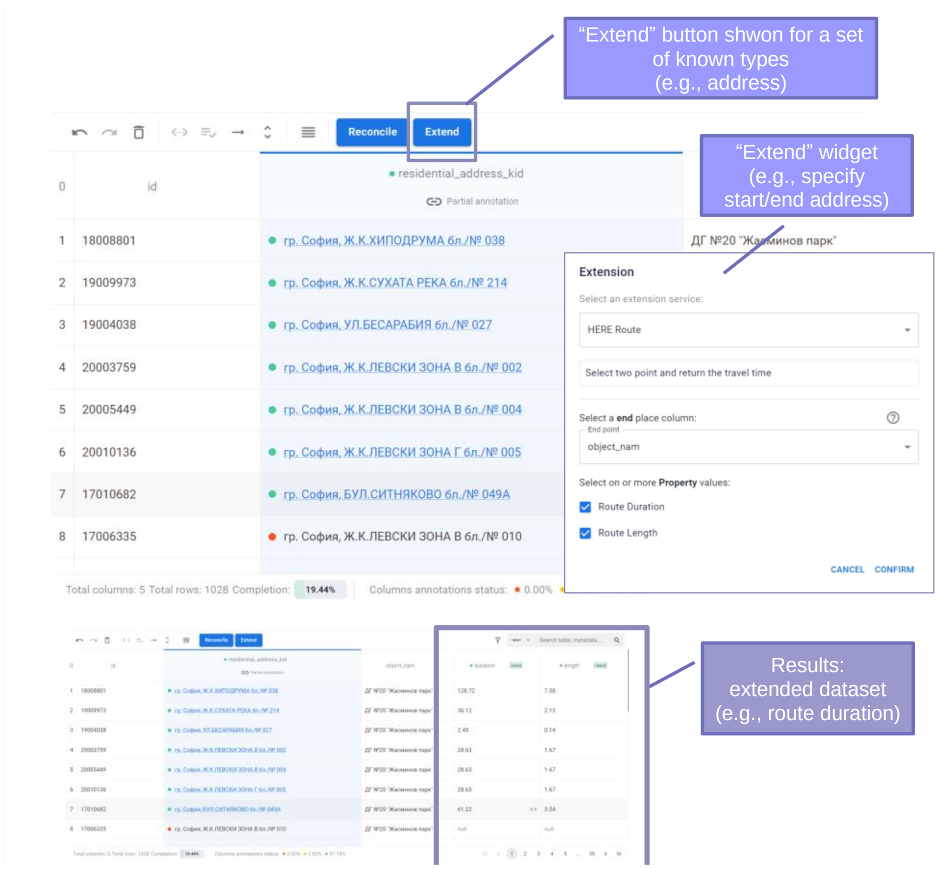


Figure 8: Example of knowledge discovery features with the SemTUI interface, one of the tools in the DiscoveR collection of enRichMyData. The picture shows a button to access data extension services compatible with a table column, the widget to select the attributes to use to fetch the data, and the outcome of the extension operation.

Typical knowledge discovery features, also exemplified in Figure 8, are:

- Reporting or recommending data extension services (either via direct database connection or API) that are compatible and relevant for the input data.
- Supporting or recommending the selection of relevant attributes for fetching data from the external source.
- Fetching the data and testing the outcome of the extension based on the selected service.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.2.2.2 Research

4.2.2.2.1 Web-scale dataset and API cataloguing and search

With respect to cataloguing and dataset search, [Chapman & al. 2020] provides an excellent survey about solutions proposed for dataset search up to that date. The contributions of the survey can be summarized as follows:

- Dataset search engines, e.g., Google Dataset Search, exist, however, research on advanced dataset search mechanisms is still quite in its infancy and growing. Two different mechanisms for dataset search are proposed: (1) issue query, return datasets; (2) issue query, build datasets. Mechanisms of the second category (e.g., tabular search) will be discussed under the knowledge discovery paragraph category. As for queries-returning-the-datasets mechanism, in addition to approaches based on common search architectures often implemented in existing data marketplaces and data portals (see also examples in Figure 5 and Figure 6), several contributions from different search sub-communities are reported, such as databases, information retrieval, entity-centric search, and tabular search (covered in the knowledge discovery paragraph). Basic centralized search is based on metadata catalogues and is powered by data portal software like CKAN, Socrata, and OpenDataSoft, and search engine technology (e.g., Solr); it is implemented in large enterprises, scientific data portals (e.g., Elsevier[48], Figshare[49], Dataverse[50]), and data marketplaces. Queries are usually formulated in Contextual Query Language (CQL)[51], and executed on dataset descriptions, with different solutions to render the results and let users refine the queries iteratively. Decentralized search would require searching over linked open data, which are at least theoretically accessible over the web, or after crawling dataset descriptions on the web. Vertical solutions have been proposed in specific domains. Surprisingly, the survey does not mention approaches based on searching and using APIs.
- The survey discuss several challenges and opportunities for improvement for each level of dataset search, i.e., query languages (going beyond keywords), query handling (differentiated access and security), data handling (automatic creation of metadata, including metadata that link datasets to ontologies - a task that is covered by tools of the LinkR collection in enRichMyData; better entity-centric search after conversion of the data or their DCAT descriptions to RDF - also a task that is covered by tools of the LinkR collection in enRichMyData); result presentation (interactivity).

Recent work after this survey has expanded the frontiers of dataset search by focusing on more advanced approaches to: extract metadata, e.g., in [Farshidi & Zhao 2022], or automatically annotate tables (covered in Section 4.2.4); tabular search (in our terminology "knowledge discovery" – discussed here below); dataset recommendation. With respect to the latter, we mention work in [Wang et al 2020], which propose to use ontologies to improve dataset retrieval in the biomedical domain, [Wang et al. 2022], which exploits citation networks, and [Santos et al. 2022], which not only evaluate similarity across datasets, but also check which table may be joinable with an input query table. Another interesting approach is provided in Voyager [Bogatu et al. 2022] a system that indexes and profiles tables and uncovers hidden relations that can be helpful for integration. The goal of the application, which comes

[48] https://www.elsevier.com/authors/tools-and-resources/research-data/open-data

[49] https://figshare.com/

[50] https://dataverse.org/

[51] https://www.loc.gov/standards/sru/cql/

with a web interface, is to support the identification and understanding of target data and the discovery of useful relationships.

There has been a body of work on describing APIs to support search, and eventually, automatic composition. This work led to elaborated semantic web languages to model web services like WSMO and OWL-S, and, eventually, lighter-weight solutions applied to APIs [Panziera et al 2012, Fariss et al 2019]. More recent approaches moved from search to full-fledged recommendation of APIs, with several approaches proposed [Nowaz et al 2022]. For example, [Almarimi & al. 2019] proposes an evolutionary approach to recommend APIs, with the goal of creating mashups; [Wei et al 2022] propose contrastive learning for API recommendations. While researchers have somehow moved from cataloguing and search to recommendation, some practical solutions to build portals for API search have emerged in the market; they will be covered in Section 4.2.2.3.

4.2.2.2.2 Deep schema insights

Data profiling techniques are used for a variety of data models to extract metadata and insights from data. Profiling activities range from ad-hoc approaches, such as eye-balling random subsets of the data or formulating aggregation queries, to systematic inference of structural information and statistics of a dataset [Abedjan et al. 2019].

Tasks deemed difficult in *structured data profiling* are dependency discovery - supported, e.g., in the Metanome prototype tool [Papenbrock et al 2015], and profiling for dynamic data. When data are very large, many propose to wisely sample a portion of data instead of scanning the full dataset; a review of such techniques is proposed in [Liu & Zhang 2020]. Approaches to profile semi-structured data, in particular JSON data, have been proposed in [Möller et al 2019] and [Loetpipatwanich & Vichitthamaros 2020].

Knowledge graphs and RDF data are often designed for being open and reused by third parties; in addition, their structure may be very complex and use different vocabularies. Therefore, the problem of understanding their structure has been largely investigated in the semantic web community. Although there is no clear definition of what a knowledge graph understanding is, for the purpose of this deliverable we refer to knowledge graph understanding as the process of gaining insights by accessing and exploring a set of simple structures that are easily understood and meaningful. Structural summarization approaches aim to create a summary to support data understanding and visualisation of complex graphs: (i) by considering a set of rules that extract subtypes and subproperties to represent many nodes and edges [Campinas et al 2012] and [Konrath et al 2012], or (ii) by extracting clusters to group a set of similar concepts and properties [Consens et al 2015]. Pattern mining methods instead extract patterns from the RDF graph that “best” represent the input graph. Such approaches produce summaries that are capable of summarizing entities in terms of their neighborhood similarity up to a certain distance and a specified bound to indicate the maximum number of the desired patterns. The summaries/patterns are then chosen to satisfy and maximize informativeness (which should capture the total amount of information; entities and their relationships in a knowledge graph) and diversity (which should cover diverse concepts with informative summaries) [Song et al 2018, Louati et al 2011, Riondato et al 2017]. Statistical methods aim to produce summaries that quantitatively represent the content of the entire RDF graph. Quantitative information might be used by users to decide if the dataset is useful for them or not. For example, LODSight [Dudáš et al 2015] displays a summary of classes, datatypes and predicates used in the dataset. The visualization allows users to quickly and easily find out what kind of data the dataset contains and its structure. It also shows how the vocabulary is used in the dataset. Similarly, SPADE [Diao et al 2019] uses OLAP-style aggregation to provide users with meaningful content of an RDF graph. It chooses aggregates that are visually interesting, a property formally based on statistical properties of the aggregation query

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

results. Users may refine a given aggregate, by selecting and exploring its subclasses. Finally, hybrid approaches developed so far combine methods from the structural, statistical, and pattern-mining classes to provide meaningful summaries. Loupe extracts types, properties, and namespaces, along with a rich set of statistics about their use within the dataset [Mihindukulasooriya et al 2015]. It offers a triple inspection functionality, which provides information about triple patterns that appear in the dataset and their frequency; patterns have a similar form and interpretation as ABSTAT's patterns shown in Figure 4. The approach in [Čebirić et al 2015] instead is a query-oriented approach. The summary enables static analysis and helps formulate and optimize queries. The scope of such summaries is to reflect whether the query has some answers against this graph or to suggest a simpler way to formulate the query. In summary, information that can be easily inferred is excluded. Such an approach has a high computational complexity.

ABSTAT, the tool included in the enRichMyData toolkit, combines patterns and statistics and is quite similar to Loupe: it does not use clustering to summarize information but patterns; however, it reduces the number of patterns extracted from a dataset based on an ontology-based minimalization technique. In this way, ABSTAT excludes the summary patterns that can be easily inferred from the ontology producing profiles that are more concise.

Structural analysis of *APIs* is performed as a subtask in the approaches that supports search and composition, so we refer to approaches cited in the "Dataset and API cataloguing and search" paragraph.

4.2.2.2.3 Knowledge discovery

Approaches reviewed in [Chapman & al. 2020] as tabular search propose solutions to augment a query table either vertically, i.e., retrieving similar data to the ones in the query table, or horizontally, i.e., extending the table with more columns based on the data available in a query table. Horizontal augmentation, the most relevant one in industry applications and when data enrichment is used to feed downstream data analysis that must be based on high-quality data, is performed either by specifying a desired attribute name, or by finding new potential column names. Examples of such systems are presented in [Cafarella et al 2008, Yakout et al 2012, Zhang & Balog 2018], and search for tables that can be joined with a query table. The *Mannheim Search Join Engine* [Lehmberg et al 2015] automatically performs such table extension operations based on a large corpus of Web data that includes HTML tables, Microdata and Linked Data.

An interesting problem for knowledge discovery is to help understand which attributes from an external source can be used, a form of exploratory search over the structured data in the external data source. Some solutions to related problems have been proposed but there is a large space for improvement. Suggestions of relevant attributes of linked data sources may come from data profiling applications: TermPicker [Schaible et al 2016] and the enRichMyData tool ABSTAT [Cutrona et al 2019] support vocabulary suggestion features that can or have been integrated into data enrichment tools. A sophisticated and interesting approach that is not been applied in the context of data enrichment applications is *Merlin* [Qarabaqi & Riedewald 2015], an interactive system that supports exploratory searching in large databases: the users specify imprecise conditions on the entities, and the system offers real-time interactive mechanisms to refine the query conditions to get the desired data.

A few approaches have been proposed that exploit knowledge graphs for tabular data extension. Based on a visual exploration paradigm, a first remarkable approach is CAVA [Cashman et al 2020]. CAVA is an application that supports the users in finding the information useful to add to an input table using knowledge graphs as external data sources. It is based on a visual analysis system designed to assist users in enriching columnar data using knowledge graphs. The system uses queries to generate new attributes, which are added to a dataset as additional columns to be used later in an analysis or modelling phase.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

CAVA uses visualizations to help users identify potential information of interest by exploring the local neighbourhood of the graph on entities in the dataset to determine commonly occurring attributes. Users can express complex operations on the knowledge graph using the graphical interface without writing code. A few approaches have been proposed that exploited graphs for tabular data extension. Differently from CAVA, another approach proposes a backend solution (without user interface) that integrates mechanisms to extend features of an input dataset using knowledge graphs into the popular framework for data analysis scikit-learn [Bucher et al 2021]; the approach does not really support the selection of relevant attributes to use, but exploit a brute force method where all attributes are combined and show that this feature enrichment improves the performance in different downstream tasks.

Finally, a family of approaches try to combine algorithms that exploit knowledge graphs in two stages: in a first stage, tabular data are annotated with terms and entities of background knowledge sources; in a second stage, links to entities in background knowledge sources are used to fetch additional data from these knowledge sources. We refer to these approaches as *extend by linking* approaches, and include solutions such as OpenRefine[52], Magic [Femke Ongenae 2021], DAGOBAH UI [Sarthou-Camy & al 2022], DataGraft with ASIA extension [Cutrona et al 2019, Ciavotta & al 2022], and Sem-TUI [Ripamonti et al 2022]. Link functionalities are based on semantic modelling, table annotation and semantic table interpretation algorithms, and will be extensively covered in Section 4.2.3.4. Extension functionalities are similar to the functionalities proposed in tabular search and CAVA, but exploit knowledge graphs (open linked data) – usually WikiData and DBpedia (OpenRefine, Magic, DAGOBAH UI, ASIA) or other services (ASIA,SemTUI). However, in most of these tools (OpenRefine, DAGOBAH UI, ASIA) more sophisticated features are implemented for the linking functionalities, with data extension functionalities covered but by quite basic features. Also, we observe that SamTUI, an enRichMyData tool and further described in Section 4.2.2.3.4, is somehow the follow-up of the application that integrates DataGraft and ASIA, offering an improved set of functionalities and a better user experience, especially for data extension. A distinguishing feature or SemTUI is the generalization of the *extend by linking* paradigm, where a larger variety of APIs can be used for linking and extension, while these other similar applications link and extend with encyclopedic data sources.

A new wave of contributions to data discovery is also expected with the application of deep learning approaches to tabular data. For example, a family of applications based on ChatGPT have started to be released in the market (e.g., SheetGPT[53]) with more applications expected to come soon. The idea of these approaches is to support a variety of language-based data extension tasks (e.g., add a column with sentiment prediction for tweets contained in an existing column). For example, a recent solution proposes a few-shot approach based fine-tuned transformers, which after linking column values to Wikipedia, extract their abstract which are thereafter encoded into features after fine-tuning latent text representations to improve classification tasks [Harari & Katz 2022]. However, we are not aware of solutions that reach out external data sources so far, while most of solutions compute inference on top of the input data. However, it is worth observing that this kind of applications is likely just begun and more approaches based on the popular "pre-train and fine-tune" paradigm developed for natural language are expected, under the emerging term of "tabular data modelling". Three surveys on approaches to pre-train tabular data models and their downstream applications have been recently published [Dong et al 2022, Borisov et al. 2022, Singh & Bedathur 2023]. A key contribution in this field, TabNet, has shown promising results on different downstream tasks [Arik & Pfister 2021].

[52] https://openrefine.org/

[53] https://sheetgpt.ai/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Finally, for API-based knowledge discovery, we refer to approaches cited in the “Dataset and API cataloguing and search” paragraph, which also implement recommendation functionalities that can be considered knowledge discovery features.

4.2.2.3 Tools and market

In the market, functionalities described above are often mixed in different products. More basic dataset search and cataloguing functionalities alone have been proposed especially in a web-scale context, where more advanced solutions are not easy to implement. We therefore start by describing **web-scale dataset search and cataloguing and API search solutions.** Otherwise, some search and cataloguing features are often combined with more advanced solutions to gain **deep schema insights** as part of dedicated applications or data management platforms. Sometimes, these solutions also include some **knowledge discovery** features, which are presented at the end of the section.

4.2.2.3.1 Web-scale dataset and API search and cataloguing

Web-scale dataset search

- **Google Dataset Search**[54] is somehow the quintessential example of a web scale infrastructure to support dataset search. Its features have been discussed as examples in the introduction of this sections as well as in the review of existing approaches.
- **Mendely data**[55] is a solution developed specially to support the discovery of research-related materials, including datasets used in scientific research. It supports full-text search and faceted search to filter our results by publication date, data type (includes code, images, etc.), source type, source. It does not provide more advanced solutions to gain schema insights for the indexed data.

API search and cataloguing

- **Rapid**[56], it’s the world’s Largest API Hub and has a mission to support open discovery of existing APIs, even if solutions for intra-organizations hubs exist. The API catalogue can be searched by category, collection, or type by using a dropdown menu, a search bar, or API Hub collections (a grouping of APIs based on similar or complementary functionality or use cases). To support the evaluation of APIs of interest, it provides reviews with popularity scores, average latency, and average success rate directly in the search results. Finally, to support integration, it provides information about each API using the endpoints page to view a list of endpoints, documentation, and code snippets to help implement the code into an existing app.
- **IBM WebSphere Application Server Liberty**[57] is a product that host web applications, including APIs and provide some features to use catalogues to support the management of large API collections within an organization. Analogous solutions also exist (e.g., Oracle WebLogic Server), with similar simple cataloguing features.

4.2.2.3.2 Deep schema insights

Data profiling and deep schema insights features are offered in **commercial graph-based data management applications**, usually as modules or components of the applications, which require a proper license.

[54] https://datasetsearch.research.google.com/

[55] https://data.mendeley.com/

[56] https://rapidapi.com/

[57] https://www.ibm.com/docs/it/was-liberty

- **TopQuadrant TopBraid Enterprise Data Government**[58] is a KG based enterprise data governance solution used for semantic data integration and management platform. It is also used to ensure compliance with data governance policies and regulations. The platform provides a comprehensive suite of functionalities for metadata management, data lineage, data quality, and data classification. It includes a profiling feature for generating summaries of the content in a knowledge graph. The TopBraid EDG Catalogue will capture all technical metadata and perform data profiling, and if needed it can capture data samples that can provide governance teams insights into data and improve automated processes for linking technical and business metadata. Apart of KG, the platform supports a wide range of data sources, including relational databases, XML, RDF, and other structured and unstructured data formats.
- **PoolParty Semantic Suite**[59] is an enterprise software suite for managing and extracting value from large-scale knowledge graphs. It provides a comprehensive set of tools and features for creating, managing, and analysing semantic data, including ontology management, taxonomy management, entity extraction, text mining, and data integration. PoolParty Semantic Suite is designed to help organizations with large amounts of data to derive insights from their data, improve search and discovery, and automate business processes. It is widely used in various industries, including finance, healthcare, government, and manufacturing.
- **Stardog**[60] is a commercial semantic graph database that supports the storage, querying, and reasoning of large-scale semantic data. The tool includes a profiling module that allows users to profile data during ingestion or after the data has been added to the database. Profiling helps to understand the shape and quality of the data, including identifying patterns, anomalies, and potential issues.
- **Amazon Neptune**[61] is a commercial graph database storing data as quads, similarly as in the RDF data model, and support different query languages. While it has several solutions for data exploration[62], features to provide deep schema insights are not reported anywhere.
- **Neo4J**[63] has query features that allow to reconstruct the schema of a dataset. A blog post from the Neo4J team describes several queries that can be used to get schema insights and a holistic view of the data[64]. The schema is somehow described as a set of schema patterns that can be easily visualized as a graph in its graph visualization interface. Also, several insights about the schema and the data distributions can be collected using dedicated queries. For example, it is possible to: check constraints and indexes, relationship types, and node types; count nodes and relationships, possibly by type, and properties; analyze uniqueness of properties, statistics about property values (e.g., mix/max/avg), most frequent values, and topological properties of the graph (e.g., node centrality).

58 https://www.topquadrant.com/

59 https://www.poolparty.biz/metadata-management

60 https://www.stardog.com/

61 https://docs.aws.amazon.com/neptune/

62 https://docs.aws.amazon.com/neptune/latest/userguide/visualization-tools.html

63 https://neo4j.com/

64 https://neo4j.com/blog/data-profiling-holistic-view-neo4j/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Deep schema insights features are offered in several **commercial data management solutions.** The website dbmstools.com[65] provides a systematic comparison of several commercial data management tools organized by category, where a tool may appear in different categories. The categories most relevant to the analysed functionality (extracting deep schema insights) are: data catalog tools[66] (31 entries), most of which contain information about the source, data usage, relationships between entities as well as data lineage; data dictionary tools[67] (53 entries), which provide a set of important information about data used within an organization (metadata), including names, definitions, and attributes about data, owners, and creators of assets, in order to provide insights into meaning and purposes of data elements; data profiling tools[68] (38 entries), which allow analysing, monitoring, and reviewing data from existing databases in order to provide critical insights. The latter category is the most relevant one. For each category, tools are described and classified, especially in terms of type of supported data (e.g., flat files, spreadsheets, NoSQL, etc.), type of supported statistics (e.g., Avg, Max, Min, Stdev), and support for data tagging. In other words, deep schema insights features are provided especially in data catalog tools and data profiling tools, but also in more comprehensive data integration platforms. In the following, we review in detail a solution that also considers knowledge graphs, and briefly summarize the most prominent solutions using the above-linked sites as a source.

- **IBM Watson Knowledge Catalogue**[69] is a cloud-based data catalog tool that supports organizations in activities such as data discovery, access, and curation, all powered by active metadata. IBM Watson Knowledge Catalog uses a combination of automated and manual profiling techniques, as well as interactive data exploration capabilities, to provide a comprehensive view of data assets and help users better understand the content of their data. This tool makes use of machine learning and natural language processing (NLP) algorithms to automatically discover and classify metadata about data assets, such as data type, format, and relationships. Moreover, it includes functionalities to preview data, perform statistical analysis, and visualize data distributions. Such functionalities have the aim to help users better understand their data. The catalogue also has specific features for semantic data sources. These features and advanced NLP solutions to extract metadata are the reasons for which it is reviewed more in detail.
- Other tools that have specific features to ***catalog data and reconstruct relationships and statistics***: Collibra Catalog[70] empowers business users to quickly discover, understand, contribute, and govern the data that matters so they can generate impactful insights that drive business value. Dataedo[71], a metadata management & data catalog tool with data profiling features, which allows to use sample data to learn what data is stored in data assets; Informatica Data Catalog[72] is a machine learning-based data catalog that lets you classify and organize data assets across any environment to maximize data value and reuse, and provides a metadata system of record for the enterprise (it also automatically scans and catalogs data across the enterprise, indexing it for

[65] https://dbmstools.com/
[66] https://dbmstools.com/categories/data-catalogs
[67] https://dbmstools.com/categories/data-dictionary-tools
[68] https://dbmstools.com/categories/data-profiling-tools
[69] https://www.ibm.com/cloud/watson-knowledge-catalog
[70] https://www.collibra.com/us/en/products/data-catalog
[71] https://dataedo.com/product/data-profiling
[72] https://www.informatica.com/products/data-catalog/enterprise-data-catalog.html

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

enterprise-wide discovery using simple, Google-like search); Global IDs Data Profiling Suite[73], which automates the discovery of data assets, supports data profiling, and provides an active inventory of all data assets; Io-Tahoe[74] (now acquired by Hitachi Vantara[75]) is a data catalog product, where the population of the data catalog is automated by using artificial intelligence and leveraging the discovery functionality and natural language analysis to automatically tag data; Truedat[76] is an open source data cataloging and governance tool that allows to quickly unify and explore combined metadata from different sources on the same interface; OvalEdge[77] is a data catalog tool that automatically organizes and catalogs your data using machine learning and advance algorithms, in such a way that data are organized using tags, usage statistics, user names, and other markers that make them retrievable with everyday language; Talend Data Catalog[78] automatically crawls, profiles, organizes, links, and enriches metadata, making it easy to search and access data, and automatically keeping up to date 80% of the information associated with the data; Octopai[79] automates the process of metadata discovery and data lineage, and quickly assemble the metadata needed to create an effective data catalog; other remarkable tools based especially on metadata management include erwin Data Catalog[80], Qlik Data Catalyst[81], Azure Data Catalog[82], Google Cloud's Dataplex[83], and Oracle Cloud Infrastructure Data Catalog[84].

- Other tools that support ***data profiling especially as means to improve data quality or support data cleaning***, but also providing some sort of schema insights: Trifacta[85] is an interactive cloud platform to profile, prepare, and pipeline data for analytics and machine learning that supports interesting data profiling features, such as automatic identification of dataset formats, schemas, attributes, relationships and metadata across datasets; Atlan[86], an open ecosystem that also provide data cataloging; Datamartist[87], an easy-to-use data profiling tool that also analyzes data formats and types; Datameer[88], a solution for data transformation in Snowflake, which also provide interesting features to generate schema insights such as system-generated recommendations, and system- and user-generated data profile information that include

---

73 https://www.globalids.com/data-profiling-discovery/

74 https://io-tahoe.com/

75 https://www.hitachivantara.com/en-us/products/dataops-software/data-catalog.html

76 https://www.truedat.io/

77 https://www.ovaledge.com/data-catalog

78 https://www.talend.com/products/data-catalog/

79 https://www.octopai.com/

80 https://www.erwin.com/products/erwin-data-catalog/

81 https://www.qlik.com/us/products/catalog-and-lineage

82 https://azure.microsoft.com/en-us/products/data-catalog/

83 https://cloud.google.com/dataplex

84 https://www.oracle.com/big-data/data-catalog/

85 https://www.trifacta.com/

86 https://atlan.com/

87 http://www.datamartist.com/

88 https://www.datameer.com/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

documentation, properties, comments, tags, and more; Kylo[89], an open source enterprise-ready data lake management software that supports search and exploration of data and metadata, lineage, and profile statistics; TIBCO Clarity[90] is data preparation, profiling, and cleansing tool that detects data patterns and data types for auto-metadata generation; Open Source Data Quality and Profiling[91], one of the few open source solutions available in the market; IBM InfoSphere Information Analyzer[92], Experian Pandora[93], Ataccama One[94].

- ***Data profiling and deep schema insights*** features are also part of commercial data management solutions addressing especially *data transformation and integration*. In fact, data profiling modules are part of most of relational database management systems and data management applications supporting ETL and other procedures to improve data quality; for example, data profiling is available in Oracle's Warehouse Builder[95] and Talend Data Fabric[96], including its solutions for data documentation[97]. In all these solutions profiling returns statistics about data distributions, estimates about duplicates and other patterns related to data quality.

Data profiling tools for RDF data on the web

- **Linked Open Vocabularies (LOV):** is a tool to search for RDF vocabulary terms, covering more than 800 vocabularies at writing time. For each vocabulary, it provides: a description and other metadata (language, creator, comment); the links the vocabulary includes to other vocabularies; statistics, tags, and expressivity class (e.g., RDFS). It supports full-text search over existing vocabularies via API and a web interface. The results for a search, e.g., "City", reports all the occurrence of vocabulary terms that match the input search (e.g., 1323 for "City") and a faceted search interface where results can be filtered by type (vocabulary, property or class, agent); for each result, the interface reports the URI, labels and other relevant properties (e.g., skos:definition), and – if available – the linked open datasets in which it is known to be used.

JSON schema generators

- In JSON, the schema can be generated and different tools providing this functionality exist. A non-exhaustive list of such tools includes[98]: online tools[99], Python libraries[100], NodeJS libraries[101] and

---

[89] https://kylo.io/

[90] https://docs.tibco.com/products/tibco-clarity-3-0-0

[91] https://sourceforge.net/projects/dataquality/

[92] https://www.ibm.com/products/infosphere-information-analyzer

[93] https://www.edq.com/data-quality-platform/data-profiling/

[94] https://www.ataccama.com/platform/data-profiling

[95] https://docs.oracle.com/cd/E18283_01/owb.112/e10935/data_profiling.htm

[96] https://www.talend.com/products/data-fabric/

[97] https://help.talend.com/r/en-US/8.0/data-catalog-user-guide/working-with-data-documentation

[98] List is extracted from the Stackoverflow post https://stackoverflow.com/questions/7341537/tool-to-generate-json-schema-from-json-data

[99]E.g., *https://www.liquid-technologies.com/online-json-to-schema-converter*, *http://www.jsonschema.net*, *https://easy-json-schema.github.io*)

[100] E.g., *https://github.com/gonvaled/jskemator*, *https://github.com/perenecabuto/json_schema_generator*

[101] E.g., *https://github.com/easy-json-schema/easy-json-schema*, *https://github.com/aspecto-io/genson-js*

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Ruby libraries[102]. However, to the best of our knowledge, no tool exists that organizes the schemas generated and supports navigation or browsing of large schema collections.

4.2.2.3.3 Knowledge discovery

Most of knowledge discovery solutions are research prototypes introduced to support the semantic annotation of semi-structured data sources; therefore, we refer to Section 4.2.4.3 for more detailed descriptions, which covers tools relevant for the LinkR collection. In this section, we only discuss the data extension functionalities of these tools, under the assumption that they can be applied to fetch data from the reference knowledge graph to which entities in table columns are linked.

- **OpenRefine:** extends with reconciliation services that match the data in the table columns to entities in external knowledge graphs.
- ***CAVA:*** *extends with* DBpedia and Wikidata
- **DAGOBAH UI:** extends with a query builder that allows users to query external knowledge graphs and integrate the results with the data in the table.
- **MAGIC:** extends with a search interface that allows users to query external knowledge graphs and retrieve additional data to enrich the original dataset.
- **Grafterizer** and **ASIA:** extends with arbitrary extension services available as APIs, which also include queries to knowledge graphs stored locally or remotely.

However, features for data extension from external sources, based on substantial human configuration, are now becoming part of several commercial applications.

- ***Python* programming environments.** The *Python* programming environment has numerous libraries specifically developed for data manipulation and extension. The most common and widely used libraries include *Pandas* [26] and *NumPy* [22]. However, these tools have a significant limitation in that they can only be used by users with advanced knowledge of data science and code writing and are not designed to support the visualization of results and user interaction. There are libraries such as Agate [1] that make code more accessible to users and improve the visualization of results, but there are also libraries such as *PandasGUI* [27], D-*Tale* [11] and *MITO* [20] that allow data to be manipulated and analysed through a graphical interface, even if with limited and incomplete functionalities. About the manipulation and querying of semantic data, Python has libraries specifically developed for this purpose, including *RDFFrames* [29]. This library allows data to be extracted from knowledge graphs by querying them efficiently, enabling users to query the KG easily.
- ***Microsoft Excel.*** Among interactive tools, *Microsoft Excel* [19] is one of the most widely used for data manipulation due to its simple graphic interface, which is suitable for less experienced users. This type of tool offers basic functionality for data manipulation and analysis. In particular, Excel allows the import of tabular data from various sources, such as files, databases, or online services, as well as offers the possibility of carrying out queries or filters on tabular data. Furthermore, it is possible to perform operations such as deleting duplicates or completing null values, as well as merging between tables to integrate data.

[102] E.g., https://github.com/maxlinc/json-schema-generator

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **SAS Data Quality**[103], **Oracle Enterprise Data Quality**[104], and similar systems. SAS Data Quality includes some sort of data extension features by supporting the extension of some new fields from table fields, such as the expansion of the gender field from the record name, to eventually support entity resolution. Oracle Enterprise Data Quality also support attributes extension to some degree[105].
- **SheetGPT**[106], and upcoming tools based on **ChatGPT**. Several tools are emerging that use text to code features of ChatGPT to automate manipulation of spreadsheet and, therefore, tabular data in general. The text to code features of ChatGPT will have impact on solutions to automate data enrichment; however, at the moment, it seems that most of the enrichment features that are supported are the ones that rely on manipulation of the data that consists of operations on the data already contained in the table, without accessing external data sources. Most of these features are expected to cover calculations (e.g., counts, aggregations, etc.), language tasks on textual fields (e.g., classification, sentiment analysis, extraction of named entities, etc.), and, likely, data cleansing (e.g., by generating scripts to rewrite textual values). These features do not provide knowledge discovery functionalities though, at least not the ones based on external information sources. However, it is worth noting that a broader variety of features could be available at limited costs by accessing models such as ChatGPT in the future, despite the well-known problems of this technology in terms of reliability, and these features may include access to external data sources or can be adequately combined with ad hoc solutions developed to access external data sources thus delivering end to end text to enrichment features.

4.2.2.3.4 Tools from the consortium

Two tools from the DiscoverR collection are provided by UNIMIB: **ABSTAT**, a tool for profiling of large RDF datasets and knowledge graphs represented with this data model, and **SemTUI**, a tool that includes both linking functionalities (see the LinkR section) and knowledge discovery functionalities.

4.2.2.3.4.1 ABSTAT

Since knowledge graphs (KGs) play a crucial role in data enrichment pipelines [Ciavotta et al 2022], either as target data sources of interest [Massri et al 2023] or as bridges to reach additional sources [Cutrona et al 2019], the first DiscoverR tool, ABSTAT [Alva Principe et al 2021], has the aim to support users along the enrichment process. ABSTAT supports users by providing a semantic profile that describes the content of the KG. The semantic profile is automatically extracted and it consists of ontology-based patterns and statistics that describe how schema elements are used in the KG. Such a profile is extracted also for very big KG [Alva Principe et al 2021]. We use ABSTAT as example of tool providing deep schema insights functionality. To help users explore the information represented in the profiles, ABSTAT provides two graphic user interfaces (GUIs). The Browse GUI lists all the schema-level connections existing in a graph as well as several statistics, thus providing a complete schema-level summary of the data stored in the KG. The interface supports humans in searching relevant data (“Which connections do the graph represent between cities and sports teams?”), and filter and browse all available connections (e.g., finding all the properties used to describe entities of the class dbo:City, or finding that the unique property connecting dbo:City and dbo:SportTeam in DBpedia is dbo:wikiPageWikiLink and that about 9000 of these

[103] https://www.sas.com/en_us/software/data-quality.html

[104] https://www.oracle.com/assets/oracle-enterprise-data-quality-ds-430148.pdf

[105] https://docs.oracle.com/en-us/iaas/analytics-for-applications/doc/augmenting-your-data.html

[106] https://sheetgpt.ai/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

connections exist). Subsequently, the user can decide to filter the data using the three available text boxes for adding constraints on the subject type, predicate, object type, and any combination of these. While the input is typed inside a text box the autocompletion feature will recommend types/predicates that occur in the patterns. While the Search GUI provides full-text search functionalities, where the user can insert any keyword and get results that match the input (patterns, types, and predicates) for all the profiles or some specific profile. Statistics, dataset names and patterns will be shown in the results of the query. However, ABSTAT supports API-based access to control the profiling process (e.g., launch a summarization, store profiles, etc.) and to make third-party applications access the profiles. APIs that provide access to the profiles support all the functionalities available with the Browse and Search GUIs. ABSTAT has been proven helpful to support users in understanding the content and quality of a KG [Spahiu et al 2023] - also by the generation of SHACL profiles for data validation [Spahiu et al 2018], but also machines, e.g., vocabulary suggestion for semantic annotation and Semantic Table Interpretation (STI) [Cutrona et al 2019], and even analytical downstream tasks such as feature selection in semantic recommender systems [Di Noia et al 2018]. ABSTAT comes as a mature tool providing APIs and GUI to let humans and machines access the summaries and the statistics (search, query, browsing), and a scalable distributed architecture to compute summaries of very large KGs [Alva Principe et al 2021]. In enRichMyData, profiles of well-established KGs like WikiData and DBpedia will be considered, but also of KGs produced and used in the project.

4.2.2.3.4.2 SemTUI

It is a framework that provides a User Interface (UI) to let users enrich tables by combining data linking and data extension services. In SemTUI, the discovery process happens while the user is enriching a data sample with the user interface: the user can discover linking algorithms that are available and use them to bridge to existing data sources (e.g., linking cities described in a column to their id in the DBpedia KG); once the links are found using the selected linking service, additional data can be fetched from the reference data source: the user can explore data available in the data source and specify the data she/he wants to add to the table (e.g., fetching the population of each city from DBpedia). Although this “link and extend” mechanism is inspired by the principles of web-based data exploration of linked open data61, SemTUI is not limited to exploiting linked data sources. For example, we tested linking and extension services from private company KGs and included the HERE62 geocoding service for linking addresses to coordinates; once two columns have geocoordinates., data can be enriched with the shortest route distance (as calculated by a route planning service). In other words, explorative functionalities offered by SemTUI support the discovery of (1) linking and extension services, (2) data fetched from external sources, and (3) possible flaws in the data enrichment process (e.g., wrong links). SemTUI implements, extends, and improves functionalities previously available in ASIA [Cutrona et al 2019] providing a better user experience, and a feature that translates enrichment operations into python code that can be manipulated also from developer-friendly interfaces like a notebook. ASIA was an application with a frontend (UI) integrated into the Grafterizer UI [Ciavotta et al 2022] and a backend that served scalable linking; SemTUI comes with its new interface and uses linking services served by ASIA’s backend, which still part of the SemTUI ecosystem.

4.2.2.4 Research & technical challenges

Dedicated tools exist to support web-scale and organization-scale *dataset search*. Innovating dataset search in an open scenario is possible but would require substantial web crawling capabilities (as in Google Dataset Search); on the other hand, for organization-scale dataset search several dedicated tools exist. Automatic data annotations are increasingly used to improve the dataset descriptions. In EMD tools dedicated to provide annotations of semi-structured data are part of the LinkR family tools. LinkR tools that return semantic annotations of semi-structured data could be coupled with existing tools for dataset search by providing these tools with additional and fine-grained metadata (e.g., column level or cell-level).

Several tools exist in the market to profile datasets in such a way to get *deep schema insights* for the more commonly used data models, e.g., relational data and semi-structured data. As it comes to graph data, vendors of graph database offer some features to provide deep schema insights that overlap with research prototypes such as ABSTAT. Some research challenges to improve profiling of knowledge graphs exist, such as better user interface (e.g., grouping schema patterns), application to wrapped sources, and extension to property-graph data models.

What we consider the most relevant challenges for data discovery are related to *tabular data search*, the task that has received most of the attention recently because deemed particularly helpful to support data science workflows, especially data wrangling and enrichment. Here several challenges can be defined that match also novel opportunities offered by recent advances in AI (e.g., generative language models as interfaces for tabular data manipulation).

Most relevant challenges for data discovery innovation

We can summarize the most relevant research and technical challenges that are also less covered in the current market as follows:

1. **Deep schema insights and dataset search** provides semantic description of semi-structured sources by annotating the sources with tags from well-known reference vocabularies (e.g., Schema.org, WikiData, etc). Priority: Low
2. **Tabular data search** provides a GUI to explore information in a table of interest and additional information sources that can be used to augment the information in the table. For these sources, provide information to understand their content and support their usage. Priority: High
3. **Tabular data search** provides language-based interfaces for exploring data augmentation functionalities for a table of interest. Support conversion of instructions given in natural language into tabular data manipulation commands. Priority: Medium

4.2.3 Data cleaning (CleanR)

4.2.3.1 Scope and example

One essential and challenging task in data science is data cleaning – the process of identifying and eliminating data anomalies and inconsistencies. Different data types, data domains, data acquisition methods, and final purposes of data cleaning have resulted in different approaches to transforming data anomalies in the literature [Sukhobok et al., 2017].

The word “data cleaning” is often used in place of other methods, like data preparation, data wrangling, and data munging, even if some differences exist between these terms. It looks useful to define these differences from the beginning, to have a clear definition of the problem; the main terms used are the following ones:

- **Data preparation** usually refers to the process from a narrow perspective, where data cleaning is just a step in the major task of cleaning and transforming raw data into data useful for analysis.
- **Data wrangling** focuses on transforming a data format, for example from a raw format to a more accessible and effective one for the analysis.
- **Data munging** is just a synonym that refers to the same concept; in the following, it will be considered one of the possible operations available in the data cleaning phase.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

A typical data cleaning workflow, defined as above, covers numerous data quality issues; basic data sanity checks are useful as initial validation to ensure accuracy, consistency, and suitability for analysis or further processing of a dataset. In particular, the main ones are:

- **Case Normalization:** converting the text in a dataset to a consistent case format. For example, converting all text to lowercase or uppercase to eliminate variations caused by different capitalization styles.
- **Trimming:** removing leading and trailing spaces from strings to avoid extra spaces at the beginning or end of a value can cause issues during analysis and comparisons.
- **Case normalization:** converting the text in a dataset to a consistent case format. It aims to eliminate variations caused by different capitalization styles and make the data uniform for easier analysis and comparison.
- **Range and Constraint Validation:** verifying that data falls within expected ranges or adheres to defined constraints.

After the application of these data sanity checks as a preliminary step, more complex data manipulations are available; in particular, the main ones are:

- **Outlier detection:** identification of records that do not respect what is considered normal behaviour.
- **Data deduplication:** removal of records that refer to the same entity.
- **Data integrity:** application of constraints to verify the domain and entity integrity.
- **Handling missing data:** identification of missing data and dealing with them.
- **Data standardization:** conversion to a data format that facilitates the analysis, like the standardization of the unit of measurement.
- **Data redundancy:** removal of redundant or irrelevant columns that do not provide relevant information or contain the same information as other variables.

To identify these kinds of problems, different approaches are available. It is mainly possible to use:

- **Statistics-based methods**, like the use of pivot tables to analyse the distribution of particular columns and detect anomalous values.
- **Distance-based methods**, for example, the use of distance measures between data points to identify outliers.
- **Model-based methods**, meaning the application of a pre-trained model on a given ground truth that can generalize on new data.

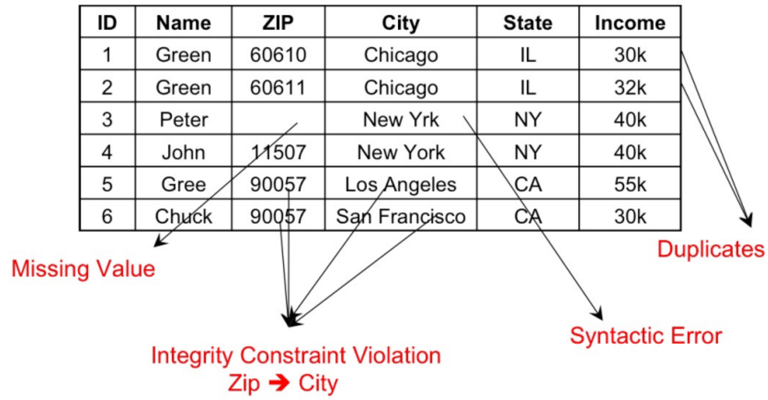


Figure 9: Example data errors [Ilyas, I. F.,2016]

After the identification of data quality issues, a data repair step is needed. Data transformation is a process of changing the format or structure of the data to solve data quality anomalies. According to [Sajid et al., 2019], it is possible to determine three main groups of data transformations:

- **Table-based:** transformations of a dataset on a structural level. These changes do not impact the record individually but regard rows and columns. Possible examples of this family are: feature selection, renaming of columns and selection of relevant row.
- **Format-based:** transformations on the individual record level, without having an impact on the general structure. Examples of these transformations are the application of lowercase, normalization of a column and rounding of values.
- **String-based:** transformations of strings or text at both cell- and column-level. This family includes examples like removing special characters, filling null values and extracting values matching those in a new column.

4.2.3.2 Research

Given the analysed data quality problems, different types of data cleaning techniques exist to automate the cleaning process according to the final user's preferences. The main techniques have been included in the list and in the table below after being selected via Google Scholar, looking at the most recent and relevant publications. These methods can be divided into the following categories:

- **Low code / visual data cleaning:** use of visual tools to identify and visualize potential data quality issues to correct them, with limited coding skills.
- **Programmatic data cleaning:** use of hard-coded tools to test hypotheses about the quality of a dataset and apply potential transformations.
- **ML-guided data cleaning:** treating data cleaning as a large-scale ML problem. They are primarily applied to a training dataset before model development and training, although some techniques may also be used to clean test data during model inference.
- **Data cleaning for ML pipeline:** the cleaning process is no longer considered as an initial step of the ETL pipeline but becomes dependent on the final application, requiring an end-to-end application-driven approach.

4.2.3.2.1 Low code / visual data cleaning

Various studies and platforms are making significant contributions to the dynamic field of data cleaning by introducing innovative approaches that enhance data quality and simplify the process using visual

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

tools, thereby reducing the need for extensive coding skills. Low code/visual data cleaning refers to the practice of utilizing visual tools and interfaces to identify, visualize, and rectify potential data quality issues without extensive coding skills. This approach allows users, including non-technical or less experienced individuals, to interact with the data directly and visually explore and manipulate it for cleaning purposes. The visual tools provide an intuitive and user-friendly interface that enables the detection of anomalies, outliers, missing values, inconsistencies, and other data quality problems. By visually representing the data and its quality issues, users can easily identify and make corrections or apply predefined cleaning operations. This approach reduces the reliance on traditional coding and scripting methods, making data cleaning accessible to a wider range of users and accelerating the cleaning process.

One such contribution is PrAVA, a pioneering visual analytics process introduced by [Milani et al., 2021], PrAVA incorporates preprocessing profiling as a critical phase within the visual analytics workflow. By combining automatic and visual analysis methods with human interaction, Visual Analytics aims to store and acquire new domain knowledge. The inclusion of the pre-processing profiling step significantly improves the system's understanding of the dataset through the utilization of statistics and metadata. This integration introduces a third feedback loop that actively involves the end user, augmenting the existing loops dedicated to visualization and model refinement.

[Carminé, R. L. A.,2021] presents TruData, a collaborative platform specifically designed to streamline data cleaning operations. TruData facilitates the efficient and effective reuse of data cleaning operators across different platforms, domains, and data types. By establishing a comprehensive repository of potential data quality operators and tailoring suggestions based on the user's specific data domain and type, TruData enhances the overall efficiency, effectiveness, and user experience associated with data cleaning tasks.

[Musleh et al., 2020] propose Coclean, an innovative crowd-in-the-loop system that empowers both lay users and power users to actively contribute to the data cleaning process. Through the adoption of a Collaborative dataset (CDF) approach, users can seamlessly share data presented as a dataframe, and data quality operations can be performed either manually or automatically. The system also provides a dedicated Python repository, enabling enhanced collaboration and traceability of data transformations.

In a case study conducted by [Petrova-Antonova et al., 2020], OpenRefine and Trifacta Wrangler were utilized to address data quality problems and evaluate cleaning techniques. Despite employing similar data cleaning methods on the same dataset, the study uncovered variations in the performance of the two tools. OpenRefine demonstrated superior performance overall, although it faced challenges when dealing with large datasets. In contrast, Trifacta excelled in handling large datasets but exhibited limitations in modifying individual records and effectively communicating the effects of operations to the end user.

These studies and platforms collectively contribute to the advancement of data cleaning methodologies and provide valuable insights into enhancing data quality. By leveraging visual tools, they offer accessible and user-friendly approaches that simplify the data cleaning process while promoting collaboration among users.

4.2.3.2.2 Programmatic data cleaning

The following solutions are tools and packages that aim to contribute to the advancement of data engineering and data cleaning practices by providing developers and data scientists with efficient and effective solutions. By leveraging generalized dataflow graphs, statistical data validation, and a cleaner API, these innovations empower users to achieve higher data quality standards and streamline their data processing workflows. Programmatic data cleaning refers to the practice of employing code-level tools and techniques to test hypotheses regarding the quality of a dataset and apply necessary transformations

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

to improve its quality. In this approach, data cleaning operations are implemented using programming languages and frameworks, allowing for precise control and customization of the cleaning process. Programmatic data cleaning involves writing code scripts or functions to perform various data cleaning tasks, such as identifying and handling missing values, removing outliers, standardizing formats, correcting errors, and validating data against predefined rules or constraints.

In this category Hamilton is one of latest available solutions. Developed by [Krawczyk et al., 2022], is an open-source Python package that facilitates the creation of dataflow frameworks for performing data transformation operations. By constructing a Directed Acyclic Graph (DAG) that captures the dependencies between data transformations required by the user, Hamilton enables the application of unit tests and evaluation of the outcome of the transformation process. This innovative approach allows for tracing data lineage dependencies and orchestrating the operations in order to achieve the desired level of data quality.

[Bantilan N., 2020] introduces pandera, a lightweight API designed for statistical data validation of Pandas dataframes, which are in-memory tables commonly used in data processing. The primary focus of pandera is to provide a convenient mechanism for defining data quality expectations in code, enabling the creation of robust data processing pipelines. The pandera Python package allows users to express both technical validation rules and domain-specific rules, providing a comprehensive tool for data validation during runtime.

Pyjanitor, developed by [Ma et al., 2019], offers a clean and intuitive API for data cleaning tasks. Inspired by the dplyr world, where functions are named using verbal expressions to enhance readability, Pyjanitor implements the functionalities of the R package janitor. While not aiming to be a universal data cleaning tool for all subject domains, Pyjanitor focuses on providing easy access to the most used data transformations. The library simplifies the process of cleaning data by offering a streamlined and user-friendly interface.

4.2.3.2.3 ML-guided data cleaning

ML-guided data cleaning refers to an approach that treats data cleaning as a large-scale machine learning (ML) problem. It involves applying ML techniques and algorithms to automatically identify and rectify data quality issues within a dataset. This approach is commonly employed on training datasets before model development and training, but it can also be used to clean test data during model inference. It provides the advantage of leveraging advanced ML algorithms and techniques to handle large and complex datasets efficiently. It allows for the identification and resolution of data quality issues that may not be easily detectable through traditional rule-based or manual approaches. This approach helps improve the quality and reliability of the data used for ML model development, leading to more accurate and robust models.

Different areas of research exist in the field, like Picket, introduced by [Liu et al., 2022], is an innovative system designed to protect against data corruption in tabular data during machine learning pipelines. During the training stage, Picket effectively identifies and removes corrupted data points from the training dataset to prevent bias in model training. This system is built upon PicketNet, a self-supervised deep learning model based on the Transformer network architecture, specifically designed for tabular data.

Baran, developed by [Mahdavi et al., 2020], employs an active learning approach to data cleaning, aiming to reduce the reliance on human experts. The system focuses on identifying a limited number of erroneous records and leveraging an ensemble of methods to determine the correct transformations. Baran effectively manages various types of errors, including value errors, vicinity of data errors, and domain-specific errors, streamlining the data cleaning process and minimizing human intervention.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Holoclean, presented by [Rekatsinas et al., 2017], takes a holistic approach to data repairs by utilizing probabilistic inference techniques. The system combines logical rules, business rules, and external catalogs, treating data cleaning as a statistical inference problem. By leveraging probability theory, Holoclean effectively combines different signals to reason about inconsistencies across datasets, enabling comprehensive and robust data cleaning.

These solutions contribute to the advancement of data cleaning practices by offering specialized techniques and approaches tailored to specific challenges. Picket safeguards against data corruption during learning and inference, Baran employs active learning for effective error correction, and Holoclean provides a holistic framework for probabilistic data repairs. By leveraging these innovative methods, data engineers and scientists can enhance data quality and mitigate biases using the machine learning theory in the context of data cleaning.

4.2.3.2.4 Data cleaning for ML pipeline

Data cleaning for an ML pipeline refers to the process of preparing and improving the quality of data specifically for machine learning tasks. Traditionally, data cleaning was considered as an initial step in the Extract, Transform, Load (ETL) pipeline, where data was cleaned and transformed before being fed into the ML model. However, with the advancement of ML techniques and the recognition of the impact of data quality on model performance, this area of research of data cleaning started to consider it as a crucial and application-dependent step in the ML pipeline. In the modern approach to data cleaning for ML, the cleaning process is no longer viewed as a standalone task but is intricately tied to the final application of the ML model. It requires an end-to-end application-driven approach, where data cleaning techniques and strategies are tailored to the specific requirements and constraints of the ML application.

An overview of the latest data cleaning techniques tailored for ML has been presented in the paper by [Whang et al., 2023]. They stress that a single cleaning algorithm cannot achieve optimal performance in all scenarios, advocating for adaptive algorithm selection based on the specific noise type. Notably, data cleaning primitives often entail significant parameters, such as thresholds, akin to hyperparameter tuning in ML. Hence, caution must be exercised when utilizing data cleaning techniques not originally designed for ML to ensure compatibility and effectiveness within the ML pipeline.

[Krishnan et al., 2019] introduce Alphaclean, a novel approach to data cleaning that incorporates multiple cleaning frameworks to create a progressive search procedure. By employing a quality function instead of relying on human-in-the-loop steps, Alphaclean selects the most suitable correction from multiple options. This quality function can also be integrated with application-specific objectives, extending the data cleaning process to encompass the entire data pipeline.

Activeclean, developed by [Krishnan et al., 2016], leverages gradients to expand outlier filtering and identify potentially erroneous data points. Through iterative cleaning of relevant observations, Activeclean effectively reduces the overall data cleaning cost. While human involvement is required to define an identity function, an update function, and correct small data batches, Activeclean automates the majority of the process.

In their work, [Ilyas, I. F., 2016] proposes an adaptive data cleaning workflow named evaluate-clean, designed to address continuous environments where manual evaluation by domain experts is impractical due to the scale of big data. The evaluate-clean workflow encompasses applying data cleaning procedures, evaluating their effects and errors, formulating data lineage hypotheses, and continuously updating repair actions. This iterative and continuous approach ensures efficient data cleaning in large-scale and dynamic data environments.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

These innovative approaches provide valuable insights and techniques for tackling data quality challenges in machine learning. They underscore the importance of adaptability, ensembling, automation, and continuous evaluation as fundamental considerations in designing effective data cleaning pipelines for ML applications.

Table 2: Overview of the most relevant data cleaning techniques in the literature

Method	Visual interface	Error detection			Error repairing			Human cost
		Human guided	Dependencies guided	Model based	Human guided	Dependencies guided	Model based	
PrAVA	Yes	Yes	No	No	Yes	No	No	High
TruData	Yes	Yes	No	No	Yes	No	No	High
CoClean	Yes	Yes	No	No	Yes	No	No	High
Hamilton	No	Yes	No	No	Yes	No	No	Medium
pandera	No	Yes	No	No	Yes	No	No	High
pyjanitor	No	Yes	No	No	Yes	No	No	High
Picket	No	Yes	No	Yes	-	-	-	Low
Baran	Yes	Yes	No	No	No	Yes	Yes	Medium
Holoclean	No	No	Yes	Yes	No	Yes	Yes	Low
Alphaclean	No	No	Yes	Yes	No	Yes	Yes	Low
Activeclean	No	Yes	No	Yes	No	Yes	Yes	Medium

4.2.3.3 Tools and market

The table below contains an analysis of the main existing tools on the market; the market is very crowded with data cleaning alternatives, so the following overview will focus on the state-of-the-art landscape without focusing on the details of each solution. These have been selected via Google Search and by looking at the libraries offered by Python and R.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

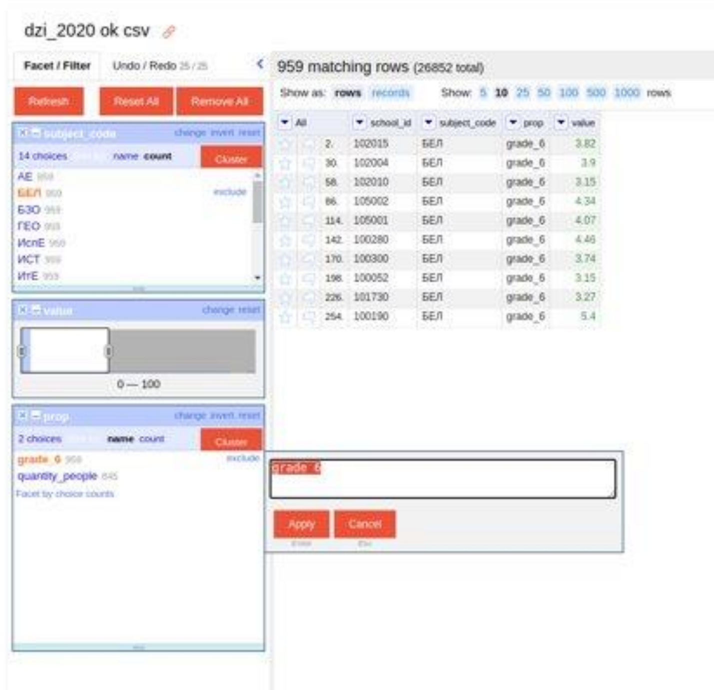


Figure 10: Example of interactive browsing by facets and batch editing of values in Ontotext Refine

The comparison has been done according to the following classifiers:

- **Error type:** data cleaning tasks that each tool can perform. Possible values are Data transformation, Integrity constraints, Data deduplication and Outlier detection.
- **Type of model:** possible types of a model are Script-based, Procedural, Event-based or a combination of them.
- **Type of language:** classification according to the computer language, either General-Purpose Language (GPL) or Domain-Specific Language (DSL).
- **Open source:** refers to software in which the source code can be accessed and modified by anybody, whereas the term "proprietary software" refers to software that is owned exclusively by the person or publisher who produced it.
- **Ease of use:** level of expertise a user needs to have for using the tool. There are three possible values for this classifier: Hard, Medium, and Easy.

Table 3: Overview of the most relevant data cleaning tools available in the market

Tool	Description	Error type	Type of model	Type of language	Open source	Ease of use
Python pyjanitor https://pyjanitor-devs.github.io/pyjanitor/	Python implementation of the R package janitor, using a set of convenient data cleaning routines	Data transformation	Script-based	GPL	Yes	Medium/Hard
Python Pandera	Python library for expressive data validation operation	Data transformation, Integrity constraints	Script-based	GPL	Yes	Medium/Hard

https://pandera.readthedocs.io/en/stable/						
Python pandas_dq https://github.com/AutoViML/pandas_dq	Python library for data quality analysis and improvement. It is based on scikit_learn functions and requires just one line of code for most operations	Data transformation, Integrity constraints, Data deduplication, Outlier detection	Script-based	GPL	Yes	Medium/Hard
R tidyverse https://www.tidyverse.org	Collection of R packages designed for data science	Data transformation Integrity constraints	Script-based	GPL	Yes	Medium/Hard
R *dplyr* *https://dplyr.tidyverse.org/*	R package that offers a grammar of data manipulation tools	Data transformation Integrity constraints Data deduplication	Script-based	GPL	Yes	Medium/Hard
OntotextRefine https://www.ontotext.com/products/ontotext-refine/	Application for automating the conversion of messy data into a knowledge graph.	Data transformation	Procedural	DSL	No	Medium
OpenRefine https://openrefine.org/	Tool for working with messy data: cleaning it; transforming it from one format into another, and extending it	Data transformation, Integrity constraints	Procedural	DSL	Yes	Medium
Grafterizer https://www.eubusinessgraph.eu/grafterizer-2-0/	Framework with interactive data cleaning and transformation functionality, and visual data profiling.	Data transformation Outlier detection	Event-based, Procedural	DSL	Yes	Medium
Tableau https://www.tableau.com/	Data visualization tool that allows non-technical users to create customized dashboards	Data transformation, Data deduplication, Outlier detection	Event-based	DSL	No	Easy
KNIME https://www.knime.com/	Data science platform integrated with most common programming languages and with intuitive GUI	Data transformation, Integrity constraints, Data deduplication, Outlier detection	Script-based, Event-based, Procedural	DSL	No	Easy
Trifacta https://www.trifacta.com/	Tool that provides data wrangling software for data exploration and self-service data preparation	Data transformation, Integrity constraints, Outlier detection	Event-based	DSL	No	Easy

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Command line *grep* https://www.gnu.org/software/grep/	Command line command for data quality and transformation	Data transformation	Script-based	GPL	Yes	Medium
Command line *awk* https://www.gnu.org/software/gawk/	Scripting language used for manipulating data and generating reports	Data transformation, Integrity constraints	Script-based	GPL	Yes	Medium
Excel *Transform Data by Example* https://www.microsoft.com/en-us/research/project/transform-data-by-example/	Excel plugin that suggests the user a possible transformation function given an example	Data transformation	Event-based	DSL	No	Easy/ Medium
Excel *QuickCode* https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/popl11-synthesis.pdf	Excel plugin that helps the user finding the desired transformation function	Data transformation	Event-based	DSL	No	Easy/ Medium

4.2.3.3.1 Tools from the consortium

Two tools from the CleanR collection are provided by the consortium: **Ontotext Refine**, by Ontotext, and **DataGraft**, by SINTEF.

4.2.3.3.1.1 Ontotext Refine

Ontotext Refine is a tool for automating the conversion of messy string data into a knowledge graph. The application is based on the popular OpenRefine tool and includes a visual RDF mapping interface, command line tool to automate all steps and integration with secured GraphDB endpoints.

As described on the Ontotext website, the tool supports analysing the input data that can be imported in different formats, including:

- Tabular formats (TSV, CSV, *SV)
- Fixed-width text files
- Excel (XLS, XLSX)
- JSON, JSON-LD, XML
- RDF: XML, Turtle/N3
- Databases (PostgreSQL, MySQL, MariaDB, SQLite)

After importing the data, different data cleaning and data transformation operations are available, using powerful row and column manipulations, faceting, and clustering. These kinds of transformations can be implemented using:

- Expressions and GREL (Google Refine Expression Language).
- Custom python (jython) or scala code, by using OpenRefine’s dedicated extensions.
- GraphDB Functions including standard SPARQL function, as well as additional SPIN functions.
- Combining datasets between Refine projects by using the *cross()* function.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- Combining multiple repositories and projects using SPARQL Federation and the virtual SPARQL endpoint of each Refine project.
- The Refine command line interface.

The cleaned data can be transformed via a visual RDF mapping. The RDF mapping visual UI is optimized to guide the user through the most popular operations, like defining URLs, choosing the right predicates and types and defining datatypes. After this step, RDF data can be exported and exposed to a virtual SPARQL endpoint that allows the user to write complex SPARQL queries.

For more complex transformation scenarios, the mapping can directly be expressed as a SPARQL query.

The resultant RDF data can be loaded into a GraphDB repository using a Federated SPARQL UPDATE query, while project configurations and mappings can also be exported in order to automate a transformation on more data with a similar structure.

4.2.3.3.1.2 DataGraft

DataGraft is a tool that provides data and database management. The platform is used to integrate Grafterizer, an interactive tool for data cleaning and transformation. It provides functionalities for suggestion-based data cleaning and transformation, as well as using visual data profiling for the user.

The Grafterizer interface allows the user to see two tables: one with the original data and one with the changes made through the user-defined transformation. Each of these steps can be saved as a pipe, in order to create a general cleaning pipeline for the whole dataset. This way of composing increases the flexibility of the process and allows complex operations to concatenate easily.

As described in the Grafterizer user guide, the main steps to using the tool are the following:

- Import data using the "Make Dataset" function, specifying column names and the presence of the header.
- Apply transformation functions. They can be operations at the row-level, like remove duplicates or filter rows, column-level, like derive or add columns, or other types, like group and aggregate.
- Eventually, apply auxiliary functions. Since it’s not possible to predefine all operations, Grafterizer can also apply complex operations hard coded by the user using the "Edit utility functions" button.
- Eventually, apply string transformation functions. The tool allows the user to apply customized transformations on strings, like trimming and replacing values, using a graphical interface.

After the data transformation step is completed, it is possible to create RDF mappings. The platform allows for real-time viewing of the nodes and corresponding relations while creating the RDF skeleton. The RDF mapping in DataGraft supports conditional mapping. This means that if the user specifies the condition, the node will be created only for the rows where this condition is satisfied.

4.2.3.4 Research & technical challenges

4.2.3.4.1 Reduce human involvement in the data cleaning process

In the context of data cleaning, there are clear differences between the approach taken by researchers and the one taken by practitioners. Most of the time, the latter group deals with data errors by applying methods like interquartile range or isolation forest that do not allow for the identification of general rules but only specific ones for the given records.

Automating this process would allow the data cleaning step to also identify general problems, like, for example, population-level errors or problems at the pipeline level that otherwise would have been ignored. In this context, ML represents a good way to generalize the process, but it also introduces issues related to data quality, and the following training may have negative consequences for the result.

4.2.3.4.2 Scalable data cleaning

The emerging increase in data volume in the context of Big Data pipelines leads to an inevitable reduction in the rule of the human in the error identification process, without in any case removing him completely. For this reason, scaling the generation and processing of big data often comes at the cost of its quality.

To avoid this effect, some methods have been proposed to use active learning for error detection tasks, like the ActiveClean paper described above. Active learning allows the final user to be involved in the process while working only on limited samples that can be handled by a human. The knowledge of the user or domain expert is also learned by the model, which continues to learn to detect eventual data shifts that would discard rules already learned. This method creates an opportunity for new data cleaning frameworks that involve the human user in all the tasks required, making the whole process scalable but still human-in-the-loop.

4.2.3.4.3 Data-centric data cleaning

Data errors in a dataset may have varied origins, like human imputation errors or pipeline dysfunctionalities. One of the most popular approaches to approaching the problem is the use of data filtering techniques to remove noisy observations. In the context of the new emerging areas of ML-based data cleaning and data cleaning for ML, dirty observations could also be useful; they allow to detect automatic constraints or rules to deal with outlier cases without removing them and putting data quality at the centre.

A possible way to integrate this data-centric vision is the investigation of the robust data cleaning area to assign different weights to each record according to its quality. Instead of filtering out noisy records, robust methods allow the noise-aware loss function to assign different weights to records according to the level of confidence in each one. This emerging class would allow the user to focus more on the data quality level to obtain better data cleaning results.

4.2.4 Data linking (LinkR)

4.2.4.1 Scope and example

Data linking is about establishing links between different identifiers, where a link is a relation between a source element and a target identifier, and an identifier can represent an entity, a property or a concept. Different linking tasks and different interpretations of the linking relation are defined depending on the nature of the sets from which the source and target identifiers are taken. We consider two main classes of linking problems that are particularly important to support data enrichment:

- ***Entity matching*** is an important class of linking problems where the elements to be linked are different records in one or different knowledge bases that represent a same real-world entity. When records belong to the same dataset, entity matching it is often referred to as *data deduplication*, because the goal is to remove duplicate records. When records belong two different data sources, entity matching is also referred to as *record linkage* (especially in the database community), *link discovery*, *instance matching, entity alignment* (the latter phrases referring especially to tasks where the source and target data are RDF data sources, and/or knowledge graphs); the goal of entity matching in this case it is to identify which identifiers in two data sources refer to the same entity to either persist the link as an explicit relation (e.g., using the owl:sameAs predicate in RDF) or to fuse information about the same entity in the two sources.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

In all above cases, it is also quite frequent to refer to the same addressed task as linking, matching, or *reconciliation*; we may use all these terms in the remainder of this section.

- ***Semantic annotation of semi-structured data*** is another important class of linking problems is the one where semi-structured data, usually tables, are linked to data represented in a different model, for example entity identifiers in knowledge base, or concept identifiers in an ontology. In this class of problems, links can also be seen as annotations of elements in the source data (e.g., a column, a value, a pair of columns) with identifiers from a reference resource, and, usually, their goal is to carry pieces of a semantic description of the source data (for example, by interpreting a table using a knowledge graph as a reference); these links, annotations or semantic descriptions can be eventually used for different downstream tasks, e.g., generation of knowledge graphs from tabular data sources or plain semantic enrichment. This class of linking problems has been addressed with different flavours by different communities: *semantic modelling* of data sources considers, in principle, arbitrary semi-structured data (not only tables) [Vu & Knoblock 2022]; *table annotation* [Zhang et al 2017], as addressed mainly in the NLP community [Nadkarni et al 2011], and, the one that can be considered the most consolidated one as of today, Semantic Table Interpretation (STI) [Liu & Chabot et al 2022]. In our discussion of state-of-the-art approaches, we will use especially the terminology proposed for STI. Finally, while in this section we focus on the semantic annotation of semi-structured data, it is worth mentioning that some tasks, e.g., linking named entities to their identifier in a reference knowledge base, share some features with analogous tasks defined for texts in natural language, which are covered in Section 4.3.1.

A common feature in both kinds of tasks is the intrinsic uncertainty of the task: entities are referred to in different ways, which include ambiguous references and clues. A linking algorithm considers this uncertainty and makes decisions that may be wrong. Different algorithms may achieve different levels of accuracy on different data.

Before discussing the state of the art in entity matching and semantic annotation of semi-structured data, we first explain these two tasks more in detail and their role in data enrichment applications.

4.2.4.1.1 Entity matching

In entity matching, the two sources are usually represented in the same data model. An example with tabular data sources is represented in Figure 11, however, a similar example could be provided with a graph-based model like RDF. The key feature of entity matching is that it is possible to compare somehow structured descriptions of the record that are matched. In a data deduplication scenario, the structure defined by the schema of the data source is obviously the same. In a record linkage or link discovery scenarios, the entity matching task is usually executed after the schemas of the two sources are matched, by specifying with attributes or predicate in the two sources can be compared (for simplicity, in the example we use the same schema).

We also consider here two special and simple cases of entity matching with limited uncertainty. In a first case, links between two systems of identifiers exist (e.g., they have been discovered in advance) and are stored in a database; for example, a very large number of *owl:sameAs* links to Wikidata are enclosed in DBpedia. In this case, linking can be seen as a plain operation that given a source identifier returns the corresponding target identifier, which is as accurate as the links that are stored. In a second case, e.g., in the geospatial domain, links can be provided by algorithms that are under certain conditions so accurate that can be trusted in most of the cases. Examples are *geocoding* (given an address, its coordinates are returned) and *reverse geocoding* (given a pair of latitude and longitude coordinates, an address is returned), which are now incorporated in services that can be trusted for many real-world applications.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

The role of entity matching in data enrichment is to deduplicate records in one data source of interest or to integrate two data sources that are structured. If two sources are structured but use different models, it is possible to apply transformations to reduce the integration problem to an entity matching problem (for example, a table can be obtained from an RDF data source as a result of a SPARQL query).

title	director	release year
jurassic world	colin trevorrow	2015
superman returns	bryan singer	2006
…	…	…
jurassic w.	c. trevorrw	2015
Jurassic park	c. trevorrw s. spielberg	1993

title	director	release year
jurassic world	colin trevorrow	2015
superman returns	bryan singer	2006
…	…	…
~~jurassic w.~~	~~c. trevorrw~~	~~2015~~
Jurassic park	c. trevorrw s. spielberg	1993

Entity matching: data deduplication

title	director	release year
jurassic world	colin trevorrow	2015
superman returns	bryan singer	2006
…	…	…

title	director	release year
jurassic w.	c. trevorrw	2015
Jurassic park	c. trevorrw s. spielberg	1993
…	…	…

Entity matching: record linkage

Figure 11: Entity matching exemplified with tabular data in the data deduplication and record linkage scenarios.

4.2.4.1.2 Semantic annotations of semi-structured data sources

In the semantic annotation of semi-structured data sources links are established between elements of a semi-structured data sources and elements of a reference knowledge base (usually a knowledge graph). The problem that has been investigated more extensively is the problem of annotating tabular data, also because several other formats, e.g., JSON data, can be converted into tables under certain conditions. To explain the relation between linking and semantic annotations we refer to the example in Figure 12. In the example, annotations of cell values consist in links to entity identifiers; each of the first two columns is linked to the concept that represents the class of entities described in the column; the pair of columns that consist of the first two columns is linked to a predicate used in the reference knowledge graph. In practice the annotations consist in *entity-level links*, i.e., links from cell values to entity identifiers in the reference knowledge base, and *schema-level links*, i.e., links from columns to concepts or categories and from column pairs to predicates, where concepts, categories and predicates are either used in a reference knowledge bases or belong to other reference vocabularies and/or ontologies. As anticipated, Semantic Table Interpretation (STI) refers to the task where annotations all refer to entities, concepts and predicates from a reference knowledge graph, typically DBpedia or Wikidata (see Section 4.2.4.2). In a very similar task referred to as table annotation [Zhang et al 2017], Wikipedia categories can be also used to label columns; in semantic modelling of semi-structured data a very similar principle is applied but considering JSON files in their native format and/or considering arbitrary ontologies for labelling the sources.

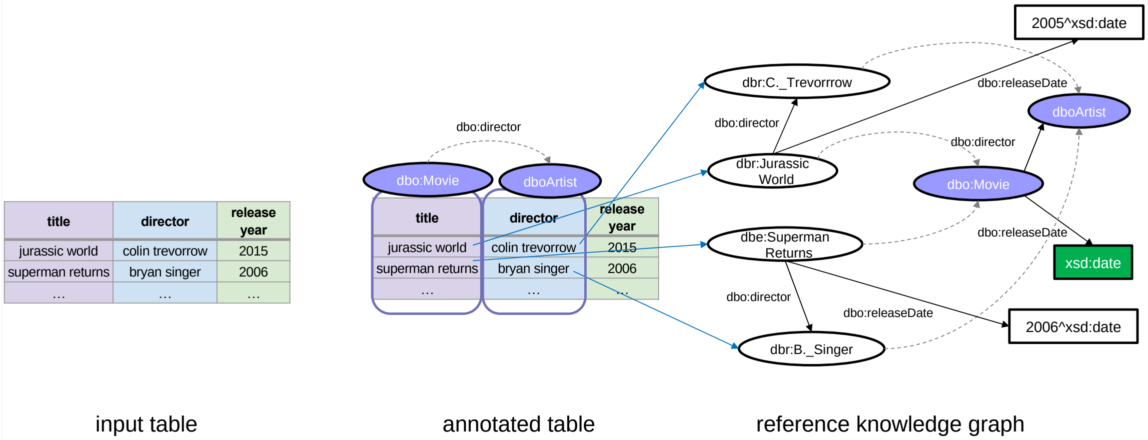


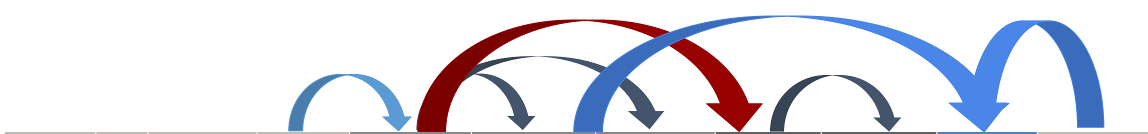
Figure 12: Semantic annotation of tabular data with a reference knowledge graph (DBpedia).

While the example shows links from all the cell values of the first two columns, in practice tables may contain cell values that refer to entities that are not present in the reference knowledge graph. Let us assume, for example, that Bryan Singer does not exist in the reference knowledge graph DBpedia. Algorithms providing entity-level links should detect that the value "bryan singer" does not correspond to any entity in DBpedia, and mark this as a novel unlinkable entity. Schema-level links however, can tell us that this novel entity is an artist (dbo:Artist) and is the director of the Superman Return movie. Eventually, we could extract a triple (e.g., <dbr:Superman_Returns, dbo:director, ex:Brian_Singer>) from the table. In other words, schema-level links support the extraction of novel information from the interpreted sources. This observation helps define the two main roles that semantic annotation of semi-structured data sources play for data enrichment:

- Construction and/or completion of knowledge graphs from semi-structured data sources: annotations can be converted in executable mappings with specific policies (e.g., how to transform values representing novel entities into IRIs) to generate a knowledge graph from semi-structured data sources or to complete an existing knowledge graph; this is the application scenario most considered in the literature [] and can be intuitively understood by looking at the example in Figure 12.
- (Plain) enrichment of tabular data: annotations can be used to bridge the gaps across different knowledge bases and systems of identifiers according to a "*link and extend*" data enrichment paradigm. In this paradigm, entity-level links to a reference data source allow to invoke a data extension functionality, e.g., issuing queries to that data source, and get more data (for data extension functionalities, see also Section 4.2.2). An example is illustrated in Figure 13, there the columns of the original table are depicted in light gray, while the new columns generated with the "link and extend" paradigm are depicted in darker shades of gray or in blue. The original table describes the performance (number of impressions, second column) of specific keywords used in digital marketing campaign (first column) in cities (third column) on certain dates (last column); each city is associated with a region and only textual labels are available for cities and regions. The goal of the data enrichment task is to collect, for each region, the average temperature on the date specified in the last column in order to train a model for weather-based impression prediction. The weather data source responds for geocoordinates and dates as input parameters, which requires bridging the gap between the spatial references (plain labels) and the respective representative coordinates. This is achieved by using a linking algorithm to reconcile region names

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

to the corresponding identifier in the GeoNames[107] knowledge base, from which latitude and longitude can be easily extracted. In addition, it is possible to extract links from Geonames to Wikidata (e.g., as stored in Wikidata) from which additional information can be extracted (area in the example). This application scenario for semantic annotation of semi-structured data sources has been increasingly investigated in the last years (see knowledge discovery in Section 5.2.2.2.3 [Cutrona et al 2019, Ciavotta et al 2023, Femke Ongenae 2021, Sarthou-Camy et al 2022]



Keyword	#im	City	Region	ID (Geonames)	Latitude (Geonames)	Longitude (Geonames)	ID (Wikidata)	Area (Wikidata)	Temp (ECMWF)	Date
194906	64	Altenburg	Thuringia				Q1205	45.6 km^2	18°	11/03/2017
517827	50	Inglostadt	Bavaria				Q980	133.35 km^2	17°	12/03/2017
459143	42	Berlin	Berlin				Q648102	891.68 km^2	17°	12/03/2017
891139	36	Munich	Bavaria				Q980	310.71 km^2	19°	11/03/2017
459143	30	Nuremberg	Bavaria				Q980	186.45 km^2	12°	10/03/2017

Figure 13: Example of data enrichment by composing different individual linking and extension services

4.2.4.2 Research

4.2.4.2.1 Entity matching

The literature about entity matching is extremely large. As anticipated similar entity matching problems are addressed in different application domains, resulting in several terms used to refer to equivalent or closely related problems. For example, record linkage, entity resolution, record deduplication and entity matching have been used mostly in the database domain, with entity matching being probably the broader term, therefore the one selected here. Instance matching, link discovery, and entity alignment have been used in the broad community that worked with ontologies, RDF and linked data, and knowledge graphs.

We distinguish between three main waves of contributions: approaches developed before the advent of machine learning (pre-ML), approaches that include traditional machine learning methods and/or other probabilistic approaches (ML/P), approaches that use deep learning approaches in some stages, including pre-trained language models (DL). Several survey papers have also been published that summarize the status in the field. In [Batini & Scannapieco 2016], several pre-ML and ML/P entity matching approaches proposed for relational databases are reviewed as part of methodologies for improving data and information quality. Pre-ML and ML/P approaches for link discovery, i.e., entity matching applied to RDF data are also reviewed in [Nentwig et al 2017]. A comprehensive review of pre-ML and ML/P and some initial DL approaches is given in [Christophides et al 2020], which focuses on challenges for entity matching in big data scenarios. A brief but very useful introduction to entity matching as means to support the construction of knowledge bases is provided in [Weikum et al 2021]. Since many DL methods have been proposed in the last years, a few surveys focus on these approaches: two surveys focus on DL approaches to match records in tabular data [Li et al 2021, Barlaug and Gulla 2021] and two surveys focus (mostly on) DL approaches to match entities in knowledge graphs, referred to as entity alignment [Zeng et al 2021, Zhang et al 2022]. In addition to a survey, a benchmarking experiment is discussed in [Zhang et al 2022],

[107] https://www.geonames.org/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

which presents some interesting results, showing that previous probabilistic entity matching methods can achieve better performance on the considered benchmarks.

To discuss aspects that may differentiate application settings and, consequently, approaches to entity matching we use the taxonomy proposed in [Christophides et al 2020], which is depicted in Figure 14. The survey refers to entity matching as entity resolution, but this taxonomy can be considered valid for understanding entity matching in broader terms. An important distinction concerns the approach to collect evidence about entity similarity: it may be *aware of the schema*, i.e., a user is assumed to specify the pairs of attributes that are compared to evaluate the similarity (schema matching is a pre-requisite), or *schema-agnostic*, i.e., the system is assumed to evaluate the similarity without prior knowledge on related schema elements in the two sources. The latter is required especially for semi-structured data, where the available attributes relevant for the comparisons may change in different entity descriptions. *Attribute-based matching techniques* examine each pair of descriptions independently from other pairs; these techniques have been applied to tabular data matching (including recent DL approaches) and to knowledge graph entities, especially in earlier approaches to link discovery [Nentwig et al 2017]. However, when entities come with graph-based descriptions, *collective matching techniques* are used to increase their matching evidence either by merging partially matched descriptions of entities or by propagating their similarity to neighbour entities via relations that will be matched in a next round. This distinction is relevant, because schema-agnostic collective inference has been applied significantly to matching entities of knowledge graphs, both in ML/P methods (e.g., in the LogMap and Paris approaches benchmarked in [Zhang et al 2022]), or in recent DL approaches to the alignment of knowledge graphs [Zeng et al 2021, Zhang et al 2022]. With respect to processing mode, we only observe that in addition to *static approaches*, where all entity descriptions are examined at once, dynamic approaches have been proposed that consider entity descriptions arriving in streaming or as queries.

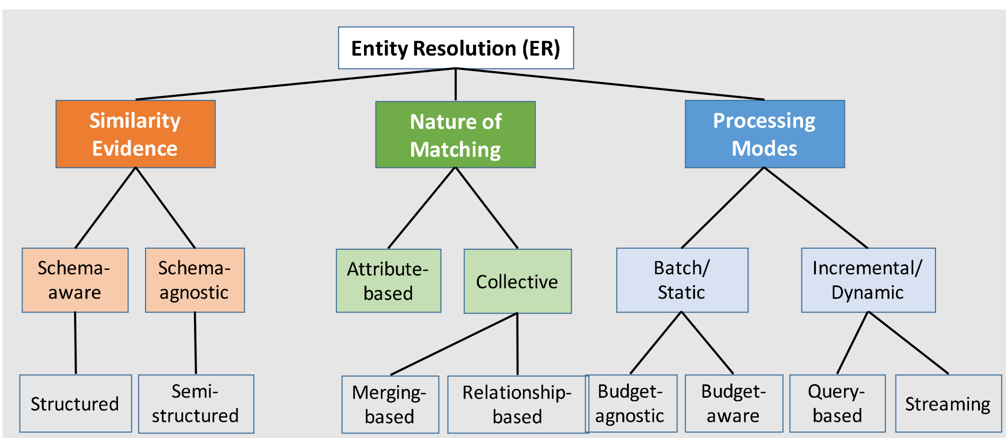


Figure 14: Taxonomy of entity matching settings and approaches (image from [Christophides et al 2020])

While it is difficult to come to a join conceptualization of the aspects that characterize an entity matching problem, there are some aspects that recur in most of the various settings in which entity matching has been studied. First, because the theoretical comparison space is quadratic (each entity in one data source should be compared to each entity of the other data source) efficiency is a key focus in most of the approaches. As a result, a first recurrent aspect considered in most of the approaches is **blocking**: each entity in one source is compared only with a subset of the entities in the other source based on a variety of heuristics and techniques. A second recurrent aspect is, obviously, **similarity evaluation**. Many string similarity measures have been introduced in pre-LM and LM/P approaches [Batini & Scannapieco 2016], while latest approaches may use machine learning to infer, or learn effective representations of the input

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

data or effective similarity measures. It is also worth noting, that in most of the cases, different similarity measures are combined (e.g., different measures for different attribute pairs, different similarity measure on the same attribute pair, different similarity measures learnt from different data modalities, etc.); as a consequence, many approaches consider both the evaluation of initial similarity scores that consider different aspects of the entity descriptions as well as the problem of their combination in a final similarity score. While in early pre-ML methods the combination of different similarity measures could be defined beforehand, in ML/P approaches the combination can be learned using, e.g., decision trees (e.g., to learn link specification functions in the LIMES framework for entity discovery [Nentwig et al 2017, Obraczka & Ngonga Ngomo 2019, Ngonga Ngomo et al 2021]), and in DL approaches the combination can be learned using complex information fusion mechanisms; for example, a function that summarize individual similarity scores for attribute pairs is proposed for tabular data (e.g., in the DITTO model [Li et al 2020]) and message passing and attention mechanisms are used in recent graph-based deep learning approaches to entity alignment [Zeng et al 2021, Zhang et al 2022]. A last aspect concerns the problem of using the similarity evaluation to **make a final decision about matching**. In some settings, the problem consists in finding an optimal threshold on a final matching score (e.g., learned from a set of labelled data in ML/P approaches), while in others the problem is interpreted as the task of clustering descriptions that refer to the same entity [Christophides et al 2020].

While we refer to the above-mentioned surveys for more details on the broad set of techniques proposed for entity matching, we report some analysis that are relevant to identify lessons to be learned from the state of the art and a few relevant open problems.

By fine-tuning large language models, DL-based approaches like Ditto achieved state of the art results in many entity matching tasks on structured data [Barlaug and Gulla 2021, Paganelli et al 2022]; these approaches usually use a schema-aware method, but are quite tolerant with missing values, which make them applicable also in cases where the schema is not filled for all records; in addition, since these methods exploit natural language descriptions at a great extent, they are quite powerful even when dealing with poor descriptions (e.g., title and category). Interestingly, different approaches to optimize approaches similar to DITTO [Li et al 2020] based on language models, are proposed in [Paganelli et al 2022], thus providing a good practical guide to exploiting these approaches in specific application domains. Following up on some of the challenges discussed in previous papers, recent work has approached two open problems with remarkable implication from the application point of view: domain adaptation with limited training data [Trabelsi et al 2022, Tu et al 2022], which paves the way for solutions that can optimize entity matching algorithms with a few user feedback.

Most of the recent approaches proposed to align entities of two knowledge graphs operated in a schema-agnostic settings and exploit graph neural networks [Liu et al 2022]. However, recent papers [Sun et al 2020, Guo et al 2022, Leone et al 2022] have also observed that the latter approaches do not outperform conventional solutions like LogMap [Jiménez-Ruiz & Cuenca Grau 2011] or, especially, PARIS [Suchanek et al 2011] yet. The authors of [Zhang et al 2022] note that the previous study missed an important feature in a pair of approaches, and, remarkably in AttrE [Trisedya et al 2019], which result as the best performing approach in their benchmark. However, it should be noted that [Zhang et al 2022] does not provide a comparison with LogMap and PARIS, leaving quite uncertain the question whether the amount or resources consumed by DL approaches to entity alignment (which impact on their scalability) is worth the gain in performance with respect to more traditional pre-ML or ML/P models proposed several years before. Another open question concerns if ML/P approaches to support link discovery, e.g., the ones based on learning link specifications [Obraczka & Ngomo 2019], are somehow outperformed by the latest generation of DL approaches; unfortunately, a systematic comparison on the same benchmark data is not provided.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.2.4.2.2 Semantic annotations of semi-structured data sources

In the literature, different terms have been proposed to describe the task of linking entities in tabular data to a knowledge graph (KG). These terms include semantic modelling of data sources, table annotation, and semantic table interpretation. The main goal of this task is to reconcile the mentions of entities within the table with a particular KG. Additionally, other related tasks may include identifying the column data types and potential relationships between columns.

According to [Liu, Chabot et al 2022], there has been a growing interest and popularity in Semantic Table Interpretation over the years, as evidenced by Figure 15. Additionally, Figure 16 illustrates the distribution of papers by topic.

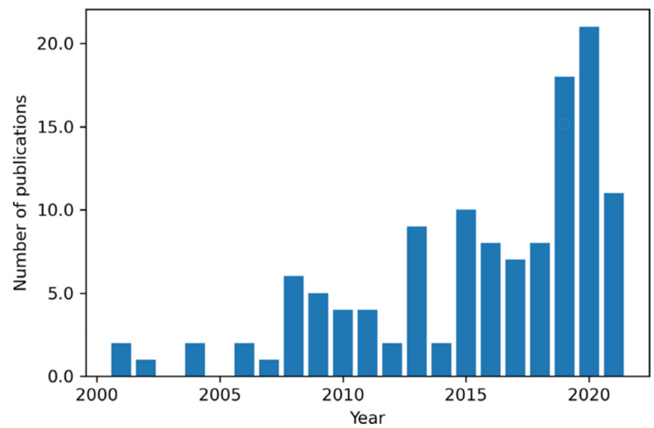


Figure 15: Number of publications of paper related to STI (image from [Liu, Chabot et al 2022])

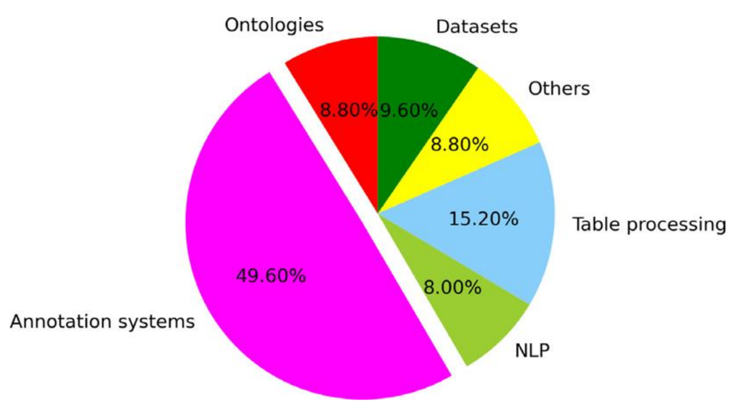


Figure 16: Distribution of the topics of the papers related to STI (image from [Liu, Chabot et al 2022])

In the field of table annotation, various approaches have been proposed, which can be broadly categorized into two main categories: heuristic-based and machine learning-based methods. However, the most important task in table annotation is entity linking (CEA), which involves associating a mention in a table with a corresponding entity in a knowledge graph.

Upon performing entity linking over mentions in tabular data, two critical annotations can be obtained: Column Type Annotations (CTA) and Column Predicate Annotations (CPA). CTA provides information about the types of entities extracted for each column, while CPA identifies the predicates or relationships between the columns, which are extracted based on the relationships between the entities.

In summary, entity linking is a crucial step in table annotation, providing valuable insights into the data represented in the table. CTA and CPA can help us better understand the relationships and connections between the entities represented in the table, supporting various tasks such as information retrieval, data integration, and knowledge discovery.

4.2.4.2.3 Heuristic based

Most approaches are children of the Semantic Table Interpretation Challenge and most of them address the problem of column classification (CEA, CTA and CPA).

Many approaches like [Nguyen et al 2021, Huynh et al 2022, Cremaschi et al 2022, Chen et al 2022, Abdelmageed et al 2021, Shigapov et al 2020, Steenwinckel et al 2021] have been proposed during international Semantic Web Challenge on Tabular Data to Knowledge Graph Matching The most relevant are MTAB [Nguyen et al 2021] that proposed an approach based on a framework which consist in two phases Structural Annotations (identify structure of the table) and Semantic Annotations (perform the annotation steps CEA, CTA and CPA). DAGOBAH [Huynh et al 2022] proposes an approach that is a two-stages annotation system consisting of an entity lookup step, followed by an entity scoring step. Finally, SELBAT [Cremaschi et al 2022] proposes an approach that basically provides an iterative process that performs Entity Linking (EL) on tables.

The approaches cited above took part in different editions of the Semantic Web Challenge and improved their systems as a result.

4.2.4.2.4 Machine learning based

Most of the approaches discussed here benefit from recent progress in machine learning and focus on column classification (task CTA). These approaches can also be useful for obtaining preliminary schema annotations of a table.

[Chen et al 2019] proposed ColNet an approach base using embeddings technique to predict the type annotations (CTA). This work uses a CNN to predict the type of a synthetic column, inspired by its successful application in text classification [Kim et al 2014].

[Suhara et al 2022] proposed a new framework called Duduo for tabular column annotation, which achieved state-of-the-art performance on two benchmark datasets. They used pre-trained Transformer language models and multi-task learning to improve performance. Duduo was also shown to be data-efficient, achieving competitive performance using only 8 tokens per column or about 50% of the training data. Their approach outperformed previous state-of-the-art methods, including Sherlock [Hulsebos et al 2019], TURL [Deng et al 2022] and Sato [Zhang et al 2019]

4.2.4.3 Tools and market

4.2.4.3.1 Entity matching

A very large number of commercial entity matching solutions for data coming in the most used formats (e.g., relational databases, JSON) exist. Since entity matching is a useful asset in the enRichMyData toolbox, but not one in which the project aims at advancing the competitive state-of-the-art tool landscape, we provide a broad overview of the available tools and related market without describing in detail each solution.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

A set of popular software tools listed on Sourceforge[108] and SlashDot.org[109] include WinPure's Clean & Match[110], DataMatch Enterprise™[111] by Data Ladder, Data Matching by FirstEigen[112], Exorbyte[113], Datactics[114], Senzing[115], Syniti's Data Matching Engine[116] (formerly Matchit), Experian Data Quality[117] and a few more. Also, database management systems and other data management solutions supporting data quality along the whole data life-cycle offer robust support for data matching; examples are Talend Studio[118], TIBCO EBX™[119], SAS Data Quality[120], and Oracle Enterprise Data Quality[121] (for Talend Studio and Oracle Enterprise Data Quality we reviewed data extensions features in Section 4.2.2.3, while in this section we only review the features of Talend that support the semantic annotation of tabular data). Most of the above-mentioned specialized software includes the possibility to define matching rules, but now also applies machine learning and/or probabilistic and fuzzy matching. Many also come with a user interface, like in Talend, which supports the definition of entity matching workflows. Most of these solutions use proprietary algorithms and it is not clear to what extent also include DL approaches and recent advances on domain adaptation with limited training data. The above-mentioned lists also include OpenRefine, an open-source software that also provides data deduplication functionalities as well as cell entity annotation functionalities (see next paragraph) and described in (see Section 4.2.3.3.1).

For DL approaches two reference research prototypes that have proven to achieve state of the art results on several research benchmarks are available as open-source software, i.e., DeepMatcher[122] [Mudgal et al 2018], and DITTO[123] [Li et al 2020]. The tools are quite easy to use at least in a controlled R&D environment.

Most of entity alignment approaches based on graph embeddings did not come as proper tools. Considered the concerns about the actual benefits of DL approaches for entity alignment we do not provide more details about these tools. However, we mention that PARIS[124] and LogMap[125] (see Section 60) are open source and can be used, at least in a controlled R&D environment. Also, it should be stressed that both systems do not align only entities, but also properties and classes in a knowledge graph.

[108] https://sourceforge.net/software/data-matching/for-freelance/

[109] https://slashdot.org/software/data-matching/?sort=rating_avg

[110] https://winpure.com/

[111] https://dataladder.com/products/datamatch-enterprise/

[112] https://firsteigen.com/data-matching/

[113] https://www.exorbyte.com/de

[114] https://www.datactics.com/

[115] https://senzing.com/

[116] https://www.syniti.com/solutions/data-matching/our-matching-engine/

[117] https://www.edq.com/

[118] https://www.talend.com/

[119] https://www.tibco.com/it/node/704336

[120] https://www.sas.com/it_it/software/data-quality.html

[121] https://www.oracle.com/assets/oracle-enterprise-data-quality-ds-430148.pdf

[122] https://github.com/anhaidgroup/deepmatcher

[123] https://github.com/megagonlabs/ditto

[124] https://github.com/dig-team/PARIS

[125] https://www.cs.ox.ac.uk/isg/tools/LogMap/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

However, these systems have not been defined with scalability in mind, despite they could have been run on several benchmark data. Otherwise, two link discover solutions designed to be more scalable and adaptable to specific matching tasks are available as open-source research prototypes: LIMES[126] [Ngonga Ngomo et al 2021], and SILK[127] [Bizer et al 2009]. SILK also comes with a user interface that can be used to define link specifications. It should be noticed that is unclear whether the most recent matching approaches published by the authors of LIMES and SILK have been incorporated into the systems.

4.2.4.3.2 Semantic annotations of semi-structured data sources

List of Semantic Modelling and STI prototypes:

- **OpenRefine** - is a powerful free, open source tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data. In addition to its data cleaning and transformation capabilities, OpenRefine offers various features for extending data with web services and external data sources. Users can utilize APIs to fetch data from web services such as Twitter, Google Maps, and Freebase, and then merge this data with their existing datasets. OpenRefine also supports the use of custom scripts written in languages such as Python, JavaScript, and Ruby, which can be used to automate complex data transformation tasks or integrate with external databases and services. Furthermore, OpenRefine provides a range of plugins that can be installed to extend its functionality even further. These plugins allow users to perform tasks such as text analysis, clustering, and data validation.
- **MAGIC** - is a data mining tool that uses Knowledge Graphs to enrich a structured file with new data and information [Steenwinckel et al 2021]. Unlike other data mining tools, MAGIC exploits an interpretable embedding approach called INK to perform CEA, CPA, and CTA tasks simultaneously, reducing the number of external calls to entity lookup services. In addition, MAGIC uses an innovative method to handle missing values within the KG, called local graph completion, which allows additional data to be integrated into the structured file even if some data in the KG is incomplete. The framework can be divided into three modules, a pre-processing module to select the appropriate annotation candidates, an INK- based processing module to obtain interpretable embedding and select the best matches, and, finally, a post-processing module to download the obtained information into the corresponding competition tasks. It also offers an intuitive and user-friendly user interface for data exploration and visualization.
- **DAGOBAH UI** – also enrichment offers a web interface that focuses on providing advanced functionality for enriching tables by exploiting the data contained within Knowledge Graph and vice versa [Sarthou-Camy & al 2022]. In contrast to Open Refine, where Cell-Entity Annotation (CEA) procedures of- ten require a manual cell annotation step, DAGOBAH UI fully automates Semantic Table Interpretation (STI) processes, allowing tables to be enriched fully automatically. DAGOBAH UI allows interaction with the system through a user-friendly user interface that enables the manipulation, interpretation, and enrichment of relational tabular data. DAGOBAH UI's pre-processing toolbox allows a table to be cleaned up, resolving any coding and cell alignment issues, and extracting table topology information (orientation, header, etc.), which is crucial to the annotation process. All three main types of annotation are performed by DAGOBAH UI, CEA, CTA, and CPA. Once annotated, tables can be enriched with additional elements from the supporting KG. The enriched table can then be exported by the user for further analysis.

[126] https://github.com/dice-group/LIMES

[127] http://silkframework.org/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

DAGOBAH UI's modal window for selecting new columns to be added to the table allows the user to preview the added values before confirming the operation, to understand which properties are most significant to the extension.

- **MTab** - is an automated tool used to annotate data in tables through the use of knowledge graphs [Nguyen et al 2021]. The software provides semantic and structural annotations to the data in the table. It can perform all STI-related activities: CEA, CTA, and CPA. The supported knowledge bases include Wikidata, DBpedia, and Wikipedia. In addition, MTab supports the handling of multiple table formats, such as Excel, CSV, TSV and markdown tables, as well as multilingual tables. The process of annotating tabular data in MTab involves several steps, starting from pre-processing the data to enriching the table with semantic annotations. The system uses search and prediction tasks to provide semantic annotations to the data in the table. In addition, MTab features a user interface that supports the annotation task and displays the results of the automatic process. The user interface also allows users to search for entities in common knowledge graphs, such as Wikidata and DBpedia. However, MTab does not allow the addition of new columns or data to the table, but only focuses on the annotation of data already present.
- **SAND** – probably it can be considered an update of the KARMA application [Gupta et al 2012], which pioneered semantic table annotations from user interfaces. It is a tool for creating semantic descriptions semi-automatically [Vu et al 2022]. SAND makes it easy to integrate with semantic modelling systems to predict or suggest semantic descriptions to the users, as well as to use different knowledge graphs (KGs). Besides its modelling capabilities, SAND is equipped with browsing/querying tools to enable users to explore data in the table and discover how it is often modelled in KGs.
- **MantisTable** [Cremaschi et al 2019] can import tables from files and create semantic descriptions to map them to KGs such as Wikidata. The system is tightly integrated with its semantic modelling algorithms and hence difficult to extend with state-of-the-art algorithms. SemTUI can be considered the application that is going to further extend MantisTable features.

In the table below the main characteristics of the tool are summarized.

Table 4: Existing semantic annotation tools analysis

Tool	Kind of tool	Level of automation	type of approach
Open Refine	Desktop app	Semi-auto	Human Driven
MAGIC	Desktop app	Auto	Heuristic
Mtab	web app	Auto	Heuristic
DAGOBAH UI	web app	Auto	Heuristic
SAND	Web app	Semi-auto	Heuristic

Table 5: Existing semantic annotation tools main functionalities

Tool	Column datatype identication	CEA	CTA	CPA	Data extension	EXPORT (annotation)
Open Refine	☑	☑	✗	✗	☑	☑
MAGIC	☑	☑	☑	☑	✗	☑
Mtab	☑	✗	☑	☑	✗	☑
DAGOBAH UI	☑	☑	☑	☑	☑	☑
SAND	☑	☑	☑	☑	✗	☑

4.2.4.4 Tools from the consortium

4.2.4.4.1 Entity matching

4.2.4.4.1.1 Ontotext Refine

See Section 4.2.3.3.1.1 for a thorough description of the tool. By means of integrating OpenRefine provides a client-side UI for any reconciliation API protocol implementation.

4.2.4.4.1.2 Ontotext Reconciliation

Ontotext reconciliation is a tool for exposing Reconciliation API over RDF data indexed in GraphDB.

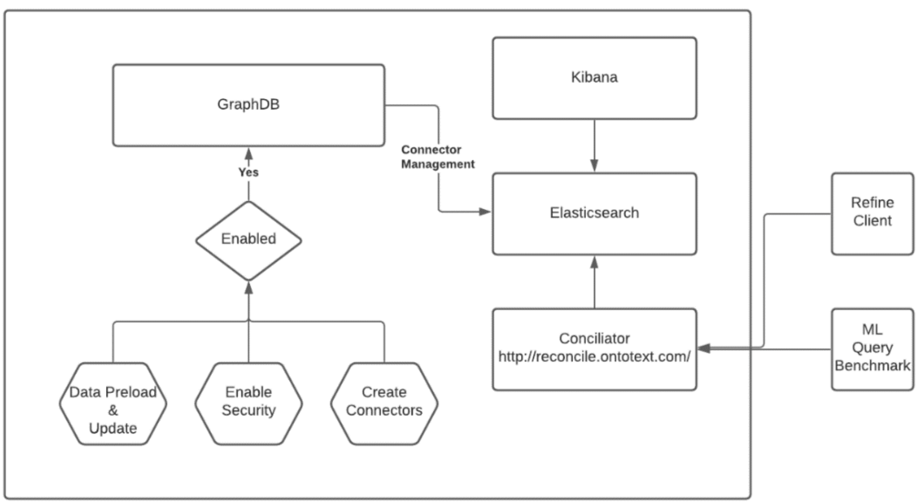


Figure 17: Ontotext Reconciliation tool

Ontotext Reconciliation tool aims to simplify the creation of a generic reconciler service compliant with OpenRefine reconciliation APIs. It minimizes user input by leveraging the ontology and content of an RDF graph. Users familiar with RDF can utilize the tool to build a robust reconciliation service without needing to modify the underlying code. Users are required to specify the types of entities to be covered by the service and can extract optional features from the graph to enhance reconciliation. The tool utilizes GraphDB's connectors, such as Lucene, Elasticsearch, and Solr, for efficient and faceted searches. Connectors ensure entity-level synchronization and support property chains, allowing users to select reconciliation candidates effectively. Key features of the connectors include automatic

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

synchronization to reflect the latest data in GraphDB, full-text search capabilities for discovering matches, and support for optional features beyond the mandatory name field.

Connectors also offer custom mapping for RDF types, ensuring accurate handling of booleans, integers, dates, and other data types. Matches suggested by the connectors are scored, sorted, and paged, enabling easy retrieval of top candidate matches. While the scoring is not limited to a specific scale by default, it would be beneficial to automatically translate the score into a [0,1] range for confidence scoring. In scenarios where connectors cannot handle complex custom features, the tool allows retrieval of all indexed values associated with a specific entry. This enables custom processing but requires extending the codebase or making changes to the Elasticsearch service if applicable.

In summary, the tool for creating generic reconciliation services should:

- Support extension for more complex reconciliation services.
- Implement OntoRefine reconciliation APIs.
- Utilize GraphDB's fast search connectors for candidate suggestion.
- Support optional reconciliation features of various data types.

4.2.4.4.1.3 Reconcile CSV

Reconcile-csv[128] is a reconciliation service for OpenRefine running from a CSV file. It uses fuzzy matching to match entries in one dataset to entries in another dataset, helping to introduce unique IDs into the system - so they can be used to join your data painlessly.

4.2.4.4.2 Semantic annotations of semi-structured data sources

4.2.4.4.2.1 Selbat

Selbat provides an algorithm to perform full-fledged STI (CEA, CPA, CTA) on an input well-formed table. It is available as API and returns a mapping from elements in the table (values, column pairs and columns) to a reference KG, e.g., WikiData. The interpretation process starts with pre-processing. Afterwards, an entity linking (EL) algorithm is applied on the textual values of the table, also referred to as *entity mentions* in the reminder of this section. The algorithm finds, for each mention, the best matching entity in the reference KG, i.e., the top candidate in a list; the list is preserved and additional data are collected. This step returns a first result for the CEA task. After the identification of a subject column S, an algorithm annotates each pair of columns <S,C>, where C is a column different from S with a property of the KG (CPA task). The approach also annotated each column with a concept in the KG that is expected to reflect the type of the data (entities or literals) contained therein (CTA task). Eventually, the algorithms re-evaluate the candidate entities for each mention by re-ranking the candidates using the evidence collected during the CPA and CTA tasks. The tool returns all the results via API and is compatible with data formats used in international challenges for STI.

4.2.4.4.2.2 SemTUI

SemTUI is a fully modular framework for the Semantic Enrichment of Tabular Data, adoptable by both experts and non-experts in the context of semantics. Nowadays, the enrichment task is at the core of almost every data analytics pipeline, and at the same time, it proves to be also costly in both time and money. For this reason, it is important to provide data scientists with tools that guide them through the

[128] https://okfnlabs.org/reconcile-csv

steps of the enrichment process, supporting them with interactive choices and visualizations. Semantics can bridge the gap in finding links across datasets and find solutions for the extension step, but it is important to include users in the annotation process. Indeed, the extension step is strongly related to the semantic annotation, but automatic algorithms that provide it can fail. However, if results are made available, interpretable, and editable, they can be reviewed and improved by the human's knowledge. A study and overview of state-of-the-art tools for both the Semantic Interpretation and Enrichment task have highlighted some of their problems. They are limited by not supporting humans in the loop of the annotation process, while also not fully providing extension steps necessary to complete the enrichment of tabular data. Most of the related works also present an entry barrier for less experienced users.

4.2.4.5 Research & technical challenges

Here are some research and technical challenges for entity linking in general, including entity matching and entity linking over tabular data:

1. **Accurate Entity Disambiguation, Contextualization, and Consolidation:** One of the biggest challenges in entity linking is disambiguating between entities with similar or identical names. For example, linking "Apple" to the technology company versus the fruit. Entities often have different meanings depending on the context in which they appear. For example, the word "bank" could refer to a financial institution or the edge of a river. Developing methods to accurately contextualize entities is a challenge in entity linking. Also, entity names that are not present in the reference KG and cannot be linked should be clustered to recognize which represent the same entity (entity resolution).
2. **Cross-lingual linking:** Entity linking becomes more challenging when dealing with multiple languages. Matching and linking entities across languages and cultures requires innovative approaches and techniques.
3. **Scalability:** As datasets grow in size and complexity, it becomes increasingly difficult to efficiently match entities. Developing algorithms and techniques that can handle large-scale datasets is a key challenge in entity linking.
4. **Robustness with respect to different data distributions:** the structure of the data can vary significantly between tables. Developing techniques to effectively link entities across different types of tabular data is a key research challenge.
5. **Evaluating performance:** Developing effective methods to evaluate the performance of entity linking algorithms is an ongoing challenge, particularly as the datasets and contexts in which entity linking is applied continue to grow in complexity.
6. **Entity linking with limited training data and human-in-the-loop approach:** In many cases, there may be limited training data available for a particular domain or context. In this case human-in-the-loop approaches with active learning to minimize human effort is a key research challenge.

4.3 Approaches to support enrichment of textual data

4.3.1 Information extraction (StructureR)

4.3.1.1 Scope and example

The objective of the StructureR is to automatically extract structured information from unstructured text documents, written by humans for humans, using human language. In the field of Natural Language Processing (NLP) this task is known as Information extraction (IE). The goal of this process is to identify relevant information within the text and convert it into a format that can be easily analysed and used by

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

computers. Common formats for storing data extracted include tables with connections between them (relational databases) and subject-predicate-object triples (triple stores). Combining knowledge extraction with powerful database systems enables the creation of sophisticated applications in industries such as healthcare, finance, marketing, and customer support.

The process of information extraction typically involves several steps:

- ***Entity recognition and linking*** which involves identifying mentions of entities such as people, organizations, products, and locations within the text and linking the recognized entity mentions to corresponding entities in a knowledge base or reference database.
- ***Relation extraction*** involves identifying the relationships between these entities such as a company producing a certain product. Open relation extraction involves identifying relations between entities without any prior knowledge of the types of relations that may exist. Closed relation extraction, on the other hand, involves identifying specific pre-defined relations between entities. In this approach, the goal is to identify only the relations that have been previously defined and are relevant to the specific task or domain at hand.
- ***Event extraction*** involves identifying events or actions that take place within the text, such as product recall announcement or agreement being reached between organizations.
- ***Temporal information extraction*** consists of extracting including both relative and absolute dates, often associated with either relations or events.

Consider the example sentence:

During yesterday's press conference, Boston Dynamics CEO Robert Playter unveiled Stretch. First announced in March 2021, Stretch, is a robot designed to automate boxes moving in warehouses and capable of carrying up to 50lbs at a time.

Example results in each step would be:

- Entity Recognition and Linking:
    1. (Boston Dynamics, Q2298325)
    2. (Robert Playter, Q108550690)
- Relation extraction (Open):
- For verb “unveil”:
- *obj_what* (“what”) = Stretch (The object of the relation “unveiled”)
- subj_who (“who”) = Robert Playter (The subject of the relation “unveiled”)
- Relation extraction (Closed):
- (Robert Playter, Boston Dynamics, employee_of)
- (Boston Dynamics, Stretch, produced_by)
- Event Extraction:
- press conference
- product launch
- Temporal information extraction:
- yesterday
- March 2021

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.3.1.2 Research

4.3.1.2.1 Entity Recognition and Linking

Entity linking is a natural language processing technique that involves identifying and linking named entities mentioned in text to their corresponding entities in a knowledge base, reference database or knowledge graph, such as Wikidata, which contain rich and precise information about entities, such as persons, locations, organizations, products as well as the relationships between them. The challenge of recognizing entities in text arises from the fact that entities can be described in many different ways, and the same entity can be referred to using different names or even different variations of the same name. For example, a person may be referred to by their full name, their first name, their last name, a nickname, or a combination of these. This variability in how entities are referred to makes it difficult for a machine learning model to accurately identify and classify them in text. Another challenge is the potential for ambiguity in the context of the text. For example, the mention of "Paris" in a sentence could refer to the city in France, or other cities, the mythological Paris of Troy, or a person whose surname or first name Paris[129]. This ambiguity can make it difficult for an entity recognition system to correctly identify the intended entity. Furthermore, named entities can often be multi-word phrases, which increases the complexity of their identification. For example, "New York City" is a named entity that consists of three words and recognizing it as a single entity requires identifying the correct boundary between the words.

The typical process involves, conceptually, the following stages:

1. Mention detection model identifies the mention boundaries in text.
2. Candidate generation creates a shortlist of possible entities (candidates) for the mention.
3. Mention encoding creates a representation of the mention (relative to the context).
4. Entity encoding creates a representation of candidates, typically, this is static created only once then retrieved whenever an entity is a candidate.
5. Entity ranking compares mention and entity representations and estimates mention-entity correspondence scores, usually, the highest scoring mention-entity representation pair is linked.

It is common to separate the mention detection and entity disambiguation (ED) stages and use different models, but it is also common to do them at the same time by the same model (joint entity detection and disambiguation, or end-to-end linking). Another key area where there are significant differences in approaches is between approaches that use local or global context. In local approaches to ED, each mention is disambiguated independently based on the immediately surrounding words, while global approaches consider the consistency between multiple entities. The final key difference we want to highlight is the difference between mono-lingual and cross-lingual architectures. Cross-lingual entity linking (XEL) faces challenges in candidate generation and mention detection due to the lack of mappings between mention strings and entities in resource-poor languages.

Entity linking in text is a very important and extremely researched task. A full survey is beyond the scope of this work. See (Sevgili, 2022) for a recent survey. Here we highlight some key works that are relevant in the context of this document:

- **Wikifier** (Brank, 2017) is a multilingual and cross-lingual approach that performs joint entity detection and disambiguation primarily using the PageRank (Brin, 1998).

[129] https://en.wikipedia.org/wiki/Paris_(disambiguation)

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **DeepType** (Raiman, 2018) is a cross-lingual global approach for entity disambiguation that performs disambiguation by creating and using a type system to constrain and check entity types, it thus operates not on the text itself but on the possible types of the entities.
- **BLINK** (Wu, 2020) is a monolingual model that performs linking by leveraging a BERT (Devlin, 2019) architecture model and the local context of the mention to do both retrieval of candidates and ranking.

4.3.1.2.2 Relation extraction

Relation extraction (RE) is the task of automatically identifying and extracting structured information about relationships between entities mentioned in natural language text. The process of relation extraction involves identifying the relevant entities mentioned in the text and then identifying the relationship between them. Given text, a relation extraction system outputs a list of triples consisting of the entity pair involved in the relationship and the relationship. If the entities are provided by a separate system (such as an entity liking system), the task can be called simply relation classification. The main difference in approaches to RE is between open and closed RE. Closed relation extraction involves extracting relations that are predefined and limited to a specific set of relation types. Open relation extraction (OpenRE) involves extracting relations without prior knowledge of the types of relations that may exist in the text. In OpenRE the relation is usually represented by its surface form, the specific text that represents the relation or a slightly normalized version such as the verb. This contrasts with closed RE which represents relations as class labels. Closed RE systems, commonly, use supervised machine learning methods, although sometimes include distant supervision or semi-supervised approaches. OpenRE systems are more diverse in this regard. Common approaches to OpenRE are:

- **Pattern-based methods:** These methods involve identifying patterns in text that indicate the presence of a relation and using these patterns to extract relations.
- **Distributional semantics:** These methods analyse the co-occurrence patterns of entities in a large corpus of text to identify potential relations.
- **Dependency parsing:** These methods use syntactic dependencies between entities to extract relations.
- **Neural network-based approaches:** These methods use deep learning models to classify pairs of entities as having a relation or not.
- **Graph-based methods:** These methods represent entities and relations as nodes and edges in a graph and use graph-based algorithms to extract relations.

Some OpenRE systems integrate more than one of these approaches. For an overview of some of the most well-known examples see [Niklaus, 2018]. In terms of closed, supervised, relation extraction, we highlight the following works:

- $\mathrm{BERT}_{\mathrm{EM}}$ [Soares, 2019] uses BERT with a classification head on top of the output embeddings corresponding to entity markers in the input at the start of the object and subject of the relationship to classify a relation while the typed marker extension () adds entity typing information to the inputs.
- (Sainz, 2021) use a model pretrained Natural Language Inference (NLI) to perform relation classification in zero-shot, few-shot, and fully supervised settings.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.3.1.2.3 Event extraction

Event extraction (EE) is the task of automatically identifying and extracting information about events described in natural language text. An event is typically defined as an occurrence in time and space that involves one or more participants or entities. This process often includes several tasks such as event detection, argument extraction, role labelling. In close-domain event extraction, a predefined structure is provided, while open-domain event extraction does not require any predefined structure. The most well-known corpus, ACE 2005[130], defines an event schema whose terminologies have been widely used in event extraction [Lai, 2022]:

- Event extent is a sentence within which an event is expressed.
- Event trigger is a word or a phrase that most clearly expresses the occurrence of the event.
- Event arguments are entities that are part of the event.
- Argument role is the relationship between an event and its arguments.

Some variations of this terminology are also common [Li, 2022]. Each of these can be considered its own task. Some approaches focus on solving only one (or a couple) of these tasks, with some systems building a full solution by sequentially solving each, while other approaches are fully end-to-end and perform joint modelling of all tasks. Among these, we highlight one of the most recent approaches, Text2Event [Lu, 2021]. It uses T5 [Raffel, 2020] to create an encoder-decoder architecture that encodes text and decodes it directly into an event structure that is a linearized form of a table or tree format represented in text form.

4.3.1.2.4 Temporal information extraction

Temporal information extraction is the task of automatically identifying and extracting information about temporal expressions mentioned in natural language text. Temporal expressions refer to any word or phrase that denotes a point in time, such as "yesterday," "in 3 days," or "January 1st, 2022". The process of temporal information extraction involves identifying the relevant text in which a temporal expression is mentioned, normalizing the expression to a standardized format, and extracting this information in a structured format, such as a timeline or calendar. For a full overview of the task see [Lim, 2019]. With regards to this task, we highlight HeidelTime [Strötgen, 2013], a multilingual rule-based system. For simpler requirements it's also feasible to handle temporal information extraction using NER based system. OntoNotes[131] such as FLAIR [Akbik, 2018].

4.3.1.3 Tools and market

We begin by giving an overview of Open Source information extraction tools:

- **Natural Language Toolkit (NLTK)[132]:** A popular Python library for working with human language data. It provides some tools for information extraction, such as named entity recognition and part-of-speech tagging.
- **Apache OpenNLP[133]:** A machine learning-based toolkit for natural language processing tasks, including information extraction. It offers pre-trained models for named entity recognition, part-of-speech tagging, dependency parsing, and coreference resolution.

---

[130] https://catalog.ldc.upenn.edu/LDC2006T06

[131] OntoNotes Release 5.0, https://catalog.ldc.upenn.edu/LDC2013T19

[132] https://www.nltk.org/

[133] https://opennlp.apache.org/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **Stanford NLP[134]:** A suite of natural language processing tools developed by Stanford University. It includes modules for named entity recognition, part-of-speech tagging, coreference resolution, and dependency parsing.
- **GATE (General Architecture for Text Engineering)[135]:** An open-source framework for developing and deploying natural language processing applications. GATE provides a wide range of tools and resources for information extraction, including entity recognition, relation extraction, and event extraction.
- **Apache UIMA (Unstructured Information Management Architecture)[136]:** An open-source framework for developing information extraction and text analytics applications. It provides a scalable and extensible infrastructure for processing unstructured data.
- **Spacy[137]:** An open-source Python library for natural language processing. Spacy offers various components for information extraction, including named entity recognition, part-of-speech tagging, and dependency parsing.
- **MITIE (MIT Information Extraction)[138]:** A library for named entity recognition, relation extraction, and coreference resolution. It is implemented in C++ and provides Python and Java interfaces.
- **TextBlob[139]:** A Python library built on top of NLTK and Pattern. It provides a simple API for common natural language processing tasks, including information extraction.
- **OpenIE (Open Information Extraction)[140]:** A system for extracting structured information from unstructured text. It uses a bootstrapping approach to identify relations and entities in text.
- **Apache Tika[141]:** A content analysis toolkit that can extract text and metadata from various file formats. It is useful for extracting information from documents, web pages, and other sources.
- **HeidelTime[142]:** Is a multilingual, domain-sensitive temporal tagger. It extracts temporal expressions from documents and normalizes them into a standard format.

In terms of commercial services, we highlight the following:

- **Google Cloud Natural Language AI[143]:** An API service that includes including sentiment analysis, entity analysis, entity sentiment analysis, content classification, syntax analysis, and entity linking to Wikipedia.

---

[134] https://opennlp.apache.org/

[135] https://gate.ac.uk/

[136] https://uima.apache.org/

[137] https://spacy.io/

[138] https://github.com/mit-nlp/MITIE

[139] https://textblob.readthedocs.io/en/dev/

[140] https://github.com/knowitall/openie

[141] https://tika.apache.org/

[142] https://github.com/HeidelTime/heideltime

[143] https://cloud.google.com/natural-language

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **IBM Watson Discovery[144]:** A suit of services that includes include entity extraction, sentiment analysis, emotion analysis, keyword extractions, category classification, concept tagging, and allows customers to build their own datasets and train models on them.
- **OpenAI API[145]:** An API that offers access to a state-of-the-art Large Language Model (LLM) and other models suitable for specific tasks, it provides a general-purpose "text in, text out" interface and the ability to generate text embeddings, the API is suitable for a wide range of NLP tasks such as summarization, question answering, classification, and information extraction.

For a broader list of commercial products and services we suggest the Gartner Insight Engines Reviews and Ratings[146].

4.3.1.3.1 Tools from the consortium

Two tools from the SructureR collection are provided by consortium. In particular, the tool provided by Expert.ai is called Expert AI Platform Document Analysis. While the tools provided by JSI are Wikifier and the EventRegistry Relation Classifier. Below is a description of each of the tools.

4.3.1.3.1.1 Expert AI Platform Document Analysis

With the Natural Language API's document analysis capabilities, you can perform deep linguistic analysis, keyphrase extraction, named entity recognition, relation extraction and sentiment analysis. More specifically, deep linguistic analysis combines text subdivision, part-of-speech tagging, morphological analysis, lemmatization, syntactic analysis and semantic analysis.

In short, our API will read and understand your text to help you make sense of your unstructured data.

Full analysis is the sum of all partial document analyses:

- Deep linguistic analysis[147]
- Keyphrase extraction[148]
- Named entity recognition[149]
- Relation extraction[150]
- Sentiment analysis[151]

Natural Language API resources carrying out full document analysis have paths like this:

- analyze/context name/language code

Boxed parts are placeholders, so for example:

- https://nlapi.expert.ai/v2/analyze/standard/en

[144] https://www.ibm.com/products/watson-discovery
[145] https://openai.com/blog/openai-api
[146] https://www.gartner.com/reviews/market/insight-engines
[147] https://docs.expert.ai/nlapi/latest/guide/linguistic-analysis/
[148] https://docs.expert.ai/nlapi/latest/guide/keyphrase-extraction/
[149] https://docs.expert.ai/nlapi/latest/guide/entity-recognition/
[150] https://docs.expert.ai/nlapi/latest/guide/relation-extraction/
[151] https://docs.expert.ai/nlapi/latest/guide/sentiment-analysis/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

is the URL of the standard context[152] resource performing the full analysis of an English text. These resources must be requested with the POST method.

In the reference section of this manual, you will find all the information you need to perform full document analysis using the API's RESTful interface, specifically:

- The format of the request[153] to be submitted to the resources.
- How to build resources' paths and full endpoints[154].
- The output format[155].

Note

Even if you consume the API through a ready-to-use client that hides low-level requests and responses, knowing the output format helps you understand and navigate the results.

Here is an example of performing full document analysis on a short English text:

(Python)

This example code uses expertai-nlapi, the open-source Python client corresponding to the nlapi-python[156] GitHub project.

The client gets user credentials from two environment variables:

- EAI_USERNAME
- EAI_PASSWORD

Set those variables with your account credentials before running the sample program below.

The program prints the number of items for each of the output's arrays.

```
from expertai.nlapi.cloud.client import ExpertAiClient
client = ExpertAiClient()

text = "Michael Jordan was one of the best basketball players of all time. Scoring was Jordan's stand-out skill, but he still holds a defensive NBA record, with eight steals in a half."
language= 'en'

output = client.full_analysis(body={"document": {"text": text}}, params={'language': language})


# Output arrays size

print("Output arrays size:");

print("knowledge: ", len(output.knowledge))
print("paragraphs: ", len(output.paragraphs))
print("sentences: ", len(output.sentences))
print("phrases: ", len(output.phrases))
```

[152] https://docs.expert.ai/nlapi/latest/guide/contexts-and-kg/

[153] https://docs.expert.ai/nlapi/latest/reference/request-format/

[154] https://docs.expert.ai/nlapi/latest/reference/endpoints/

[155] https://docs.expert.ai/nlapi/latest/reference/output/full-analysis/

[156] https://github.com/therealexpertai/nlapi-python

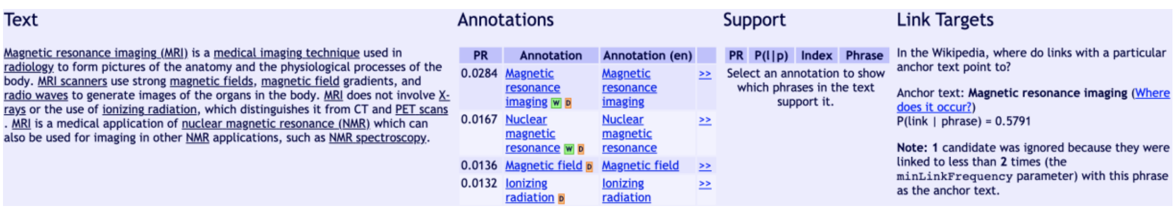
THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

```
print("tokens:                    ", len(output.tokens))
print("mainSentences:             ", len(output.main_sentences))
print("mainPhrases:               ", len(output.main_phrases))
print("mainLemmas:                ", len(output.main_lemmas))
print("mainSyncons:               ", len(output.main_syncons))
print("topics:                    ", len(output.topics))
print("entities:                  ", len(output.entities))
print("entities:                  ", len(output.relations))
print("sentiment.items: ", len(output.sentiment.items))
```

4.3.1.3.1.2 Wikifier

Wikifier is a semantic annotation service for 100 languages. Primarily it provides Entity Linking to Wikipedia (Wikification). As part of its internal workings, it also performs other tasks such as language identification, part-of-speech tagging, and tokenization. It also performs other helpful tasks such as resolving the links to Wikidata, and adding Wikidata and DBpedia classes to each entity linked.

Wikifier is implemented as a web service, available at http://wikifier.org together with the documentation and example code[157]. Here we provide an example from the web interface as well as part a service request response.




```
{
 "annotations": [
 {
 "title": "Magnetic resonance imaging",
 "url": "http://en.wikipedia.org/wiki/Magnetic_resonance_imaging",
 "lang": "en",
 "secLang": "en",
 "wikiDataItemId": "Q161238",
 "wikiDataClasses": [
 {
 "itemId": "Q2671652",
 "enLabel": "medical test"
 }
 ]
 }, …
}
```


[157] https://wikifier.org/info.html

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.3.1.3.1.3 Event Registry Relation Classifier

This service, developed by Event Registry, performs unified relation and event classification from text on a semi-open set of classes. Below is an example of a request that results in the class "product-recall".

Request:


```
{
 "text": "Seoul - South Korea said Thursday it had ordered Volkswagen to recall 125 500 diesel vehicles, after tests confirmed the German automaker had faked emissions results from the cars' engines.",
  "h": {
 "mention": "Volkswagen",
 "label": "ORG",
 "chFrom": 49,
 "chTo": 59,
 "src": "wikifier",
 "title": "Volkswagen",
 "_id": "Q246",
 "url": "http://en.wikipedia.org/wiki/Volkswagen"
 },
 "t": {
 "mention": "diesel vehicles",
 "label": "PRODUCT",
 "chFrom": 78,
 "chTo": 93,
 "src": "wikifier",
 "title": "Diesel engine",
 "_id": "Q174174",
 "url": "http://en.wikipedia.org/wiki/Diesel_engine"
 },
}
```


Reply: "product-recall"

4.3.1.4 Research & technical challenges

4.3.1.4.1 Entity Recognition and Linking

1. **Ambiguity:** Entities often have multiple interpretations or can be referred to using different names or variations. Resolving the ambiguity and correctly disambiguating entities is a significant challenge in entity linking.
2. **Knowledge Base Coverage:** The effectiveness of entity linking heavily relies on the coverage and accuracy of the underlying knowledge base or reference database. Expanding and maintaining comprehensive and up-to-date knowledge bases across different domains and languages is a constant challenge.
3. **Cross-lingual Entity Linking:** Linking entities across different languages introduces additional complexity due to variations in language structures, cultural references, and the lack of cross-lingual resources.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.3.1.4.2 Relation extraction

1. **Data Availability and Annotation:** Obtaining labelled training data for relation extraction can be challenging, especially for specific domains or fine-grained relation types.
2. **Fine-Grained Relation Extraction:** Traditional relation extraction focuses on a limited set of predefined relation types. However, there is a need for more fine-grained relation extraction that can capture a broader range of relationships with higher specificity and granularity.
3. **Cross-Domain and Cross-Lingual Relation Extraction:** Extending relation extraction models to handle different domains or languages presents challenges due to variations in language use, domain-specific terminologies, and cultural context. Developing methods that can generalize and adapt across domains and languages is an ongoing research challenge.

4.3.1.4.3 Event extraction

1. **Multilingual and Cross-Domain Event Extraction:** Extending event extraction techniques to handle multilingual text and diverse domains poses challenges due to variations in language use, cultural differences, and domain-specific terminologies.
2. **Limited Annotated Data:** The availability of large-scale annotated datasets for training event extraction models is limited, especially for specific domains or fine-grained event types. Developing large-scale annotated datasets with labelled arguments and their attributes is a time-consuming and resource-intensive task.
3. **Efficiency:** Efficient algorithms or pipelines, especially for argument extraction, that can handle the large, big data in an efficient way remains an open problem.

4.3.1.4.4 Temporal information extraction

1. **Multilingual Temporal Information Extraction:** Extending temporal extraction techniques to handle multilingual text presents challenges due to language-specific temporal expressions, cultural differences, and variations in date formats.
2. **Noise and Contextual disambiguation:** Temporal information, may contain errors, inconsistencies, or incomplete expressions. Contextual disambiguation (e.g., “Good Friday Agreement” vs “(a) Good Friday)” can also result in noise. Resolving temporal expressions' ambiguity and correctly assigning the appropriate time reference remains a challenge.

4.3.2 Classification (ClassifieR)

4.3.2.1 Scope and example

Document classification, a fundamental task in NLP, involves categorizing text documents into predefined classes or categories based on their content. The goal is to automatically assign the most appropriate category or label to each document, enabling efficient organization, retrieval, and analysis of large text corpora. The most relevant types of document classification, to our context, include topic classification, news categorization, sentiment analysis, emotion classification, and intent prediction.

Document classification can be divided into the following:

1. **Multiclass Classification:** In multiclass classification, each document is assigned to one and only one class or category. The goal is to accurately assign each document to a single predefined class label from a set of multiple mutually exclusive classes. For example, if there are three classes (A, B, C), a multiclass classifier would assign each document to one of these three classes.
2. **Multilabel Classification:** In multilabel classification, each document can be assigned to multiple class labels simultaneously. Instead of being limited to a single class label, a document may belong

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

to multiple categories or have multiple attributes. The classifier assigns a binary label to each class, indicating whether the document belongs to that class or not. This allows for more flexibility and captures the possibility of documents having multiple topics or attributes. For instance, a document might be labelled as belonging to both "Sports" and "Entertainment" categories.

3. **Hierarchical Classification:** Hierarchical classification involves organizing classes or categories in a hierarchical or tree-like structure. Instead of directly assigning documents to specific classes, the classifier operates in a hierarchical manner, making decisions at different levels of the hierarchy. Each class is organized into parent and child relationships, where the child classes represent specific subcategories or attributes of the parent classes. This approach allows for a more structured and granular classification scheme. For example, in a hierarchical classification system for news articles, the top-level classes could be "Sports," "Politics," and "Entertainment," with further subcategories such as "Football," "Basketball," "Elections," "Legislation," "Movies," and so on.

In summary, multiclass classification assigns each document to a single class label, multilabel classification allows for multiple class labels per document, and hierarchical classification organizes classes in a hierarchical structure to provide a more structured and granular classification scheme.

4.3.2.2 Research

The state of the art in document classification has moved to pretrained Large Language Model Transformer based Neural Networks after the introduction of BERT. That said, depending on the specific task, classical machine learning approaches such as Support Vector Machines (SVMs) (Cortes, 1995) and Naïve Bayes classifiers, remain competitive or at least represent a valid trade-off in speed versus accuracy. With respect to multilingual data, we highlight XLM-Roberta (Conneau, 2020) and mT5 (Xue, 2021).

Document classification commonly relies on supervised datasets, created to fit a particular taxonomy. These datasets match each document to a label or set of labels. Very often, creating these datasets and the associated taxonomy is the hardest part of the problem. To solve this challenge, tools that rely on machine learning techniques have been proposed over the years. One example of such a tool was the original Ontogen (Fortuna, 2005), the latest derivative of which is InfoMiner[158]. These rely on unsupervised clustering algorithms such as k-means (Lloyd, 1982; MacQueen, 1967) and k-means++ (Arthur, 2007) to organize documents into groups by similarity and on few-shot Active Learning, with techniques such as classifier uncertainty sampling (Lewis, 1994), to simultaneously build the taxonomy and the dataset from a pool of unlabeled documents. More recently, Natural Language Inference based Zero-Shot (Yin, 2019). and Few-Shot (Pasunuru, 2021) classification methods based on Natural Language Inference (Dagan, 2006) have also shown potential to help create classifiers and datasets.

4.3.2.3 Tools and market

Most of the tools providing text classification features also provide information extraction features. We therefore refer to the tool list discussed in Section 4.3.1.3 also for text classification.

4.3.2.3.1 Tools from the consortium

4.3.2.3.1.1 Expert AI Platform Document Classification

Analyse text to label and identify media topics, emotional traits, geographical references, and more.

Document classification determines what a text is about in terms of categories of a taxonomy.

---

[158] https://github.com/Infominer-JSI

Available taxonomies are:

Table 6: Available taxonomies in Expert AI Platform Document Classification

Taxonomy	English	Spanish	French	German	Italian
iptc	✓	✓	✓	✓	✓
geotax	✓	✓	✓	✓	✓
emotional-traits	✓			✓	
behavioral-traits	✓			✓	

In the Natural Language API terminology, taxonomy "x" is both a specific set of categories and the name of the API resources capable of classifying a text according to that set.

Taxonomies' resources have paths like this:

- categorize/taxonomy name/language code

Boxed parts are placeholders, so for example:

- https://nlapi.expert.ai/v2/categorize/iptc/en

is the URL of the iptc resource capable of performing the IPTC Media Topics[159] classification of English texts.
These resources must be requested with the POST method, submitting the text to classify.

In the reference section of this manual, you will find all the information you need to perform document classification using the API's RESTful interface, specifically:

- The format of the request[160] to be submitted to the resources.
- How to build resources' paths and full endpoints[161].
- The output format[162].

Note

Even if you consume the API through a ready-to-use client that hides low-level requests and responses, knowing the output format helps you understand and navigate the results.

Here is an example of performing IPTC Media Topics classification of a short English text:

(Python)

This example code uses expertai-nlapi, the open-source Python client corresponding to the nlapi-python[163] GitHub project.

The client gets user credentials from two environment variables:

- EAI_USERNAME

---

[159] https://docs.expert.ai/nlapi/latest/guide/classification/iptc-media-topics/

[160] https://docs.expert.ai/nlapi/latest/reference/request-format/

[161] https://docs.expert.ai/nlapi/latest/reference/endpoints/

[162] https://docs.expert.ai/nlapi/latest/reference/output/classification/

[163] https://github.com/therealexpertai/nlapi-python

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- EAI_PASSWORD

Set those variables with your account credentials before running the sample program below.

For each output category, the program prints the ID and the hierarchy.

```
from expertai.nlapi.cloud.client import ExpertAiClient
client = ExpertAiClient()

text = "Michael Jordan was one of the best basketball players of all time. Scoring was Jordan's stand-out skill, but
he still holds a defensive NBA record, with eight steals in a half."
taxonomy = 'iptc'
language= 'en'

output = client.classification(body={"document": {"text": text}}, params={'taxonomy': taxonomy, 'language':
language})

print("Tab separated list of categories:")

for category in output.categories:
    print(category.id_, category.hierarchy, sep="\t")
```

The following articles describe the capabilities of the available taxonomies.

Self-documentation resources

The API provides self-documentation resources to programmatically discover available taxonomies and their features. Learn more about these resources in the dedicated article[164].

4.3.2.3.1.2 InfoMiner

InfoMiner, an offshoot of the original Ontogen, provides a modern web user interface with useful visualizations underpinned by data analysis and machine learning algorithms with the objective of rapidly constructing labelled datasets, their taxonomies, and classifiers.

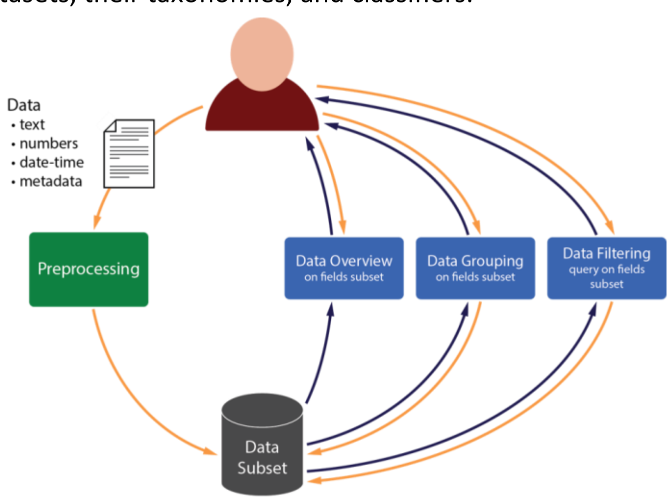


Figure 18: InfoMiner

Data Grouping features allow the user to quickly identify similar documents, with user defined metrics. This is supported by automatic methods such as clustering.

[164] https://docs.expert.ai/nlapi/latest/guide/classification/taxonomies-info/

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE
RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

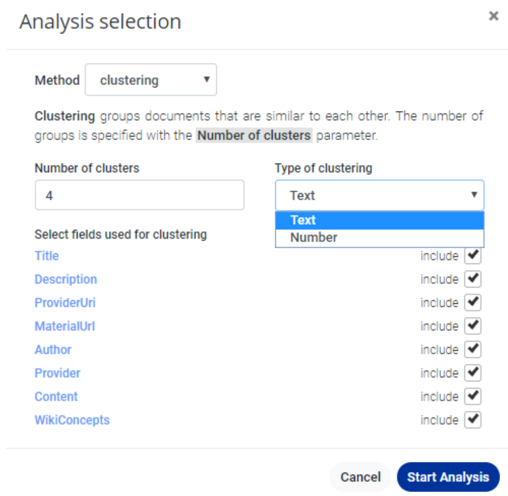


Figure 19: Infominer – Data Grouping

Smart Visualization techniques allow the user to quickly understand the data. InfoMiner uses centroid based methods to summarize each cluster. It also automatically creates visualizations such as word clouds, treemaps, and timelines.

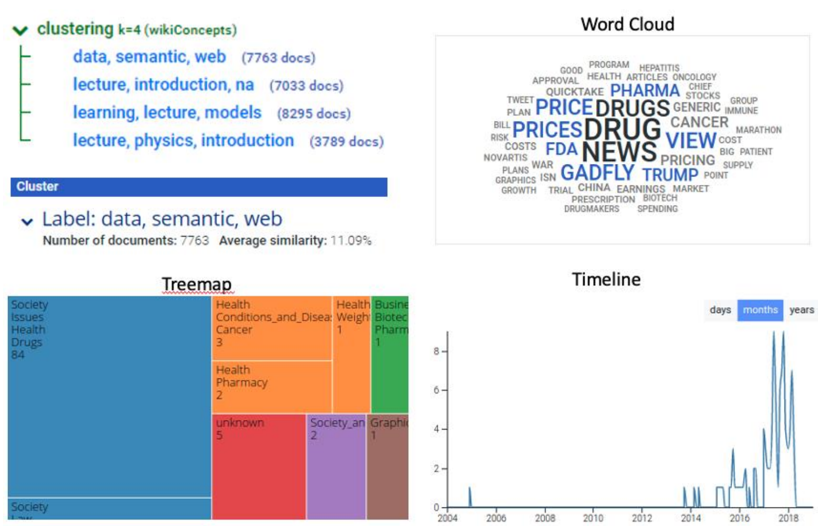


Figure 20: InfoMiner – Smart Visualisation

Data Filtering allows the user to query data over its set of properties, the most important of which typically being its textual content or metadata.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

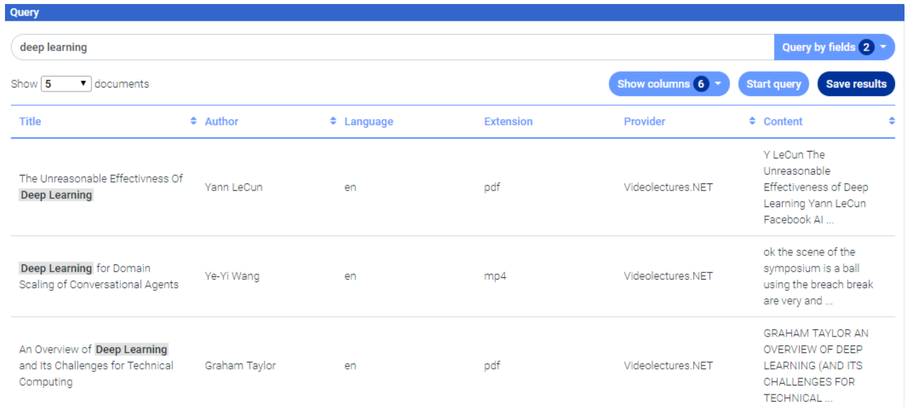


Figure 21: InfoMiner – Data Filtering

Taxonomy creation allows the user to use all the previously mentioned methods (analysis, grouping, filtering) to create a taxonomy and navigate.

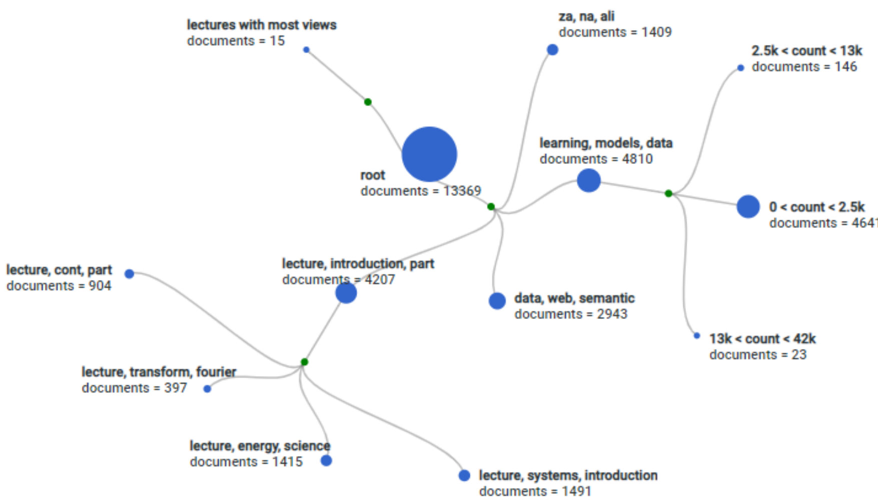


Figure 22: InfoMiner – Taxonomy

Dataset and Classifier based on the taxonomy typically begin from grouped documents. Active Learning based techniques can be used to further refined the grouped documents into a labelled dataset. A classifier can be automatically created for any taxonomy and used directly via a REST API to classify new documents.

4.3.2.4 Research & technical challenges

The primary challenges in document classification research we want to highlight are:

1. **Insufficient and Imbalanced Data:** Obtaining a large, diverse, and balanced dataset with sufficient annotated examples for each class can be a challenge, especially for niche domains or fine-grained classification tasks. Data scarcity and class imbalance can affect the performance and generalizability of classification models.
2. **Domain Adaptation:** Classification models trained on one domain may not generalize well to different domains or new, unseen data. Adapting models to handle domain shift or developing domain adaptation techniques to improve transferability is a research challenge.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

3. **Multilingual and Cross-lingual Classification:** Extending document classification techniques to handle multilingual text or cross-lingual scenarios poses challenges due to language differences, cultural variations, and the availability of labelled data in different languages. Developing methods that can effectively generalize and adapt across languages is an ongoing research challenge.
4. **Label Noise, Annotation Consistency and Quality:** Ensuring consistent and high-quality annotations across the dataset is crucial. In automatic or semi-automatic dataset building it is important to automatically detect likely errors. When dealing with human annotators, mistakes happen too and different annotators may have variations in labelling criteria or interpretations, leading to inconsistencies. Providing automatic feedback to human annotators is also important.
5. **Multi-label and Hierarchical Classification:** Building datasets for multi-label or hierarchical document classification requires careful annotation design and consideration of relationships between labels. Defining clear guidelines for assigning multiple labels to documents or capturing hierarchical relationships adds complexity to dataset construction.

4.4 Other relevant aspects for data enrichment pipelines

4.4.1 Scaling, replication and adaptation (ScalR and ReusR)

4.4.1.1 Scope and example

Executing cleaning, transformation, and linking at large scale requires infrastructural components that allow for scalability. A definition of scalability is given by [Herbst et al., 2013], which states that “scalability is the ability of a system to sustain increasing workloads by making use of additional resources”. The implementation of a system with this characteristic is an essential step in a big data pipeline to avoid common performance bottlenecks. Usual issues arise in the following three areas:

- **CPU usage:** is the most common bottleneck. This issue happens when the pipeline works correctly but the processing power of the CPU is not enough to handle the whole process.
- **Memory usage:** the server does not have enough memory to organize the pipeline flow. This issue can also indicate a memory leak in the process.
- **Disk usage:** this happens when the volume of disk space is fully occupied by the processed data.

Another common reason for implementing a scalable process is the flexibility of the new infrastructure. It allows one to change the priorities of the process, for example, by focusing more on one step rather than another, without losing the initial investment. In addition, a scalable process is future proof for an eventual resize in the future.

After analysing the need for a scalable process, there are two main ways of scaling:

- **Scaling up, or vertical scaling:** means using more powerful hardware and more memory. This method offers the best performance, since everything works on the same machine. A possible limitation could be related to the speed of growth of the process; for a fast process, it represents just a short-term solution, and frequent updates became more and more expensive due to hardware limitations.
- **Scaling out, or horizontal scaling:** means adding new power across the infrastructure and not in the same machine. This solution uses parallel computing to increase the performance of the infrastructure and is valid also in the long term. At the same time, moving from a single machine to a distributed system leads to lower speed and higher complexity.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

The main goal of the ScalR is to provide horizontal scalability of data enrichment pipelines using software containers, and support for management of the different procedures associated with the execution of data enrichment pipelines flexibly on heterogeneous computing infrastructures.

The goal can be achieved by promoting the reuse and modification of existing data enrichment pipelines by exposing them as an integrated deployable unit, as opposed to ad-hoc, non-reusable pieces of code. For this reason, most of the tools that handle scalability can also be considered data orchestrator tools, since, in general, it is usually more convenient to use an automatic tool to orchestrate a pipeline, rather than combine different scripts. This option allows the user to use different languages, like batch or Python files, and use them in the same pipeline with no difficulties. A data orchestrator can receive as input different types of files, handle the scalability, create and schedule the progress of the entire data pipeline. Other advantages of the use of a data orchestrator are the good readability of the data flow, a limited space for error since a great number of activities are managed automatically, and better data quality visibility.

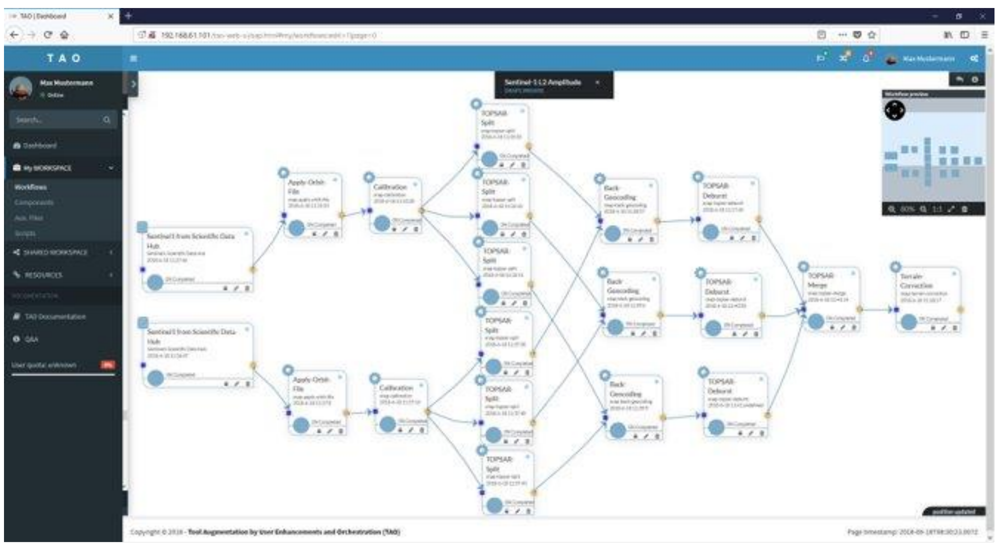


Figure 23: Example of existing workflow using TAO interface

To improve the reuse across use cases of the described pipelines, the goal of the ReusR is to provide the possibility to reuse the existing processing components to build new workflows or to reuse the existing workflows in other business cases (with or without other adaptations, depending on the needs).

4.4.1.2 Research

Given the analysed data scaling problems, different directions of research exist. The main topics have been included in the list below after being selected via Google Scholar, looking at the most recent and relevant publications. These areas can be divided into the following categories:

- **Benchmarking scalability:** definition of universally accepted metrics to create a general procedure for comparison of data scaling tools.
- **Scalability architecture:** study of possible evolution of the standard monolithic architecture of data scaling tools using cloud and containers.
- **Workflow architecture:** focus on the organization and scheduling of data scaling activities using workflow management tools.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.4.1.2.1 Benchmarking scalability

Benchmarking scalability refers to the process of establishing universally accepted metrics and procedures for comparing the performance and efficiency of data scaling tools or systems. The goal is to define a standardized framework that allows for objective evaluation and comparison of different tools or techniques used for scaling and processing large volumes of data.

The primary objective of benchmarking scalability is to provide a fair and consistent basis for assessing the capabilities of data scaling solutions across various dimensions, such as processing speed, resource utilization, scalability, and performance under different workloads or data sizes. By defining commonly agreed-upon metrics and methodologies, benchmarking scalability enables researchers, developers, and users to make informed decisions when selecting or comparing different data scaling tools.

In this area of research, the work from [Henning et al., 2022] proposes a configurable method for benchmarking the scalability of cloud-native applications. The study acknowledges the significance of scalability in cloud-native architectures but highlights the lack of a universally accepted benchmarking approach. To address this gap, the researchers emphasize the importance of metrics that focus on resource demand and load capacity. The method incorporates various search algorithms and Service Level Objectives (SLOs) to enable comprehensive scalability evaluation of cloud-native applications.

In [Henning et al., 2021], the authors delve into measuring the scalability of distributed stream processing engines. They define metrics specifically tailored for comparing scalable architectures, with a particular emphasis on distributed stream processing. The study introduces two primary metrics: the demand metric, which characterizes the evolution of resource demands as the load increases, and the capacity metric, which describes how the load capacity evolves with the provision of additional resources. These metrics serve as valuable indicators for assessing the scalability of distributed stream processing engines and facilitating informed comparisons between different architectures.

4.4.1.2.2 Scalability architecture

Scalability architecture refers to the exploration and analysis of potential advancements in the traditional monolithic architecture of data scaling tools through the utilization of cloud computing and containerization technologies. It involves studying the ways in which data scaling tools can evolve to handle increasing workloads and growing data volumes effectively. By leveraging cloud services and containerization, scalability architecture aims to enhance the scalability, flexibility, and efficiency of data scaling tools by enabling them to dynamically allocate resources, scale horizontally, and distribute workloads across multiple instances or nodes. This approach allows organizations to meet the demands of ever-growing data processing requirements and achieve optimal performance and resource utilization in a scalable and cost-effective manner.

In their study, [Blinowski et al., 2022] provide an insightful evaluation of the performance and scalability of monolithic and microservice architectures. Through experimental scenarios, they observe that performance tends to degrade when scaling out beyond a certain number of instances. Interestingly, they find that there is no clear winner between the two architectures in every context. While scalable monolithic infrastructure remains the preferable choice for small-sized systems, the evaluation highlights the need to consider various factors when determining the optimal architecture for scalability.

Another important aspect of scalability in cloud architectures is network performance variability, as addressed by [De Sensi et al., 2022]. Their work defines a methodology to evaluate the impact of network noise on application scalability across different cloud solutions and varying instance numbers. By simulating potential scalability performance losses, this methodology enables users to assess the

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

influence of network performance on overall system scalability, providing valuable insights for designing and optimizing cloud architectures.

Lastly, in the context of executing large-scale data workflows, [Dessalk et al., 2020] present an innovative approach that utilizes software container technologies and message-oriented middleware (MOM). Their solution enables highly scalable workflow execution by leveraging the benefits of containerization. The authors compare their approach with the widely used Argo Workflow tool, showcasing the advantages and scalability achieved through the implementation of software containers in executing big data workflows. This research contributes to the exploration of scalable workflow execution techniques, offering insights into the effectiveness of software containers in handling large-scale data processing tasks.

4.4.1.2.3 Workflow architecture

Workflow architecture refers to the design and structure of the organization and scheduling of data scaling activities using workflow management tools. It involves defining the sequence of tasks, their dependencies, and the overall flow of data and operations within a data scaling process. The workflow architecture aims to optimize and streamline the execution of tasks, ensuring efficient data processing and scalability. By utilizing workflow management tools, such as workflow engines or orchestration frameworks, the workflow architecture provides a systematic approach to automate and manage complex data scaling activities. It enables the coordination and synchronization of tasks, facilitates error handling and recovery, and provides monitoring and control capabilities throughout the data scaling process. Overall, the workflow architecture plays a crucial role in orchestrating and managing the execution of data scaling activities, improving efficiency and scalability in data processing workflows.

In their work, [Nikolov et al., 2021] present a conceptualization and scalable execution approach for big data workflows using domain-specific languages (DSL) and software containers. This approach enables the definition of complex data processing activities through a DSL while ensuring scalability by leveraging container-based technology and message-oriented middleware solutions. By combining these techniques, the authors provide a robust framework for handling large-scale data workflows efficiently.

Additionally, [Corodescu et al., 2021] propose an architecture for software container-centric big data workflow orchestration that emphasizes data locality. Their solution takes into account data locality information, utilizes long-lived containers for executing workflow steps, and effectively manages interactions with diverse data sources through containers. By prioritizing data locality, the proposed architecture aims to optimize performance and efficiency in big data workflows.

In the study conducted by [Mitchell et al., 2019], an exploration of emerging features in workflow management systems (WMSs) from user perspectives is presented. The focus is on data-oriented WMSs that are surpassing traditional task-oriented WMSs. These emerging tools aim to integrate container-based and cloud solutions, catering to various use cases such as machine learning and big data analytics. The study provides insights into the evolving landscape of workflow management systems and the increasing demand for data-driven workflows across different domains.

These works contribute to the advancement of workflow architecture by incorporating domain-specific languages, software containers, and data locality considerations. By leveraging these techniques, researchers and practitioners can develop scalable and efficient solutions for managing and executing big data workflows, addressing the challenges of data processing and analytics in complex environments.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.4.1.3 Tools and market

The table below contains an analysis of the existing tools on the market. These have been selected via Google Search and by looking at the products offered by the main players in the field, like Apache and Google Cloud. The comparison has been done according to the following classifiers:

- Description
- Pros
- Cons
- Reuse of components

Table 7: Overview of the most relevant scaling tools available in the market

Tool	Description	Pros	Cons	Reuse of components
Apache Airflow https://airflow.apache.org/	Open-source platform to schedule and monitor workflows, originally developed by Airbnb	Dependencies management using DAGs, Scheduler, Great UI, Current market leader	Difficult to scale, Not easy to pass data between steps, No ETL functionality	Yes
Argo Workflows https://argoproj.github.io/argo-workflows/	Open-source container native workflow engine for orchestrating parallel jobs on Kubernetes	Parallel workflows, Used for ETL, Scheduler	Requires YAML format. Difficult to integrate third-party operators	Yes
Dagster https://dagster.io/	Cloud-native open-source orchestrator for the whole development lifecycle	Easy to setup, Good UI, High level of testability	Transformation oriented, Limited support to external services	Yes
Prefect https://www.prefect.io/	Cloud-native open-source orchestrator for the whole development lifecycle	Easy data transfer, Use of dynamic workflows	Limited UI, Limited support to external services	Yes
Luigi https://luigi.readthedocs.io/en/stable/index.html	Developed by Spotify, Python package for pipeline of batch jobs	Heavier code than Airflow	Limited logs, Based on input-output concept, Limited scheduling options	Yes
DBT https://www.getdbt.com/	Transformation workflow for building data pipeline in a CI/CD manner	Good UI, Fast orchestrator	Limitation to SQL, Covers only the T of ETL procedure	Yes
Flyte https://flyte.org/	Workflow automation platform for complex mission-	Environment and dependency isolation, Use of DAGs	High maintenance cost, Difficult to configure	Yes

	critical data and ML processes at scale			
Apache NiFi https://nifi.apache.org/	Apache system to process and distribute data using scalable DAGs	Low code service, ETL functionalities, Visual interface	Memory intensive, Focus on stream processing	Yes
AWS Step Functions https://aws.amazon.com/step-functions/	Amazon orchestrator for distributed applications using visual workflows	Cloud, Handle scalability, High availability	Max request size limit of 256KB Requires workflows with the proprietary Amazon States Language	No
Azure Data Factory https://azure.microsoft.com/en-us/products/data-factory	Data pipeline orchestrator and ETL tool in the Microsoft Azure cloud ecosystem	Cloud, Good UI, Handle scalability	Focus on ETL, Limited to the Azure environment	Yes
Google Cloud Composer https://cloud.google.com/composer	Google's fully managed workflow orchestration solution built on Apache Airflow	Cloud, Quick to start and scale, Easy to use, No vendor lock-in	Pay-for-use, Limits to the number of services and integrations supported	Yes
TAO https://www.c-s.ro/industries/space/earth-observation-2/tool-augmentation-by-user-enhancements-and-orchestration-tao/	TAO (Tool Augmentation by user enhancements and Orchestration) is an open source, lightweight, generic, extensible and distributed orchestration framework. Processing workflows can be defined, processed and distributed by the final user	Handle scalability, Easy integration with additional processing modules, Visual interface	Limited monitor dashboard	Yes

4.4.1.3.1 Tools from the consortium

4.4.1.3.1.1 TAO

A tool from the ScalR collection is provided by CS GROUP–ROMANIA: **TAO,** which stands for Tool Augmentation by user enhancements and Orchestration.

TAO is an open source, lightweight, generic, extensible, and distributed orchestration framework. It allows to reuse (i.e., integrate) commonly used toolboxes (such as, but not limited to some EarthObservation processing tools like SNAP, Orfeo Toolbox, GDAL, PolSARPro, etc.). This framework allows for processing composition and distribution in such a way that end users could define processing workflows by themselves and easily integrate additional processing modules, without any programming knowledge requirements.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

TAO platform provides a means for orchestrating heterogeneous processing components and libraries to process scientific data. This is achieved in the following steps:

- Preparation of resources (including processing components or tools) and data input.
- Definition of a workflow as a processing chain.
- Execution of workflows, scaling up from one to as many nodes as made available.
- Retrieval and visualization of the results, which allows one to see what is executing in the system and where it is executing, as well as the system resource usage (CPU, memory, storage space).

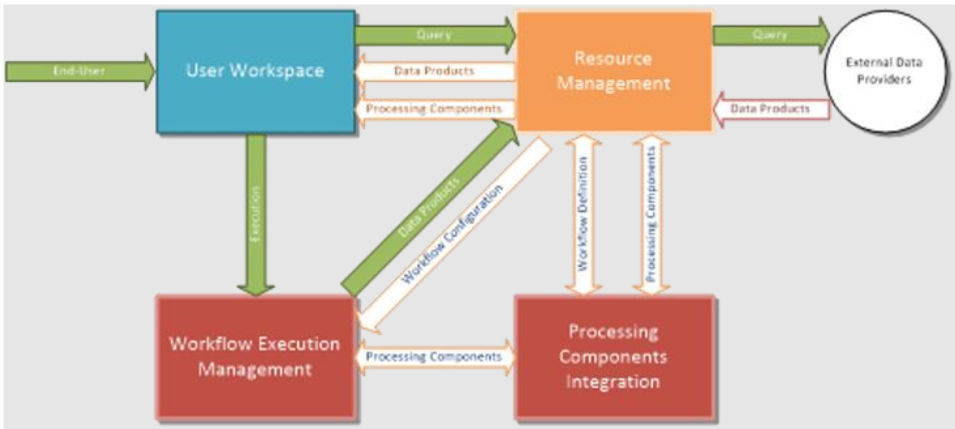


Figure 24: Architecture of TAO tool

The processing component, shown in the figure above, represents a standalone application (or tool) defined by the following parameters:

- **Input description:** type of data that the component accepts as an input source (e.g., CSV, JSON, etc.).

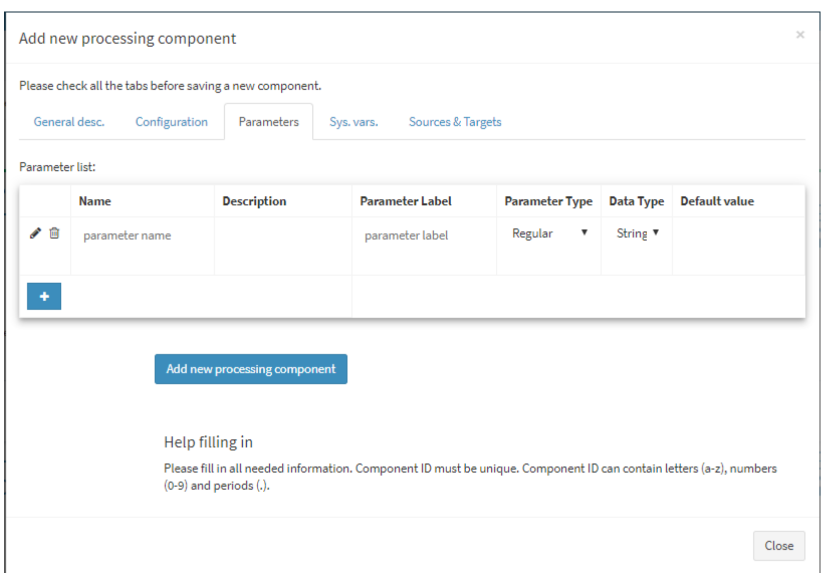


Figure 25: TAO Processing component parameters definition

- **Processing operation with execution parameters:** the operation that the component executes with the list of accepted parameters,

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

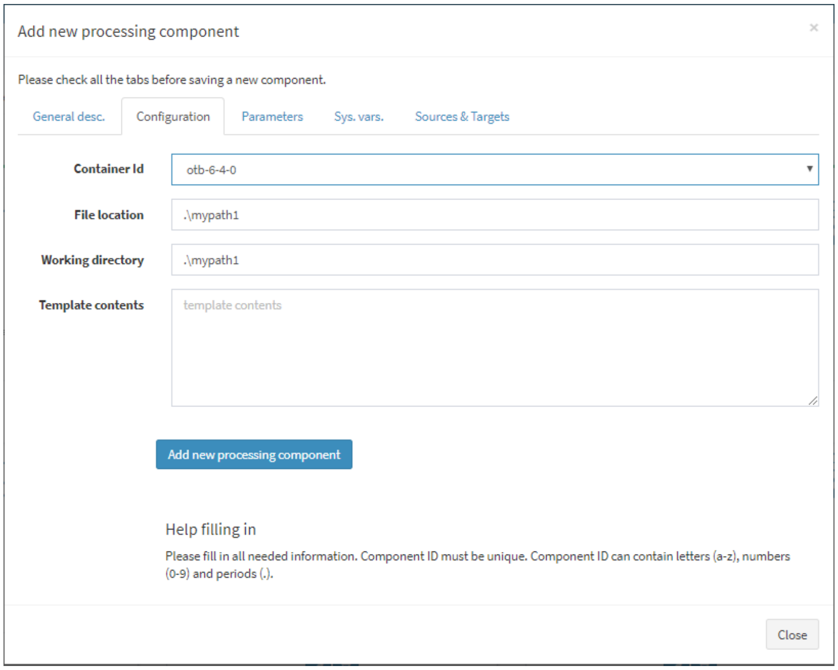


Figure 26: TAO Processing component configuration

- **Output description:** type of data provided as a processing operation result. Can consist of one or more files.

From the TAO web interface, the user has the possibility to edit a workflow by adding the desired data source to be used as the initial data provider in the workflow and also to chain the desired processing modules. The chaining of the modules is performed by the user by drag and dropping the available processing components on the drawing canvas and chain them using directed connectors.

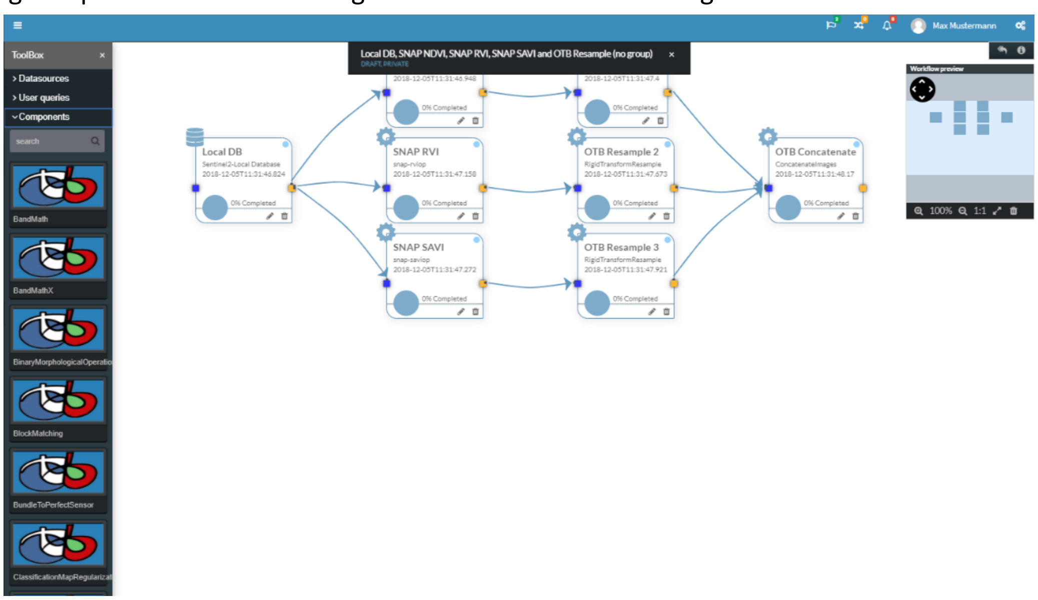


Figure 27: TAO workflow design

Another relevant component is the DRMAA (Distributed Resource Management Application API), which provides a standardized access to the DRM systems for execution resources. It is focused on job

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

submission, job control, reservation management, retrieval of jobs, and machine monitoring information. Currently, there are supported DRMAA implementations for local, remote SSH, Torque and SLURM executions. Support for Kubernetes and CWL (Common Workflow Language) implementation is also in progress. The DRMAA implementations are provided as plugins that allow a high flexibility as the current implementation can be changed easily with another.

The TAO framework allows the following deployment modes:

- **Standalone** mode where only a set of core components are deployed allowing execution of orchestrated workflows from an external toolbox that does not possess orchestration. The standalone mode can be assimilated also with the service mode of the TAO platform. In this mode, all the executions will take place on the local machine where TAO is installed.
- **Platform** mode where the full platform is deployed, allowing users management, resource and catalogue and distributed execution on multiple nodes. In this mode, there will exist administrator users that will perform the configuration of the nodes and will set add accounts or set quotas for the users. The regular users will have the possibility to create, manage and execute their own workflows and also to visualize the data produced.

The TAO monitoring dashboard is the first page that is displayed to the user after the login and within this view are available the following information:

- The recent notification messages.
- Information about the status of the TAO master node like the last start time, the number of processors of the master node, the memory used and a graphic about the processor and memory usage.
- A summary information about the load of each processing node defined in TAO.
- A list of workflows executing in this moment and the percentage of completion.

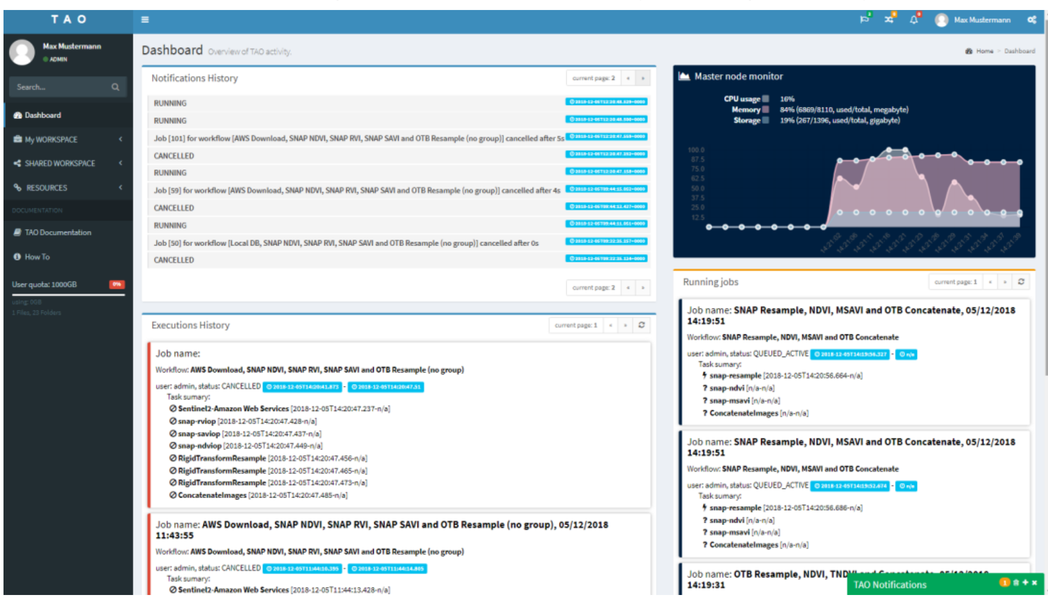


Figure 28: TAO Dashboard

4.4.1.4 Research & technical challenges

4.4.1.4.1 Definition of a general scalability benchmarking methodology

As seen in the previous section, there is an area of research that is active in the development of a universal metric to be used as a benchmarking methodology. A need for this kind of metric is real since, at the moment, it is difficult to evaluate the differences between different tools without resorting to trial and error procedures or simulations.

The papers reported above can be considered a starting point for the definition of this kind of tool. A challenge for the future is the integration of other parameters, like the difference in the architecture of a cloud system or the consumption of memory or computational power for different solutions, to obtain a wider comparison.

4.4.1.4.2 Definition of a data workflow management system for container-based system

In the market, different workflow management systems exist, and more recently they have focused on the integration of cloud and container technologies. The use of these technologies allows the use of non-physical resources, theoretically without limitation in scaling up.

The diffusion of this container-based system could lead to new workflows for data pipelines, to exploit the high availability and scalability guaranteed by the cloud. Actual tools are good at integrating cloud technologies into traditional workflows, often with limitations on third party applications, while different structures could be possible. This new area of research could also overcome some of the actual challenges in the field, like network noise that limits the potential scalability.

4.4.2 Streaming data (StreamR)

4.4.2.1 Scope and example

There are two primary types of streaming data considered in this section: time series in the industrial application context and text.

Multivariate time series refers to a type of data in which multiple variables are recorded over time in an industrial context. In industrial applications, various sensors and devices continuously capture data from multiple sources, such as temperature, pressure, humidity, vibration, and energy consumption. This data is recorded at regular intervals, creating multivariate time series datasets. Analysing multivariate time series data in industrial applications is crucial for monitoring, predictive maintenance, process optimization, and anomaly detection. It enables the identification of patterns, trends, and correlations among the different variables over time, providing valuable insights into the behaviour and performance of industrial systems. Analysing a collection of time series data to observe a complex system is a challenging task. Artificial intelligence provides us with tools to analyse such data, which may be difficult for humans to comprehend in its raw form.

Streaming text data analysis refers to the process of extracting valuable insights, patterns, and trends from continuous and real-time text data streams. It involves applying various techniques and algorithms to analyse the textual content as it arrives, enabling timely decision-making and response. The analysis of streaming text data typically involves the following steps:

1. **Data Collection:** Streaming text data is collected from various sources, such as social media platforms and news feeds. The data is continuously retrieved.
2. **Text Classification and Categorization:** Classification algorithms are applied to categorize the streaming text data into predefined or dynamically evolving classes or topics. This can involve techniques like text classification, topic modelling, or sentiment analysis.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

3. **Named Entity Recognition and Linking:** Extracting named entities, such as person names, locations, organizations, or other specific entities, from the streaming text data is important for understanding the context and identifying relevant information.
4. **Event Detection:** Identifying events or significant occurrences in the streaming text data is crucial for real-time monitoring and situational awareness. Event detection techniques help to identify important events, trends, or anomalies.
5. **Topic Tracking and Trend Analysis:** Tracking topics or trends in the streaming text data helps to understand the changing interests, emerging discussions, or shifts in public opinion. Techniques like topic modelling, trend detection, or time series analysis can be applied for this purpose.
6. **Visualization and Dashboards:** Visualizing the streaming text data analysis results through interactive dashboards or visual representations helps in understanding and exploring the data in real-time. Visualizations can include word clouds, sentiment heatmaps, temporal graphs, or network graphs.

4.4.2.2 Research

As a fully integrated solution, Streamstory [Stopar, 2018] is unique as it encompasses time series clustering, state modelling, visualization, statistical analysis, and predictive analytics. Time series clustering is a methodology with broad applications. For a broad survey see [Alqahtani, 2021]. For applications of clustering and classification of time series related to data visualization see [Ali, 2019]. For an overview of time series clustering in machine state detection, which is the main use of clustering within Streamstory, see [Henning, 2020] while [Guillaume, 2020] shows a similar methodology for an application like Streamstory's original purpose - predictive maintenance. [Javed, 2020] provides a comparison of different time series clustering algorithms, including K-means, which is used in Streamstory. We also want to highlight the use of time series clustering in creating efficient nearest neighbour search systems such as in [Vanderkam, 2013] which can efficiently find correlated series among a very large set of time series. With regards to other work modelling time series state using Markov chains in industrial applications, we can point to [Zang, 2018]. Another important set of methods with a long history of use in the context of sensor-derived and industrial time series are Neural Networks. Here we highlight the work [Sadaei, 2019] in short term energy use forecasting, [Wang, 2022] in anomaly detection in manufacturing, and the more generic work of [Sen, 2019] which nevertheless is benchmarked against time series from electricity use and transportation.

In terms of text, let's go here into clustering / sentence transformers, multilingual and into deep passage ranking.

4.4.2.3 Tools and market

The following tables give an overview of the tools available in the market as either strictly commercial software, services, or open-source tools. The tables below list tools designed to handle streaming data and provide a summary of pros and cons.

Table 8: Data streaming management tools

Tool	Description	Type	Pros	Cons	Open source
Amazon Kinesis https://aws.amazon.com/kinesis/	Amazon Kinesis Data Streams enables you to build your own custom applications that process or	Real-time Data Processing	High throughput	Limited to the AWS environment	N

	analyse streaming data for specialized needs		High scalability	Some latency in data processing	
Amazon SQS https://aws.amazon.com/sqs/	Amazon Simple Queue Service is a message queuing service that decouples the components of a cloud application and transmits any volume of data	Message Queue	Easy to use High reliability	Limited to the AWS environment Some latency in data processing	N
Apache Beam https://beam.apache.org/	A unified programming model for batch and streaming data processing, using parallel processing data pipelines	Big Data Management	Portability across different platforms Easy API for configuration	Not easy to start working with Limited documentation	Y
Apache Flink https://flink.apache.org/	An open source system for fast and versatile data analytics in clusters. Flink supports both batch and streaming analytics	Big Data Management	High scalability Allows SQL on streams Exactly-once state consistency	Not easy to configure Recent framework Supports only Java and Scala	Y
Apache Flume https://flume.apache.org/	A service designed to efficiently collect, aggregate, and move large amounts of log data	Log Management	Simple architecture High reliability	Limited scalability Limited compatibility with external services	Y
Apache Kafka https://kafka.apache.org/	Stream processing platform. It is a distributed, partitioned, and replicated commit log service that provides the functionality of a messaging system	Message Queue	High scalability High durability Good fault tolerance	Not easy to configure Lacks a monitoring to ol	Y
Apache Pulsar https://pulsar.apache.org/	A distributed messaging solution developed and released to open source at Yahoo. Pulsar supports both pub-sub messaging and queuing in a platform	Message Queue	High scalability Easy to use High throughput	Recent framework Not easy to configure	Y
Apache Samza https://samza.apache.org/	A distributed streaming machine learning framework that contains a programming abstraction for distributed streaming ML algorithms without directly dealing with the complexity of the underlying distributed stream processing engines	Message Queue	Easy to use High performance	Limited to Apache and Java environments	Y
Apache Spark Streaming https://spark.apache.org/	Spark Streaming brings Apache Spark's language integrated API to	Big Data Management	High performance	Not easy to configure	Y

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

	stream processing, letting the user writes streaming jobs the same way he writes batch jobs		High throughput Integration of different data sources	Resource intensive	
Apache Storm https://storm.apache.org/	A distributed real-time computation system that makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing	Real-time Data Processing	High reliability Compatible with any programming language	Not easy to configure Resource intensive	Y
AtScale https://www.atscale.com/	A data warehouse virtualization solution that creates a live connection between people and data without moving it, regardless of whether it is on-premise or in the cloud, turning the data warehouse into a multidimensional analytics engine.	Big Data Management	Integration of different data sources Good UI	Costly solution Dependent on Apache Hadoop	Y
Azure Stream Analytics https://azure.microsoft.com/en-us/products/stream-analytics	An event processing engine by Microsoft for real-time analytics on multiple streams of data	Big Data Management	SQL-based for fast processing High volume of data	Some latency in data processing Limited documentation	Y
Confluent Platform https://developer.confluent.io/	A full-scale data streaming platform that enables the user to easily access, store, and manage data as continuous, real-time streams, built by the creators of Apache Kafka	Message Queue	High scalability High durability Good fault tolerance	Limited documentation Lack user support	N
EsperTech https://www.espertech.com/	Esper is a language, compiler and runtime for complex event processing (CEP) and streaming analytics, available for Java as well as for .NET	Real-time Data Processing	Low latency High throughput	Limited documentation Lack user support	Y
Google Cloud Dataflow https://cloud.google.com/dataflow	A data processing service built on Apache Beam for transforming and enriching streaming and batch data	Real-time Data Processing	Autoscaling High consistency	Quotas limitations Expensive solution	N
Google Cloud Pub/Sub https://cloud.google.com/pubsub?hl=en	A real-time messaging service that allows the user to send and receive messages between independent applications hosted on Google Cloud Platform or elsewhere on the Internet	Message Queue	Low latency Auto scalability High availability	Costly solution Limited documentation	N

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

IBM InfoSphere streams https://www.ibm.com/docs/en/streams	A software platform that enables the development and execution of applications that process information in data streams that generates massive volumes of moving data	Real-time Data Processing	Easy to use Low latency	Costly solution Vendor lock-in	N
Photon https://www.databricks.com/product/photon	A query execution engine designed to accelerate complex workloads on Delta Lakes. It is built on Apache Spark and improves the execution of SQL and machine learning workloads	Big Data Management	High performance Designed for real-time processing	User-defined functions not supported Limited sources supported	N
Redpanda https://redpanda.com/	A Kafka-compatible streaming data platform that is proven to be faster and cheaper. It is also JVM and ZooKeeper-free	Real-time Data Processing	High performance High scalability	Recent framework Limited user support	Y
SQLstream https://sqlstream.com/	Creates streaming Kafka and Kinesis applications with continuous SQL queries to discover, analyze and act on data in real time	Big Data Streaming Analytics	Good UI Low latency	Costly solution Vendor lock-in	N
StramSets Data Collector https://streamsets.com/products/data-collector-engine/	A data pipeline engine for building reliable, smart data pipelines for both streaming and batch data from a wide variety of sources and destinations	Big Data Management	Good UI Easy to apply schema changes	Vendor lock-in Limited user support	Y

Table 9: Data streaming management research tools

Tool	Description	Type	Pros
Cloudets https://ieeexplore.ieee.org/abstract/document/7259407	An elastic framework for big data visual analytics in the cloud. The Cloudet is a self-adaptive cloud-based platform that treats both data and compute nodes as elastic objects	Big Data Streaming Analytics	High scalability High elasticity
CQELS https://arxiv.org/abs/2202.13958	CQELS (Continuous Query Evaluation over Linked Streams) is a platform-agnostic federated execution framework for semantic stream fusion	Real-time Data Processing	Handle data integration problems Support different data formats (including video)
ETALIS https://link.springer.com/chapter/10.1007/978-3-642-19724-6_5	An expressive language for specifying and combining complex events. Two main approaches are possible: a non-logic approach and a logic rule-based one, where the latest one achieves the best performance	Real-time Data Processing	Definition of a declarative semantics for complex event patterns

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

			Use of reasoning on events
Google MillWheel https://research.google/pubs/pub41378/	A framework for building low-latency data-processing applications. MillWheel provides fault tolerance at the framework level, where any node in the topology can fail at any time without affecting the correctness of the result	Real-time Data Processing	Fault-tolerance Guarantees Low latency
MavEStream https://ieeexplore.ieee.org/document/4270595	A stream processing system with an event processing system that combines event and stream data processing	Real-time Data Processing	Identification of events from streaming data

Table 10: Data streaming management prototypes tools

Tool	Description	Type	Pros	Cons
Arroyo https://github.com/ArroyoSystems/arroyo	A distributed stream processing engine in Rust, designed to efficiently perform stateful computations on streams of data. It joins Apache Kafka streams using SQL	Real-time Data Processing	High scalability Stateful operations	Limited documentation Limited UI
Benthos https://github.com/benthosdev/benthos	A stream processor able to connect various sources and sinks and perform hydration, enrichments, transformations and filters on payloads	Real-time Data Processing	High performance High resiliency	Limited metric monitoring
Brooklin https://github.com/linkedin/brooklin	A distributed system intended for streaming data between heterogeneous source and destination systems, designed for multi-tenancy	Message Queue	High reliability High throughput	No auto-scaling property Limited number of data sources supported
C-SPARQL https://github.com/streamreasoning/CSPARQL-engine	An extension of SPARQL whose distinguishing feature is the support of continuous queries, i.e. queries registered over RDF data streams and then continuously executed	Real-time Data Processing	Allows real-time integration between SPARQL and streaming flows	Limited to RDF format
Conduit https://github.com/ConduitIO/conduit	A data streaming tool written in Go for building and running real-time data pipelines. It can be seen as a Kafka Connect replacement	Real-time Data Processing	High consistency Good UI and observability components	Limited number of data connectors supported

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Memphis https://github.com/memphisdev/memphis	An alternative to traditional message brokers that enables the building of modern queue-based applications that require large volumes of streamed and enriched data	Message Queue	High scalability Natively integrated with Kubernetes and Docker Low-code	Limited integrations with third party elements
Strimzi https://github.com/strimzi/strimzi-kafka-operator	A way to run an Apache Kafka cluster on Kubernetes or OpenShift in various deployment configurations	Message Queue	Easy to use Increase scalability using cloud native technologies	Since it is based on Apache Kafka, it is difficult to configure at first

4.4.2.3.1 Tools from the consortium

4.4.2.3.1.1 Streamstory

Streamstory[165] [Stopar, 2018] is a tool designed to meet the needs of complex system observation, primarily within the context of smart industry. It is designed to handle sensor data as multivariate time series and to provide automatic hierarchical abstractions and automatically form typical states and identify transactions between them. This allows for unparalleled automatic abstraction and analysis of complex systems. It achieves this by implementing various machine learning clustering algorithms to analyse the data and determine typical states of the system followed by Markov chains to model the transitions within that system. The tool produces results on multiple levels, which means that the system can be modelled in a generalized way or explored in more details. The analysis outcome is a visualization of the input data, helping an expert to better understand and interpret the behaviour of the observed system.

The system attempts to solve automatically solve the problem of visualization and interpretation of multivariate time series data. The goal of the model is to identify different states and the transitions between those states (Figure 29). This system allows us to visualize time series data in a way that is easy, or at least easier, to interpret. Figure 30 shows an example of the tool's interface displaying visualizations of a system built automatically from a dataset.

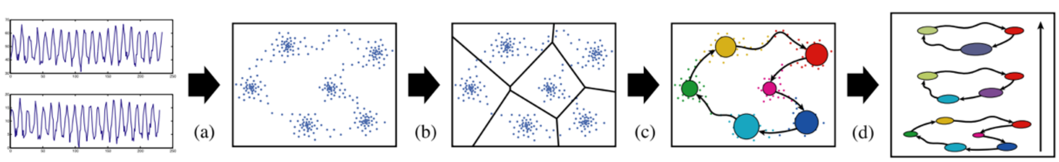


Figure 29: Schematic display of steps. (a) constructing a point cloud, (b) Identifying states by partitioning the ambient space, (c) Modelling the transitions between states and (d) Creating the hierarchy of states and transitions [1].

[165] https://github.com/E3-JSI/StreamStory2

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Figure 30: Example of Streamstory's user interface and graphical visualization automatically built from a dataset.

Streamstory has been developed through the course of several European Research projects including: FP7 NRG4CAST, FP7 Proasense, H2020 OPTIMUM, H2020 NAIADES, H2020 Infinitech and, H2020 FACTLOG. In these projects it has found diverse applications from predictive failure analysis, optimization of transportation, energy demand forecasting, and water management.

4.4.2.3.1.2 Event Registry

Event Registry[166] (Leban, 2014) is the world's leading news intelligence platform, empowering organizations to keep track of world events and analyse their impact. It tracks over 150,000 news sources in 50+ languages, and processes articles within minutes of being published. It's continuous stream of articles, additionally grouped into events, represents a very large stream of text enriched with metadata, both extracted directly from the article text, the source HTML, or automatically added using machine learning methods. Using its API[167] it's possible to directly follow the stream of articles or events, get the full news content as well as information about the mentioned entities, topics, and sentiment. Filter using

166 https://www.eventregistry.org/

167 https://newsapi.ai/

any of the available properties to get only the content needed. The API is available as a REST API[168], wrapped in a node.js library[169], or python library[170]. Some common use cases for the stream include:

- Getting the latest news mentions of a person, company, location, etc.
- Supporting a platform that requires real-time news content.
- Extract market trends, analyse specific industry, track competitors and customers.
- Track news reach and virality of a person, company, product or topic of interest.
- Track news across languages and locations (Figure 31).

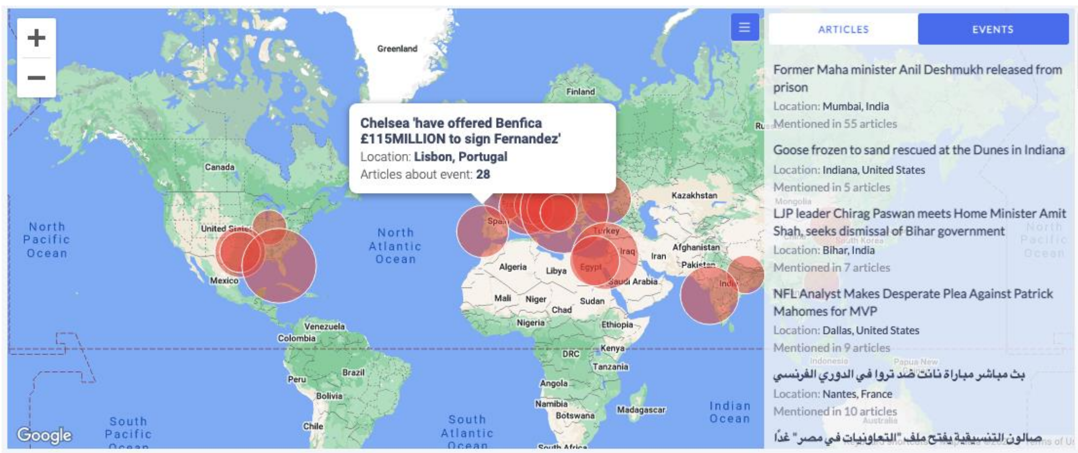


Figure 31: Visualizing the locations of events in Event Registry.

Although it is possible to use the stream data to create visualizations, Event Registry also works as a platform with archive that can be queries and can also display built-in visualizations. Figure 32 shows an example that combines the data and metadata into creating a series of visualizations.

[168] https://newsapi.ai/documentation?tab=introduction

[169] https://github.com/EventRegistry/event-registry-node-js

[170] https://github.com/EventRegistry/event-registry-python

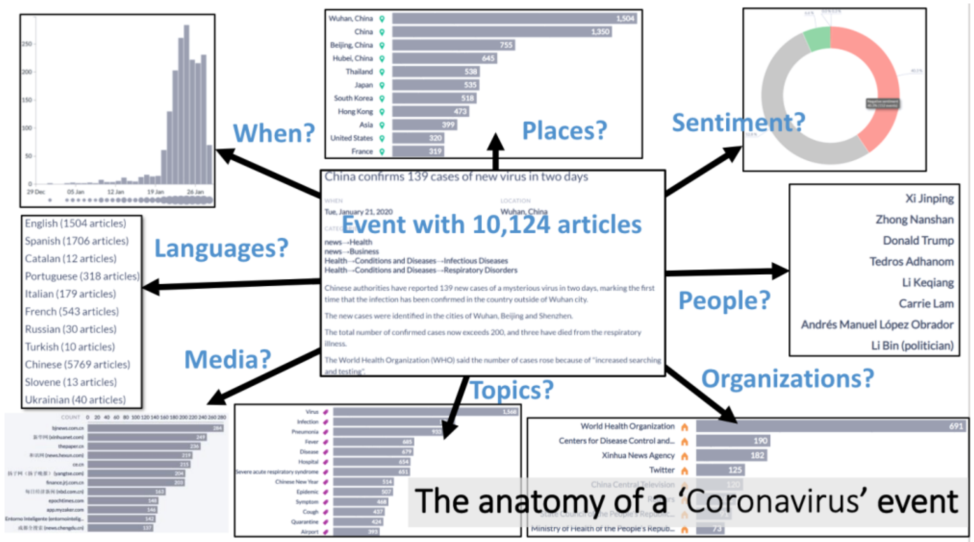


Figure 32: Example of Event Registry data and metadata with built-in visualizations.

4.4.2.4 Research & technical challenges

Challenges in handling streaming multivariate time series data include:

1. **Real-Time Processing:** Streaming data requires real-time analysis to extract timely insights and enable prompt decision-making. Processing large volumes of streaming multivariate time series data in real-time poses computational and latency challenges.
2. **Data Quality and Noise:** Industrial data streams can be prone to noise, outliers, missing values, and data quality issues. Handling data imperfections and ensuring data quality during real-time analysis is crucial for accurate results.
3. **Anomaly Detection:** Identifying abnormal behaviour or anomalies in multivariate time series data is critical for early detection of system faults, equipment failures, or safety concerns. Developing effective anomaly detection algorithms for streaming data is challenging due to the dynamic nature of industrial processes.
4. **Interpretability and Explainability:** In industrial applications, the interpretability and explainability of multivariate time series analysis models are crucial. Understanding the factors driving predictions and decisions is important for gaining trust and confidence in the analysis results.
5. **Adaptability to Diverse Domains:** Different industrial domains have unique characteristics, processes, and requirements. Developing multivariate time series analysis techniques that can adapt and generalize across diverse industrial domains is challenging due to the domain-specific knowledge and data characteristics.
6. **Model Maintenance and Continuous Learning:** Industrial systems and processes evolve over time, requiring models to adapt and continuously learn from new data. Developing techniques for model maintenance, retraining, and online learning with streaming data is an ongoing research challenge.

Challenges in handling streaming text data include:

1. **Real-Time Processing:** Processing streaming text data in real-time requires efficient algorithms and systems capable of handling data as it arrives. Developing scalable and low-latency processing techniques to handle high data velocities is a challenge.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

2. **Concept Drift and Evolving Topics:** Streaming text data is prone to concept drift, where the underlying concepts or topics change over time. Adapting to concept drift and detecting emerging topics in real-time is a research challenge. Maintaining up-to-date models and adapting to evolving linguistic patterns is important.
3. **Cross-Lingual Analysis and Transfer Learning:** Performing meaningful analysis across multiple languages poses challenges due to linguistic and cultural variations. Developing cross-lingual models and transfer learning techniques that can leverage knowledge from one language to improve performance in others is an important capability and a significant challenge.
4. **Data Sparsity and Imbalance:** Multilingual text streams often exhibit data sparsity and class imbalance across different languages. Collecting sufficient training data for each language and addressing the challenges of imbalanced classes in multilingual settings are ongoing research challenges.

4.4.3 Energy efficiency assessment and improvement (GreeneR)

4.4.3.1 Scope and example

The GreenR tool aims to provide infrastructure components to support the monitoring of data enrichment pipelines in terms of their environmental impact. It monitors the carbon footprint of the various components in the pipeline and provides the results to the use through a dashboard to log and modulate the environmental impact due to the heavy computations within the pipelines.

The capabilities provided by GreenR capabilities will align the proposed data pipeline infrastructure with the UN sustainability goals that build "resilient and sustainable infrastructure". GreenR will denominate most of the individual components in the pipeline, with two high level contributions: monitoring and intervention. The services developed in GreenR are used to profile the energy efficiency of individual components, measured in terms of their carbon footprint. GreenR services is based on Carbontracker, which is currently implemented as a Python based tool that can track and predict the energy consumption of processing- and memory- devices. The measured carbon footprint is then used to improve the overall efficiency of the pipeline. This is achieved by reducing wasteful recomputations that incur large carbon costs.

4.4.3.2 Research

It is no secret that the computing footprint of machine learning has rapidly increased in the past decade. In fact, we’ve seen two step jumps in recent years in terms of total compute for ML – one around 2010, and one around 2016 (Sevilla et al. 2022). Billion parameter deep models are becoming the norm, at least for anyone with access to massive computational resources and a penchant for large language models. This has happened while the carbon emissions of the information sector as a whole are estimated to be as high as 2.1-3.9% of total greenhouse gas (GHG) emissions, on par with global aviation.

As of now we don’t exactly know the contributions of ML to the GHG. Accurate reporting and measurement of carbon emissions is necessary for transparency and assessing the environmental impact of ML, understanding the emissions of different parts of the ML development life cycle, and can guide research on how to make ML more sustainable. This can also help spur innovation in related areas such as the compute efficiency of ML. But it is entirely non-trivial to define what emissions to measure and how, and depending on what factors one includes in their measurements and how they measure them, one can come to radically different conclusions about the environmental impact of ML.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.4.3.2.1 Assessing Emissions

The common paradigm used to assess the carbon emissions for a product or service is Life-Cycle Assessment (LCA) [Klöpffer 1997]. The basic idea of LCA is that all environmental burdens connected with a product or service have to be assessed, back to the raw materials and down to the waste removal. In the context of product development, this includes the emissions produced by manufacturing, transport, product use, and recycling, and within ML, the focus is on manufacturing (building and utilizing the data, hardware, and methods for creating models) and product use (building and utilizing the data, hardware, and methods for downstream use of models) [Wu et al. 2022]. Within those broad categories, we can differentiate two sources of emissions which need to be accounted for. Operational carbon emissions are the emissions produced by hardware use i.e., from energy consumed when running any computation on actual hardware. Quantifying this involves knowledge of the power consumption of the device(s) the job is being run on (e.g. in Kilowatts (kW); when run in a data centre, one should also factor in the power usage effectiveness (PUE) of that data centre), the total time that the job is running (e.g. in hours (h)), and the carbon intensity of the power grid (e.g. in equivalent grams of CO2 per kilowatt hour i.e. gCO2eq/kWh). Embodied carbon emissions are emissions produced by manufacturing and maintaining the hardware used for computation e.g., producing raw materials, manufacturing, packaging, and assembly of hardware, maintaining and cooling the data centres where hardware sits, etc. It is extremely tricky to measure this – you need to have knowledge of the emissions produced from manufacturing your hardware stack. Additionally, to determine the embodied emissions for a particular job, you need to know what percentage of the manufacturing emissions to use, which is dependent on the lifespan of the hardware and how long your job will run.

Actually, accounting for all of the emissions within these categories is indeed a rabbit hole: within operational emissions, do we only care about the emissions produced by the GPU where our model computations are executed, or do we also care about all of the supporting computation on e.g., CPUs and network cards which run in conjunction with model computations? Within embodied emissions, how do we account for manufacturing costs when the manufacturers don't release this data, let alone how that cost can or will be amortized across the lifespan of that hardware? We don't have any widely adopted standards for this in our field yet; as a result, there are vastly different estimates, opinions, and projections about the environmental impact of ML.

To highlight this and offer some perspective, we look at four recent papers which examine and account for emissions in different stages of the ML life cycle.

Luccioni et al. (2022) reports on the estimated carbon footprint of BLOOM [Luccioni 2022]. BLOOM is essentially the open-source answer to GPT-3, the approximately equivalently sized large language model released by OpenAI (and the original base model for ChatGPT). For some more background on BLOOM: the BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) is a 176 billion parameter language model. It was trained on 1.6 terabytes of data in 46 natural languages and 13 programming languages as part of the BigScience workshop, a year-long initiative that lasted from May 2021 to May 2022 and brought together over a thousand researchers from around the world. The paper looks at emissions during the whole life cycle of developing BLOOM, namely on: (1) equipment manufacturing, (2) model training, and (3) model deployment.

Given the difficulty of quantifying embodied emissions (1), they have to rely on estimates: the total emissions of the production of their compute cluster was not available, and the emissions associated with producing the A100 GPUs that they used were not available either.

They also do not have access to real time power consumption of the GPUs, so they estimate this using the thermal power design (TDP) of the GPUs used, which serves as an upper bound on the energy

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

consumption. Additionally, they don't have access to the real-time carbon intensity of the grid, so they rely on the average carbon-intensity of the energy grid in France (where the hardware is located) during 2020.

Something related to (2) which is unique is the idle power consumption of the compute cluster where the model was trained. This takes into account the emissions associated with running all of the hardware in the compute cluster not directly involved in running the model (i.e., all hardware that is not a GPU or device where model computation is taking place but which is running nonetheless e.g., network components, the CPUs, etc.), giving a more holistic view of the carbon emissions produced by BLOOM.

Finally, for (3), they deploy the model to google cloud platform (GCP) and measure the real time power consumption of the instance over an 18-day period using CodeCarbon [Lacoste et al. 2019].

Their main findings can be summarized as follows:

- They observe 50.5 tonnes of CO2eq over the development life cycle of BLOOM.
- This can be broken down into 11.2 tonnes (22.2%) embodied emissions, 24.69 tonnes (48.9%) dynamic power consumption, and 14.6 tonnes (28.9%) idle power consumption.
- For inference, BLOOM emitted 19kgCO2eq per day when deployed on a GCP instance in the uscentral-1 region (calculated based on the average hourly carbon intensity in that region).
- When considering the BigScience workshop as a whole, taking into account the whole experimentation and prototyping of BLOOM, the total emissions add up to 66.29 tonnes of CO2eq operational emissions + 73.32 tonnes embodied and idle emissions.

The paper presents an honest view of the emissions produced from developing BLOOM. They rely on estimates where necessary, so it is difficult to say exactly how accurate the figures in the paper are. But the figure about idle consumption is particularly striking to me – it constituted almost 30% of the total emissions from training.

Next, [Wu et al. 2022] takes a holistic perspective of the carbon footprint of ML at Facebook (I'll focus mostly on their retrospective analysis in Section III). Like in [Luccioni 2022], they try to account for the whole life cycle of ML development, including data, experimentation, training and inference. Similarly, they also look at both the operational and embodied footprint of each stage of ML development.

They first look at and compare the emissions from 5 Transformer based language models used for text translation at Facebook, as well as some open-source models (e.g., Meena, GPT-3, T5). For their in-house models, they are also able to break this down into emissions from offline training (experimentation and training of models on a fixed dataset), online training (updating a model with new training data), and inference, where inference and online training costs are measured over an equal time-period to offline training. They observe equal costs for inference and training for recommendation use cases and greater inference cost for general language modelling e.g., translation. This demonstrates the need to consider the potential cost of inference for a model, while a majority of work that I've come across mostly considers the training cost.

One thing to note in the above is that, despite being a centralized entity with their own data centres, they still appear to use only estimates of carbon intensity at their data centres as opposed to real-time intensity, relying on the 2020 Facebook sustainability report. They additionally rely on rough estimates for the embodied emissions, potentially highlighting just how challenging it is to quantify these emissions:

We assume GPU-based AI training systems have similar embodied footprint as the production footprint of Apple's 28core CPU with dual AMD Radeon GPUs (2000kg CO2). For CPU-only systems, we assume half the embodied emissions. Based on the characterization of model training and inference at Facebook, we assume an average utilization of 30-60% over the 3- to 5-year lifetime for servers.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

They then conclude that the split between operational and embodied emissions is 30%/70% for large scale language modelling tasks at Facebook, and that embodied emissions will likely continue to dominate the overall emissions of ML going forward.

Finally, they look at how efficient ML techniques can potentially help reduce the operational energy footprint of large LMs at Facebook. The main conclusion here is that at Facebook, efficient methods have helped to reduce the energy consumed as a percentage of their total energy consumption, but increasing demand has still led to increasing energy consumption from ML systems over time, an example of Jevon's paradox i.e., increased efficiency lowers the cost of a resource which leads to increased demand, so consumption of that resource increases overall. Finally, they offer several insights and suggestions for reducing both embodied and operational emissions going forward.

Another industry paper, this time from Google, [Patterson et al. 2022] look specifically at operational emission reductions at Google through two case studies and noting four factors (called 4M) which, if adopted across the industry, would contribute to reductions in carbon emissions.

The 4M factors are:

- **Model:** selecting an efficient model architecture can reduce computation and thus energy consumption and potentially emissions.
- **Machine:** specialized hardware such as new GPUs and TPUs can improve performance/Watt by factors of up to 13.7x.
- **Mechanization:** computing in cloud data centres can reduce energy costs by a factor of 1.4x.
- **Map:** Selecting a better location for computation can reduce the gross carbon footprint due to a better energy mixture.

Case study 1 looks at how the 4M factors reduce the "end-to-end CO_2e" when using the original Transformer model from 2017 trained in an average (non-google) data centre as a reference point. Essentially, (1) looks at improvements in model efficiency (Transformer → Evolved Transformer → Primer), (2) adds improvements in hardware (P100 → TPUv2 → TPUv4), (3) adds better data centre efficiency (google data centres have better power usage effectiveness over the average data centre), and (4) adds selecting a data centre in a location with the lowest carbon intensity. They report that, by adopting the 4M factors, it is possible to achieve a 747x reduction in end-to-end CO_2e. This does not take into account factors related to use e.g., how long it will take to train a model, how long it will be deployed for, how many people will be using it, etc. The improvements in model architectures also do not take into account the emissions from neural architecture search (NAS) which were required to find them. The basic message is: use more efficient architectures on more efficient hardware in less carbon intensive locations to reduce carbon emissions. Their conclusion is that, given this figure, it is inaccurate to use current day figures to predict the future carbon emissions of ML systems.

Case study 2 compares gross carbon emissions from training two large language models: OpenAI's GPT-3 and Google's GLaM [Du et al. 2022]. They report a 14x reduction in the emissions from training, which can be attributed to the model being more efficient (though GLaM has more parameters than GPT-3, it uses mixture-of-experts to select which parts of the network to use) and the fact that the model was trained in a more efficient data centre with a better energy mixture. Again, this case study is intended to support the position that using current estimates of carbon emissions are likely to overestimate predictions of emissions going forward.

They then perform a retrospective analysis of the percentage of total compute related to machine learning at Google data centres, looking at one specific week in April for 2019, 2020, and 2021. They find that as a percentage of compute, machine learning is holding steady, though overall compute increased over those

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

three years. They argue that improvements in ML efficiency kept ML compute percentage steady despite increased demand (note: another example of Jevon's paradox at work).

Finally, they argue that previous estimates of the cost associated with particular models (e.g., Evolved Transformer and Primer), were severely overestimated due the inclusion of NAS in the calculation, and that NAS should have an overall net positive effect due to finding more efficient architectures. In line with this, they echo the work of [Wu et al. 2022] to argue that the primary cost is at inference, which constitutes 60% of the ML computation at Google.

Their overall conclusion is that if the 4M factors they list are widely adopted, and innovation in these areas continues, the carbon footprint of ML will plateau and then shrink.

The work by [Luccioni et al. 2023] seeks to quantify and compare the carbon emissions produced during training by ~100 state of the art models from the past decade. The paper is thorough – in addition to gleaning necessary details from the papers for each of the models, they contact and retrieve additional information from the authors, including:

- Where the models were trained.
- The hardware used.
- The total training time of the models.

Given this, they are able to characterize and compare:

- The main sources of energy (based on location) used to train those models.
- Estimations of the carbon emissions produced by each model.
- The evolution of carbon emissions over time in the field.
- The relationship between performance and carbon emissions.

While they rely on estimation by necessity, and it is hard to say how accurate such estimations are, the paper gives a view into what sorts of insights can be gained by performing standardized comparisons of the carbon emissions of a wide set of models across time. The caveat being that they too don't look holistically: they look only at model training, and only a subset of models in the field (albeit very important ones).

They also provide evidence which contradicts [Patterson et al. 2022]: based on their analysis, the carbon footprint of ML is actually increasing over time.

If we look at the aggregated data from all tasks, we can observe that overall, the carbon emissions per model have increased by a factor of about 100 (two orders of magnitude) from 2012 to recent years, with slight fluctuations, as in 2020. In fact, the last three years of our sample (2019-2021), have seen models that have emitted orders of magnitude more carbon than before.

Note the similarity with the results from [Sevilla et al. 2022], included at the beginning of this essay: a step jump in model size around 2016 and a step jump in carbon emissions around 2019.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

4.4.3.3 Tools and market

Table 11: Methods for estimating the carbon footprint of Deep Learning

Tool	Track	Predict	Report	Pip	Vis.	API	GPU	CPU
ML Emissions Calculator	✗	✗	✓	✗	✗	✗	✓	✗
Experiment-Impact-Tracker	✓	✗	✓	✓	✗	✗	✓	✓
CarbonTracker	✓	✓	✓	✓	✗	✗	✓	✓
CodeCarbon	✓	✗	✓	✓	✓	✓	✓	✓

1. **Machine Learning Emissions Calculator (MLEC)** [Lacoste et al. 2019]: This is a self-reporting interface where users input training time, hardware and geographic location post hoc to estimate the carbon emissions. It currently only estimates based on the energy consumption of only the graphics processing unit (GPU).
2. **Experiment-Impact-Tracker (EIT)** [Henderson et al. 2020]: This tool can be used to track the energy costs of CPU and GPU during the model training by embedding few lines of code into the training scripts. The tool can be tweaked to fetch real-time carbon intensity; currently this feature is supported for California (US).
3. **CodeCarbon** [Henderson et al. 2020]: This is the most recent addition to the carbon tracking tools and is the best maintained. It provides tracking capabilities like EIT and Carbontracker, along with comprehensive visualizations. The application programming interface (API) integration capabilities are best supported in CodeCarbon which can be useful for extending additional functionalities.

4.4.3.3.1 Tools from the consortium

The main tool to be offered from the consortium will be based on **Carbontracker** [Anthony et al. 2020]. A brief outline of its features is provided below.

4.4.3.3.1.1 Carbontracker

Carbontracker [Anthony et al. 2020]: This tool is similar to EIT in several aspects. In addition to the tracking capabilities of EIT, Carbontracker can also predict the energy consumption based on even a single run of the model configuration. This can be useful for performing model selection based on carbon emissions without having to train all the way. This tool currently provides real-time carbon emissions for Denmark and the United Kingdom, and can be extended to other locations.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

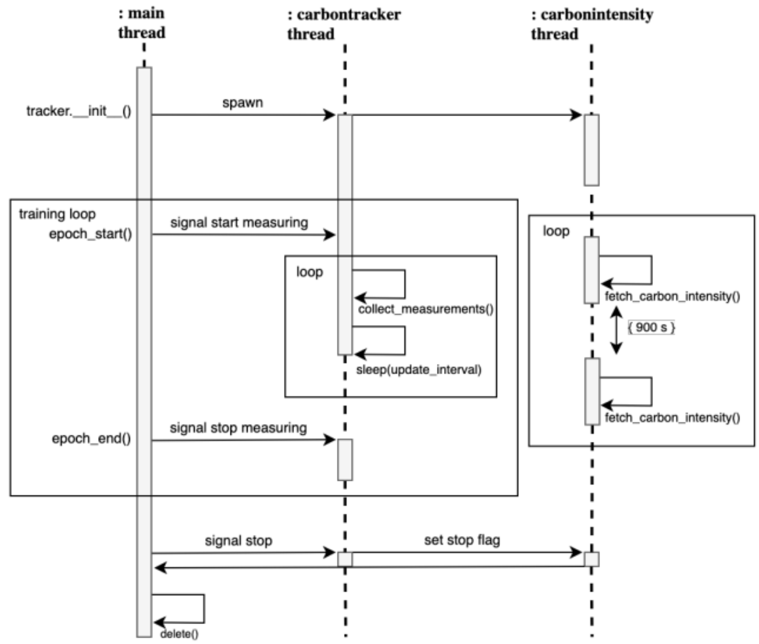


Figure 33: Visualization of the Carbontracker control flow (image from (Anthony et al. 2020))

4.4.3.4 Research & technical challenges

Carbontracker is a tool that is primarily designed for the tracking and prediction of the carbon footprint of Machine Learning pipelines. The hardware and software support of this tool is currently limited and focuses on the ML community. Extending these functionalities to hardware/software agnostic pipelines will be a technical challenge that needs to be addressed.

Integrating the GreenR features into all business cases and other tools will be challenging due to the diversity of needs and requirements.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

5 Requirements for Tool Collections

EMD tool collections will be developed in accordance with market requirements and business-case requirements. This requires addressing the challenges and issues that emerged from the state-of-the-art analysis and the BC definitions, as specified in D4.1.

5.1 Components for the enrichment of semi-structured data

5.1.1 WrappR

Requirement ID	RQ-WrappR-1
Requirement Short Title	Virtual Knowledge Graph
Requirement Description	The WrappR tool must be able to expose a relational database dataset as a virtual Graph that can be queried using SPARQL without copying the source data and keeping synchronized with it.
Source	SoTA / BC1, BC4, BC6 (RQ-BC1-16, RQ-BC6-04, RQ-BC6-06)
Relevant enRichMyData tools	GraphDB / Semantic Objects
Means of verification	Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-WrappR-2
Requirement Short Title	Full-Text-Search Data Sync
Requirement Description	The WrappR tool must be able to keep full text search and faceted search index(es) synchronized with the data in a semantic repository . The FTS index should be seamlessly accessible via the same data interface (SPARQL/ GraphQL, REST) as the data.
Source	SoTA / BC2, BC4 (RQ-BC2-07, RQ-BC4-02)
Relevant enRichMyData tools	GraphDB / Semantic Search
Means of verification	Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-WrappR-3
Requirement Short Title	Large Language Model (LLM) interaction
Requirement Description	The WrappR Tool should be able to index Knowledge Graph entities for use by a Large Language Model (LLM), e.g. by using a vector database, as part of an architecture involving a LLM

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Source	SoTA / BC4 (RQ-BC4-07)
Relevant enRichMyData tools	Ontotext GraphDB
Means of verification	Validation with a Business Case / Collaboration between team members
Depends on	
Priority	Should have

5.1.2 ResourcR

Requirement ID	RQ-ResourcR-1
Requirement Short Title	Scalability
Requirement Description	The entity retrieval component should efficiently handle large knowledge graphs (KG).
Source	SoTA / BC4
Relevant enRichMyData tools	lamAPI
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-ResourcR-2
Requirement Short Title	Caching / Response Optimization
Requirement Description	The entity retrieval module should ensure a predefined response time for queries by implementing an efficient caching strategy.
Source	SoTA / BC1 / BC4
Relevant enRichMyData tools	lamAPI
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-ResourcR-3
Requirement Short Title	Heterogeneity
Requirement Description	The entity retrieval module should be capable of indexing diverse sources and knowledge graphs (KGs) that may be provided in various formats and use different ontologies.
Source	SoTA / BC1 / BC4

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Relevant enRichMyData tools	lamAPI
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-ResourcR-4
Requirement Short Title	Privacy and security
Requirement Description	The entity retrieval module must incorporate privacy and security measures in compliance with GDPR regulations. This can be achieved through the implementation of authentication mechanisms or token-based methods to safeguard the data.
Source	SoTA / BC1 / BC4
Relevant enRichMyData tools	lamAPI
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-ResourcR-5
Requirement Short Title	Contextualized Retrieval
Requirement Description	The entity retrieval module should be capable of providing contextualized data retrieval based on various scenarios or use cases.
Source	SoTA / BC1 / BC4
Relevant enRichMyData tools	lamAPI
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Should have

5.1.3 DiscoverR

Requirement ID	RQ-DiscoverR-1
Requirement Short Title	Tabular data search on data about weather, calendar events, organizations
Requirement Description	DiscoverR tool must be able to support the discovery of data related to weather, calendar events, and organizations, which are relevant to enrich tabular data; the discovery process must help users understand which data features are available and specify criteria to match data that are relevant for each record in the original table

Source	BC1 / BC4 (RQ-BC1-04, RQ-BC1-18, RQ-BC4-02)
Relevant enRichMyData tools	SemTUI
Means of verification	Validation with business cases, User satisfaction
KPIs affected	User satisfaction in business cases
Depends on	RQ-DiscoverR-2, RQ-DiscoverR-3
Priority	Must have

Requirement ID	RQ-DiscoverR-2
Requirement Short Title	GUI for tabular data search
Requirement Description	DiscoverR tool must provide a GUI to ease the exploration of information in a table of interest and additional information sources that can be used to augment the information in the table. For these sources, provide information to understand their structure, content and support their usage.
Source	SoTA
Relevant enRichMyData tools	SemTUI
Means of verification	User satisfaction in completing benchmark enrichment task
KPIs affected	User satisfaction
Depends on	
Priority	Must have

Requirement ID	RQ-DiscoverR-3
Requirement Short Title	Natural language interfaces for tabular data search
Requirement Description	DiscoverR tool should provide language-based interfaces for exploring available data augmentation functionalities for a table of interest. In other words, the tool should support conversion of instructions given in natural language into tabular data manipulation commands.
Source	SoTA
Relevant enRichMyData tools	SemTUI
Means of verification	User satisfaction in completing benchmark enrichment task
KPIs affected	User satisfaction
Depends on	
Priority	Should have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-DiscoverR-4
Requirement Short Title	Semantic description of semi-structured sources not in RDF
Requirement Description	The DiscoverR should allow the construction of semantic descriptions of semi-structured sources by annotating the sources with tags from well-known reference vocabularies (e.g., Schema.org, WikiData, etc).
Source	SoTA
Relevant enRichMyData tools	ABSTAT, selBat
Means of verification	Quantitative analysis
KPIs affected	Accuracy of predicted labels (P/R/F-1)
Depends on	RQ-LinkR-5, RQ-LinkR-6
Priority	Could have

5.1.4 CleanR

Requirement ID	RQ-CLEANR-1
Requirement Short Title	Data sanitization and harmonization
Requirement Description	CleanR tool must be able to support data formatting and harmonization operations, such as whitespace trimming, splitting of columns, and handling multivalued fields
Source	SoTA, BC6
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-CLEANR-2
Requirement Short Title	Handling missing and unwanted values
Requirement Description	CleanR tool must support operations to handle missing and unwanted values, like null entries and duplicates
Source	SoTA, BC1, BC6
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must Have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-CLEANR-3
Requirement Short Title	String manipulations
Requirement Description	CleanR tool must support basic operations related to string manipulation, like replacement of values or regex operators. Common fixes are related to money symbols, grammar symbols and date formats
Source	SoTA, BC1, BC4
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-CLEANR-4
Requirement Short Title	Interactive data cleaning
Requirement Description	CleanR tool must be able to support a visual interface, in order to manipulate data while exploring it and to work with filtered subsets for specific transformations
Source	SoTA, BC2
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	RQ-CLEANR-1, RQ-CLEANR-2, RQ-CLEANR-3
Priority	Must Have

Requirement ID	RQ-CLEANR-5
Requirement Short Title	Identification of abnormal data
Requirement Description	CleanR tool must be able to help the user identify abnormal data generated in the process of mineral processing, including real-time data by sensors, images and videos. A more efficient data cleaning step will significantly improve the efficiency of the system in this use case
Source	BC6
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-CLEANR-6
Requirement Short Title	Generalisability of data analysis
Requirement Description	CleanR tool must be generalizable for different data analysis procedures. Each ML model is typically developed for a specific dataset, so a generalizable procedure is desirable for economic and strategic reasons
Source	BC2
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Nice to have

Requirement ID	RQ-CLEANR-7
Requirement Short Title	Limit human intervention in data cleaning
Requirement Description	CleanR tool must limit the quantity of manual or rule-based interventions. The tool must support a combination of techniques such as lookup to validated values, active learning, fuzzy matching, limiting the human intervention to a few cases
Source	SoTA, BC4
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-CLEANR-8
Requirement Short Title	Near-real-time data cleaning
Requirement Description	CleanR tool should be able to apply data transformation and data cleaning activities in near-real-time, in order to limit the delay between the collection of the data and the use of it. This functionality would be helpful to enhance the functions of the business case
Source	BC6
Relevant enRichMyData tools	Ontotext Refine
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Nice to have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

5.1.5 LinkR

Requirement ID	RQ-LinkR-1
Requirement Short Title	Integration into existing pipelines
Requirement Description	The LinkR should be able to add enriched features to current processes. Customized input from stored data (e.g., in PostgreSQL Server on AWS). Output (deliver augmented information) to JSON files.
Source	SoTA / BC1 / BC4
Relevant enRichMyData tools	s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Should have

Requirement ID	RQ-LinkR-2
Requirement Short Title	Uncertain outputs reporting
Requirement Description	LinkR should provide detailed quality reporting, including relevant information to report uncertain outputs (e.g., uncertain types, properties and entities).
Source	SoTA / BC1 / BC4
Relevant enRichMyData tools	s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-LinkR-3
Requirement Short Title	Uncertain outputs handling
Requirement Description	LinkR should support reviewing uncertain outputs and providing corrective feedback to support human-in-the-loop linking mechanisms.
Source	SoTA / BC4
Relevant enRichMyData tools	SemTUI / s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	RQ-LinkR-12
Priority	Must have

Requirement ID	RQ-LinkR-4
Requirement Short Title	Ability to process a large amount of data
Requirement Description	The linking approach should be capable of efficiently handling a large amount of data. This can be achieved through parallel processing and dividing the data into batches
Source	SoTA / BC4
Relevant enRichMyData tools	SemTUI / s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-LinkR-5
Requirement Short Title	Ability to deal with various languages
Requirement Description	The linking approach should have the ability to handle data in various languages (e.g., English, French)
Source	SoTA / BC4
Relevant enRichMyData tools	SemTUI / s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	
Priority	Must have

Requirement ID	RQ-LinkR-6
Requirement Short Title	Reconciliation against any sources / KGs.
Requirement Description	The linking approach should be able to reconcile data against any data sources or KGs.
Source	SoTA / BC4
Relevant enRichMyData tools	SemTUI / s-elBat
Means of verification	Synthetic datasets / Validation with a Business Case
Depends on	RQ-ResourcR-1, RQ-ResourcR-3
Priority	Should have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

5.2 Components for the enrichment of textual data

5.2.1 StructureR

Requirement ID	RQ-StructureR-1
Requirement Short Title	PDF Parsing
Requirement Description	StructureR must include a component for the parsing and segmentation of PDF documents, often ranging several hundred pages that need to be split into smaller sections.
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Must have

Requirement ID	RQ-StructureR-2
Requirement Short Title	Semantic text analysis in all major language
Requirement Description	StructureR should include a component for semantic text analysis in all major languages . For each passage we keep and index: 1.Document, page, geo-coordinates and offset of the passage 2.Passage full-text index (with symbolic metadata such as lemmas, entity tags, etc.) 3.Passage neural embedding
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-StructureR-3
Requirement Short Title	Semantic text analysis in English Language
Requirement Description	StructureR must include a component for semantic text analysis in English language. For each passage we keep and index: 1.Document, page, geo-coordinates and offset of the passage 2.Passage full-text index (with symbolic metadata such as lemmas, entity tags, etc.) 3.Passage neural embedding

Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Must have

Requirement ID	RQ-StructureR-4
Requirement Short Title	Semantic analysis of natural language questions in all major languages. Each question should be: •analyzed by symbolic NLP processor that identifies aspects as: •symbolic concepts (related to a given ontology/knowledge-graph), eg: synonyms <methodology,method,procedure,process> belong to a specific unique concept •related concepts (e.g. <extinguisher, fire extinguisher> is more specific than <device> but less specific than <carbon dioxide fire extinguisher>) that can be used to expand or restrict the query •lemmas, entities, etc
Requirement Description	.
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-StructureR-5
Requirement Short Title	Semantic analysis of natural language questions in English language
Requirement Description	StructureR must include a component for the semantic analysis of Natural language requests in English language. Each question must be: •analyzed by symbolic NLP processor that identifies aspects as: •symbolic concepts (related to a given ontology/knowledge-graph), eg: synonyms <methodology,method,procedure,process> belong to a specific unique concept •related concepts (e.g. <extinguisher, fire extinguisher> is more specific than <device> but less specific than <carbon dioxide fire extinguisher>) that can be used to expand or restrict the query •lemmas, entities, etc

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Must have

Requirement ID	RQ-StructureR-6
Requirement Short Title	Question Answering in all major languages
Requirement Description	StructureR should include a component for question answering in all major languages. In particular, Symbolic Passage Retrieval and Ranker: •Uses the Question Analyzer to generate a symbolic query for the Passage Storage •Executes the query which “selects” and “ranks” a first set of possible answers
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-StructureR-7
Requirement Short Title	Questions Answering in English language
Requirement Description	StructureR must include a component for question answering in English language. In particular, Symbolic Passage Retrieval and Ranker: •Uses the Question Analyzer to generate a symbolic query for the Passage Storage •Executes the query which “selects” and “ranks” a first set of possible answers
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Must have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-StructureR-8
Requirement Short Title	PDF Answer Navigator for all major languages
Requirement Description	Structured records should be linked to sections of documents that best describe how the described problem can be resolved. This should work for all major languages
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-StructureR-9
Requirement Short Title	PDF Answer Navigator for English language
Requirement Description	Structured record should be linked to sections of documents that best describe how the described problem can be resolved in English language
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-StructureR-10
Requirement Short Title	Off-line operation
Requirement Description	The entire pipeline should be deployable in a private environment (either on-premise or in a secure cloud environment accessible only by the business partner)
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document Analysis
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-StructureR-11
Requirement Short Title	Identify Companies in News
Requirement Description	Extracting company mentions from news articles and linking it to Wikipedia
Source	BC5
Relevant enRichMyData tools	Wikifier
Means of verification	Validation with BC5
Depends on	RQ-StreamR-1
Priority	Must have

Requirement ID	RQ-StructureR-12
Requirement Short Title	Identify AI related Events in News
Requirement Description	Identifying AI related news from the news stream and organization of AI news according to AI specific taxonomies
Source	BC5
Relevant enRichMyData tools	ER Relation Classifier
Means of verification	Validation with Business Case 5.
Depends on	RQ-StreamR-1
Priority	Must have

5.2.2 ClassifiR

Requirement ID	RQ-ClassifiR-1
Requirement Short Title	Document classification
Requirement Description	Documents should be classified according to keywords/known problem descriptions/AI related topics.
Source	BC2 / BC3 /BC5
Relevant enRichMyData tools	Expert AI Platform Document classification, InfoMiner
Means of verification	Validation with documents related to Business Cases 2, 3, 5
Depends on	
Priority	Should have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-ClassifiR-2
Requirement Short Title	Offline Classification
Requirement Description	The entire pipeline should be deployable in a private environment (either on-premise or in a secure cloud environment accessible only by the business partner)
Source	BC2 / BC3
Relevant enRichMyData tools	Expert AI Platform Document classification, InfoMiner
Means of verification	Validation with documents related to Business Case 2 and Business Case 3
Depends on	
Priority	Should have

Requirement ID	RQ-ClassifiR-3
Requirement Short Title	Keyword extraction
Requirement Description	Identify core keywords that define categories or concepts
Source	BC1
Relevant enRichMyData tools	InfoMiner
Means of verification	Validation with documents related to Business Case 1
Depends on	
Priority	Should have

5.3 Components covering other relevant aspects for data enrichment pipelines

5.3.1 ScalR

Requirement ID	RQ-ScalR-1
Requirement Short Title	User workflow execution
Requirement Description	ScalR shall be able to execute workflows (pipelines) defined by the user.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-ScalR-2
Requirement Short Title	Executable tools integration and execution
Requirement Description	ScalR shall be able to invoke programs from Open Container Initiative (Docker) images. These are mainly the enRichMyData tools, for which processing component descriptions will be created.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-ScalR-3
Requirement Short Title	Web services integration and execution
Requirement Description	ScalR shall allow the programs it runs to call Web services. These web services can be remotely hosted, but also run on the local infrastructure, in containers (preferable, when possible). These are mainly the enRichMyData tools, for which processing component descriptions will be created.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-ScalR-4
Requirement Short Title	Tool execution parallelization
Requirement Description	ScalR shall be able parallelize the program, when possible, as defined by the workflow.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-ScalR-5
Requirement Short Title	Flexibility on heterogeneous components
Requirement Description	ScalR shall be flexible enough to support the use of heterogeneous computing infrastructures in the data enrichment pipelines.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-ScalR-6
Requirement Short Title	Horizontal scalability
Requirement Description	ScalR shall be able to provide horizontal scalability of data enrichment pipelines using software containers.
Source	SoTA , BC3
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-SCALR-7
Requirement Short Title	ScalR Kubernetes execution
Requirement Description	ScalR shall be able to execute programs on a Kubernetes cluster.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-SCALR-8
Requirement Short Title	Handling big data
Requirement Description	ScalR shall allow the tools it runs to handle large amounts of data ("Big Data") in the data enrichment process.
Source	SoTA, BC3
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	RQ-SCALR-1
Priority	Must have

Requirement ID	RQ-SCALR-9
Requirement Short Title	Data enrichment pipelines as integrated deployable units
Requirement Description	ScalR shall promote the reuse and modification of data enrichment pipelines by exposing them as integrated deployable units, rather than as-hoc, non-reusable code.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	RQ-SCALR-4
Priority	Must have

Requirement ID	RQ-SCALR-10
Requirement Short Title	Automatic scalability
Requirement Description	ScalR shall scale automatically, up to defined limits, in order to have an infrastructure that can withstand traffic spikes and be available all the time, allowing real-time data collection.
Source	SoTA / BC3 / BC4
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-SCALR-11
Requirement Short Title	ScalR behaviour on errors
Requirement Description	ScalR shall prevent or stop the execution of the workflow on critical errors (e.g. incompatibility between step inputs and outputs, disk full errors, etc).
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-SCALR-12
Requirement Short Title	ScalR execution monitoring
Requirement Description	ScalR shall support monitoring the status and health of running workloads, and infrastructure (e.g. cluster).
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-SCALR-13
Requirement Short Title	ScalR execution logging
Requirement Description	ScalR shall produce detailed logs, allowing for any internal problems to be identified and diagnosed quickly.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

5.3.2 ReusR

Requirement ID	RQ-REUSR-1
Requirement Short Title	ReusR component usage
Requirement Description	ReusR would be to reuse existing available processing components to build new pipelines or to reuse the existing workflows in other business cases (with or without adaptation, depending on the needs).
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

Requirement ID	RQ-REUSR-2
Requirement Short Title	ReusR component versioning
Requirement Description	ReusR should be able to keep a history of processing component (tools) descriptors and workflows, to enable different versions of the same component to co-exist in different workflows. It should support restoring to a previous version when possible.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Synthetic datasets, Validation with business cases
Depends on	None
Priority	Must have

5.3.3 StreamR

Requirement ID	RQ-StreamR-1
Requirement Short Title	News access
Requirement Description	StreamR must provide access to news archives and streams , focusing on companies and public entities.
Source	BC4 / BC5
Relevant enRichMyData tools	Event Registry
Means of verification	Validation with Business Cases 4 and 5.
Depends on	
Priority	Must have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-StreamR-2
Requirement Short Title	Tabular Data Support
Requirement Description	StreamR should support access to data in tabular format
Source	BC2 / BC6
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Cases 2, 6.
Depends on	
Priority	Should have

Requirement ID	RQ-StreamR-3
Requirement Short Title	Weather data support
Requirement Description	StreamR should support weather data
Source	BC1
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Cases 1
Depends on	
Priority	Should have

Requirement ID	RQ-StreamR-4
Requirement Short Title	Time series correlation
Requirement Description	StreamR should be able to calculate correlation between time series.
Source	BC1 / BC6
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Cases 1, 6.
Depends on	
Priority	Should have

Requirement ID	RQ-StreamR-5
Requirement Short Title	Prediction
Requirement Description	StreamR should be able to predict values for welding quality processes, and possibly others

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Source	BC2
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Case 2
Depends on	
Priority	Should have

Requirement ID	RQ-StreamR-6
Requirement Short Title	Anomaly detection
Requirement Description	StreamR should detect anomalous states in the context of a mineral processing plant
Source	BC6
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Case 6
Depends on	
Priority	Should have

Requirement ID	RQ-StreamR-7
Requirement Short Title	Sensor data cleaning
Requirement Description	StreamR should support automatic methods for cleaning sensor data
Source	BC6
Relevant enRichMyData tools	StreamStory
Means of verification	Validation with Business Case 6
Depends on	
Priority	Should have

5.3.4 GreenR

Requirement ID	RQ-GreenR-1
Requirement Short Title	Live Carbon Intensity
Requirement Description	Fetch live carbon emissions data for the specific geographic location when running on local devices. For cloud instances, the information provided by service provider will be used or resort to default values.
Source	SoTA

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Relevant enRichMyData tools	GreenR
Means of verification	Historical data. Forecasting on held out historical data
Depends on	None
Priority	Should have

Requirement ID	RQ-GreenR-2
Requirement Short Title	Visual dashboard
Requirement Description	Total runtime, carbon emissions and energy consumption are estimated. These statistics should be depicted in a graphical format other than text logging.
Source	SoTA
Relevant enRichMyData tools	None
Means of verification	Synthetic data
Depends on	N
Priority	Should have

Requirement ID	RQ-GreenR-3
Requirement Short Title	Interface with other services
Requirement Description	Support for interfacing with other tools and services should be improved
Source	All BCs
Relevant enRichMyData tools	All EMD tools
Means of verification	Synthetic data
Depends on	All EMD tools
Priority	Should have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

6 System-wide Requirements

System-wide requirements are requirements that define necessary system quality attributes such as performance, usability and reliability, as well as global functional requirements that are not captured in behavioural requirements artifacts such as use cases.

6.1 System-wide Functional Requirements

The following sub-sections present requirements concerning security and privacy, and functional requirements concerning processing component description and workflow administration on the integration platform.

6.1.1 Security and Privacy Requirements

A minimum level of security can be achieved by means of an [Authentication & Authorization] component. Each user can access the platform by using, at least, a login and a password. The user accounts are created by the platform administrator, while the user passwords are set up by users themselves.

User-defined components and/or workflows, and the results of such workflows should be visible only to the authoring users, unless explicitly shared by the authors.

In addition, there are operations (i.e., functionalities) that should not be accessible to all users (for example, the definition of a system topology). Therefore, an access control has to be put in place in order to restrict some features to an administrator. Moreover, users should not be able to edit certain system artifacts (for example pre-defined processing components cannot be edited or deleted by users), but they should be able to use them.

Requirement ID	REQ-SYS-1
Requirement Short Title	Authentication
Requirement Description	The integrated toolbox shall be usable by a user by means of an account with a password.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	REQ-SYS-2
Requirement Short Title	Type of users
Requirement Description	The system should allow at least two types of users: admin and regular user.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Depends on	None
Priority	Must Have

Requirement ID	REQ-SYS-3
Requirement Short Title	Account Management
Requirement Description	The users shall be able to change the password of their respective accounts by themselves.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	REQ-SYS-4
Requirement Short Title	Access control
Requirement Description	An access control mechanism shall be implemented so that certain features can be used only by an administrator.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	REQ-SYS-5
Requirement Short Title	Artifact Access
Requirement Description	The user-defined artifacts, together with their results, shall be by default private (i.e. visible only to their author).
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	REQ-SYS-6
Requirement Short Title	Artifacts Sharing
Requirement Description	The platform shall allow a user to share one or more of its artifacts and results with other users.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

6.1.2 Resource administration

Before any utilization by the public users, the platform should have an initial preparation of resources, made by Administrator, resources that will be exposed to everyone. In addition, each Public User will have the possibility to define his own resources and decide if those are visible to everyone (public mode) or not (visible only for his private use).

The resources are used in definition of complex workflows for processing being linked. A resource can be removed only if there are no references to them.

Type of resources:

- **Processing component**: application / toolbox for processing input data
- **User**: user with different profiles to access resources
- **Data**: descriptors of input source, intermediary or final result from a workflow.

All resources are collected into a catalog, grouped by categories, type of access, etc.

By default, for resources added by Public User there is space quota predefined for saving them. With [User Quota Management] module, the Administrator should be able to configure the maximum amount of data that a user is allowed to have (either downloaded or produced) in his/her workspace.

6.1.2.1 Processing components

A **processing component** (on short called component) represents a standalone **application** (or module) defined by the following parameters:

- **Input description:** type of data that the component accepts as input source (e.g., file type etc.)
- **Processing operation with execution parameters:** the operation that the component executes with the list of accepted parameters.
- **Multi-thread flag:** indicate the component capability to execute the operation in a parallelized way. This information is just a flag and it is Administrator's responsibility to know if the component operation can be executed in parallel.
- **Output description:** type of data provided as processing operation result. Can consist in one or more files.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

A component is viewed as a generic execution resource and should permit, through its parameters list, the possibility to define input location (the place where it expects to retrieve input data for consuming) and output location (the place where the processing results are persisted).

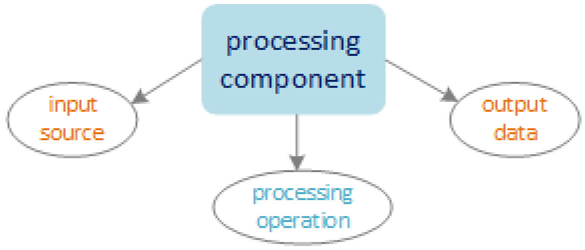


Figure 34: Processing component definition

Each processing component has to be described in terms of its expected input, its processing module and its output. The possible operations for this are addition, deletion, editing and cloning of components.

Requirement ID	RQ-RAD-1
Requirement Short Title	Processing component administration
Requirement Description	It shall be possible for the user (Administrator or Public User) to add, edit or delete resources stored in the system
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-RAD-2
Requirement Short Title	Processing component deletion
Requirement Description	A resource can be deleted from the system only if there are no references to it
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-RAD-3
Requirement Short Title	Processing components access

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement Description	Resources added by Administrator can be used by all users
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

6.1.2.2 Workflows

By workflow it is understood a sequence of processing operations performed on a given input, having at least one output. A workflow is described by a formal flow diagramming technique, with directed flows between components.

The processing operations in a workflow are handled by processing components.

The [Workflow Management] feature copes with all operations necessary to define such workflows and parameters of their components.

After defining / describing processing components, these can be glued together to form a workflow. A workflow definition will then comprise of the set of constituent modules and the rules that link them.

For a better user experience, a workflow will be graphically/visually created by dragging and dropping processing components (boxes) on a drawing canvas, and then connecting them with directed (arrowed) lines. Then, for each component, the parameters of the component can be modified.

A workflow can be created from scratch, can be edited or deleted, and/or can be created (cloned) from an existing workflow.

For definition of a workflow, we distinguish three phases:

- **Design:** the workflow is graphically described by a graph into system UI.
- **Preparation:** for each component is allocated an eligible node where the workflow can be executed. The workflow is interpreted as a job, composed by multiple tasks. The processing operation of a component together with the precondition for execution is translated into a task that will be executed on the allocated node.
- **Execution:** the tasks list within a job are executed.

At design time phase, the workflow is composed by components and continuity conditions when is necessary to introduce the cycles between different components or to conditionally continue the flow. The condition can be a standalone component or a script, which is capable to make an evaluation of output data. The workflow is persisted into database into a structured form.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

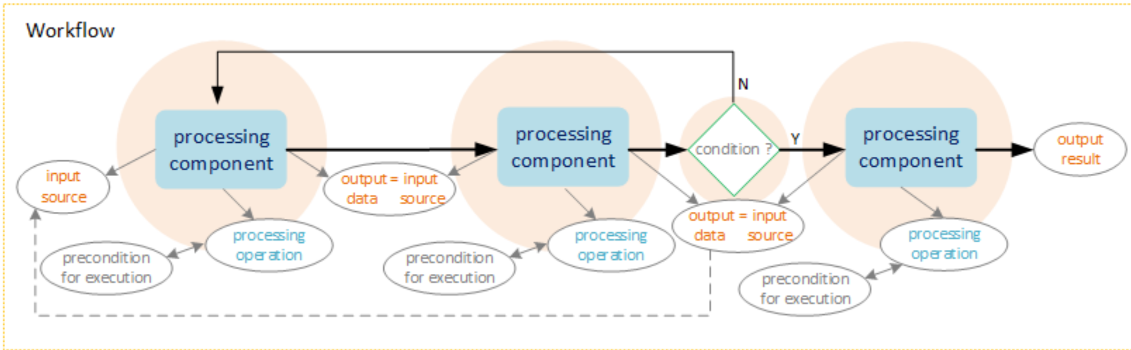


Figure 35: Workflow definition at design time

Components can only be linked together if the output of the previous (set of) component(s) matches the input requirements of the following component. By example, it cannot link a component with output type "json" with another component with input source type "xml".

For a component, we define the invocation descriptor, which must be filled when pick-up a component from system or user components lists, when a workflow is drawn:

- Describe how the component should be invoked: the relative file path to executable, the name of the executable, etc.
- Define the precondition for processing operation execution (if necessary): a precondition that says if the processing operation should be executed or not.
- Choose the input data types (from the list of accepted types from component definition).
- Define the input source according with component's input description (file path, stream, web service, etc.).
- Choose the output data types results (from the list of provided types from component definition).
- Define the output result location according with component's output description.
- Define if the output results are temporary (no cleaning action necessary) or persisted (in order to be reused later).

At execution time phase, the workflow (represented as a job) is executed; meaning the list of its tasks is executed.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

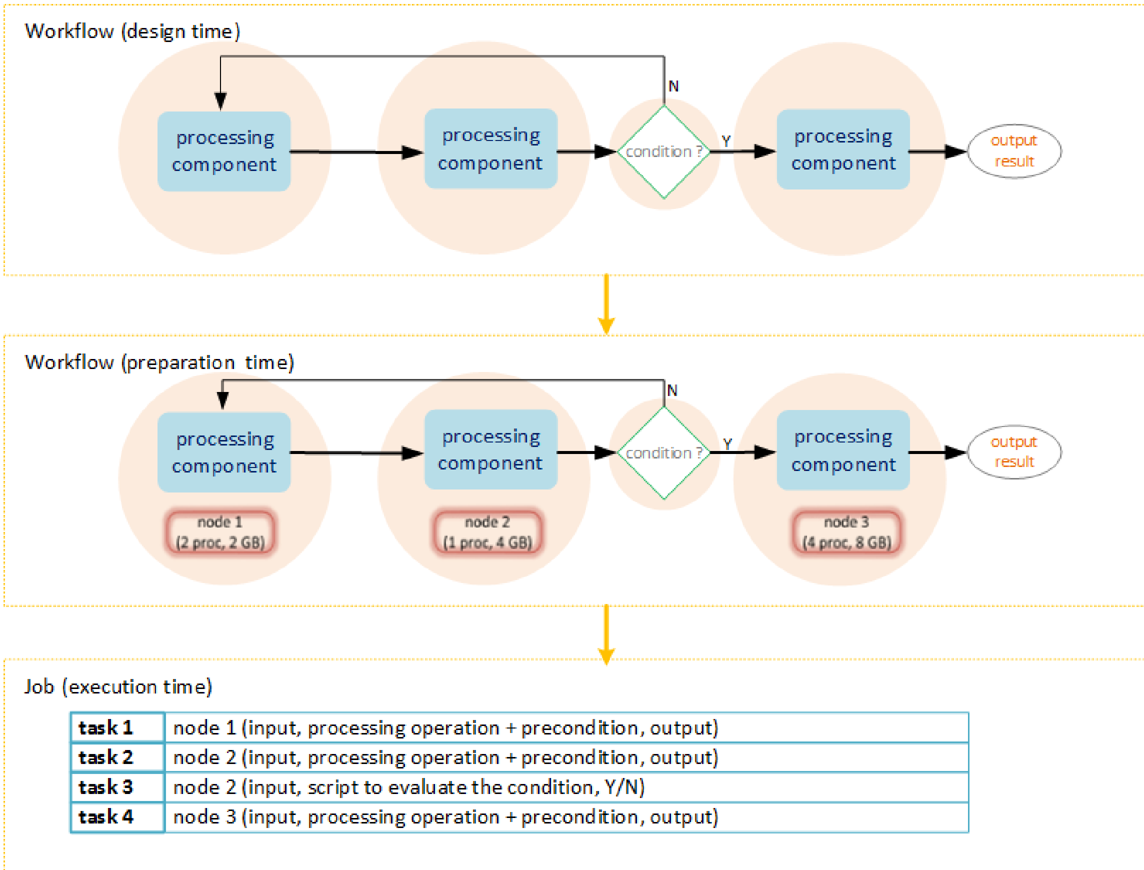


Figure 36: Job and tasks definition

Requirement ID	RQ-UWS-1
Requirement Short Title	Workflow definition and management
Requirement Description	The workflow manager shall allow the definition and management of workflows
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-UWS-2
Requirement Short Title	Workflow validation
Requirement Description	The Workflow Manager shall allow the link between two components only if the output of first component matches the input requirements of the second component

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-UWS-3
Requirement Short Title	Workflow execution
Requirement Description	The Workflow Manager shall be able to convert workflow represented graphically into jobs to be executed
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

6.2 System interfaces

This section presents the requirements concerning the interfaces that the enRichMyData platform presents to the users but also the external programming interfaces.

6.2.1 User interfaces

The interaction of the users with enRichMyData/TAO as a platform shall be done via web pages.

There will be one or more web pages for each module depicted in the logical model. They will allow the user (or the administrator) to perform the respective operations. The design (layout) of these pages will be established during the design phase.

At minimum, the system web user interface shall be able to perform the following functionally grouped operations:

- **Processing Components:** define new processing component, edit an existing processing component, delete a processing component, view existing processing components.
- **Workflow Management:** add a new workflow, add a processing component to a workflow, edit processing component instance parameters, remove a processing component from a workflow, clone a workflow, view existing workflow definitions, execute a workflow, interact with a workflow, remove a workflow.
- **User Management:** add a user, remove a user.
- **Authentication:** login, logout, password recovery.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **Monitoring Dashboard:** view currently used resources (CPU, memory, time), view execution history, view platform statistics.

Requirement ID	RQ-SUI-1
Requirement Short Title	User interface access
Requirement Description	The system shall be accessible to the users via a web interface
Source	SoTA
Relevant enRichMyData tools	ScalR, ReusR
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-SUI-2
Requirement Short Title	User interface look and feel
Requirement Description	The system shall allow the users to modify the look and feel: font size and colour theme.
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-SUI-3
Requirement Short Title	User interface dashboard
Requirement Description	The system shall present to the user a dashboard where the user can see the status of activities relevant to him (executions, history, notifications and statistics).
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-SUI-4
Requirement Short Title	User interface workflows
Requirement Description	The user shall be able to create new workflows from defined tool descriptions, using the graphical interface
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

6.2.2 Interfaces with External Systems

Requirement ID	RQ-SAPI-1
Requirement Short Title	System access via API
Requirement Description	The system shall offer a REST API for allowing interactions from third party systems for the basic operations: launch new execution jobs, get job status, get job results, etc
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

Requirement ID	RQ-SAPI-2
Requirement Short Title	Integrating components via REST or OGC WPS API
Requirement Description	The system should be able to integrate tool components that can be invoked via a REST API or a standardized Open Geospatial Consortium Web Processing Service (OGC WPS) API
Source	SoTA
Relevant enRichMyData tools	TAO
Means of verification	Validation with business cases
Depends on	None
Priority	Must Have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Requirement ID	RQ-SAPI-3
Requirement Short Title	Integration with GreenR
Requirement Description	The system should be able to retrieve and display information from GreenR about measured carbon emissions and other statistics.
Source	SoTA
Relevant enRichMyData tools	TAO, GreenR
Means of verification	Validation with business cases
Depends on	RQ-GreenR-1, RQ-GreenR-2, RQ-GreenR-3
Priority	Must Have

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

7 Conclusions

In this deliverable we presented a comprehensive overview of state-of-the-art solutions, which cover different aspects of data enrichment. We considered several aspects, grouped in three main categories:

- Aspects related to the enrichment of structured and semi-structured data:
    - Model transformations, data wrapping and indexing
    - Data discovery
    - Data cleaning
    - Data linking
- Aspects related to the enrichment of unstructured data, focusing on textual data
    - Information extraction
    - Classification
- Other aspects of data enrichment
    - Scaling, replication and adaptation
    - Streaming data
    - Energy efficiency assessment and improvement

For each aspect, we covered both research work and tools, i.e., remarkable research prototypes and community-driven project, as well as tools available in the market. In the list, we also briefly summarized the features of tools provided by EMD partners. More importantly, for each aspect, we identified main challenges, with the objectives of highlighting where innovation is more needed because of the gap exiting between scientific research and solutions available in the market.

In the second part of the deliverable, we described requirements for each family of tool that will populate the EMD toolbox. The support for these requirements comes from two main sources of evidence: the analysis of state of the art, and the analysis of the business case requirements. The latter, discussed in D4.1, is the result of the cooperation of tool providers and business partners by means of a specific methodology. The requirements are drawn from these two information sources, aiming at identifying features that tools must have to be effective in industry use cases – based on business requirements, and features that are expected to innovate products to support data enrichment.

In the presentation of the actual requirements, we considered requirements specific to tool collections, and system-wide requirements. As a result, for the collections of tools that target the enrichment of semi-structured and textual data, we collected a total of 41 requirements: WrappR (3), ResourcR (5), DiscoverR (4), CleanR (8), LinkR (6), StructureR (12), ClassifieR (3). For other relevant aspects of data enrichment, we collected a total of 25 requirements: ScalR (13), ReusR (2), StreamR (7), GreenR (3). In addition, a total of 19 system-wide requirements have been collected.

Some of the features considered in the requirements are needed to cover business needs or to support interoperability across tools and/or external components; some other features derived from SoTA are likely to be more innovative; we summarize the most relevant features (must and should have) by tool collection as follows:

- **WrappR:** providing virtualized graph-based access to relational data; sync full-text search and facets features; provide entity indexes based LLMs.
- **ResourcR:** provide IR-based matching functionalities on arbitrary large knowledge graphs with optimized response time and secure access; provide contextualized search features.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

- **DiscoverR:** provide tabular data search features via a GUI and, possibly, via natural language interfaces exploiting LLMs.
- **CleanR:** support typical data cleaning operations such as handling missing and unwanted values, string manipulation, identification of abnormal data, etc., possibly in near real-time; support more general interactive cleaning mechanisms, and limit user intervention using machine learning.
- **LinkR:** support the interpretation and management of uncertainty on large amounts of data to deliver full-ledged human-in-the-loop linking solutions, possibly against arbitrary target data sources.
- **StructureR:** support PDF parsing and navigation (documents and answers to requests), semantic analysis of documents and requests in English and other languages; support question answering applications.
- **ClassifieR:** support document classification, also off-line and keyword extraction.
- **ScalR:** support the execution of workflows specified by a user also integrating external tools and web services; parallelize the execution of workflows, guarantee horizontal scalability on heterogenous computing infrastructures and Kubernetes clusters with robust behaviour against errors; support the automatic scalability of ML pipelines; produce; produce execution logs.
- **ReusR:** make it possible to reuse existing components (and their versions) and workflows to adapt them.
- **StreamR:** support processing in the business cases, by providing, especially, support for tabular data, time series correlation and predictive analytics, anomaly detection, and sensor data cleaning.
- **GreenR:** provide data about carbon emission adjusted to specific locations; provide a visual dashboard; interface with other enRichMyData tools.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

8 References

8.1 Model transformations, data wrapping and indexing

[Ali et al 2022] W. Ali, M. Saleem, B. Yao, A. Hogan, and A.-C. N. Ngomo, "A survey of RDF stores & SPARQL engines for querying knowledge graphs," The VLDB Journal, vol. 31, no. 3, pp. 1–26, May 2022, doi: 10.1007/s00778-021-00711-3.

[Ambritta et al 2016] P. N. Ambritta, Y. S. Hande, and S. Darade, "A Survey on Access Control Models and Applications," presented at the IoT, Jul. 2016.

[Arenas et al 2012] Arenas, M., Bertails, A., Prud'hommeaux, E., & Sequeda, J. A direct mapping of relational data to RDF, Sept. 2012. URL http://www. w3. org/TR/2012/REC-rdb-direct-mapping-20120927.

[Avogadro et al 2022] Avogadro, R., Cremaschi, M., D'adda, F., De Paoli, F., & Palmonari, M. (2022). LamAPI: a Comprehensive Tool for String-based Entity Retrieval with Type-base Filters. In 17th ISWC workshop on ontology matching (OM).

[Azad et al 2019] Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. Information Processing & Management, 56(5), 1698-1735.Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. Information Processing & Management, 56(5), 1698-1735.

[Bereta et al 2019] K. Bereta, G. Xiao, and M. Koubarakis, "Ontop-spatial: Ontop of geospatial databases," Journal of Web Semantics, vol. 58, p. 100514, Oct. 2019, doi: 10.1016/j.websem.2019.100514.

[Botoeva et al 2018] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, and G. Xiao, "A Generalized Framework for Ontology-Based Data Access," in AI*IA 2018 – Advances in Artificial Intelligence, C. Ghidini, B. Magnini, A. Passerini, and P. Traverso, Eds., in Lecture Notes in Computer Science, vol. 11298. Cham: Springer International Publishing, 2018, pp. 166–180. doi: 10.1007/978-3-030-03840-3_13.

[Botoeva et al 2019] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, and G. Xiao, "Ontology-based data access – Beyond relational sources," IA, vol. 13, no. 1, pp. 21–36, Aug. 2019, doi: 10.3233/IA-190023.

[Brandt et al 2018] S. Brandt, E. G. Kalaycı, V. Ryzhikov, G. Xiao, and M. Zakharyaschev, "Querying Log Data with Metric Temporal Logic (Technical Report)." arXiv, Aug. 16, 2018. Accessed: Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/1703.08982

[Calvanese et al 2016] D. Calvanese et al., "Ontop: Answering SPARQL queries over relational databases," SW, vol. 8, no. 3, pp. 471–487, Dec. 2016, doi: 10.3233/SW-160217.

[Cutrona et al 2021] Cutrona, V., Puleri, G., Bianchi, F., & Palmonari, M. (2021, September). NEST: Neural Soft Type Constraints to Improve Entity Linking in Tables. In SEMANTiCS (pp. 29-43).

[De Cao et al 2022] De Cao, N., Wu, L., Popat, K., Artetxe, M., Goyal, N., Plekhanov, M., ... & Petroni, F. (2022). Multilingual autoregressive entity linking. Transactions of the Association for Computational Linguistics, 10, 274-290.

[Floridi et al 2020] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30, 681-694.

[Ghufran et al 2017] Ghufran, M., Bennacer, N., & Quercini, G. (2017, May). Wikipedia-based extraction of key information from resumes. In 2017 11th international conference on research challenges in information science (RCIS) (pp. 135-145). IEEE.

[Hambarde et al 2023] Hambarde, K. A., & Proenca, H. (2023). Information Retrieval: Recent Advances and Beyond. arXiv preprint arXiv:2301.08801.

[Haw et al 2017] S.-C. Haw, J. W. May, and S. Subramaniam, "Mapping Relational Databases to Ontology Representation: A Review," in Proceedings of the International Conference on Digital Technology in Education - ICDTE '17, Taipei, Taiwan: ACM Press, 2017, pp. 54–58. doi: 10.1145/3134847.3134852.

[Hellmann et al 2010] Hellmann, S., Unbehauen, J., Zaveri, A., Lehmann, J., Auer, S., Tramp, S., ... & Nagy, H. (2010). Lod2 deliverable 3.1. 1: Report on knowledge extraction from structured sources. Technical Report 257943.

[Hert et al 2011] M. Hert, G. Reif, and H. C. Gall, "A Comparison of RDB-to-RDF Mapping Languages," in Proceedings of the 7th International Conference on Semantic Systems, in I-Semantics '11. New York, NY, USA: ACM, 2011, pp. 25–32. doi: 10.1145/2063518.2063522.

[Kalayci et al 2019] E. Güzel Kalayci, S. Brandt, D. Calvanese, V. Ryzhikov, G. Xiao, and M. Zakharyaschev, "Ontology–based access to temporal data with Ontop: A framework proposal," International Journal of Applied Mathematics and Computer Science, vol. 29, no. 1, pp. 17–30, Mar. 2019, doi: 10.2478/amcs-2019-0002.

[Konstantinou et al 2014] N. Konstantinou, D. Kouis, and N. Mitrou, "Incremental Export of Relational Database Contents into RDF Graphs," in Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), in WIMS '14. New York, NY, USA: ACM, 2014, p. 33:1-33:8. doi: 10.1145/2611040.2611082.

[Kontchakov et al 2014] R. Kontchakov, M. Rezk, M. Rodríguez-Muro, G. Xiao, and M. Zakharyaschev, "Answering SPARQL Queries over Databases under OWL 2 QL Entailment Regime," in The Semantic Web – ISWC 2014, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 552–567. doi: 10.1007/978-3-319-11964-9_35.

[Lanti et al 2016] D. Lanti, G. Xiao, and D. Calvanese, "Data Scaling in OBDA Benchmarks: The VIG Approach." arXiv, Jul. 29, 2016. Accessed: Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/1607.06343

[Li et al 2020] Li, B. Z., Min, S., Iyer, S., Mehdad, Y., & Yih, W. T. (2020). Efficient one-pass end-to-end entity linking for questions. arXiv preprint arXiv:2010.02413.

[Michel et al 2014] Michel, F., Montagnat, J., & Zucker, C. F. (2014). A survey of RDB to RDF translation approaches and tools (Doctoral dissertation, I3S).

[Onoe et al 2020] Onoe, Y., & Durrett, G. (2020, April). Fine-grained entity typing for domain independent entity linking. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 8576-8583).

[Pequeno et al 2014] V. M. Pequeno, V. M. P. Vidal, M. A. Casanova, L. E. T. Neto, and H. Galhardas, "Specifying complex correspondences between relational schemas and RDF models for generating customized R2RML mappings," in Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14, Porto, Portugal: ACM Press, 2014, pp. 96–104. doi: 10.1145/2628194.2628233.

[Raiman et al 2018] Raiman, J., & Raiman, O. (2018, April). Deeptype: multilingual entity linking by neural type sy stem evolution. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[Sahoo et al 2009] Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., ... & Ezzat, A. (2009). A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Report, 1, 113-130.

[Sequeda et al 2011] J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker, "Survey of directly mapping SQL databases to the Semantic Web," The Knowledge Engineering Review, vol. 26, no. 04, pp. 445–486, Dec. 2011, doi: 10.1017/S0269888911000208.

[Spanos et al 2012] Spanos, D. E., Stavrou, P., & Mitrou, N. (2012). Bringing relational databases into the semantic web: A survey. Semantic Web, 3(2), 169-209.

[Tenney et al 2019] Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. arXiv preprint arXiv:1905.05950.

[Xiao et al 2019] G. Xiao, L. Ding, B. Cogrel, and D. Calvanese, "Virtual Knowledge Graphs: An Overview of Systems and Use Cases," Data Intellegence, vol. 1, no. 3, pp. 201–223, Jun. 2019, doi: 10.1162/dint_a_00011.

8.2 Data discovery

[Abbas et al. 2021] Abbas, A. E., Agahari, W., van de Ven, M., Zuiderwijk, A., & de Reuver, M. (2021). Business Data Sharing through Data Marketplaces: A Systematic Literature Review. Journal of Theoretical and Applied Electronic Commerce Research, 16(7), 3321–3339. https://doi.org/10.3390/jtaer16070180

[Abedjan et al. 2019] Abedjan, Z., Golab, L., Naumann, F., & Papenbrock, T. (2019). Data Profiling Tools. In Data Profiling (pp. 97-101). Cham: Springer International Publishing.

[Almarimi & al. 2019] Almarimi, N., Ouni, A., Bouktif, S., Mkaouer, M. W., Kula, R. G., & Saied, M. A. (2019). Web service API recommendation for automated mashup creation using multi-objective evolutionary search. Applied Soft Computing, 85, 105830.

[Arik & Pfister 2021] Arik, S. Ö., & Pfister, T. (2021, May). TabNet: Attentive interpretable tabular learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 8, pp. 6679-6687).

[Bogatu et al. 2022] Bogatu, A., Paton, N. W., Douthwaite, M., & Freitas, A. (2022, March). Voyager: Data Discovery and Integration for Data Science. In Proceedings 25th International Conference on Extending Database Technology (EDBT 2022).

[Borisov et al. 2022] Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). Deep neural networks and tabular data: A survey. IEEE Transactions on Neural Networks and Learning Systems.

[Brickley et al. 2019] Brickley, D., Burgess, M., & Noy, N. (2019, May). Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In The World Wide Web Conference (pp. 1365-1375).

[Bucher et al 2021] Bucher, T. C., Jiang, X., Meyer, O., Waitz, S., Hertling, S., & Paulheim, H. (2021). scikit-learn Pipelines Meet Knowledge Graphs: The Python kgextension Package. In The Semantic Web: ESWC 2021 Satellite Events: Virtual Event, June 6–10, 2021, Revised Selected Papers 18 (pp. 9-14). Springer International Publishing.

[Cafarella et al 2008] Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: Webta- bles: exploring the power of tables on the web. Proc. VLDB Endow. 1(1), 538–549 (2008). https://doi.org/10.14778/1453856. 1453916

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Campinas et al 2012] Campinas, S., Perry, T. E., Ceccarelli, D., Delbru, R., & Tummarello, G. (2012, September). Introducing RDF graph summary with application to assisted SPARQL formulation. In 2012 23rd International Workshop on Database and Expert Systems Applications (pp. 261-266). IEEE

[Cashman et al 2020] Cashman, D., Xu, S., Das, S., Heimerl, F., Liu, C., Humayoun, S. R., ... & Chang, R. (2020). Cava: A visual analytics system for exploratory columnar data augmentation using knowledge graphs. IEEE Transactions on Visualization and Computer Graphics, 27(2), 1731-1741.

[Čebirić et al 2015] Čebirić, Š., Goasdoué, F., & Manolescu, I. (2015, June). Query-oriented summarization of RDF graphs. In Data Science: 30th British International Conference on Databases, BICOD 2015, Edinburgh, UK, July 6-8, 2015, Proceedings (pp. 87-91). Cham: Springer International Publishing

[Chapman & al. 2020] Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L. D., Kacprzak, E., & Groth, P. (2020). Dataset search: a survey. The VLDB Journal, 29(1), 251-272.

[Ciavotta & al 2022] Ciavotta, M., Cutrona, V., De Paoli, F., Nikolov, N., Palmonari, M., & Roman, D. (2022). Supporting Semantic Data Enrichment at Scale. In Technologies and Applications for Big Data Value (pp. 19-39). Cham: Springer International Publishing.

[Consens et al 2015] Consens, M. P., Fionda, V., Khatchadourian, S., & Pirro, G. (2015). S+ epps: construct and explore bisimulation summaries, plus optimize navigational queries; all on existing sparql systems. Proceedings of the VLDB Endowment, 8(12), 2028-2031

[Cutrona et al 2019] Cutrona, V., Ciavotta, M., De Paoli, F., & Palmonari, M. (2019). ASIA: a Tool for Assisted Semantic Interpretation and Annotation of Tabular Data. In ISWC Satellites (pp. 209-212).

[Das Sarma et al 2012] Das Sarma, A., Fang, L., Gupta, N., Halevy, A., Lee, H., Wu, F., Xin, R., Yu, C.: Finding related tables. In: Proceedings of the 2012 ACM SIGMOD International Conference on Manage- ment of Data, pp. 817–828. ACM (2012). https://doi.org/10.1145/ 2213836.2213962

[Diao et al 2019] Diao, Y., Guzewicz, P., Manolescu, I., & Mazuran, M. (2019). Spade: A modular framework for analytical exploration of RDF graphs. Proceedings of the VLDB Endowment (PVLDB), 12(12), 1926-1929

[Dong et al 2022] Dong, H., Cheng, Z., He, X., Zhou, M., Zhou, A., Zhou, F., ... & Zhang, D. (2022). Table pretraining: A survey on model architectures, pretraining objectives, and downstream tasks. arXiv preprint arXiv:2201.09745.

[Dudáš et al 2015] Dudáš, M., Svátek, V., & Mynarz, J. (2015). Dataset summary visualization with lodsight. In The Semantic Web: ESWC 2015 Satellite Events: ESWC 2015 Satellite Events, Portorož, Slovenia, May 31–June 4, 2015, Revised Selected Papers 12 (pp. 36-40). Springer International Publishing

[Fariss et al 2019] Fariss, M., El Allali, N., Asaidi, H., & Bellouki, M. (2019). Review of ontology based approaches for web service discovery. In Smart Data and Computational Intelligence: Proceedings of the International Conference on Advanced Information Technology, Services and Systems (AIT2S-18) Held on October 17–18, 2018 in Mohammedia 3 (pp. 78-87). Springer International Publishing.

[Farshidi & Zhao 2022] Farshidi, S., Zhao, Z. (2022). An Adaptable Indexing Pipeline for Enriching Meta Information of Datasets from Heterogeneous Repositories. In: Gama, J., Li, T., Yu, Y., Chen, E., Zheng, Y., Teng, F. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2022. Lecture Notes in Computer Science(), vol 13281. Springer, Cham. https://doi.org/10.1007/978-3-031-05936-0_37

[Femke Ongenae 2021] Ongenae, F. (2021). MAGIC: Mining an Augmented Graph using INK, starting from a CSV.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Harari & Katz 2022] Harari, A., & Katz, G. (2022, May). Few-shot tabular data enrichment using fine-tuned transformer architectures. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1577-1591).

[Konrath et al 2012] Konrath, M., Gottron, T., Staab, S., & Scherp, A. (2012). Schemex—efficient construction of a data catalogue by stream-based indexing of linked data. Journal of Web Semantics, 16, 52-58.

[Lehmberg et al 2015] Lehmberg, O., Ritze, D., Ristoski, P., Meusel, R., Paulheim, H., Bizer, C.: The mannheim search join engine. J. Web Semant. 35, 159–166 (2015). https://doi.org/10.1016/j.websem.2015.05.001

[Liu & Zhang 2020] Liu, Z., & Zhang, A. (2020). Sampling for big data profiling: A survey. IEEE Access, 8, 72713-72726.

[Loetpipatwanich & Vichitthamaros 2020] Loetpipatwanich, S., & Vichitthamaros, P. (2020, September). Sakdas: A Python Package for Data Profiling and Data Quality Auditing. In 2020 1st International Conference on Big Data Analytics and Practices (IBDAP) (pp. 1-4). IEEE.

[Louati et al 2011] Louati, A., Aufaure, M. A., Lechevallier, Y., & Chatenay-Malabry, F. (2011, October). Graph Aggregation: Application to Social Networks. In HDSDA (pp. 157-177).

[Mihindukulasooriya et al 2015] Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., & Gómez-Pérez, A. (2015). Loupe-An Online Tool for Inspecting Datasets in the Linked Data Cloud. ISWC (Posters & Demos), 1, 1.

[Möller et al 2019] Möller, M. L., Berton, N., Klettke, M., Scherzinger, S., & Störl, U. (2019). jHound: large-scale profiling of open JSON data. BTW 2019.

[Nowaz et al 2022] Nawaz, M. S., Khan, S. U. R., Hussain, S., & Iqbal, J. (2022). A study on application programming interface recommendation: state-of-the-art techniques, challenges and future directions. Library Hi Tech, (ahead-of-print).

[Panziera et al 2012] Panziera, L., Comerio, M., De Paoli, F., & BATINI, C. (2012). Quality-driven extraction, fusion and matchmaking of semantic web api descriptions. Journal of Web Engineering, 247-268.

[Papenbrock et al 2015] Papenbrock, T., Bergmann, T., Finke, M., Zwiener, J., & Naumann, F. (2015). Data profiling with metanome. Proceedings of the VLDB Endowment, 8(12), 1860-1863.

[Qarabaqi & Riedewald 2015] Qarabaqi, B., & Riedewald, M. (2015). Merlin: Exploratory analysis with imprecise queries. IEEE Transactions on Knowledge and Data Engineering, 28(2), 342-355.

[Riondato et al 2017] Riondato, M., García-Soriano, D., & Bonchi, F. (2017). Graph summarization with quality guarantees. Data mining and knowledge discovery, 31, 314-349

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

[Santos et al. 2022] A. Santos, A. Bessa, C. Musco and J. Freire, "A Sketch-based Index for Correlated Dataset Search," 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022, pp. 2928-2941, doi: 10.1109/ICDE53745.2022.00264.

[Santos et al. 2022] A. Santos, A. Bessa, C. Musco and J. Freire, "A Sketch-based Index for Correlated Dataset Search," 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022, pp. 2928-2941, doi: 10.1109/ICDE53745.2022.00264.

[Sarthou-Camy & al 2022] Sarthou-Camy, C., Jourdain, G., Chabot, Y., Monnin, P., Deuzé, F., Huynh, V. P., ... & Troncy, R. (2022). DAGOBAH UI: a new hope for semantic table interpretation. In The Semantic Web: ESWC 2022 Satellite Events: Hersonissos, Crete, Greece, May 29–June 2, 2022, Proceedings (pp. 107-111). Cham: Springer International Publishing.

[Schaible et al 2016] Schaible, J., Gottron, T., & Scherp, A. (2016). Termpicker: Enabling the reuse of vocabulary terms by exploiting data from the linked open data cloud. In The Semantic Web. Latest Advances and New Domains: 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29--June 2, 2016, Proceedings 13 (pp. 101-117). Springer International Publishing.

[Singh & Bedathur 2023] Singh, R., & Bedathur, S. (2023). Embeddings for Tabular Data: A Survey. arXiv preprint arXiv:2302.11777.

[Song et al 2018] Song, Q., Wu, Y., Lin, P., Dong, L. X., & Sun, H. (2018). Mining summaries for knowledge graph search. IEEE Transactions on Knowledge and Data Engineering, 30(10), 1887-1900.

[Wang et al. 2020] Xu Wang, Zhisheng Huang, and Frank van Harmelen. Ontology-based semantic similarity approach for biomedical dataset retrieval. In Health Information Science, pages 49–60. Springer International Publishing, 2020

[Wang et al. 2022] Xu Wang, Frank van Harmelen, and Zhisheng Huang. Recommending scientific datasets using author networks in ensemble methods. Data Science, 5(2):167–193, July 2022

[Wei et al 2022] Wei, M., Harzevili, N. S., Huang, Y., Wang, J., & Wang, S. (2022, May). Clear: contrastive learning for api recommendation. In Proceedings of the 44th International Conference on Software Engineering (pp. 376-387).

[Yakout et al 2012] Yakout, M., Ganjam, K., Chakrabarti, K., Chaudhuri, S.: Info- gather: entity augmentation and attribute discovery by holistic matching with web tables. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIG- MOD '12, pp. 97–108. ACM, New York, NY, USA (2012). https:// doi.org/10.1145/2213836.2213848

[Zhang & Balog 2018] Zhang, S., Balog, K.: Ad hoc table retrieval using semantic similarity. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23–27, 2018, pp. 1553–1562 (2018). https://doi.org/10.1145/3178876. 3186067

8.3 Data cleaning

[Alva Principe et al 2021] Alva Principe, R. A., Maurino, A., Palmonari, M., Ciavotta, M., & Spahiu, B. (2021). ABSTAT-HD: a scalable tool for profiling very large knowledge graphs. The VLDB Journal, 1-26.

[Bantilan N., 2020] Bantilan, N. (2020). pandera: Statistical data validation of pandas dataframes. In Proceedings of the Python in Science Conference (SciPy) (pp. 116-124).

[Carminé, R. L. A.,2021] Carminé, R. L. A. (2021). TruData: A Collaborative Platform for Data Cleaning.

[Ciavotta et al 2022] Ciavotta, M., Cutrona, V., De Paoli, F., Nikolov, N., Palmonari, M., & Roman, D. (2022). Supporting Semantic Data Enrichment at Scale. In Technologies and Applications for Big Data Value (pp. 19-39). Cham: Springer International Publishing.

[Cutrona et al 2019] Cutrona, V., Ciavotta, M., De Paoli, F., & Palmonari, M. (2019). ASIA: a Tool for Assisted Semantic Interpretation and Annotation of Tabular Data. In ISWC Satellites (pp. 209-212).

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Cutrona et al 2019] Cutrona, V., De Paoli, F., Košmerlj, A., Nikolov, N., Palmonari, M., Perales, F., & Roman, D. (2019). Semantically-enabled optimization of digital marketing campaigns. In The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18 (pp. 345-362). Springer International Publishing.

[Cutrona et al 2020] Cutrona, V., Bianchi, F., Jiménez-Ruiz, E., & Palmonari, M. (2020, November). Tough tables: Carefully evaluating entity linking for tabular data. In International Semantic Web Conference (pp. 328-343). Springer, Cham.

[Di Noia et al 2018] Di Noia, T., Magarelli, C., Maurino, A., Palmonari, M., & Rula, A. (2018, June). Using ontology-based data summarization to develop semantics-aware recommender systems. In European Semantic Web Conference (pp. 128-144). Springer, Cham.

[Di Noia et al 2018] Tommaso di Noia, Andrea Maurino, Corrado Magarelli, Matteo Palmonari, and Anisa Rula. Using ontology-based data summarization to develop semantics-aware recommender systems. In The Semantic Web - ESWC 2018, 2018

[Ilyas, I. F., 2016] Ilyas, I. F. (2016). Effective Data Cleaning with Continuous Evaluation. IEEE Data Eng. Bull., 39(2), 38-46.

[Krawczyk et al., 2022] Krawczyk, S., ben Izzy, E., & Quinn, D. (2022). Hamilton: enabling software engineering best practices for data transformations via generalized dataflow graphs.

[Krishnan et al., 2016] Krishnan, S., Wang, J., Wu, E., Franklin, M. J., & Goldberg, K. (2016). Activeclean: Interactive data cleaning while learning convex loss models. arXiv preprint arXiv:1601.03797.

[Liu et al., 2022] Liu, Z., Zhou, Z., & Rekatsinas, T. (2022). Picket: guarding against corrupted data in tabular data during learning and inference. The VLDB Journal, 1-29.

[Ma et al., 2019] Ma, E. J., Barry, Z., Zuckerman, S., & Sailer, Z. (2019). Pyjanitor: A cleaner API for cleaning data. In Proceedings of the 18th Python in Science Conference (pp. 50-53).

[Mahdavi et al., 2020] Mahdavi, M., & Abedjan, Z. (2020). Baran: Effective error correction via a unified context representation and transfer learning. Proceedings of the VLDB Endowment, 13(12), 1948-1961.

[Massri et al 2023] Massri, M.Besher, Spahiu, Blerina, Grobelnik, Marko, Alexiev, Vladimir, Palmonari, Matteo, & Roman, Dumitru. (2023). Towards InnoGraph: A Knowledge Graph for AI Innovation. 3rd International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment (Sci-K 2023), Austin, Texas, USA. https://doi.org/10.1145/3543873.3587614

[Milani et al., 2021] Milani, A. M. P., Loges, L. A., Paulovich, F. V., & Manssour, I. H. (2021). PrAVA: Preprocessing profiling approach for visual analytics. Information Visualization, 20(2-3), 101-122.

[Musleh et al., 2020] Musleh, M., Ouzzani, M., Tang, N., & Doan, A. (2020, June). Coclean: Collaborative data cleaning. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (pp. 2757-2760).

[Petrova-Antonova et al., 2020] Petrova-Antonova, D., & Tancheva, R. (2020). Data cleaning: A case study with OpenRefine and Trifacta Wrangler. In Quality of Information and Communications Technology: 13th International Conference, QUATIC 2020, Faro, Portugal, September 9–11, 2020, Proceedings 13 (pp. 32-40). Springer International Publishing.

[Rekatsinas et al., 2017] Rekatsinas, T., Chu, X., Ilyas, I. F., & Ré, C. (2017). Holoclean: Holistic data repairs with probabilistic inference. arXiv preprint arXiv:1702.00820.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Sajid et al., 2019] Sajid, S., von Zernichow, B. M., Soylu, A., & Roman, D. (2019). Predictive data transformation suggestions in grafterizer using machine learning. In Metadata and Semantic Research: 13th International Conference, MTSR 2019, Rome, Italy, October 28–31, 2019, Revised Selected Papers (pp. 137-149). Springer International Publishing.

[Spahiu et al 2016] Spahiu, B., Porrini, R., Palmonari, M., Rula, A., & Maurino, A. (2016, May). ABSTAT: ontology-driven linked data summaries with pattern minimalization. In European Semantic Web Conference (pp. 381-395). Springer, Cham.

[Spahiu et al 2018] Spahiu, B., Maurino, A., & Palmonari, M. (2018, October). Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL. In ISWC (Best Workshop Papers) (pp. 103-117).

[Spahiu et al 2023] Spahiu, B., Palmonari, M., Alva Principe, R. A., & Rula, A. Understanding the structure of knowledge graphs with ABSTAT profiles. Semantic Web, (Preprint), 1-27

[Sukhobok et al., 2017] Sukhobok, D., Nikolov, N., & Roman, D. (2017, August). Tabular data anomaly patterns. In 2017 International Conference on Big Data Innovations and Applications (Innovate-Data) (pp. 25-34). IEEE.

[Whang et al., 2023] Whang, S. E., Roh, Y., Song, H., & Lee, J. G. (2023). Data collection and quality challenges in deep learning: A data-centric ai perspective. The VLDB Journal, 1-23.

8.4 Data linking

[Abdelmageed et al 2021] Abdelmageed, N., & Schindler, S. (2021, June). Jentab: A toolkit for semantic table annotations. In Second International Workshop on Knowledge Graph Construction.

[Barlaug and Gulla 2021] Barlaug, Nils, and Jon Atle Gulla. "Neural networks for entity matching: A survey." ACM Transactions on Knowledge Discovery from Data 15.3 (2021)

[Batini & Scannapieco 2016] Batini, C., & Scannapieco, M. (2016). Data and information quality. Cham, Switzerland: Springer International Publishing.

[Bizer et al 2009] Volz, J., Bizer, C., Gaedke, M., & Kobilarov, G. (2009). Silk-a link discovery framework for the web of data. Ldow, 538, 53.

[Chen et al 2019] Chen, J., Jiménez-Ruiz, E., Horrocks, I., & Sutton, C. (2019, July). Colnet: Embedding the semantics of web tables for column type prediction. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 29-36).

[Kim et al 2014] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746-1751). Doha, Qatar: Association for Computational Linguistics.

[Chen et al 2022] Chen, S., Karaoglu, A., Negreanu, C., Ma, T., Yao, J. G., Williams, J., ... & Lin, C. Y. (2022). LinkingPark: An automatic semantic table interpretation system. Journal of Web Semantics, 74, 100733.

[Christophides et al 2020] Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2020). An overview of end-to-end entity resolution for big data. ACM Computing Surveys (CSUR), 53(6), 1-42.

[Cremaschi et al 2019] Cremaschi, M., Rula, A., Siano, A., & De Paoli, F. (2019). MantisTable: a tool for creating semantic annotations on tabular data. In The Semantic Web: ESWC 2019 Satellite Events: ESWC

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

2019 Satellite Events, Portorož, Slovenia, June 2–6, 2019, Revised Selected Papers 16 (pp. 18-23). Springer International Publishing.

[Deng et al 2022] Deng, X., Sun, H., Lees, A., Wu, Y., & Yu, C. (2022). Turl: Table understanding through representation learning. ACM SIGMOD Record, 51(1), 33-40.

[Guo et al 2022] Guo, L., Han, Y., Zhang, Q., & Chen, H. "Deep Reinforcement Learning for Entity Alignment." Findings of ACL 2022. (2022)

[Gupta et al 2012] Gupta, S., Szekely, P., Knoblock, C. A., Goel, A., Taheriyan, M., & Muslea, M. (2015). Karma: A system for mapping structured sources into the semantic web. In The Semantic Web: ESWC 2012 Satellite Events: ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012. Revised Selected Papers (pp. 430-434). Berlin, Heidelberg: Springer Berlin Heidelberg.

[Hulsebos et al 2019] Hulsebos, M., Hu, K., Bakker, M., Zgraggen, E., Satyanarayan, A., Kraska, T., ... & Hidalgo, C. (2019, July). Sherlock: A deep learning approach to semantic data type detection. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1500-1508).

[Huynh et al 2022] Huynh, V. P., Chabot, Y., Labbé, T., Liu, J., & Troncy, R. (2022). From Heuristics to Language Models: A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH. Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab).

[Jiménez-Ruiz & Cuenca Grau 2011] Jiménez-Ruiz, E., & Cuenca Grau, B. (2011). Logmap: Logic-based and scalable ontology matching. In The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10 (pp. 273-288). Springer Berlin Heidelberg.

[Koleva & al 2022] Koleva, A., Ringsquandl, M., Buckley, M., Hasan, R., & Tresp, V. (2022). Named Entity Recognition in Industrial Tables using Tabular Language Models. arXiv preprint arXiv:2209.14812.

[Leone et al 2022] Leone, M., Huber, S., Arora, A., García-Durán, A., & West, R. (2022). A critical re-evaluation of neural methods for entity alignment. Proceedings of the VLDB Endowment, 15(8), 1712-1725.

[Li et al 2020] Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W. C. (2020). Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment, 14(1), 50-60.

[Li et al 2021] Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., & Tan, W. C. (2021). Deep entity matching: Challenges and opportunities. Journal of Data and Information Quality (JDIQ), 13(1), 1-17.

[Liu & Chabot et al 2022] Liu, J., Chabot, Y., Troncy, R., Huynh, V. P., Labbé, T., & Monnin, P. (2022). From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. Journal of Web Semantics, 100761.

[Liu & Huynh et al 2022] Liu, J., Huynh, V. P., Chabot, Y., & Troncy, R. (2022, October). Radar Station: Using KG Embeddings for Semantic Table Interpretation and Entity Disambiguation. In The Semantic Web–ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings (pp. 498-515). Cham: Springer International Publishing.

[Liu et al 2022] Liu, X., Hong, H., Wang, X., Chen, Z., Kharlamov, E., Dong, Y., & Tang, J. "SelfKG: Self-Supervised Entity Alignment in Knowledge Graphs." Proceedings of the ACM Web Conference (2022)

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Mudgal et al 2018] Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., ... & Raghavendra, V. (2018, May). Deep learning for entity matching: A design space exploration. In Proceedings of the 2018 International Conference on Management of Data (pp. 19-34).

[Nadkarni et al 2011] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5), 544-551.

[Nentwig et al 2017] Nentwig, M., Hartung, M., Ngonga Ngomo, A. C., & Rahm, E. (2017). A survey of current link discovery frameworks. Semantic Web, 8(3), 419-436.

[Ngonga Ngomo et al 2021] Ngonga Ngomo, A. C., Sherif, M. A., Georgala, K., Hassan, M. M., Dreßler, K., Lyko, K., ... & Soru, T. (2021). LIMES: a framework for link discovery on the semantic web. KI-Künstliche Intelligenz, 1-11.

[Nguyen et al 2021] Nguyen, P., Yamada, I., Kertkeidkachorn, N., Ichise, R., & Takeda, H. (2021, October). SemTab 2021: Tabular Data Annotation with MTab Tool. In SemTab@ ISWC (pp. 92-101).

[Obraczka & Ngomo 2019] Obraczka, D., & Ngomo, A. C. N. (2019). Dragon: Decision tree learning for link discovery. In Web Engineering: 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11–14, 2019, Proceedings 19 (pp. 441-456). Springer International Publishing.

[Paganelli et al 2022] Paganelli, M., Buono, F. D., Baraldi, A., & Guerra, F. "Analyzing how BERT performs entity matching." VLDB Endowment 15.8 (2022)

[Shigapov et al 2020] Shigapov, R., Zumstein, P., Kamlah, J., Oberländer, L., Mechnich, J., & Schumm, I. (2020). bbw: Matching csv to wikidata via meta-lookup. In CEUR Workshop Proceedings (Vol. 2775, pp. 17-26). RWTH Aachen.

[Singh et al 2021] Singh, G., Singh, S., Wong, J., & Saffari, A. (2021). Relation Extraction from Tables using Artificially Generated Metadata. arXiv preprint arXiv:2108.10750.

[Steenwinckel et al 2021] Steenwinckel, B., De Turck, F., & Ongenae, F. (2021). MAGIC: Mining an Augmented Graph using INK, starting from a CSV. In SemTab@ ISWC (pp. 68-78).

[Suchanek et al 2011] Suchanek, F. M., Abiteboul, S., & Senellart, P. (2011). PARIS: Probabilistic Alignment of Relations, Instances, and Schema. Proceedings of the VLDB Endowment, 5(3).

[Suhara et al 2022] Suhara, Y., Li, J., Li, Y., Zhang, D., Demiralp, Ç., Chen, C., & Tan, W. C. (2022, June). Annotating columns with pre-trained language models. In Proceedings of the 2022 International Conference on Management of Data (pp. 1493-1503).

[Suhara et al 2022] Suhara, Y., Li, J., Li, Y., Zhang, D., Demiralp, Ç., Chen, C., & Tan, W. C. (2022, June). Annotating columns with pre-trained language models. In Proceedings of the 2022 International Conference on Management of Data (pp. 1493-1503).

[Sun et al 2020] Sun, Z., Zhang, Q., Hu, W., Wang, C., Chen, M., Akrami, F., & Li, C. "A benchmarking study of embedding-based entity alignment for knowledge graphs." Proceedings of the VLDB Endowment 13.12 (2020)

[Trabelsi et al 2022] Trabelsi, M., Heflin, J., & Cao, J. (2022, February). DAME: Domain Adaptation for Matching Entities. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (pp. 1016-1024).

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Trisedya et al 2019] Trisedya, B. D., Qi, J., & Zhang, R. (2019, July). Entity alignment between knowledge graphs using attribute embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 297-304).

[Tu et al 2022] Tu, J., Han, X., Fan, J., Tang, N., Chai, C., Li, G., & Du, X. (2022). Dader: hands-off entity resolution with domain adaptation. Proceedings of the VLDB Endowment, 15(12), 3666-3669.

[Vu & Knoblock 2022] Vu, B., & Knoblock, C. A. (2022). SAND: A Tool for Creating Semantic Descriptions of Tabular Sources. In The Semantic Web: ESWC 2022 Satellite Events: Hersonissos, Crete, Greece, May 29–June 2, 2022, Proceedings (pp. 63-67). Cham: Springer International Publishing.

[Vu et al 2019] Vu, B., Knoblock, C., & Pujara, J. (2019, May). Learning semantic models of data sources using probabilistic graphical models. In The world wide web conference (pp. 1944-1953).

[Vu et al 2019] Vu, B., Knoblock, C., & Pujara, J. (2019, May). Learning semantic models of data sources using probabilistic graphical models. In The world wide web conference (pp. 1944-1953).

[Vu et al 2021] Vu, B., Knoblock, C. A., Szekely, P., Pham, M., & Pujara, J. (2021). A graph-based approach for inferring semantic descriptions of wikipedia tables. In The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20 (pp. 304-320). Springer International Publishing.

[Vu et al 2021] Vu, B., Knoblock, C. A., Szekely, P., Pham, M., & Pujara, J. (2021). A graph-based approach for inferring semantic descriptions of wikipedia tables. In The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20 (pp. 304-320). Springer International Publishing.

[Vu et al 2022] Vu, B., & Knoblock, C. A. (2022). SAND: A Tool for Creating Semantic Descriptions of Tabular Sources. In The Semantic Web: ESWC 2022 Satellite Events: Hersonissos, Crete, Greece, May 29–June 2, 2022, Proceedings (pp. 63-67). Cham: Springer International Publishing.

[Weikum et al 2021] Weikum, G., Dong, X. L., Razniewski, S., & Suchanek, F. (2021). Machine knowledge: Creation and curation of comprehensive knowledge bases. Foundations and Trends® in Databases, 10(2-4), 108-490.

[Zeng et al 2021] Zeng, K., Li, C., Hou, L., Li, J., & Feng, L. (2021). A comprehensive survey of entity alignment for knowledge graphs. AI Open, 2, 1-13.

[Zhang et al 2017] Zhang, Z. (2017). Effective and efficient semantic table interpretation using tableminer+. Semantic Web, 8(6), 921-957.

[Zhang et al 2019] Zhang, D., Suhara, Y., Li, J., Hulsebos, M., Demiralp, Ç., & Tan, W. C. (2019). Sato: Contextual semantic type detection in tables. arXiv preprint arXiv:1911.06311.

[Zhang et al 2022] Zhang, R., Trisedya, B. D., Li, M., Jiang, Y., & Qi, J. (2022). A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. The VLDB Journal, 31(5), 1143-1168.

[Zhou et al 2021] Zhou, Y., Singh, S., & Christodoulopoulos, C. (2021, October). Tabular data concept type detection using star-transformers. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 3677-3681).

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

8.5 Information Extraction

[Akbik et al., 2018] Akbik, Alan, Duncan Blythe, and Roland Vollgraf. "Contextual string embeddings for sequence labeling." In Proceedings of the 27th international conference on computational linguistics, pp. 1638-1649. 2018.

[Brank et al., 2017] Brank, Janez, Gregor Leban, and Marko Grobelnik. "Annotating documents with relevant wikipedia concepts." Proceedings of SiKDD 472 (2017).

[Brin et al., 1998] Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." Computer networks and ISDN systems 30, no. 1-7 (1998): 107-117.

[Kenton et al., 2019] Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In Proceedings of NAACL-HLT, pp. 4171-4186. 2019.

[Lai et al., 2022] Lai, Viet Dac. "Event Extraction: A Survey." arXiv preprint arXiv:2210.03419 (2022).

[Li et al., 2022] Li, Qian, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng et al. "A Survey on Deep Learning Event Extraction: Approaches and Applications." IEEE Transactions on Neural Networks and Learning Systems (2022).

[Lim et al., 2019] Lim, Chae-Gyun, Young-Seob Jeong, and Ho-Jin Choi. "Survey of temporal information extraction." Journal of Information Processing Systems 15, no. 4 (2019): 931-956.

[lu et al., 2021] Lu, Yaojie, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. "Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction." In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2795-2806. 2021.

[Niklaus et al., 2018] Niklaus, Christina, Matthias Cetto, André Freitas, and Siegfried Handschuh. "A Survey on Open Information Extraction." In Proceedings of the 27th International Conference on Computational Linguistics, pp. 3866-3878. 2018.

[Poerner et al., 2020] Poerner, Nina, Ulli Waltinger, and Hinrich Schütze. "E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT." In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 803-818. 2020.

[Raffel et al., 2020] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21, no. 1 (2020): 5485-5551.

[Raiman et al., 2018] Raiman, Jonathan, and Olivier Raiman. "Deeptype: multilingual entity linking by neural type system evolution." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[Sainz et al., 2021] Sainz, Oscar, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. "Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 1199-1212. 2021.

[Sevgili et al., 2022] Sevgili, Ö., Shelmanov, A., Arkhipov, M., Panchenko, A. and Biemann, C., 2022. Neural entity linking: A survey of models based on deep learning.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

[Shen & al 2014] Shen, Wei, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions." IEEE Transactions on Knowledge and Data Engineering 27.2 (2014): 443-460.

[Soares et al., 2019] Soares, Livio Baldini, Nicholas Fitzgerald, Jeffrey Ling, and Tom Kwiatkowski. "Matching the Blanks: Distributional Similarity for Relation Learning." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2895-2905. 2019.

[Strötgen et al., 2013] Strötgen, Jannik, and Michael Gertz. "Multilingual and cross-domain temporal tagging." Language Resources and Evaluation 47 (2013): 269-298.

[Zhou et al., 2021] Zhou, Wenxuan, and Muhao Chen. "An improved baseline for sentence-level relation extraction." arXiv preprint arXiv:2102.01373 (2021).

8.6 Classification

[Arthur et al., 2007] Arthur, David, and Sergei Vassilvitskii. "K-means++ the advantages of careful seeding." In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027-1035. 2007.

[Conneau et al., 2020] Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. "Unsupervised Cross-lingual Representation Learning at Scale." In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440-8451. 2020.

[Cortes et al., 1995] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20 (1995): 273-297.

[Dagan et al., 2006] Dagan, Ido, Oren Glickman, and Bernardo Magnini. "The pascal recognising textual entailment challenge." In Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment: First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers, pp. 177-190. Springer Berlin Heidelberg, 2006.

[Fortuna et al., 2006] Fortuna, Blaž, Dunja Mladenič, and Marko Grobelnik. "Semi-automatic construction of topic ontologies." In Semantics, Web and Mining: Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3-7, 2005, Revised Selected Papers, pp. 121-131. Springer Berlin Heidelberg, 2006.

[Lewis et al., 1994] Lewis, David D., and Jason Catlett. "Heterogeneous uncertainty sampling for supervised learning." In Machine learning proceedings 1994, pp. 148-156. Morgan Kaufmann, 1994.

[Lloyd et al., 1982] Lloyd, Stuart. "Least squares quantization in PCM." IEEE transactions on information theory 28, no. 2 (1982): 129-137.

[MacQueen et al., 1967] MacQueen, J. "Classification and analysis of multivariate observations." In 5th Berkeley Symp. Math. Statist. Probability, pp. 281-297. Los Angeles LA USA: University of California, 1967.

[Pasunuru et al., 2021] Pasunuru, Ramakanth, Veselin Stoyanov, and Mohit Bansal. "Continual few-shot learning for text classification." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 5688-5702. 2021.

[Xue et al., 2021] Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer." In

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 483-498. 2021.

[Yin et al., 2019] Yin, Wenpeng, Jamaal Hay, and Dan Roth. "Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3914-3923. 2019.

8.7 Scaling, replication and adaptation

[Blinowski et al., 2022] Blinowski, G., Ojdowska, A., & Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. IEEE Access, 10, 20357-20374.

[Corodescu et al., 2021] Corodescu, A. A., Nikolov, N., Khan, A. Q., Soylu, A., Matskin, M., Payberah, A. H., & Roman, D. (2021). Big data workflows: Locality-aware orchestration using software containers. Sensors, 21(24), 8212.

[De Sensi et al., 2022] De Sensi, D., De Matteis, T., Taranov, K., Di Girolamo, S., Rahn, T., & Hoefler, T. (2022). Noise in the Clouds: Influence of Network Performance Variability on Application Scalability. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 6(3), 1-27.

[Dessalk et al., 2020] Dessalk, Y. D., Nikolov, N., Matskin, M., Soylu, A., & Roman, D. (2020, November). Scalable execution of big data workflows using software containers. In Proceedings of the 12th International Conference on Management of Digital EcoSystems (pp. 76-83).

[Henning et al., 2021] Henning, S., & Hasselbring, W. (2021, April). How to measure scalability of distributed stream processing engines?. In Companion of the ACM/SPEC international conference on performance engineering (pp. 85-88).

[Henning et al., 2022] Henning, S., & Hasselbring, W. (2022). A configurable method for benchmarking scalability of cloud-native applications. Empirical Software Engineering, 27(6), 143.

[Herbst et al., 2013] Herbst, N. R., Kounev, S., & Reussner, R. H. (2013, June). Elasticity in Cloud Computing: What It Is, and What It Is Not. In ICAC (Vol. 13, No. 2013, pp. 23-27).

[Mitchell et al., 2019] Mitchell, R., Pottier, L., Jacobs, S., da Silva, R. F., Rynge, M., Vahi, K., & Deelman, E. (2019, December). Exploration of workflow management systems emerging features from users perspectives. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 4537-4544). IEEE.

[Nikolov et al., 2021] Nikolov, N., Dessalk, Y. D., Khan, A. Q., Soylu, A., Matskin, M., Payberah, A. H., & Roman, D. (2021). Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers. Internet of Things, 16, 100440.

8.8 Streaming data

Ali, Mohammed, Ali Alqahtani, Mark W. Jones, and Xianghua Xie. "Clustering and classification for time series data in visual analytics: A survey." IEEE Access 7 (2019): 181314-181338.

Alqahtani, Ali, Mohammed Ali, Xianghua Xie, and Mark W. Jones. "Deep time-series clustering: A review." Electronics 10, no. 23 (2021): 3001.

Guillaume, Antoine, Christel Vrain, and Elloumi Wael. "Time series classification for predictive maintenance on event logs." arXiv preprint arXiv:2011.10996 (2020).

Hennig, Martin, Manfred Grafinger, Detlef Gerhard, Stefan Dumss, and Patrick Rosenberger. "Comparison of time series clustering algorithms for machine state detection." Procedia CIRP 93 (2020): 1352-1357.

Javed, Ali, Byung Suk Lee, and Donna M. Rizzo. "A benchmark study on time series clustering." Machine Learning with Applications 1 (2020): 100001.

Leban, Gregor, Blaz Fortuna, Janez Brank, and Marko Grobelnik. "Event registry: learning about world events from news." In Proceedings of the 23rd International Conference on World Wide Web, pp. 107-110. 2014.

Sadaei, Hossein Javedani, Petrônio Cândido de Lima e Silva, Frederico Gadelha Guimaraes, and Muhammad Hisyam Lee. "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series." Energy 175 (2019): 365-377.

Sen, Rajat, Hsiang-Fu Yu, and Inderjit S. Dhillon. "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting." Advances in neural information processing systems 32 (2019).

Stopar, Luka, Primoz Skraba, Marko Grobelnik, and Dunja Mladenic. "Streamstory: exploring multivariate time series on multiple scales." IEEE transactions on visualization and computer graphics 25, no. 4 (2018): 1788-1802.

Vanderkam, Dan, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. "Nearest neighbor search in google correlate." (2013).

Zang, Dong, Jinhai Liu, and Huaizhen Wang. "Markov chain-based feature extraction for anomaly detection in time series and its industrial application." In 2018 Chinese Control And Decision Conference (CCDC), pp. 1059-1063. IEEE, 2018.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

8.9 Energy efficiency assessment and improvement

[Anthony et al. 2020] Anthony, Lasse F. Wolff, Benjamin Kanding, and Raghavendra Selvan. "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models." arXiv preprint arXiv:2007.03051 (2020).

[Du et al. 2022] Du, Nan, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun et al. "Glam: Efficient scaling of language models with mixture-of-experts." In International Conference on Machine Learning, pp. 5547-5569. PMLR, 2022.

[Henderson et al. 2020] Henderson et al. Towards the systematic reporting of the energy and carbon footprints of machine learning. 2020

[Klöpffer 1997] Klöpffer, Walter. "Life cycle assessment: From the beginning to the current state." Environmental Science and Pollution Research 4 (1997): 223-228.

[Lacoste et al. 2019] Lacoste, Alexandre, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. "Quantifying the carbon emissions of machine learning." arXiv preprint arXiv:1910.09700 (2019).

[Luccioni et al. 2022] Luccioni, Alexandra Sasha, Sylvain Viguier, and Anne-Laure Ligozat. "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model." arXiv preprint arXiv:2211.02001 (2022).

[Luccioni et al. 2023] Luccioni, Alexandra Sasha, and Alex Hernandez-Garcia. "Counting Carbon: A Survey of Factors Influencing the Emissions of Machine Learning." arXiv preprint arXiv:2302.08476 (2023).

[Patterson et al. 2022] Patterson, David, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. "The carbon footprint of machine learning training will plateau, then shrink." Computer 55, no. 7 (2022): 18-28.

[Schmidt et al. 2021] Schmidt et al. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. 2021

[Sevilla et al. 2022] Sevilla, Jaime, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. "Compute trends across three eras of machine learning." In 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2022.

[Wu et al. 2022] Wu, Carole-Jean, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang et al. "Sustainable ai: Environmental implications, challenges and opportunities." Proceedings of Machine Learning and Systems 4 (2022): 795-813.

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.