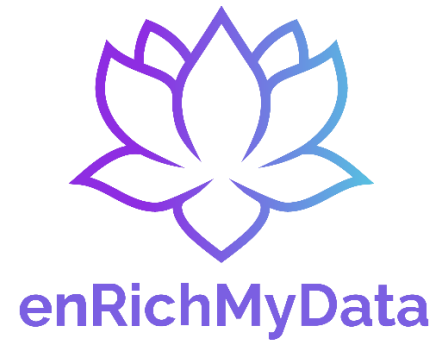


Enabling Data Enrichment Pipelines for AI-driven Business Products and Services

HORIZON-CL4-2021-DATA-01-03

D3.2 enRichMyData integrated toolbox v2, and testing and evaluation report

Work Package 3

Type of document:	Other
Dissemination level:	Public
Lead beneficiary:	CS
Authors:	Cosmin Udroiu, Cosmin Cara, Laurentiu Nicola (editors), all other technical partners contributors
Version:	1.0
Due Date of document:	30.06.2025
Delivery Date of document:	30.06.2025

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	05.03.2024	Cosmin Udroiu (CS)	Structure and ToC
0.2	01.10.2024	All partners	Version 1.0 corresponding to the enRichMyData Integration Toolbox v2.0
0.3	15.06.2025	All partners	Version 2.0 corresponding to the enRichMyData Integration Toolbox v3.0
1.0	30.06.2025	Dumitru Roman	Final editing, version submitted

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

Executive summary

The comprehensive assessment and understanding of the enRichMyData toolbox is best achieved by jointly considering the two key documents, the D2.2 – "enRichMyData Tools" and D3.2 – "Integrated toolbox – Validation plan and testing and evaluation report". These documents have been developed in parallel to ensure a coherent and traceable development lifecycle from requirement definition through to validated deployment.

The D2.2 – "enRichMyData Tools v2" document serves as the foundation as it describes the key improvements and changes in each tool since the previous reporting period, associated KPIs used to monitor tool performance and licensing and openness considerations for reuse and integration. It establishes the initial baseline for functionality and performance expectations based on user needs and system requirements. The deliverable also presents the tool collections in the project that include (i) data discovery (DiscoverR), (ii) data wrapping (WrappR), (iii) data cleaning and transformation (CleanR), (iv) linking and extension (LinkR), (v) data structuring (StructR) and (vi) data classification (ClassifiR), whereas services include (i) interoperable access to datasets (ResourcR), (ii) support for scalable data enrichment pipelines (ScaleR), (iii) support to share and reuse data enrichment pipelines (ReusR), (iv) analysis of streaming data (StreamR) and (v) estimates of energy consumption and CO2 emission of data enrichment pipelines, thus supporting the implementation of "green" (i.e., energy efficient) data enrichment pipelines (GreenR).

Complementing this, the D3.2 – "Integrated toolbox – Validation plan and testing and evaluation report" captures the iterative development and release cycles of the toolbox, including evolutions, tool updates and added features in each release of the toolbox. It documents the verification and validation processes used to assess compliance with the user requirements and the previously defined KPIs. This includes test cases, validation scenarios, and an evaluation of results against expectations.

Taken together, these documents provide a complete view of the toolbox lifecycle. They define what the toolbox is intended to do, the tool collections and expectations from each tool, while the validation report demonstrates how those intentions were implemented and verified in practice.

Readers are therefore encouraged to consider these documents as complementary and interdependent, especially when evaluating the maturity, quality, and impact of the toolbox.

This document thus represents the validation plan, and the testing and evaluation report of the enRichMyData Integration Toolbox.

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.

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

List of figures

n/a

List of abbreviations

Abbreviation	Description
AI	Artificial intelligence
Apache2.0	Apache License, Version 2.0, see also: https://opensource.org/license/apache-2-0/
API	Application Programming Interface
BSD2Clause	The 2-Clause BSD License, see also: https://opensource.org/license/bsd-2-clause/
CO2	Carbone dioxide
GPL 3.0	GNU General Public License version 3, see also: https://opensource.org/license/gpl-3-0/
EMD	enRichMyData
KPI	Key Performance Indicators
LLM	Large Language Model
MIT	The MIT License, see also: https://opensource.org/license/mit/
RDF	Resource Description Framework
REST API	Representational State Transfer API
SSH	Secure Shell
STI	Semantic Table Interpretation
TAO	Tool Augmentation by User Enhancements and Orchestration
TRL	Technology Readiness Level
UI	User interface

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

1 Deliverable overview

1.1 Deliverable purpose, scope and context

This document represents the Validation plan and testing and evaluation report of the enRichMyData toolbox. It provides an overview of the enRichMyData toolbox release versions and the testcases used to validate each of the released versions.

The purpose of this deliverable is to formally document the validation strategy, testing procedures, and evaluation results for the enRichMyData toolbox, a modular system composed of several independent but interoperable tools. These tools are designed to function both as standalone components and as part of an integrated workflow, enabled by a common integration framework that ensures consistency, data exchange, and coordinated execution.

This deliverable D3.2 serves multiple objectives:

- to ensure that each tool meets its individual functional and non-functional requirements.
- to verify that the tools operate correctly when integrated through the shared framework.
- to evaluate the performance of the toolbox against the predefined KPIs from deliverable D2.2, which reflect measurable quality, performance, usability, and efficiency targets.
- to provide transparency on the testing process, results, and the overall readiness of the toolbox for operational use or deployment.

This validation and evaluation report covers the following elements:

- component-level validation - testing of each tool separately to confirm that it satisfies its specified requirements and behaves as expected under different input conditions.
- integration testing – evaluation of how well the tools interoperate within the integration framework, with focus on data flow, compatibility, and compliance to interface standards.
- pipelines validation - testing the toolbox in real or representative end-to-end use cases to assess system-level performance and functional correctness.
- KPI evaluation - analysis of actual values obtained during validation against the expected KPI values defined during the design phase. This includes identifying any deviations, assessing their impact, and proposing corrective or optimization actions where necessary.
- test environment documentation - description of the testing infrastructure, configurations, and any tools or scripts used to support automation or monitoring.

The scope includes both functional validation (e.g., correctness, expected behavior) and non-functional evaluation (e.g., performance, scalability, usability, robustness). However, this deliverable does not include full software user manual documentation or deployment documentation, which are provided along with the toolbox installation package.

The enRichMyData toolbox supports modular development, extensibility, and ease of integration, allowing individual components to evolve independently while maintaining system-wide coherence through the integration framework.

This validation and evaluation effort is part of the quality assurance process defined in the project's lifecycle and contributes to the achievement of the following key objectives:

- compliance with technical and functional specifications
- demonstration of interoperability among tools
- verification of the integration framework's effectiveness
- delivery of a reliable and performance-validated enRichMyData toolbox to end users

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

1.2 Target audience

This deliverable is specifically designed for the stakeholders of the enRichMyData platform, encompassing professionals including data engineers, data scientists, full-stack developers, and scientists.

1.3 Deliverable structure

This deliverable contains the following sections:

- Section 2 provides a description of the scope and purpose of the software developed in enRichMyData, i.e., the components of the toolbox.
- Section 3 describes the validation plan describing a synthesis of the requirements described in D1.1, the foreseen toolbox version and evolutions, the validation process and the test cases specification.
- Section 4 presents the validation and evaluation report of the tools into the enRichMyData toolbox.

1.4 Applicable documents

- D1.1 – Technical and Market requirements
- D1.2 - enRichMyData architecture
- D2.1 and D2.2 - enRichMyData tools

1.5 Updates of this deliverable

This deliverable corresponds to version 3.0 of the enRichMyData integration toolbox. The document has been evolved during the project according to the evolutions of the toolbox. The version 1.0 of this document was internally delivered and reviewed at M18, corresponding to the version 2.0 of the enRichMyData toolbox.

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

2 Purpose of the software

The purpose of the software toolbox delivered through the enRichMyData project is to provide practical, robust, and scalable components that support organizations in enriching their data with reference data they may have limited knowledge of, as well as supporting data providers in making their data reusable and available in data enrichment processes. The toolbox lowers the technological entry barriers by providing support for the definition of highly scalable and replicable data enrichment pipelines through a set of tools and infrastructure services related to capabilities needed during the lifecycle of enrichment pipelines. The toolbox makes the data enrichment process accessible to a broader set of stakeholders by reducing the required expertise and enhancing the tool support level.

The collection of tools and the tools in each collection are defined in the deliverable D2.2.

The toolbox is developed according to interoperability principles, where each tool can be used in stand-alone mode, and each component that processes data can be reused in more articulated data processing pipelines. Inspired by this principle, the toolbox also provides an integration framework that can be used to design, execute and scale pipelines using the components and, potentially, other tools. The enRichMyData toolbox is composed mainly of tools that can be executed through the command line or a REST API and can be integrated into pipelines allowing automated execution. Nevertheless, there exist also tools that have their own user interface which can be used in the exploration phase but also in runtime or during continuous enhancements (as defined in deliverable D1.2 – enRichMyData Architecture) but not necessarily being integrated in the automated execution pipelines that are implemented in the integration framework.

The integration framework is derived from the "Tool Augmentation by User Enhancements and Orchestration" (TAO) framework that was initiated by the European Space Agency (ESA) with the aim to provide extension and reusability (i.e., integration) of commonly used toolboxes in the Earth Observation domain.

The TAO framework core started to be refactored in the context of enRichMyData project to contain a generic core component that is no longer coupled to Earth Observation data model and that would facilitate the integration of enRichMyData tools (and other tools) and their composition in the desired pipelines. Following principles defined in the OGC[1] Application Packaging standard, TAO enables interfaces to be defined between the tools and packages the integrated tools into independent Docker[2] containers, facilitating the integration of tools and supporting a high degree of interoperability.

This framework allows for processing composition and distribution in such a way that end users can define processing workflows by themselves and easily integrate additional processing components.

enRichMyData Integration platform provides a means for the orchestration of heterogeneous processing components and libraries in order to process enRichMyData specific data. This can be achieved in following steps:

- Preparation of resources. This includes execution nodes, processing components, data sources and data sinks.
- Definition of a workflow as a processing chain.
- Execution of the workflow.
- Retrieval / visualization of the results.

enRichMyData integration framework will contain, by default, an initial set of tools and applications that can be combined into workflows. This default set of tools and applications has been updated

[1] https://docs.ogc.org/bp/20-089r1.html

[2] https://www.docker.com/

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

during the project's lifetime with newly developed ones. Additionally, users can add their own developed tools, applications or scripts to the framework.

The execution of the workflows can be performed on a single local machine or distributed on several machines that were defined as execution nodes in the enRichMyData frameworks, using one of the supported resource managers (Torque[3], SLURM[4], SSH, Kubernetes[5]).

It is worth noting that in enRichMyData, TAO is part of the ScalR and ReusR tools collections.

[3] https://github.com/xrg/torque

[4] https://slurm.schedmd.com/documentation.html

[5] https://kubernetes.io/

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

3 Validation plan

3.1 Purpose

This validation plan outlines the strategies and processes to ensure that the enRichMyData toolbox meets its intended requirements and functions as expected. This document is important for confirming that the enRichMyData toolbox, which is a collection of software tools used for implementing the pipelines, operates correctly, reliably, and aligns with predefined specifications.

The validation plan provides the following key elements in the next sections:

- A synthesis of the requirements as defined in D1.1, in Section 3.2.
- The release methodology containing the original release plan description, performed releases and the evolutions of the tools, in Section 3.3.
- The validation process, including its implementation, the validation activities, the identification of the validation tasks, and the validation of requirements , in Section 3.4.
- The test cases and scenarios for testing and validating the toolbox, in Section 3.5.
- The validation of the test cases and scenarios is performed via the validation and KPI evaluation report, for each enRichMyData toolbox version that was released during the project, and presented in Section 7.

3.2 Summary of requirements

All requirements below are reported in D1.1 and provided in the tables hereafter for reference to make this deliverable self-contained and help the reader to check evolutions and validation plans.

3.2.1 WrappR

RQ-WrappR-1	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.
RQ-WrappR-2	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.
RQ-WrappR-3	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

3.2.2 ResourcR

RQ-ResourcR-1	The entity retrieval component should efficiently handle large knowledge graphs (KG).
RQ-ResourcR-2	The entity retrieval module should ensure a predefined response time for queries by implementing an efficient caching strategy.
RQ-ResourcR-3	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.
RQ-ResourcR-4	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.
RQ-ResourcR-5	The entity retrieval module should be capable of providing contextualized data retrieval based on various scenarios or use cases.

3.2.3 DiscoverR

RQ-DiscoverR-1	DiscoverR tool must be able to support the discovery of data related to weather, and organizations, which are relevant to enrich tabular data; the discovery process must help users understand which

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

	data features are available and specify criteria to match data that are relevant for each record in the original table
RQ-DiscoverR-2	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.
RQ-DiscoverR-3	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.
RQ-DiscoverR-4	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).

3.2.4 CleanR

RQ-CleanR-1	CleanR tool must be able to support data formatting and harmonization operations, such as whitespace trimming, splitting of columns, and handling multivalued fields
RQ-CleanR-2	CleanR tool must support operations to handle missing and unwanted values, like null entries and duplicates
RQ-CleanR-3	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
RQ-CleanR-4	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
RQ-CleanR-5	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
RQ-CleanR-6	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
RQ-CleanR-7	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
RQ-CleanR-8	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

3.2.5 LinkR

RQ-LinkR-1	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.
RQ-LinkR-2	LinkR should provide detailed quality reporting, including relevant information to report uncertain outputs (e.g., uncertain types, properties and entities).
RQ-LinkR-3	LinkR should support reviewing uncertain outputs and providing corrective feedback to support human-in-the-loop linking mechanisms.
RQ-LinkR-4	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
RQ-LinkR-5	The linking approach should have the ability to handle data in various languages (e.g., English, French)
RQ-LinkR-6	The linking approach should be able to reconcile data against any data sources or KGs.

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

3.2.6 StructureR

RQ-StructureR-1	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.
RQ-StructureR-2	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
RQ-StructureR-3	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
RQ-StructureR-4	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
RQ-StructureR-5	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
RQ-StructureR-6	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
RQ-StructureR-7	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
RQ-StructureR-8	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
RQ-StructureR-9	Structured record should be linked to sections of documents that best describe how the described problem can be resolved in English language
RQ-StructureR-10	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)
RQ-StructureR-11	Extracting company mentions from news articles and linking it to Wikipedia

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

RQ-StructureR-12	Identifying AI related news from the news stream and organization of AI news according to AI specific taxonomies

3.2.7 ClassifiR

RQ-ClassifiR-1	Documents should be classified according to keywords/known problem descriptions/AI related topics.
RQ-ClassifiR-2	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)
RQ-ClassifiR-3	Identify core keywords that define categories or concepts

3.2.8 ScalR

RQ-ScalR-1	ScalR shall be able to execute workflows (pipelines) defined by the user.
RQ-ScalR-2	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.
RQ-ScalR-3	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.
RQ-ScalR-4	ScalR shall be able to parallelize the program, when possible, as defined by the workflow
RQ-ScalR-5	ScalR shall be flexible enough to support the use of heterogeneous computing infrastructures in the data enrichment pipelines.
RQ-ScalR-6	ScalR shall be able to provide horizontal scalability of data enrichment pipelines using software containers.
RQ-ScalR-7	ScalR shall be able to execute programs on a Kubernetes cluster.
RQ-ScalR-8	ScalR shall allow the tools it runs to handle large amounts of data ("Big Data") in the data enrichment process.
RQ-ScalR-9	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.
RQ-ScalR-10	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.
RQ-ScalR-11	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).
RQ-ScalR-12	ScalR shall support monitoring the status and health of running workloads, and infrastructure (e.g. cluster).
RQ-ScalR-13	ScalR shall produce detailed logs, allowing for any internal problems to be identified and diagnosed quickly.

3.2.9 ReusR

RQ-ReusR-1	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).
RQ-ReusR-2	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.

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

3.2.10 StreamR

RQ-StreamR-1	StreamR must provide access to news archives and streams, focusing on companies and public entities.
RQ-StreamR-2	StreamR should support access to data in tabular format
RQ-StreamR-3	StreamR should support weather data
RQ-StreamR-4	StreamR should be able to calculate correlation between time series.
RQ-StreamR-5	StreamR should be able to predict values for welding quality processes, and possibly others
RQ-StreamR-6	StreamR should detect anomalous states in the context of a mineral processing plant
RQ-StreamR-7	StreamR should support automatic methods for cleaning sensor data

3.2.11 GreenR

RQ-GreenR-1	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.
RQ-GreenR-2	Total runtime, carbon emissions and energy consumption are estimated. These statistics should be depicted in a graphical format other than text logging.
RQ-GreenR-3	Support for interfacing with other tools and services should be improved

3.2.12 System-wide requirements

3.2.12.1 Security and Privacy Requirements

REQ-SYS-1	The toolbox integration framework shall be usable by a user by means of an account with a password.
REQ-SYS-2	The system should allow at least two types of users: admin and regular user.
REQ-SYS-3	The users shall be able to change the password of their respective accounts by themselves.
REQ-SYS-4	An access control mechanism shall be implemented so that certain features can be used only by an administrator.
REQ-SYS-5	The user-defined artifacts, together with their results, shall be by default private (i.e. visible only to their author).
REQ-SYS-6	The platform shall allow a user to share one or more of its artifacts and results with other users.

3.2.12.2 Resource administration

RQ-RAD-1	It shall be possible for the user (Administrator or Public User) to add, edit or delete resources stored in the system
RQ-RAD-2	A resource can be deleted from the system only if there are no references to it
RQ-RAD-3	Resources added by Administrator can be used by all users
RQ-UWS-1	The workflow manager shall allow the definition and management of workflows

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

RQ-UWS-2	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
RQ-UWS-3	The Workflow Manager shall be able to convert workflow represented graphically into jobs to be executed

3.2.12.3 User interfaces

RQ-SUI-1	The system shall be accessible to the users via a web interface
RQ-SUI-2	The system shall allow the users to modify the look and feel: font size and colour theme.
RQ-SUI-3	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).
RQ-SUI-4	The user shall be able to create new workflows from defined tool descriptions, using the graphical interface

3.2.12.4 Interfaces with External Systems

RQ-SAPI-1	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
RQ-SAPI-2	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
RQ-SAPI-3	The system should be able to retrieve and display information from GreenR about measured carbon emissions and other statistics.

3.3 enRichMyData Toolbox Release Methodology

3.3.1 General

During the early stages of the project, before M12, the first draft of the current document was initiated containing the methodology for performing the releases, the initial release plan of the toolbox and the foreseen evolutions for each tool in the toolbox, during each planned release.

The document has been updated during the project to keep track of the evolutions foreseen and performed for each release of the toolbox. Thus, at month 18, we also released a version of the document that contained the foreseen evolutions of the tools in the toolbox and their status at that moment.

3.3.2 Release procedure

During the project, two major releases were foreseen and performed for enRichMyData toolbox and tools:

- enRichMyData Toolbox V2.0, released at mid of September (M24)
- enRichMyData Toolbox V3.0, planned for M34

An initial version of the toolbox was created prior M12 but without a public release, this version being considered the first version of the toolbox. This version was not released because it had not undergone full verification and validation. It was intended solely for presentation and demonstration purposes, making it easier to gather feedback at this stage. This version of the toolbox was deployed on an internal virtual machine, without providing an explicit installation package. The tools available at that moment were integrated using TAO as an integration platform and creating the initial SpendNetwork pipeline for demonstration purposes.

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

For each version, the evolutions that were introduced in the toolbox for each tool were identified and traced with the corresponding identifiers. These evolutions were linked to existing requirements already reported in D1.1 or to other evolutions relevant to the project. In this deliverable, we refer to each tool with the name used for the specification of requirements. If the evolution of tools results in branching or the addition of new tools, these details are documented in the enRichMyData Toolbox V3.0.

Two weeks before the toolbox release, all tool provider partners were asked to update their repositories and create the Docker images or the exposed services. Also, they provided information about whether the foreseen evolutions were implemented or not and, if not, they were asked for a new schedule. If the evolutions were critical for the implementation of the enRichMyData pipelines, a new intermediate version of the toolbox was scheduled to be released later. The section 0 of the current document was thus updated accordingly.

Before releasing a new version, after all the tools are integrated, a verification of the test cases will be performed to check that integrated tools contain the expected evolutions. For the commercial tools the test cases will not be executed but only the change log of the release will be checked as it is expected that the partner releasing the tool already performed the corresponding tests for the tool release. More details about the verification and validation process is provided in section 3.4.

3.3.3 enRichMyData Toolbox Evolution to V3.0

3.3.3.1 ABSTAT (DiscoverR)

No evolution foreseen in the project. However, the results of a large-scale user study to evaluate the usefulness of ABSTAT for understanding the structure of knowledge graphs has been completed and published on the Semantic Web Journal[6].

3.3.3.2 OntoText GraphDB (WrappR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-GRDB-1	Release 10.6 addresses fine-grained security, YasGUI (SPARQL editor and result browser) improvements, dynamic addition of cluster nodes, general usability improvements.	RQ-WrappR-1	Validated internally by OntoText. No test cases needed, to be only checked in release notes. Released in V2 of toolbox	DONE
EVL-GRDB-2	Release 10.5 addresses JSON-LD 1.1 integration, Azure Blob storage backup, cluster stability, SHACL improvements, JDBC connector improvements	RQ-WrappR-2	Validated internally by OntoText. No test cases needed, to be only checked in release notes. Relased in V2 of toolbox	DONE
EVL-GRDB-3	Release 10.4 addresses LLM integration, “Talk to Your Graph” improvements, named graph and quad-based security, assigning stable IDs to each transaction, better Amazon (AWS) integration and operations guide, RDF4J API parameter improvements, SHACL improvements, fixes for backups in cluster, fixes for SPARQL COALESCE, security	RQ-WrappR-3	Validated internally by OntoText. No test cases needed, to be only checked in release notes. Released in V2 of toolbox	DONE

[6] https://content.iospress.com/articles/semantic-web/sw223181

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

	improvements, JDBC (SQL table) improvements			
EVL-SEMOBJ-1	Semantic objects fused with Ontotext Graph DB	RQ-WrappR-2	Validated internally by OntoText. No test cases needed, to be only checked in release notes Released in V2 of the toolbox	DONE

3.3.3.3 Ontotext Refine (CleanR/LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-ONTREF-1	Release Ontotext Refine Command Line Intetrface Declarative SPARQL definition for RDF transformations Upgrade to latest stable Open Refine release, security improvements.	RQ-CleanR-1 RQ-CleanR-2 RQ-CleanR-3 RQ-CleanR-4 RQ-CleanR-5 RQ-CleanR-6 RQ-CleanR-7 RQ-CleanR-8	Validated internally by OntoText.	DONE

3.3.3.4 Ontotext Reconcile (LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-ONTREC-1	Release of Ontotext Reconcile v1 as a standalone tool	RQ-LinkR-6 RQ-LinkR-4	Validated internally by OntoText. No test cases needed, to be only checked in release notes	DONE

3.3.3.5 SemT-X[7](DiscoverR/LinkR)

Note: the tool previously known as SemT-UI has evolved into SemT-X, a framework consisting of a backend (SemT-backend) that provides frontends with a Web API for table reconciliation and extension. SemT-UI is a browser-based user interface that allows users to discover datasets for data enrichment, test them with source tables by applying enrichment operations, and review the results. Its ultimate goal is to identify useful operations for constructing an enrichment pipeline. SemT-Py is a Python library designed for use in notebooks, enabling the actual execution of these pipelines. These

[7] Previously referred to as SemT-UI; the new name SemT-X reflects the evolution into a more comprehensive framework that includes SemT-UI and SemT-Py.

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

pipelines facilitate the creation of Docker components, which can serve as building blocks for cloud-based execution in environments such as TAO.

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-SEMTUI-1	SemT-Py: a library for programmatic design of enrichment pipelines - Integration with Python notebook - Creation of pipeline components in Docker - Preliminary tests on BCs	RQ-LinkR-3 RQ-LinkR-4 RQ-LinkR-5 RQ-LinkR-6 RQ-DiscoverR-1 RQ-DiscoverR-2 RQ-DiscoverR-4	Validated on data from Eureka (SN) and MarKEn (JOT) business cases	DONE
EVL-SEMTUI-2	Interface to support full-fledged HITL reconciliation A) Extension of human manipulations to multiple values B) From revision to learning from the user feedback	RQ-LinkR-2 RQ-LinkR-3	Development of feature A is completed and tested. Development of feature B has been down prioritized in favor of interfaces with LLM-based linking approaches.	A- DONE B- CANELLED
EVL-SEMTUI-3	Novel data enrichment services - LLM-based reconciliation through the interaction with Lion-Linker	RQ-LinkR-2 RQ-LinkR-5 RQ-LinkR-6 RQ-DiscoverR-3	Validated on data from Eureka (SN) business case	DONE
EVL-SEMTUI-4	Code generation from the UI - Mapping between operations on the UI and python code	RQ-LinkR-1 RQ-LinkR-3 RQ-LinkR-4 DiscoverR-3	Validated on MarKEn (JOT) business case	DONE
EVL-SEMTUI-5	Data extension services - Support for relational algebra in SPARQL-based, multi-value data extension services - Extension with NLP services for textual values (ClassifiR)	RQ-LinkR-2 RQ-LinkR-4 RQ-LinkR-5	Validated on real data from Eureka (SN) business case	DONE
EVL-SEMTUI-6	Code generation from the UI - Generation of RML mappings for KG generation	RQ-LinkR-2 RQ-LinkR-3		DONE

3.3.3.6 LamAPI (ResourcR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status

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

EVL-LAMAPI-1	Filtering based on coarse-grain types with two different methods implemented. In particular: - Allowing to look up for entities by NER types (ORG, PERS, LOC and OTHERS) - Filtering strategies based on soft and hard constraint, given the Transitive Clousure of the Wikidata types	RQ-ResourcR-5		DONE
EVL-LAMAPI-2	Code refactoring to simplify usage of LamAPI, which added - possibility to specify the language - possibility of specifying QIDs must be in response	RQ-ResourcR-5	Tested on benchmark data against SOTA approaches based on (L)LM. Optimization has been completed	DONE

3.3.3.7 Alligator [s-elBat] (LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-SELBAT-1	Algorithms to support full-fledged HITL reconciliation - From revision to learning from the user feedback - Advanced decision heuristics to identify issues to be submitted to users - Fine-tuning of ML algorithms from users' feedback	RQ-LinkR-3	HITL features have been migrated to the development of Lion-Linker, as HITL can be declined to the interaction the user has when specifying additional prior information in the prompt	CANCELLED
EVL-SELBAT-2	Modular execution over large amount of data - Optimization of the s-elBat pipeline execution - Available as Python library - Full STI: CEA, CTA, CPA	RQ-LinkR-4	Tested on benchmark data against SOTA approaches based on (L)LM. Optimization has been completed. Python library is ready.	DONE
EVL-SELBAT-3	Fine-tuning of the algorithm on domain data - E.g., company names (Eureka BC), locations (MarKEn BC)	RQ-LinkR-6	In-domain fine-tuning for new data distributions has been tested (see reference paper). Type-specific fine tuning has been discarded due to unsatisfactory results and switched to LLM-based reconciliation. This feature has migrated to the development of Lion-linker.	CANCELLED

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

EVL-SELBAT-4	Extra: LLM-based Tabular Data Enrichment and Annotation - Analysis of TURL, TableLlama, and LLMs on STI data - New models based on these ones	RQ-LinkR-4; RQ-LinkR-6	Analysis: completed. The "new models" feature has migrated to the development of Lion-linker.	DONE

3.3.3.8 Crocodile (LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-croco-1	Implementation of a python library to perform entity linking over tabular data by using traditional algorithms	RQ-LinkR-4		DONE
EVL-croco-2	Support pandas data frame as input as well as support to process different inputs, single row, a column, single cell etc.. in general a given subset of a table	RQ-LinkR-4		DONE

3.3.3.9 Koala-UI (LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-koala-1	Ability of supporting different linking algorithms by connecting the api endpoint (e.g. Crocodile and Alligator)	RQ-LinkR-1		DONE
EVL-koala-2	Getting real time data regarding the status of the annotation process	RQ-LinkR-1		DONE
EVL-koala-3	Possibility to change the annotations by the user	RQ-LinkR-1		DONE

3.3.3.10 Lion-Linker (LinkR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-lion-1	Implementation of python library to perform entity linking over tabular data by using LLMs	RQ-LinkR-1		DONE
EVL-lion-2	Support different default prompt templates and possibility to specify examples for few shots learning	RQ-LinkR-2 RQ-LinkR-3		DONE
EVL-lion-3	Ollama[8] integration	RQ-LinkR-1 RQ-LinkR-4 RQ-LinkR-5		DONE
EVL-lion-4	OpenRouter[9] integration	RQ-LinkR-1		DONE

[8] https://ollama.com/

[9] https://openrouter.ai/

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

		RQ-LinkR-4 RQ-LinkR-5		
EVL-lion-5	TableLlama[10] integration	RQ-LinkR-1 RQ-LinkR-4 RQ-LinkR-5		DONE
EVL-lion-6	Full STI	RQ-LinkR-1 RQ-LinkR-4		DONE

3.3.3.11 Wikifier (StructR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-WIKIF-2	Continuous Wikipedia data update, application in InnoGraph Business case	RQ-StructureR-2, RQ-StructureR-3, RQ-StructureR-11, RQ-StructureR-8, RQ-StructureR-9	Wikifier extracts annotations with specific references to location in the document and links to Wikipedia pages in original (and additionally in English) language	DONE

3.3.3.12 Categorizer (ClassifiR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-CLSFR-1	Development of CPC based taxonomy (for AI innovations) intended to be used within Classifier component	RQ-ClassifiR-1, RQ-ClassifiR-3	Initial development of CPC based taxonomy for AI innovations Implemented in V2 of the toolbox.	DONE
EVL-CATEG-1	Categorization development involving several relevant vocabularies	RQ-ClassifiR-1, RQ-ClassifiR-3		DONE

[10] https://osu-nlp-group.github.io/TableLlama/

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

3.3.3.13 Streamstory (StreamR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-STRST-2	Data loading component update, API availability for general public, application withing enRichMyData business cases	RQ-StreamR-2, RQ-StreamR-3, RQ-StreamR-6, RQ-StreamR-5		DONE

3.3.3.14 Time Series Explorer Tool (StreamR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-TSTOOL-1	Implementation of initial data loading and processing techniques	RQ-StreamR-2, RQ-StreamR-4		DONE
EVL-TSTOOL-2	Data preprocessing, development of noise detection and addressing techniques, application within EnrichMyData business cases	RQ-StreamR-2, RQ-StreamR-7		DONE

3.3.3.15 ER Relation Classifier (StructR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-ERCLSF-1	First version implemented, testing in InnoGraph Business case	RQ-StructureR-12	First version implementation Identifying AI related news from the news stream and organization of AI news according to AI specific taxonomies. Implemented in version 2 of the toolbox.	DONE
EVL-ERCLSF-2	Component update, application within InnoGraph business case	RQ-StructureR-12		DONE

3.3.3.16 Event Registry (StreamR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status

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

EVL-ERCLSF-2	Continuous news data crawling and update, application in InnoGraph Business case	RQ-StreamR-1		DONE

3.3.3.17 Expert.ai Platform Document Analysis (StructureR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-DOCAN-1	Integration with ChatGPT - Available in Expert.AI Expert System generator no integration is envisaged	RQ-StructureR-7	New version allowing interaction via API. Implemented in version 2 of the toolbox	Ver-API-1.0 available
EVL-DOCAN-2	Development and integration with Expert.ai Language Model (offline). Available in Expert.ai Expert System generator no integration is envisaged	RQ-StructureR-7	New version allowing interaction via API. Implemented in version 2 of the toolbox	Ver-API-1.0 available
EVL-DOCAN-4	Human in the loop: new functionalities Available in Expert.ai Expert System generator no integration is envisaged	RQ-StructureR-7		DONE
EVL-DOCAN-5	Deployed on use case partners' premises (Philips and Bosch)	RQ-StructureR-10	Integrated in Expert System Generator, which is deployed privately	DONE

3.3.3.18 Expert.ai Platform Document Classification (ClassifiR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-DOCLA-1	From short descriptions, manually annotated, train a model for classifying new descriptions according to the previously adopted taxonomy using a pure-ML approach.	RQ-ClassifiR-1		DONE
EVL-DOCLA-2	Deployed on use case partners' premises (Philips and Bosch)	RQ-ClassifiR-2	Integrated in Expert System Generator, which is deployed privately	DONE

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

3.3.3.19 Expert.ai Expert System generator (Q&A) (StructureR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-DOCAN-4	API available to generate an Expert System (Retrieval Augmented Generation on top of processed document collections). To generate an Expert System on a given topic, it is necessary to provide PDFs that contain knowledge on that topic.	RQ-StructureR-1, RQ-StructureR-2, RQ-StructureR-3, RQ-StructureR-4	Evaluation by Expert of domain, the only ones who can assess whether the correct answer to the question asked	DONE
EVL-DOCAN-5	API available that can receive responses from the expert system for a data query related to the ES topic of interest. The questions are asked in natural language as are the answers; mechanisms are then included to track and explain the answers received through links to the parts of the pdf that contain useful elements to generate the answer.	RQ-StructureR-5 RQ-StructureR-6, RQ-StructureR-7, RQ-StructureR-8, RQ-StructureR-9	Evaluation by Expert of domain, the only ones who can assess whether the correct answer to the question asked	DONE

3.3.3.20 SIM-PIPE (ScalR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-SIMP-1	Splitting of input data for dry-runs	RQ-ScalR-6	Manual splitting of data only.	Partially done
EVL-SIMP-2	Perform predictions of resource requirements (frontend)	RQ-ScalR-13	Released in version 2 of the toolbox (SIMPIPE v-0.4.4)	DONE
EVL-SIMP-3	Perform predictions using SIM-PIPE API	RQ-SAPI-1	Released in version 2 of the toolbox (SIMPIPE v-0.4.4)	DONE
EVL-SIMP-4	Return predictions using SIM-PIPE API	RQ-SAPI-1	Released in version 2 of the toolbox (SIMPIPE v-0.4.4)	DONE
EVL-SIMP-5	Emulate resources on the computing continuum	RQ-ScalR-6 RQ-ScalR-7 RQ-ScalR-8 RQ-ScalR-9	Released in version 3 of the toolbox	DONE

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

EVL-SIMP-6	Integration of CarbonTracker (GreenR)	RQ-ScalR-6 RQ-ScalR-7 RQ-ScalR-8 RQ-ScalR-9 RQ-SUI-3 RQ-SUI-4 RQ-SAPI-1 RQ-SAPI-3	Released in version 3 of the toolbox	DONE

3.3.3.21 TAO (ScalR/ReusR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-TAO-1	Export workflows into Argo format	RQ-ScalR-9		DONE
EVL-TAO-2	Finalize and test the Kubernetes DRM	RQ-ScalR-7 RQ-ScalR-10		DONE
EVL-TAO-6	Integration with SIM-PIPE	RQ-ScalR-6 RQ-ScalR-7 RQ-ScalR-8 RQ-ScalR-9		DONE
EVL-TAO-4	Finish the data model generalization and specialization implementations for enrichmydata (scaling tabular data improvements, visualization, data sources, configuration, error handling)	RQ-ScalR-1 RQ-ScalR-2 RQ-ScalR-3 RQ-ScalR-4 RQ-ScalR-11		DONE
EVL-TAO-5	Versioning mechanism to components and sharing components between users.	RQ-ReusR-1 RQ-ReusR-2 REQ-Sys-2		DONE
EVL-TAO-8	GreenR integration	RQ-ScalR-12		DONE
EVL-TAO-9	Analyse the possibility of concurrent resource managers and implementation	RQ-ScalR-7		DONE
EVL-TAO-10	Update the documentation (how to section in website, development, integration and SUM)	RQ-ScalR-13		DONE

3.3.3.22 CarbonTracker (GreenR)

ID	Evolution	D1.1 Covered Reqs.	Remarks	Current status
EVL-CT-2	Total runtime, carbon emissions and energy consumption are estimated. These statistics should be depicted in a graphical format other than text logging.	RQ-GreenR-2	Estimated but not depicted in graphical format other than text logging.	DONE

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

EVL-CT-3	Support for interfacing with other tools and services should be improved	RQ-GreenR-3	CLI tool allows interface with any other tools for estimating energy consumption and carbon emissions. Implemented in version 2.0 of the toolbox.	DONE
EVL-CT-4	Fetch live carbon emissions data for the specific geographic location when using Carbontracker. This utilizes electricitymaps api.	RQ-GreenR-1	If users have electricitymaps API, this feature is available.	DONE
EVL-CT-6	Implement CLI tool	RQ-GreenR-3	Completed in Carbontracker release v.1.2.1	DONE
EVL-CT-7	Track usage of Carbontracker by the community.	RQ-GreenR-3	This is done at https://carbontracker.info/	DONE
EVL-CT-8	Support VM instances	RQ-GreenR-2	Access to power measurements for CPU generally poses a security risk. Might have to resort to estimates based on hardware specs.	DONE
EVL-CT-8	Testing suite for all functionalities	RQ-GreenR-3	>99% coverage is implemented.	DONE

3.4 Verification and Validation process

3.4.1 General

This section describes the validation process that is used to validate the enRichMyData Toolbox. This process and concepts are inspired from the various standards defining the validation and verification process like for ex. ECSS-E-ST-10-02C[11] from ESA, ISO/IEC/IEEE 29119[12] and others. Nevertheless, these standards are defined from all kinds of applications, with various levels of criticality. In our project we adopted a lightweight tailored application of these standards with only high-level verification and validation activities, the tools in the enRichMyData not being critical applications that require detailed coverage testing.

The software validation process confirms that the functional requirements are correctly and completely implemented in the final enRichMyData toolbox. The base document for defining the validation activities is the document D1.1 Technical and Market Requirements.

The validation procedures with respect to the functional specifications will be run before each major version of the toolbox in order to validate the developed enRichMyData toolbox. The results of this validation activity will be documented in the Validation and Evaluation Report document.

3.4.2 Validation process implementation

The enRichMyData toolbox will be validated by testing against functional requirements presented in D1.1 and copied in Section 3.2 “Summary of requirements”.

[11] ECSS-E-ST-10-02C Rev.1 – Verification - https://ecss.nl/wp-content/uploads/2018/02/ECSS-E-ST-10-02C-Rev.1(1February2018).pdf

[12] ISO/IEC/IEEE 29119-1:2022 - https://www.iso.org/standard/81291.html

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

3.4.3 Validation tasks identification

The validation activities described in this section will be used to validate the developed toolbox with respect to the functional specifications.

The validation activities will be performed by test (running test cases) and when validation by test scripts or tests using directly the application is not applicable, the validation will be performed by analysis, inspection or review of design (according to section 3.2 Summary of requirements).

Additional validation of the enRichMyData tools is carried out during internal testing sessions with users from the consortium, who have not been directly involved in the development of the tool; but also during other external workshops and conferences where external users can execute and perform independent validation of the tools, according to the instructions and scenarios provided.

The validation scenarios will be defined according to the requirements defined in D1.1.

3.4.4 Requirements validation

The test cases and procedures for validation will cover the test scenarios defined in the Test cases specifications section. These tests will be run before M33 and will cover the functional requirements specified in “Requirements” section of each test.

Another important component of the validation processes consists in implementing the pipelines defined by the business use cases and assessing of the execution and the results produced by the toolbox on relevant datasets by the use case providers and experts in the domain.

3.4.5 KPIs evaluation methodology

Following the execution of the enRichMyData Toolbox validation process, the actual values for each of the predefined KPIs have been collected and analyzed. In the Section “7.2 enRichMyData Toolbox V3.0 KPI Evaluation Report”, we present the evaluation of these KPI results against the expected values or targets established during the KPI definition phase and that were defined for each tool in the deliverable D2.2.

Thus, each KPI was compared to its corresponding expected value or performance thresholds. This comparison aims to determine whether the toolbox meets its performance, reliability, and usability objectives. The analysis focuses on identifying:

- KPIs fully achieved or exceeded - indicating successful implementation and alignment with design expectations.
- KPIs partially achieved - suggesting areas where the toolbox is functional but may require optimization or refinement.
- KPIs not achieved - highlighting performance gaps or issues that may need corrective actions.

For KPIs where actual results significantly deviate from the expected values, further analysis was conducted to identify possible causes. These may include:

- Technical limitations or bottlenecks in specific modules.
- Variations in input data characteristics that were not fully accounted for during KPI definition.
- Environmental differences between development and testing setups.

In each case, root cause analysis was used to distinguish between anomalies, structural issues, or evolving requirements.

In cases where KPI results suggest that the original targets were unrealistic or based on assumptions that no longer hold or other reasons, a justification is provided.

To facilitate transparent evaluation, in section “7.2 enRichMyData Toolbox V3.0 KPI Evaluation Report” is provided a summary table that includes:

- KPI name

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

- Expected value
- Actual value obtained
- Status (Achieved / Partially Achieved / Not Achieved)
- Comments / Justifications / Actions

3.4.6 Test case pass/fail criteria

The item pass/fail criterion depends on the validation method for the item.

According to the verifications and validation standards mentioned in section 3.4.1, possible verification methods are:

- **Test verification method [T]** - The method is referred to as "Test" when requirements have to be validated by measuring product performance and function under various simulated environments. These measurements may require the use of special equipment, instrumentation, and simulation techniques. Established principles and procedures shall be used to determine conformance to requirements.
- **Inspection verification method [I]** - The method is referred as "Inspection" when verification is achieved by visual determination of physical characteristics (such as construction features, hardware conformance to document drawing or workmanship requirements).
- **Analysis verification method [A]** - The method is as "Analysis" when verification is achieved by performing theoretical or empirical evaluation by accepted techniques. The analytical techniques shall be selected amongst systematic, statistical and qualitative design analysis, modelling and computational simulation. Verification by similarity is considered part of Analysis. It shall be applied if it can be shown that the article under verification is similar to another article that has already been verified to equivalent or more stringent requirements.
- **Review of design verification [R]** - The method is referred as "Review" when verification is achieved by validation of records or by evidence of validated design documents or when approved design reports, technical descriptions, engineering drawings unambiguously show that the requirement is met.

All tests have a number of steps. The general criteria is:

- a test **passed** if the expected results are achieved for all the steps of the procedure related to this test.
- a test **failed** if the expected results are not achieved for at least one step of the procedure related to this test.

3.5 Test Case specification and Test Procedures results

3.5.1 General

This chapter contains a specification for each test case that has to be tested against the requirements identified in D1.1 and for each of these test cases, the following information is specified:

Test case identifier: Title	The test case unique identifier followed by a title
Test item	A short description of the test case purpose
Covered requirements	Covered specification requirements
Input specification	The inputs to execute the test case
Expected results	The expected results
Pass/fail criteria	The criteria to decide whether the test has passed or failed

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

Environmental needs	The environment used for the execution of the test case. This can include the configuration, and the setup of the facility used to execute the test case as well as the utilization of any special test equipment; the configuration of the software utilized to support the test conduction;
Dependencies	List of all the test cases to be executed before this test case

Table 3-1 : Test case organization

While the test cases define what is tested (requirements covered, inputs, expected results and criteria), the test procedures define how a test is performed and present the actual results of the executions. A test procedure can cover several test cases.

The template for a test procedure is as follows:

Test ID	Expected Result	Actual Result	Pass/Fail
The ID of the test procedure (ex. **TP-WrappR-01)**	Expected result as defined in the test case(s) covered.	Actual result obtained	PASS or FAIL

Table 3-2 : Test procedure organization

In order to increase the readability of the document, the test procedures are defined and assessed for each tool in the same sections, after the definition of the test cases.

The test cases were defined when document was initiated and updated upon major evolutions added to the tools. The tests procedures were created or updated upon the addition of new test cases, but the actual results and "Pass/Fail" status were assessed before each of the release of the toolbox, if the tool was evolved during that release.

Whenever possible, KPIs were measured directly as part of the test procedures. For instance, execution times, memory usage, and data processing accuracy were recorded during functional and performance tests. In cases where direct measurement was not feasible within the test setup, alternative evaluation methods were applied. This included expert analysis of tool design, review of code quality and documentation, and structured user feedback collected through post-evaluation surveys following tool testing performed by the partners during consortium plenary meetings.

The last execution statuses corresponding to the version 3.0 of the enRichMyData toolbox for each tool are presented in the next sections while a summary of the test procedure results are presented in Section 7.1.

3.5.2 ABSTAT (DiscoverR)

No evolutions. Tests performed and tool validated outside and prior to the enRichMyData project, through demos and within other projects. No test case and tests procedures were defined during the project.

3.5.3 OntoText GraphDB (WrappR)

3.5.3.1 Test cases

TC-WrappR-01: GraphDB evolutions integration	
Test item	Test to validate the integration and evolutions of the OntoText GraphDB.
Covered requirements	**EVL-GRDB-1, EVL-GRDB-2, EVL-GRDB-3**

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

Input specification	Retrieve the latest version of the GraphDB Docker image. Check that the release notes contain the mentioned evolutions.
Expected results	Docker image for the current GraphDB version is available and has release notes containing the evolutions. Executing the provided test scripts produce the expected outputs.
Pass/fail criteria	The release notes of the current version contain the evolutions and tests are successful.
Environmental needs	N/A
Dependencies	N/A

3.5.3.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-WrappR-01	Retrieve the latest version of OntoText GraphDB from the website and check the release notes.	The release notes contained the evolution of mentioned.	PASS
UV-INT-Ontotext Talk To Your Graph	Ability of the system to generate valid SPARQL queries based on a custom ontology.	System performs as expected. Tested on internal benchmarks and demonstrated on a public demo from Spend Network use case data	PASS

3.5.4 OntoText Refine (CleanR/LinkR)

3.5.4.1 Test cases

TC-WrappR-03: Semantic Search evolutions integration	
Test item	Test to show the integration and evolutions of the OntoText Refine.
Covered requirements	**EVL-ONTREF-1**
Input specification	Retrieve the latest version of the Refine Docker image. Check that the release notes contain the mentioned evolutions.
Expected results	Docker image for the current Refine version is available and has release notes containing the evolutions. Executing the provided test scripts produce the expected outputs.
Pass/fail criteria	The release notes of the current version contain the evolutions and tests are successful.
Environmental needs	N/A
Dependencies	N/A

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

3.5.4.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-WrappR-02	Retrieve the latest version of OntoText Refine from the website and check the release notes.	The release notes contained the evolution of mentioned. OntoRefine also integrated into TAO and executed into JOT pipeline. Results validated by JOT.	PASS
UV-EXT-Ontotext	OntoRefine is tested concurrently by almost 30 people in the DataWeek workshop session as part of the "Talk to Your Graph" scenario. The steps are provided in a tests scenarios document.	The users are successfully following the steps, and the application is behaving as expected.	PASS

3.5.5 OntoText Reconcile (LinkR)

TC-WrappR-04: Semantic Search evolutions integration	
Test item	Test to show the integration and evolutions of the OntoText Reconcile.
Covered requirements	**EVL-ONTREC-1**
Input specification	Retrieve the latest version of the Reconcile Docker image. Check that the release notes contain the mentioned evolutions.
Expected results	Docker image for the current Reconcile version is available and has release notes containing the evolutions. Executing the provided test scripts produce the expected outputs.
Pass/fail criteria	The release notes of the current version contain the evolutions and tests are successful.
Environmental needs	N/A
Dependencies	N/A

3.5.5.1 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-WrappR-03	Retrieve the latest version of OntoText Reconcile from the website and check the release notes.	The release notes contained the evolution of mentioned.	PASS

3.5.6 SemT-X (DiscoverR/LinkR)

3.5.6.1 Test cases

TC-SEMTUI-01	
Test item	Test the annotation process
Covered requirements	EVL-SEMTUI-1

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

Input specification	Upload a table, annotate it and at the end of the process export the annotated table
Expected results	Annotated tables
Pass/fail criteria	Pass: The table has been annotated with active links Fail: No annotation for the table was provided
Environmental needs	Sem-TUI running instance, examples of tables
Dependencies	N/A

TC-SEMTUI-02	
Test item	Test for view of the results
Covered requirements	EVL-SEMTUI-2
Input specification	Annotated table
Expected results	Filtered results
Pass/fail criteria	Pass: Results can be sorted by confidence and single labels can be modified Fail: Results cannot be sorted by confidence and/or single labels cannot be modified
Environmental needs	Sem-TUI running instance, annotated table loaded in the tool
Dependencies	N/A

3.5.6.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-SemTUI-01	Test the annotation and filtering process using SemTUI according to TC-SEMTUI-01 and TC-SEMTUI-02.	SemTUI was integrated into TAO and executed in the JOT pipeline. Results validated by JOT.	PASS
UV-INT-SemTUI	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Test SemTUI following the tasks provided in the document in order to perform reconciliation, refinement, extension, search and filtering.	All operations were performed as expected although. Users filled in the feedback survey presented in the annex.	PASS
UV-EXT-SemTUI	SemTUI is tested concurrently by almost 30 people in the DataWeek workshop session. The steps are provided in a document.	The users are successfully following the steps and the application is behaving as expected.	PASS

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

3.5.7 Alligator (s-elBat) (LinkR)

3.5.7.1 Test cases

TC-SELBAT-01: Accuracy of the Algorithm for STI annotations	
Test item	Test to evaluate the algorithms supporting STI, on both benchmark and business datasets, against State-of-The-Art LLM-based algorithms.
Covered requirements	EVL-SELBAT-1
Input specification	Provide a dataset for reconciliation with ground truth. Use Alligator to perform full STI annotations.
Expected results	The system effectively returns STI annotations with a performance on-par or better than LLMs-based counterparts.
Pass/fail criteria	Pass: The system shows measurable improvement in reconciliation accuracy Fail: No significant improvement or incorrect annotations.
Environmental needs	Alligator instance running; Dataset for reconciliation.
Dependencies	N/A

TC-SELBAT-02: Efficient Execution Over Large Amounts of Data	
Test item	Test to ensure that Alligator efficiently handles and processes large amounts of data.
Covered requirements	EVL-SELBAT-2
Input specification	Load a large dataset into Alligator; Execute reconciliation processes and monitor system performance, resource usage, and processing time
Expected results	Alligator processes large datasets efficiently, with acceptable performance metrics and resource usage.
Pass/fail criteria	Pass: The system maintains performance within acceptable limits and uses resources efficiently, completing the processing of 100 rows in around 70 seconds. Fail: The system does not meet performance expectations if it significantly exceeds around 70 seconds to process 100 rows or if resource usage exceeds acceptable limits.
Environmental needs	Alligator instance running; Large dataset for testing
Dependencies	N/A

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

3.5.7.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-SELBAT-01	Perform the tests and check the expected results according to TC-SELBAT-01 and TC-SELBAT-02.	Tested through the developed test scripts. Average Accuracy (ED) = 0.819, on six benchmark datasets. These values are aligned with best approaches for entity disambiguation based on LLMs (TableLlama scores 0,812). Regarding CTA and CPA evaluations, on two SemTab Challenge datasets: CTA: - HTR1: F1 = 0.927 - HTR2: F1 = 0.817 CPA: - HTR1: F1 = 0.840 - HTR2: F1 = 0.847 Regarding the speed of the annotation, on a sample table of 1000 rows and 11 columns, with 4 of them containing named-entities, with a total of 4000 mentions to annotate, Alligator takes: W/o cache: 274.2s for 4000 mentions - 0.07 s/mention W/ cache: 29.2s for 4000 mentions - 0.007 s/mention Hyperparameters: - 50 candidates - 8 workers - 16 rows per worker	PASS

3.5.8 Crocodile (LinkR)

3.5.8.1 Test cases

TC-croco-01: Crocodile Test Case	
Test item	Test the library itself
Covered requirements	EVL-croco-1
Input specification	Provide CSV as input and verify that the library works and provides the annotated table as output
Expected results	Annotated table
Pass/fail criteria	The table was annotated / no annotation was provided
Environmental needs	Docker
Dependencies	N/A

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

3.5.8.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-Croco-01	Provide a CSV as input to crocodile. The table annotated accordingly using crocodile.	Crocodile was integrated into TAO and executed as part of the SpendNetwork and InnoGraph use cases. The results were as expected and used by the subsequent steps.	PASS

3.5.9 Lion-linker (LinkR)

3.5.9.1 Test cases

TC-lion-01: Lion-Linker Test Case	
Test item	Test the library behavior
Covered requirements	EVL-lion-1
Input specification	Provide CSV as input and verify that the library works and provides the annotated table as output
Expected results	Annotated table
Pass/fail criteria	The table was annotated / no annotation was provided
Environmental needs	Docker, Ollama
Dependencies	N/A

3.5.9.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
UV-INT-LionLinker	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Follow the steps in the test description document by installing Ollama and try the notebook provided.	The operations are performed with success and the execution is as expected.	PASS

3.5.10 Koala-UI (LinkR)

3.5.10.1 Test cases

TC-koala-01: Koala-UI Test Case	
Test item	Test the web app
Covered requirements	EVL-koala-1
Input specification	Upload a CSV file through the web-UI and verify that it’s possible to get the annotated table
Expected results	Annotated table in the UI with the metadata
Pass/fail criteria	The table was annotated / no annotation was provided
Environmental needs	Docker

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

Dependencies	N/A

3.5.10.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
UV-INT-KoalaUI	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Follow the steps in the two cases (movies table and SpendNetwork table) presented in the document provided for annotation of the table, exploring linking results and functionalities.	The application behaves as expected. Only some small delays on upload due to the large number of concurrent users and small VM used.	PASS

3.5.11 Wikifier (StructR)

3.5.11.1 Test cases

TC-WIKIF-01: Wikifier Test Case	
	Test the annotation of Wikipedia related information from text (GitHub repositories, news etc.)
Covered requirements	RQ-StructureR-11
Input specification	The user can test Wikifier either at https://wikifier.org website or via REST API. At website the user inputs textual content in the window, submits the input and observes the results.
Expected results	Text annotated with Wikipedia concepts.
Pass/fail criteria	Text was annotated / no annotation was provided
The table was annotated / no annotation was provided	For interface testing: working browser. For API testing: any programming environment supporting REST APIs.
Dependencies	N/A

3.5.11.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-WIKIFIER-01	Perform the operations specified in the TC-WIKIFIER-01	Tests performed by providing inputs to https://wikifier.org. The provided text is annotated as expected. Wikifier included also in TAO and tested in Philips, Bosch and InnoGraph pipelines.	PASS

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

Test ID	Expected Result	Actual Result	Pass/Fail
UV-INT-Wikifier	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Test the Wikifier using the website interface and then using the API, according to the instructions in the test document provided.	Running Wikifier from the website and from cli led to the expected results. Users filled in the feedback survey presented in the annex.	PASS

3.5.12 Categorizer (ClassifiR)

3.5.12.1 Test cases

TC-CATEG-01: Categorizer Test Case	
Test item	Test the offline classification
Covered requirements	RQ-ClassifiR-2
Input specification	Textual input (specifically the user prepares a set of textual excerpts for classification); testing can be performed in Python environment based on obtained developed classification model.
Expected results	List of classes with confidence level assigned.
Pass/fail criteria	Pass: successful model utilization, fail: any displayed errors
Environmental needs	Python, Python environment (for instance, Google Colab etc.)
Dependencies	N/A

3.5.12.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-CATEG-01	Perform the operations specified in the TC-CATEG-01	Test it in version 3.0 of the toolbox through the Python interface.	PASS

3.5.13 StreamStory (StreamR)

3.5.13.1 Test cases

TC-STREAMST-01: StreamStory Test Case	
Test item	Testing data loading and visualization with StreamStory
Covered requirements	RQ-StreamR-2
Input specification	Data from project business case in tabular form (loaded in StreamStory interface app)
Expected results	Presented visualization
Pass/fail criteria	Pass: data successfully loaded, presented in visual form; fail: any potential errors displayed to user
Environmental needs	StreamStory web application:

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

	http://streamstory.ijs.si
Dependencies	N/A

3.5.13.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-STREAMST-01	Perform the operations specified in the TC-STREAMST-01	Tested in the InnoGraph use case.	PASS
UV-INT-StreamStory	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Follow the steps in the Stream Story scenario document, using the data provided in the test package.	The application is behaving as described in the testing scenario document.	PASS

3.5.14 Time Series Explorer Tool (StreamR)

3.5.14.1 Test cases

TC-TSERT-01: Time Series Explorer Tool Test Case	
Test item	Test the sensor data cleaning
Covered requirements	RQ-StreamR-7
Input specification	Data input into time series tool component (CSV). Testing can be done in Python or Python environment after obtaining the component implementation repository/resources on GitHub.
Expected results	Cleaned data (CSV)
Pass/fail criteria	Pass: data cleaning performed; fail: any potential errors displayed to user
Environmental needs	Python/Python environment (for instance, Google Colab etc.)
Dependencies	StreamStory Test Case

3.5.14.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-TSERT-01	Perform the operations specified in the TC-TSERT-01	As expected with the CSV file provided by JSI.	PASS

3.5.15 ER Relation Classifier (StructR)

3.5.15.1 Test cases

TC-ERRELC-01: ER Relation Classifier Test Case	
Test item	Test relation extraction from the news - specifically the extraction of "technology use" and "AI" related news.
Covered requirements	RQ-StructureR-12

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

Input specification	User with ER credentials uses Event Registry API in order to get news in JSON format (specific query parameters include "technology use" classification parameter). Link to ER Registry API (on GitHub): https://github.com/EventRegistry/event-registry-python
Expected results	News in JSON format
Pass/fail criteria	Pass: successful response from ER service, Fail: failed response from ER service
Environmental needs	Python
Dependencies	Event Registry Test Case

3.5.15.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-ERRELC-01	Perform the operations specified in the TC-ERRELC-01	A script event_registry_test.py was created and executed successfully according to the examples on GitHub.	PASS

3.5.16 Event Registry (StreamR)

3.5.16.1 Test cases

TC-EVTREG-01: Event Registry Test Case	
Test item	StreamR must provide access to news archives and streams. Test ER access by API.
Covered requirements	RQ-StreamR-1
Input specification	User with ER credentials uses Event Registry API in order to get news in JSON format (specific query parameters include time period and concepts of interest – for instance, news about "Artificial intelligence" for the last month). Link to ER Registry API (on GitHub): https://github.com/EventRegistry/event-registry-python
Expected results	News in JSON format
Pass/fail criteria	Pass: successful response from ER service, Fail: failed response from ER service
Environmental needs	Python
Dependencies	Wikifier Test Case

3.5.16.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-EVTREG-01	Perform the operations specified in the TC-EVTREG-01	A script event_registry_test.py was created and executed successfully according to the examples on GitHub.	PASS

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

3.5.17 Expert.ai Expert System generator (Q&A) (StructureR)

3.5.17.1 Test cases

TC-LLM_INT-01: Integration With ChatGPT	
Test item	Test to show Integration with OpenAI APIs.
Covered requirements	**RQ-StructureR-7; EVL-DOCAN-1; EVL-DOCAN-3**
Input specification	The user asks a question
Expected results	The system responds with a punctual response
Pass/fail criteria	Pass: After user asked a question, the system responds with a punctual response. Fail: no response from system or not a valid response.
Environmental needs	ANSWERS platform is up and running
Dependencies	N/A

TC-LLM_INT-02: Integration With off-line LLM	
Test item	Test to show Integration With off-line LLM
Covered requirements	**RQ-StructureR-7; EVL-DOCAN-2; EVL-DOCAN-3**
Input specification	The user asks a question
Expected results	The system responds with a punctual response, without taking data out of where the ANSWERS platform is installed
Pass/fail criteria	After user asked a question, the system responds with a punctual response.
Environmental needs	ANSWERS platform is up and running
Dependencies	N/A

3.5.17.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-LLM_INT-01	Perform the operations specified in the TC-LLM_INT-01 and TC-LLM_INT-02	The Expert.ai services were used for testing the new features and results were as expected.	PASS
UV-INT-Expert System Generator	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Follow the instructions indicated by the Expert.AI presenter. Try additional options of the application and check the results.	The application is responsive and generates the results as expected based on the documents provided.	PASS

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

Test ID	Expected Result	Actual Result	Pass/Fail
UV-INT-ExpertAI-Answers	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Perform the steps from the tutorial and upload the files provided into the archives. Create folder, document collection, configure generative AI and other settings. Start asking questions.	All operations performed as expected. Documents were uploaded successfully, and relevant answers were retrieved.	PASS

3.5.18 Expert.ai Platform Document Classification (ClassifiR)

No evolutions. Tests performed and tool validated outside and prior to the enRichMyData project, through demos and within other projects. Nevertheless, the tool was tested through the initial SpendNetwork pipeline implemented in TAO as part of this pipeline.

3.5.19 Expert.ai Platform Document analysis (StructureR)

No evolutions. Tests performed and tool validated outside and prior to the enRichMyData project, through demos and within other projects. Nevertheless, the tool was tested through the initial SpendNetwork pipeline implemented in TAO as part of this pipeline.

3.5.20 LamAPI (ResourcR)

3.5.20.1 Test cases

TC-LAMAPI-01: Test to validate the entity lookup functionality by NER types in LamAPI	
Test item	Test to validate the entity lookup functionality by NER types or an arbitrary Wikidata type in LamAPI
Covered requirements	RQ-ResourcR-5
Input specification	Provide a set of text data containing entities categorized as ORG, PERS, LOC, and OTHERS or an arbitrary Wikidata type. Use LamAPI to perform NER/type lookup on the provided text data
Expected results	The system returns entities correctly classified into their respective NER (ORG, PERS, LOC, OTHERS) or arbitrary types
Pass/fail criteria	Pass: The output matches the expected classification of entities. Fail: The output does not match the expected classification of entities
Environmental needs	LamAPI instance running, Text data with known entities for testing
Dependencies	N/A

TC-LAMAPI-02: Performance and Scalability Test for Large Knowledge Graphs	
Test item	Test to evaluate the performance and scalability of LamAPI when handling large knowledge graphs
Covered requirements	RQ-ResourcR-1, RQ-ResourcR-2, RQ-ResourcR-3

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

Input specification	Load a large knowledge graph dataset into LamAPI. Perform multiple complex queries to evaluate response time and system stability under load
Expected results	LamAPI efficiently handles large knowledge graphs with acceptable response times and system stability during multiple complex queries. The average response time is expected to be around 2 to 3 seconds for 100 concurrent requests.
Pass/fail criteria	Pass: The system maintains performance within acceptable limits and remains stable under load. Fail: The system shows significant performance degradation or instability
Environmental needs	LamAPI instance running; Large knowledge graph dataset
Dependencies	N/A

TC-LAMAPI-03: Security and Privacy Compliance Test	
Test item	Test to ensure that LamAPI complies with GDPR regulations by implementing authentication mechanisms or token-based methods to safeguard the data
Covered requirements	RQ-ResourcR-4
Input specification	Perform a series of access attempts using valid and invalid credentials. Monitor the system's response to unauthorized access attempts and ensure proper logging and handling of these events
Expected results	LamAPI correctly restricts access based on authentication credentials (or tokens), logs unauthorized access attempts, and ensures data privacy and security
Pass/fail criteria	Pass: Unauthorized access attempts are logged and denied; valid access is allowed. Fail: Unauthorized access is granted or valid access is denied incorrectly
Environmental needs	LamAPI instance running; User accounts (or tokens) with varying levels of access
Dependencies	N/A

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

3.5.20.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-LAMAPI-01 **TP-LAMAPI-02**	Execute the script validate_LamAPI.py that tests 100 simultaneous requests to LamAPI endpoints and performs the security and performance analysis.	LamAPI also integrated into TAO and used in the SpendNetwork pipeline. Results validated by SpendNetwork. With respect to the quality of the retrieved candidates, expressed reporting the Coverage (i.e., the number of times the correct candidate is found in the retrieved list of entities), LamAPI achieves the following: - Coverage = 0.90 for top-30 candidates on four out of six datasets from the SemTab challenge - Coverage = 0.80 for top-100 candidates on two out of six datasets from the SemTab challenge - Coverage on TURL-2K-red for top-50 candidates is 88.17 for LamAPI and 71.75 for Wikidata Lookup Service Additionally, the specification of the NER or specific Wikidata type improves the coverage as follows: - W/o types: 0,784 - W/ types: 0.814 Tested on six benchmark datasets from the SemTab Challenge Regarding the Average Retrieval Time (ART) expressed in s/mention, LamAPI, from a remote REST API, achieves an ART of 0.036 s/mention, which is equal to 3.6s for 100 mentions	PASS

3.5.21 SIM-PIPE (ScalR)

3.5.21.1 Test cases

TC-SIMP-01: Splitting of input data for dry-runs	
Test item	Test to show that SIM-PIPE is splitting the input data for dry-runs.
Covered requirements	**EVL-SIMP-1, RQ-ScalR-6**
Input specification	Provide to SIM-PIPE dry run execution an input dataset that can be split (large CSV file).
Expected results	The dry run is splitting the input data.
Pass/fail criteria	Pass: during the dry run the input data is split; Fail: No split is done or errors during the dry run.
Environmental needs	SIM-PIPE is up and running
Dependencies	N/A

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

TC-SIMP-02: Perform predictions of resource requirements (frontend)	
Test item	Test to show that the SIM-PIPE is performing predictions of resource requirements.
Covered requirements	**EVL-SIMP-2, EVL-SIMP-3, EVL-SIMP-4**
Input specification	Execute a dry run and wait for finishing.
Expected results	SIM-PIPE is displaying into the web interface a prediction of the resources needed to execute the dry run. Query also the SIM-PIPE API in order to retrieve the information about the prediction of resources
Pass/fail criteria	Pass: The resources needed are displayed in the SIM-PIPE web interface and are retrieved via the API. Fail: error displaying resources or no resources displayed.
Environmental needs	SIM-PIPE is up and running
Dependencies	N/A

TC-SIMP-03: Emulate resources on the computing continuum	
Test item	Test to show that the SIM-PIPE is able to emulate resources on the computing continuum
Covered requirements	**EVL-SIMP-5**
Input specification	Hardware architecture specification
Expected results	Emulated resource consumption (CPU, Memory, Network, Duration) is close to resource consumption on actual hardware.
Pass/fail criteria	Pass: Obtain resource consumption from emulated hardware Fail: no resources received or error encountered.
Environmental needs	SIM-PIPE is up and running
Dependencies	N/A

TC-SIMP-04: Integration of carbontracker	
Test item	Test to show that the SIM-PIPE is able to perform integration of CarbonTracker
Covered requirements	**EVL-SIMP-6**
Input specification	Argo workflows data pipeline
Expected results	Pass: Obtain resource consumption in CO2 equivalents for a finished data pipeline dry run Fail: error retrieving the resource consumption in CO2 equivalents or they are all 0.
Pass/fail criteria	CO2 equivalents for a dry run is obtained
Environmental needs	SIM-PIPE is up and running

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

Dependencies	N/A

3.5.21.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-SIMP-01	Test that the dry run is splitting the input data (TP-SIMP-01), that is performing predictions of resource requirements (TP-SIMP-02) and that SIM-PIPE is able to emulate resources on the computing continuum (TP-SIMP-03).	SIM-PIPE integrated with TAO. TAO exports pipelines as Argo workflows and created projects and dry runs into SIM-PIPE. SIM-PIPE returned the prediction of resource requirements after splitting the input data.	PASS
TP-SIMP-02	Test to show that the SIM-PIPE is able to perform integration of CarbonTracker	For the pipeline exported from TAO, SIM-PIPE returned the CO2 equivalents for the dry run created for the pipeline provided as Argo workflow generated from TAO.	PASS

3.5.22 TAO (ScalR/ReusR)

3.5.22.1 Test cases

TC-ScalR -01: Workflows executions	
Test item	Test to show that the toolbox can execute workflows (pipelines) defined by the user and that is able to invoke programs from Open Container Initiative (Docker) images. These are mainly the enRichMyData tools, for which processing component descriptions will be created. Also, tests show that it allows the programs it runs to call Web services. The toolbox also supports monitoring the status and health of running workloads, and infrastructure (e.g. cluster) and produces detailed logs, allowing for any internal problems to be identified and diagnosed quickly.
Covered requirements	**EVL-TAO-4, RQ-ScalR-6, RQ-ScalR-9, RQ-ScalR-12, RQ-ScalR-13**
Input specification	In the web interface of the toolbox the user creates a component that invokes an external Web service then implements a pipeline containing components added into the toolbox. Execute the workflow and check the execution
Expected results	A workflow is created and executed. The step executions can be monitored in the dashboard and checking the toolbox logs, the Docker invocation commands can be checked.
Pass/fail criteria	Pass: No error occurs during workflow execution and the logs can be monitored into dashboard. The Docker invocation commands are available in logs. Fail: no execution performed or execution failed.
Environmental needs	TAO platform is up and running
Dependencies	N/A

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

TC-ScalR -02: Workflow executing in Kubernetes cluster	
Test item	Test to show that the toolbox is able to execute programs on a Kubernetes cluster.
Covered requirements	**RQ-ScalR-7, EVL-TAO-2**
Input specification	Create a workflow and select the option to be executed on a Kubernetes cluster.
Expected results	The workflow is executed successfully.
Pass/fail criteria	Pass: No errors occur when workflow is executed during the Kubernetes cluster. Fail: error executing in a cluster.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ScalR-03: Execution stopping on critical errors	
Test item	Test to show that the execution of the workflow on critical errors (e.g., incompatibility between step inputs and outputs, disk full errors, etc.) is stopped.
Covered requirements	**RQ-ScalR-11**
Input specification	Create a pipeline where a component is creating a JSON file in another format than the one expected by the next processing component.
Expected results	The pipeline execution is stopped after the error of the component receiving invalid input.
Pass/fail criteria	Pass: The execution is stopped on error. Fail: The execution continues even on error, with errors on all subsequent steps.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ScalR-04: Workflow steps parallelization	
Test item	Test to show that ScalR is able to parallelize the programs, when possible, as defined by the workflow
Covered requirements	**RQ-ScalR-4**
Input specification	Create a pipeline where several steps in the workflow have the same level for execution. Start the execution of this workflow.
Expected results	During the execution, in the dashboard can be seen that the steps having the same level are executed at the same time.
Pass/fail criteria	Pass: The steps are executed in parallel. Fail: Error on step executions or they are executed one by one.

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

Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ScalR-05: Usage of heterogenous infrastructure	
Test item	Test to show that ScalR can support the use of heterogeneous computing infrastructures in the data enrichment pipelines.
Covered requirements	**RQ-ScalR-5**
Input specification	Configure TAO to execute a pipeline on a local machine. Execute a simple pipeline. Next, configure TAO to use Kubernetes as resource manager. Execute a simple pipeline.
Expected results	Checking the logs and also the system load, can be checked that the steps are executed locally in the first case, and using the Kubernetes cluster in the second case.
Pass/fail criteria	Pass: The pipeline is executed without errors in both configurations. Fail: errors during executions.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ScalR-06: Handling large amounts of data	
Test item	Test to show that ScalR allows the tools it runs to handle large amounts of data ("Big Data") in the data enrichment process.
Covered requirements	**RQ-ScalR-8**
Input specification	Create a pipeline and provide as input a very large file and then execute the workflow using these data.
Expected results	The big file is used as input to the pipeline and used by processing nodes, even if executed on external nodes or in Kubernetes cluster.
Pass/fail criteria	Pass: The pipeline is executed without errors using the large file provided as input. Fail: errors occurred during execution.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ScalR-07: Handling large amounts of data	
Test item	Test to show that ScalR scales 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
Covered requirements	**RQ-ScalR-10**

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

Input specification	In a cloud environment, execute a pipeline with several steps in parallel, having at the beginning only one execution node available.
Expected results	TAO creates additional VMs, using the OpenStack API, and allocates steps executions on the new created VMs. At the end of the execution, it deallocates the dynamically allocated VMs.
Pass/fail criteria	Pass: The VMs are allocated, and steps are executed successfully. The VMs are deallocated after a while (according to the timeout configured). Fail: Errors during execution or creation or deletion of the VMs.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC- ReusR -01: Component reuse	
Test item	Test to show that it is possible 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).
Covered requirements	**REQ-ReusR-1, RQ-ScalR-9**
Input specification	From the processing components collection, the user can drag and drop components in the workflows dashboard and build new pipelines. A created pipeline can be cloned and then modified without affecting the original one
Expected results	A workflow is created and cloned
Pass/fail criteria	Pass: No error occurs during workflow creation and its cloning. Fail: Errors occurred.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-ReusR-02: Component versioning	
Test item	Test to show that it is possible 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.
Covered requirements	**REQ-ReusR-2**
Input specification	In the editor of the processing components, containers and workflows descriptions can be set the current version of the component.
Expected results	The component version can be set and saved.
Pass/fail criteria	Pass: After saving the component, the version is preserved. Fail: errors occurred during saving.
Environmental needs	TAO platform is up and running

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

Dependencies	N/A

3.5.22.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-TScalR-01	Create a pipeline and executed as mentioned in TC-ScalR-01. Execute the pipeline. Check the monitoring dashboard so that the pipeline executes OK.	Several pipelines implemented for the use cases in the project (JOT, SpendNetwork, Philips, Bosch, InnoGraph). The pipelines were executed successfully and final results were created. Checking the logs, the performance KPI was evaluated - less than 0.5 between scheduling two pipelines steps by TAO.	PASS
TP-TScalR-02	First execute an existing pipeline with the default executor (SSH executor in OpenStack environment as per TC-ScalR-7). The steps are executed in parallel (as defined in TC-ScalR-04) in the cluster and logs available in the dashboard. Then, as in as in TC-ScalR-02, run the same pipeline using Kubernetes as resource manager for steps execution. The steps execute in the cluster upon selection of the new resources manager, as per TC-ScalR-05	In the OpenStack environment, new VMs are dynamically created and steps executed on the created VMs. At the end of the execution of the pipeline, the VMs are deallocated. For K8s, a cluster with 4 nodes was created. Steps were executed on the nodes in the cluster. The KPI for scalability was evaluated, on pipelines where parallelization was possible, performances were increased by more than 75% (an overhead being introduced by the data transfer between nodes). Nevertheless, on pipelines not supporting parallelization, no improvement was observed (as expected)	PASS
TP-TScalR-03	The execution is stopped during critical errors (disk space, no execution resources available) as per TC-ScalR-03.	When executed on a local VM with no disk space available, the execution stops at the first step when no files can be created. When executed by selecting K8s resource manager, but no cluster was created, the execution stops at the first step.	PASS
TP-ReusR-01	Test the TC-ReusR-01 and TC-ReusR-02 by creating multiple pipelines that contain common tool components. The pipelines are executed OK even if executed in the same time.	The execution of pipelines containing same components do not interfere one with other.	PASS
UV-INT-TAO	During the physical plenary meetings prepare testing scenarios for participating partners in consortium. Perform the steps detailed in the testing scenarios document	Pipelines were successfully created by the participants, executed, running status updated and results available from the web interface. Users filled in the feedback survey presented in the annex.	PASS

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

Test ID	Expected Result	Actual Result	Pass/Fail
	concerning the creation of a pipeline in TAO, its parametrization, execution, monitoring and results visualization.	Usability KPI was also evaluated and TAO was appreciated by all users as easy to use.	

3.5.23 Carbontracker (GreenR)

3.5.23.1 Test cases

TC-GreenR-01: Real-time carbon emission	
Test item	Test to validate fetching of live, region-specific, carbon emissions data
Covered requirements	**EVL-CT-1**
Input specification	Run CarbonTracker CLI on a task with an electricitymaps API key.
Expected results	Live carbon emission data and logs specifying the found region and carbon intensity value at the given time.
Pass/fail criteria	Pass: Carbon emission data matches the location of task execution. Fail: invalid emission data received, using fallback data.
Environmental needs	CarbonTracker package
Dependencies	API key from electricitymaps.

TC-GreenR-02: Graphical overview of tracked activities	
Test item	Test to show a graphical overview of measured carbon emissions data
Covered requirements	**EVL-CT-2**
Input specification	Run CarbonTracker CLI on a task.
Expected results	Highlight carbon emission and energy consumption per time period, e.g., epoch, during task activity, compare consumption over time from different hardware components, e.g., CPU and GPU.
Pass/fail criteria	Pass: A graphical, i.e., not text logging, overview of carbon emission and energy consumption data for compute task can be generated from logs. Fail: No measurements in log file to collect.
Environmental needs	CarbonTracker package
Dependencies	N/A

TC-GreenR-03: Interfacing with other tools through CLI	
Test item	Test to show CarbonTracker functionalities can be used through CLI
Covered requirements	**EVL-CT-3**
Input specification	Run CarbonTracker CLI on a task.

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

Expected results	Regular CarbonTracker execution incl. Logging of results interfaced with execution of other tools run from the command line.
Pass/fail criteria	CarbonTracker does not interfere with execution of other programs when interfaced with other tools from the command line. Logging of activities.
Environmental needs	CarbonTracker package
Dependencies	N/A

3.5.23.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-GreenR-01	Execute the script greenr.sh and check the expected results mentioned in TC-GreenR-01, TC-GreenR-02 and TC-GreenR-03	As expected for TC-GreenR-01, TC-GreenR-02 and TC-GreenR-03.	PASS

3.5.24 System-wide requirements

3.5.24.1 Test cases

3.5.24.1.1 Security and Privacy Requirements

TC-SYS-01: login on platform	
Test item	Test to show the user can login into the integration platform represented by TAO using his credentials
Covered requirements	**REQ-SYS-1**
Input specification	User login into platform with username / password
Expected results	Successful login
Pass/fail criteria	Pass: User is active, the platform recognizes the credentials and creates the session for user. Fail: fail to login
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-SYS-03: login on platform	
Test item	Test to show the user is able to change the password of their respective accounts by themselves.
Covered requirements	**REQ-SYS-3**
Input specification	User changes its password
Expected results	Password changed
Pass/fail criteria	Pass: After logout, the new password can be used. Fail: The password was not changed

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

Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-SYS-04: components availability and sharing	
Test item	Test to show that certain features can be used only by an administrator.
Covered requirements	**REQ-SYS-2, REQ-SYS-4, REQ-SYS-5, REQ-SYS-6**
Input specification	Login as administrator, then logout and login as a regular user.
Expected results	Processing components and Container sections are available only when logged as administrator. When logged as regular user, only own workflows are visible.
Pass/fail criteria	Pass: The sections are available only when logged as admin and only own workflows are seen, unless shared by other users. Fail: Sections missing.
Environmental needs	TAO platform is up and running
Dependencies	N/A

3.5.24.1.2 Resource administration

TC-RAD-01: Resources administration	
Test item	Test to show that the user can add, edit or delete resources stored in the system and that resources shared by the administrator can be used by other users.
Covered requirements	**RQ-RAD-1, RQ-RAD-2, RQ-RAD-3**
Input specification	In the user product repositories section, the user can add, edit, delete resources. The user can also delete his own workflows and can create workflows with the processing components added by the administrator
Expected results	The user can create, delete or move files within his repositories and can add and delete his own workflows.
Pass/fail criteria	Pass: No errors occur during the operations. Fail: errors occurred during these operations.
Environmental needs	TAO platform is up and running
Dependencies	N/A

3.5.24.1.3 User interfaces

TC-UI-01: components availability and sharing	
Test item	Test to show that system is accessible to the users via a web interface and that the user can change look and feel, see results of activities in dashboard and create new workflows.
Covered requirements	**RQ-SUI-1, RQ-SUI-2, RQ-SUI-3, RQ-SUI-4**

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

Input specification	Access TAO URL as specified in the User Manual and login with the user and password. Create a workflow and execute it.
Expected results	The user can access TAO's webpage and perform the login. The workflow is edited in the dashboard and can be executed. The messages from the execution are displayed in the dashboard.
Pass/fail criteria	Pass: TAO's webpage is accessible and login successful. Workflow is created and executed. Messages during execution are visible in the dashboard. Fail: any error during the operations performed.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC-UWS-01: Workflows editing and execution	
Test item	Test to show that the user can add and edit workflows in a visual manner, and that the created workflows can be executed.
Covered requirements	**RQ-UWS-1, RQ-UWS-2, RQ-UWS-3**
Input specification	The user creates a workflow by dragging and dropping processing components on the workflow table and then interconnecting them. Once the workflow is saved, it can be executed, and the execution steps are shown in the dashboard.
Expected results	The workflow is created, and the user cannot interconnect incompatible components (when the output of the component is incompatible with the input of another component). The steps of execution are shown in the dashboard.
Pass/fail criteria	Pass: The workflow is successfully created and executed, while the steps are shown and updated in the dashboard. Fail: errors during workflow execution
Environmental needs	TAO platform is up and running
Dependencies	N/A

3.5.24.1.4 Interfaces with External Systems

TC- SAPI-01: Accessing through the API	
Test item	Test to show that the system offers 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.
Covered requirements	**RQ-SAPI-1**
Input specification	Using curl or other REST API client, requests to login, launch execution and get job status endpoints are invoked.
Expected results	The platform responds with successful results.
Pass/fail criteria	Pass: The requests are successful.

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

	Fail: Requests failed.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC- SAPI-02: Accessing external services	
Test item	Test to show that the system is 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
Covered requirements	**RQ-SAPI-2**
Input specification	Create a processing component that invokes an external service, add it to a workflow and execute the workflow.
Expected results	The external service is invoked during the workflow execution.
Pass/fail criteria	Pass: The request to the external service is performed successfully. Fail: errors during request.
Environmental needs	TAO platform is up and running
Dependencies	N/A

TC- SAPI-03: Accessing external services	
Test item	Test to show that the system able to retrieve and display information from GreenR about measured carbon emissions and other statistics
Covered requirements	**RQ-SAPI-3**
Input specification	Execute a workflow having components that can provide carbon emissions and other statistics provided by GreenR.
Expected results	The carbon emissions received are displayed in the dashboard for the executed job.
Pass/fail criteria	Pass: The GreenR statistics are displayed successfully. Fail: errors generating statistics or statistics are 0.
Environmental needs	TAO platform is up and running
Dependencies	N/A

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

3.5.24.2 Test procedures

Test ID	Expected Result	Actual Result	Pass/Fail
TP-SYS-01	Open TAO web interface and perform the operations mentioned in the test cases TC-SYS-03, TC-SYS-04, TC-RAD-01, TC-UWS-01, TC-UI-01. The user can login, workflows can be created and executed. The executions are viewed in Dashboard and result files in the repositories can be open copied or downloaded.	As expected.	PASS
TP-SYS-02	From the same machine or another machine, execute the script test_tao_api.sh that invokes TAO API and invokes external services according to TC- SAPI-01 and TC- SAPI-02. All curl executions performed from script are successful.	As expected.	PASS
TP-SYS-03	In the web interface perform the steps described in test cases TC-SYS-01, TC-SYS-03, TC-SYS-04, TC-RAD-01, TC-UWS-01, TC-UI-01 (login as admin and regular user, change password, create and execute workflows)	The system behaves as expected in the mentioned test cases.	PASS
TP-SYS-04	Execute a pipeline and check the Carbontracker statistics.	The carbon tracker statistics are aggregated for jobs, from each step executed.	PASS

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

4 Toolbox releases

4.1 General

Information and installation packages of the toolbox can be found at:

- enRichMyData Changelog – what's new in the current version
  https://s3.waw3-2.cloudferro.com/swift/v1/EMD/CHANGELOG.md
- enRichMyData installation ReadMe.txt file
  https://s3.waw3-2.cloudferro.com/swift/v1/EMD/ReadMe.txt
- enRichMyData toolbox user manual with information about installation and usage
  https://s3.waw3-2.cloudferro.com/swift/v1/EMD/D3.2_User_Manual.docx
- enRichMyData toolbox on GitHub:
  https://enrichmydata.github.io/toolbox/
- A video demonstrating the usage of the toolbox is available here:
  https://www.youtube.com/watch?v=E8rmOVOFC-0&t=64s

4.2 enRichMyData Toolbox V2.0

Information and installation packages of the toolbox can be found at:

- enRichMyData toolbox installation package
  https://s3.waw3-2.cloudferro.com/swift/v1/EMD/20241015_EmdInstallPackage.zip
- Deployed instance of the enRichMyData toolbox:
  https://64.225.143.9:8443/ui/login.html

The toolbox offers the following functionalities:

- Integration framework represented by TAO (TAO Project · GitHub)
- Several tools already integrated as Docker images (LamAPI, OntoRefine, SemT-X, Alligator) along with a set of additional helper tools used for different operations (ex. open-meteo invocation, filtering CSV columns, merging CSV files etc.). See https://enrichmydata.github.io/toolbox/ for github locations of the integrated tools.
- Demonstration pipelines for SpendNetwork and JOT use cases are already included into the system database as proof of concept. Please check the above mentioned ReadMe.txt file to see how you can have these pipelines in your own installed TAO instance.

4.3 enRichMyData Toolbox V3.0

Information and installation packages of the toolbox can be found at:

- enRichMyData toolbox installation package
  https://s3.waw3-2.cloudferro.com/swift/v1/EMD/20250630_EmdInstallPackage.zip
- Deployed instance of the enRichMyData toolbox:
  https://64.225.140.189:8443/ui/login.html

The toolbox offers the following functionalities:

- Integration framework represented by TAO (TAO Project · GitHub)
- All enRichMyData tools providing a command line interface or accessible through a REST API were integrated as Docker images on the integration platform. These tools include LamAPI, Crocodile, Lion-Linker, Wikifier, OntoRefine, SemT-X, Alligator, Expert.AI tools etc. When tools are provided as external services and if was not possible to deploy them directly on the

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

integration platform, connectors to them were added into TAO allowing these external services to be used as processing components into pipelines. Along with the tools, there was also a set of additional helper or custom tools used for different operations (ex. open-meteo invocation, filtering CSV columns, merging CSV files etc.). The custom tools included are usually developed by the partners into consortium as part of their corresponding pipeline. It's worth mentioning that some of the tools that are used in the exploration phase and that are mainly tools where graphical interface is playing an important role while using the tool (like Human-In-The-Loop activities) were not added as processing components in TAO. Nevertheless, the results (intermediate or final) from the pipelines executions can be used by these tools.

- Demonstration pipelines for Philips, Bosch, SpendNetwork, JOT and InnoGraph use cases are already included into the system database as proof of concept. Please check the above-mentioned ReadMe.txt file to see how you can have these pipelines in your own installed TAO instance.

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

5 Deployment infrastructure

Information about each of the enRichMyData tools deployment can be found on the toolbox page: https://enrichmydata.github.io/toolbox/.

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

6 Toolbox openness

The enRichMyData toolbox tools openness is detailed in deliverable D2.2 for each individual tool in the toolbox so in this section we focus more on the integration framework openness.

The TAO framework, which is used for the integration of tools and creation of pipelines, is available as opensource on GitHub in the following address https://github.com/tao-org. It has an open architecture allowing external tools to be integrated and used in the pipelines. The software user manual (SUM) provides detailed documentation on how new tools can be integrated. For the integration of the tools, there is no limitation, any tool that can be executed (either through command line or through the invocation of an API, no matter if they are internal or 3rd party tools that are necessary for business cases) can be integrated as TAO will always use Docker images encapsulating the tools. All the Docker images of tools are added into TAO through their descriptors, which define what applications are available and provide information about their invocation and interconnectivity. In TAO, tools, both commercial and open-source, can be used as long as the license constraints concerning the usage and distribution of the tools are respected.

The execution and orchestration framework represented by TAO allows also to be used via the exposed REST API interface, all the functionalities available from the web interface being available also through the REST API. Additionally, TAO is implementing an OGC WPS (Web Processing Service) client and server allowing both to invoke external tools providing this interface but also to be invoked by external application in order to invoke the implemented pipelines. To enable execution of workflows implemented in TAO on other external environments or platforms, the export from TAO internal format into CWL, JSON, Bash or Argo Workflows is also supported.

As TAO has a pluggable architecture, new resource managers, data sources or processing services can be further added. Thus, for example, the current list of resource managers represented by the local invocations, SSH, Slurm, Torque or Kubernetes, can be extended programmatically with new ones.

During each release, an installation package is created containing all the installation scripts and the needed libraries to quickly set up a fully functional enRichMyData integration toolbox instance. The installation package will contain all dependencies needed (database creation, additional packages installation) but also all the Docker images for the tools supported in that version. The installation package is published and available for free download, installation and usage on local premises or in cloud environments. The installation of the toolbox is straightforward and very simple, by running a single script all installations and configurations being automatically performed.

Beside TAO, the installation package contains already some of the enRichMyData tools that are deployed either as services (like LamAPI, Crocodile, Lion-Linker, SemT-X, etc.) or as Docker images that are included as TAO application containers. These tools can be freely used within existing pipelines deployed with TAO or in new pipelines that can be developed. Additionally, the tools deployed as services are used both by the pipelines created but also can be used by other third-party applications.

A public deployment was made available for testing purposes (with certain limitations, e.g., datasets, processing, etc.) and intermediate internal versions were released.

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

7 Validation and KPI evaluation report

7.1 enRichMyData Toolbox V3.0 Validation Report

This chapter provides a detailed summary of the validation and evaluation activities performed on the tools included in the project toolbox. The objective of this process was to verify the correct implementation of the tool functionalities, validate their fitness for use, and assess compliance with quality and performance expectations defined at the project level. The validation followed a structured approach, based on the progressive evolution of each tool, and was supported by test procedures developed in accordance with a simplified process defined in some of the well-known verification and validation standards like ECSS-E-ST-10-02C and ISO/IEC/IEEE 29119. The verification and validation process that was applied for the enRichMyData project was presented in section 3.4.

For each tool, a set of test cases was developed to systematically verify its core functionalities. These test cases were defined during the development lifecycle in close alignment with the initial requirements and user expectations. Each test case was linked to specific functional or non-functional requirements and was implemented in the corresponding test procedures. These procedures included the definitions of inputs, test environments, execution steps, expected outputs, and pass/fail criteria.

The validation process was executed iteratively, with formal test sessions conducted at different development milestones or during the physical plenary meetings of the consortium.

The test cases and test procedures along with the results obtained were presented in section 3.5, for each tool.

7.2 enRichMyData Toolbox V3.0 KPI Evaluation Report

The KPIs for each tool were defined in the deliverable “D2.2 – enRichMyData Tools”.

These KPIs were also evaluated during the validation process. Whenever possible, KPIs were measured directly as part of the test procedures. For instance, execution times, memory usage, and data processing accuracy were recorded during functional and performance tests. In cases where direct measurement was not feasible within the test setup, alternative evaluation methods were applied. This included expert analysis of tool design, review of code quality and documentation, and structured user feedback collected through post-evaluation surveys.

Surveys were distributed to users across the consortium following the testing phases, with the goal of capturing practical insights related to usability, integration into workflows, and overall satisfaction. The survey responses provided important qualitative validation data and were used to complement the quantitative test results. In particular, user feedback contributed to the evaluation of usability-related KPIs and provided a basis for prioritizing future enhancements. The results of the survey for the tools evaluated are presented in 8 Annex – Tools evaluation questionnaires results.

The next table presents the evaluation report for all KPIs defined in “D2.2 – enRichMyDataTools”:

Tool	KPI	Target	Actual	Status	Comment
SemT-UI	Usability	User Experience Questionnaire (UEQ) – results above the 0.8 positive threshold for Attractiveness, Pragmatic Quality, and Hedonic Quality	Attractiveness = 1.67 Pragmatic Quality = 1.39 Hedonic Quality = 1.46	Achieved	
OntoText GraphDB	Performance	*Accuracy* values compared to the baseline of the benchmark. Target value >85% accurate queries corresponding to natural language questions. Evaluated by hand.	Accuracy consistently > 85%	Achieved	Evaluation of the TTYG component on query rewriting

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

	Integration	No regression w.r.t existing functionalities of GraphDB	Integration successful	Achieved	This concerns the integration of Semantic Object into GraphDB
OntoText Refine	Usability	Use the tool in EMD use cases and handle the large amounts of data supplied by the use cases.	Same	Achieved	
s-elBat (Alligator)	Quality of annotations (Linking/CEA)	Entity linking accuracy on diverse benchmark datasets comparable to SOTA (mostly LLM-based approaches). Accuracy is measured by assuming the correct candidate is always present in the list of candidates to choose from. CEA Accuracy >= 0.8	Tested on 6 different datasets (https://arxiv.org/pdf/2408.06423): Alligator-FT, the fine-tuned version of alligator on domain-specific data, performs better then the original Alligator. • Alligator-FT: 0.800 • Alligator: 0.737 And performs as or better then LLM based linkers: • TURL: 0.445 • TableLlama: 0.812 • GPT-4o: 0.862	Achieved	
	Full STI (CTA and CPA)	Ability to perform full STI: Cell Entity Annotation (CEA), Column Type Annotation (CTA) and Column Property Annotation (CPA) *CEA: see above* *CTA F1 >= 0.8* *CPA F1 >= 0.8*	CTA: • HTR1- F1: 0.927 • HTR2- F1: 0.817 CPA: • HTR1- F1: 0.840 • HTR2- F1: 0.847 https://arxiv.org/pdf/2408.06423	Achieved	

	Efficiency	Average processing time: ability to process 1000 rows in less than 500s	Test on a sample table of 1000 rows and 11 columns (with 4 of them containing entities)
			• W/o cache: 297.2s
			  ○ 3.36 rows/s
			  ○ 13.46 mentions/s
			• W/ cache: 29.2s
			  ○ 34.23 rows/s
			  ○ 82.98 mentions/s
			Hyperparameters:
			• 50 candidates
			• 8 workers
			• 16 rows per worker
			• 2 ml workers
				Achieved	

Crocodile	Disambiguation Accuracy	Entity linking accuracy on benchmark datasets comparable to Alligator. Accuracy is measured by assuming the correct candidate is always present in the list of candidates to choose from. Specifically used: HTR1 and HTR2	HTR1 Accuracy: 0.9154 HTR2 Accuracy: 0.8444 Hyperparameters: • 10 candidates	Achieved	

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

	Efficiency	Average processing time: ability to process 1000 rows in less than 400s	Test on a sample table of 1000 rows and 11 columns (with 4 of them containing entities)
			• W/o cache: 274.2s
			  ○ 3.65 rows/s
			  ○ 14.59 mentions/s
			• W/ cache: 8.7s
			  ○ 115.51 rows/s
			  ○ 462.03 mentions/s
			Hyperparameters:
			• 50 candidates
			• 8 workers
			• 16 rows per worker
			• 2 ml workers
				Achieved	

Lion-Linker	LLMs models interoperability	Employment of 2+ Large Language models	• Possibility to ue all Ollama models • Possibility to use all OpenAI models • Possibility to use Grok models	Achieved	
	Retriever Flexibility	3+ candidate retrievers	• Wikidata Lookup Service • LamAPI • OpenRefine	Achieved	
Koala-UI	Usability	Tool usability was evaluated through a user questionnaire		Achieved	
	Integration		All tested endpoints functioned as expected	Achieved	
OntoText Reconcile	Performance	Benchmark. Ability to process 10K lines of input data in less than 1 minute	Same	Achieved	
Wikifier	Scalability	Scalability documents (news, code, etc.): **Values:** minimum 1.000.000 documents	Currently annotated over 500.000 code documents + over 400.000 science documents + around 2.000.000 news documents	Achieved	
	Annotation precision	Measuring Precision Top N for semantic annotation/wikification task based on developed gold standard; Annotation Precision Top 1 > 0.8	Precision Top 1 (All) 0.921 Precision Top 3 (All) 0.842 Precision Top 10 (All) 0.801 Precision Top 1 (PageRank > 0.01) 0.975 Precision Top 3 (PageRank > 0.01) 0.965 Precision Top 10 (PageRank > 0.01) 0.960	Achieved	
Expert.ai Platform Document Analysis	Quality of context identification	Values for the measure (concrete values): Number of correct and acceptable answers > 80%	>80% value obtained from the average of several experiments conducted over the years https://docs.expert.ai/platform/1.10/authoring/how-to/categorization/manage/interpret-results/history/	Achieved	
ER Relation Classifier	Scalability	Number of instances processed over 10.000.000	Currently 1,287,332 (as of October 1, 2024), 4.500.000 (as of June 2025) identified	Achieved	

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

		Relevant news articles identified > 1.000.000	Processed news instances: over 14.000.000		
	Performance	F1 measure (per class) > 0.7	Technology-use: **0.782** Technology-use + AI: **0.9085** Technology-use + Blockchain: **0.91**	Achieved	
Categorizer	Performance	F1 measure > 0.7	Taxonomy (auto), Macro-F1 Score, Val: 0.93 Taxonomy (auto), Macro-F1 Score, Test: 0.87 Taxonomy (auto), Micro-F1 Score, Val: 0.90 Taxonomy (auto), Micro-F1 Score, Test: 0.70 Taxonomy (manual), Macro-F1 Score, Val: 0.93 Taxonomy (manual), Macro-F1 Score, Test: 0.86 Taxonomy (manual), Micro-F1 Score, Val: 0.91 Taxonomy (manual), Micro-F1 Score, Test: 0.71	Achieved	
Expert.ai Platform Document Classification	Quality of document classification	F1 >80%	F1 > 80 value obtained from the average of several experiments conducted over the years https://docs.expert.ai/platform/1.10/authoring/how-to/extraction/manage/interpret-results/compare/	Achieved	
Expert.ai Expert System generator (Q&A)	Quality of context identification	Values for the measure (concrete values): Number of correct and acceptable answers > 80% MRR> 85%	>80%	Achieved	
LamAPI	Efficiency	Retrieval time of mentions/s	147.71s for 4000 mentions, i.e. 27.08 mentions/s	Achieved	
	Quality of entity retrieval	Coverage >= 0.8	Coverage = 0.90 for top-30 candidates on four out of six datasets from the SemTab challenge Coverage = 0.80 for top-100 candidates on two out of six datasets from the SemTab challenge Coverage on TURL-2K-red for top-50 candidates is 88.17 for LamAPI and 71.75 for Wikidata Lookup Service		
	Impact of types during entity retrieval	Candidate retrieval with enhanced type search (given by the transitive closure) improves coverage and MRR on domain specific data	W/o Transitive Closure: 0,784 W/ Transitive Closure: 0,814	Achieved	

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

		Coverage >= 0.8			
TAO	Performance	Max 2 seconds overhead between scheduling two pipeline steps	The actual values obtained is less than 0.5 between scheduling two pipelines steps	Achieved	
	Scalability	Improve performances with >= 25% when having an additional node	On pipelines where parallelization was possible, performances were increased by more than 75% (an overhead being introduced by the data transfer between nodes). Nevertheless, on pipelines not supporting parallelization, no improvement was observed (as expected)	Achieved	
	Usability	Easiness in creating pipelines, executing and monitoring	TAO was appreciated by all users as easy to use.	Achieved	Advanced features might be sometimes difficult to use but usually those kind of operations are made by the admin or IT-oriented people.
SIM-PIPE	Usability	Manual tests (pass/fail). Is TAO able to use SIM-PIPE API to: Execute dry-runs Perform resource prediction Retrieve resource prediction Retrieve resource prediction without any further processing of the data	True (Pass) True (Pass) True (Pass) True (Pass)	Achieved	Integration with TAO API has been implemented
Event Registry	Scalability	Number of documents provided > 1.000.000	Number of provided documents: around 2.000.000 news documents	Achieved	
Stream Story	Usefulness		Majority of respondents identified tool as useful to their profile	Achieved	
Time Series Explorer Tool	Performance	ROC AUC >0.7 (BGRIMM business case, predicting the power of the SAG mill with machine learning and graph methods)	Experiment where all data was embedded with TS2Vec models, shift to predict 0: 0.9948 ROC AUC Experiment where all data was embedded with TS2Vec models, shift to predict 30: 0.9951 ROC AUC Experiment where all data was embedded with TS2Vec models, shift to predict 50: 0.9940 ROC AUC Experiment where all data was embedded with TS2Vec models, shift to predict 70: 0.9934 ROC AUC	Achieved	

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

			Experiment where all data was embedded with TS2Vec models, shift to predict 90: 0.9925 ROC AUC		
CarbonTracker	Interoperability	**Platform support:** Compatible with Intel, AMD, NVIDIA, and Apple silicon.	n/a	Achieved	
	Task-agnostic tool	**Tool usage:** Provides both CLI and Python API for flexible integration.	n/a	Achieved	
	Accuracy	**Real-time accuracy:** Real-time *electricitymaps* ensure accurate carbon intensity measurements.	n/a	Achieved	
	Integration & Fallback	**Fallback:** Uses a predefined set of average CPU energy consumption values when power data is unavailable.	n/a	Achieved	
ABSTAT	Efficiency	Speed: On a 1 node distributed infrastructure, ABSTAT-HD is on average: 3x faster than ABSTAT in core profiling mode; 42x faster than ABSTAT in full profiling mode. Graph size: experiments with ABSTAT were run, with a 5-node distributed infrastructure, on an 8.18B triple graphs, 6x larger than graphs profiled with DistLODStat (efficient profiling method computing statistics but not patterns)		Achieved	Tool was not further developed in the project after M12.
	Conciseness	Compression rate = 0,56 on DBpedia 2014 (566M triples).		Achieved	
	Usefulness	Macro-averaged increase in query completion accuracy vs Protege = +10% Macro-averaged decrease in time required for query completion = -49,29%		Achieved	

In conclusion, the validation and evaluation of the toolbox were performed through a comprehensive and methodical process based on formal test procedures and verification standards. All tools were evaluated against their intended requirements, and test results confirmed a high level of compliance, functional completeness, and readiness for operational deployment. The validation also highlighted areas for improvement in future development cycles. This structured approach ensures that the toolbox meets the quality and performance expectations of the enRichMyData project and provides a reliable foundation for its use in real-world applications.

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

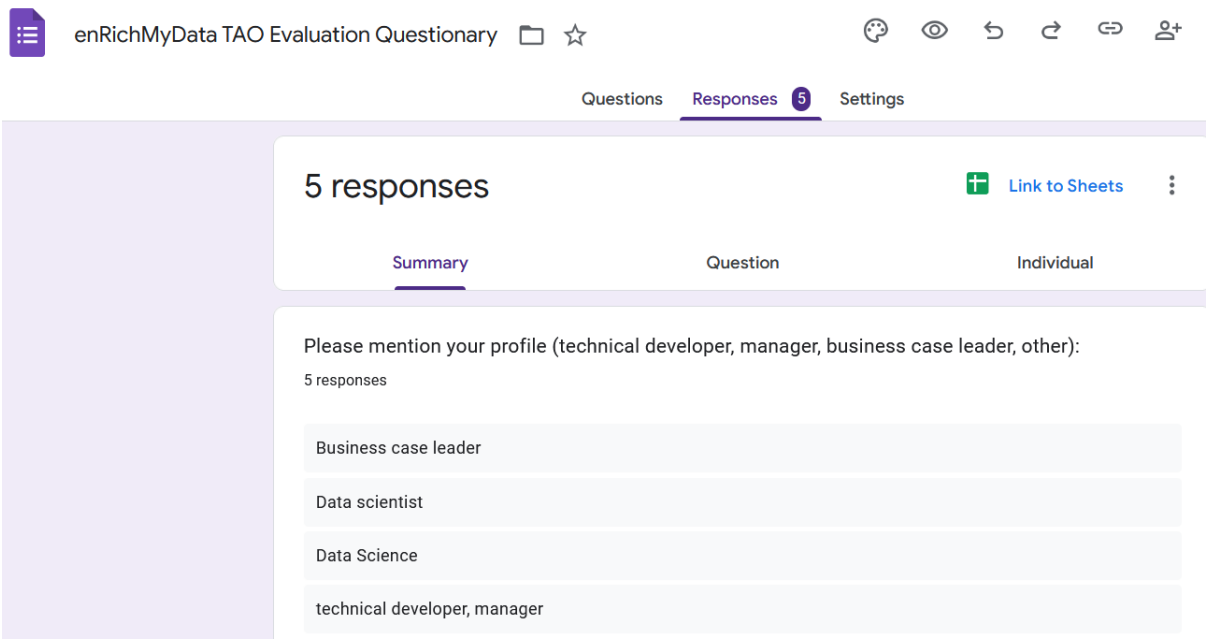
8 Annex – Tools evaluation questionnaires results

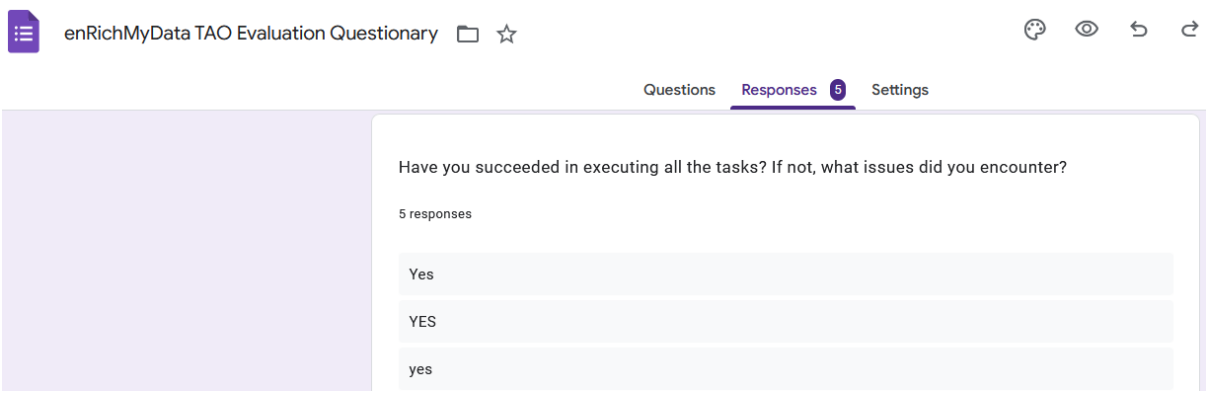
This annex presents a comprehensive overview of the feedback collected from consortium members who participated in testing and evaluating some of the tools included in the enRichMyData toolbox. The surveys were distributed to users across all participating members to the physical plenary meetings, following the evaluation phase, with the aim of assessing the usability, effectiveness, and overall user satisfaction with each tool.

The annex includes both quantitative and qualitative data derived from structured questionnaires. Key metrics evaluated include ease of use, integration into existing workflows, performance, reliability, and perceived added value. Users were also encouraged to provide open-ended comments and suggestions for improvement.

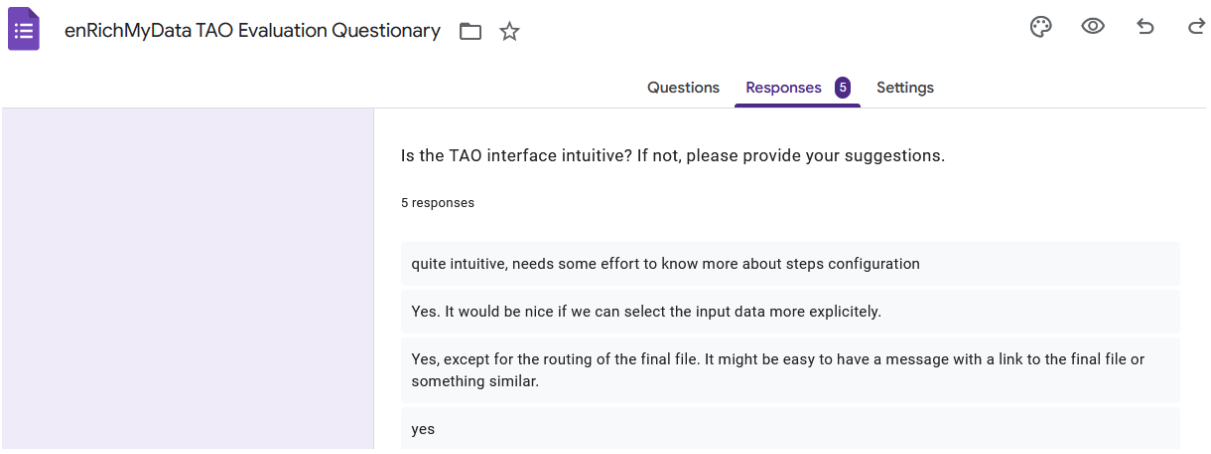
This annex serves as a valuable reference for understanding user perspectives, identifying areas for enhancement, and guiding future development and dissemination strategies within the consortium and beyond.

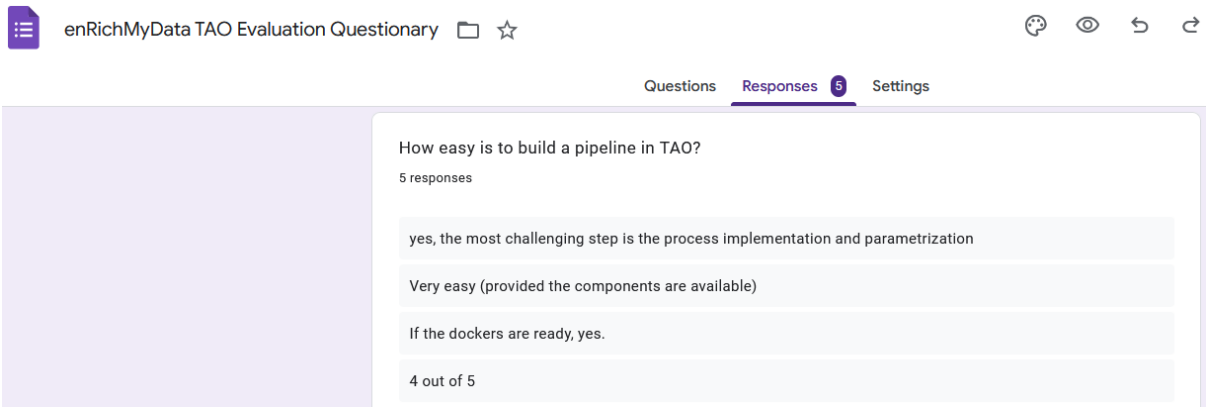
8.1 TAO

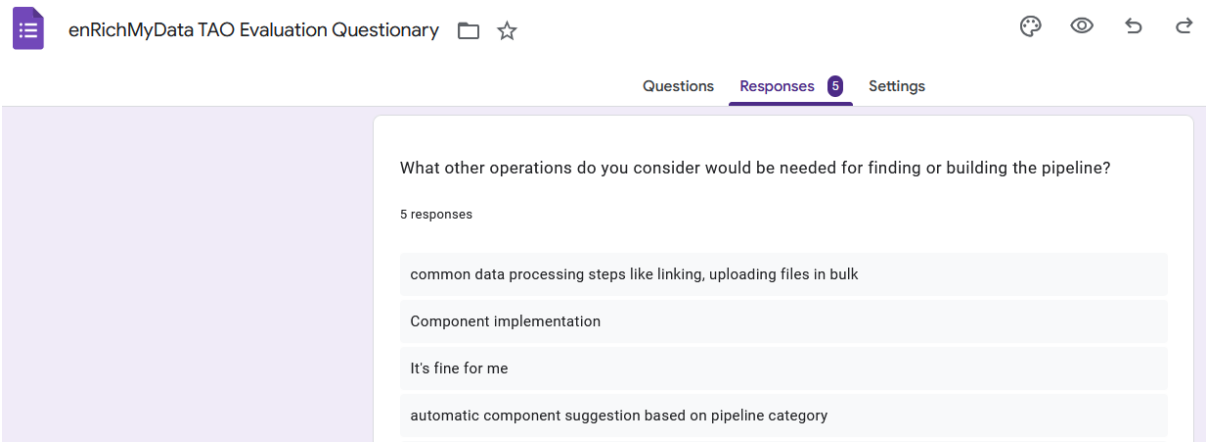


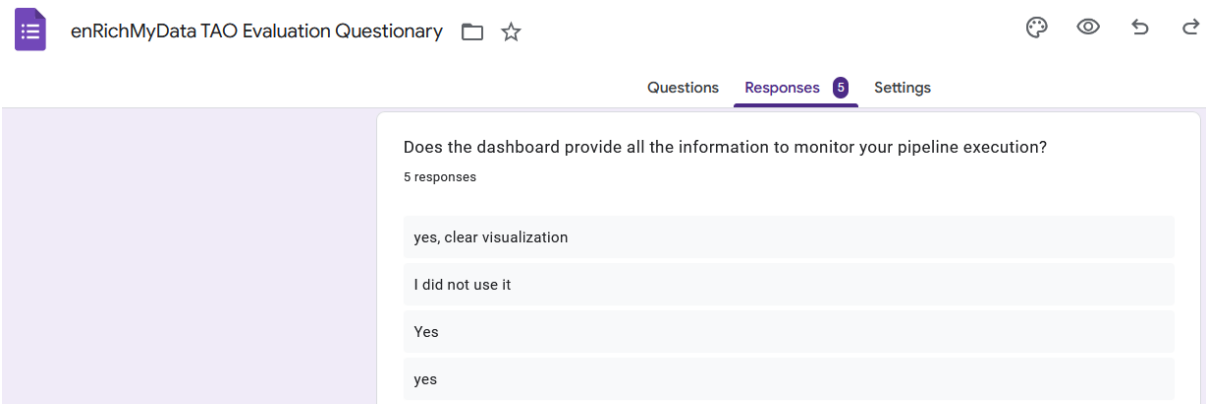


THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.

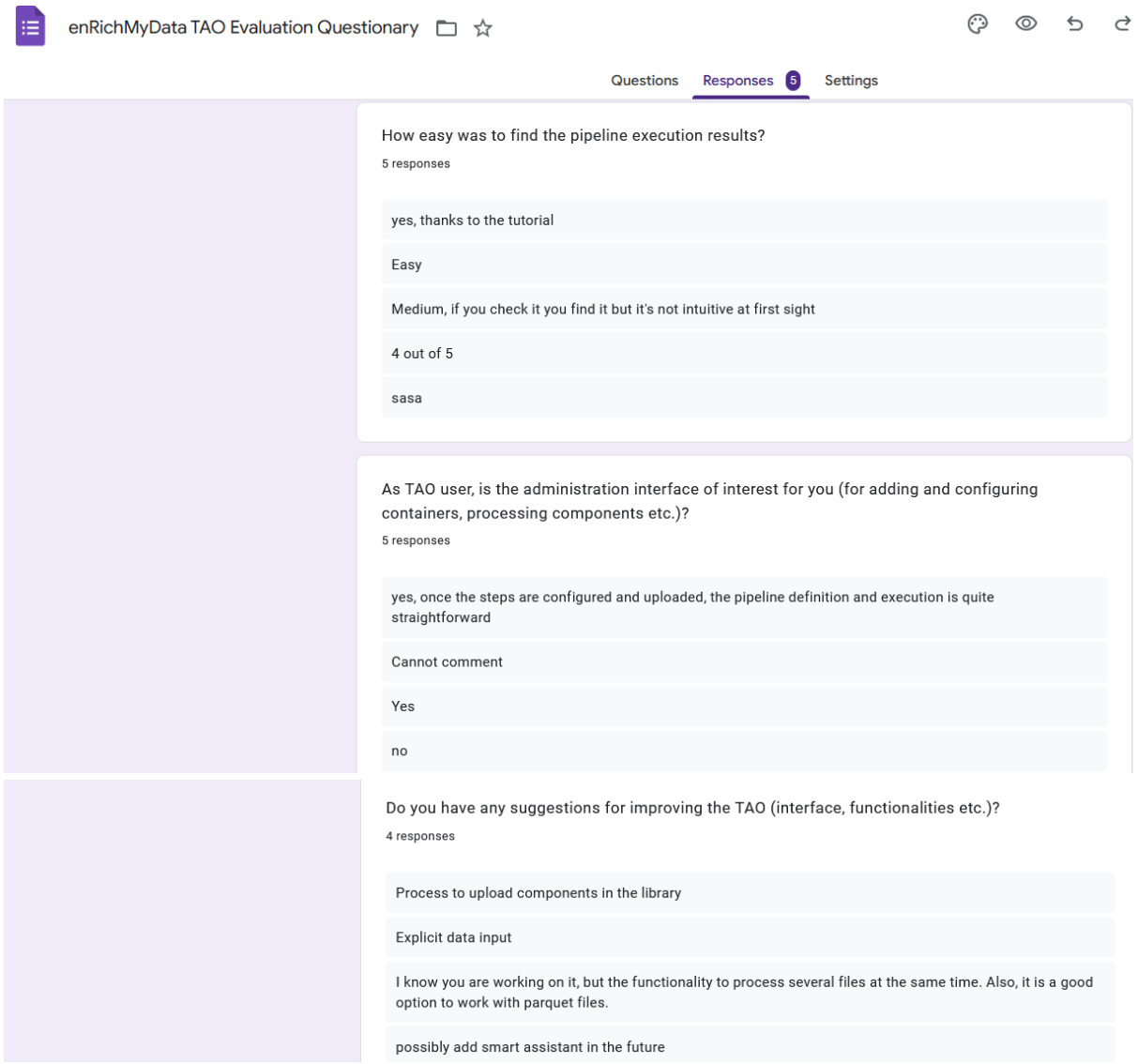








THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON EUROPE RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 101070284.



enRichMyData TAO Evaluation Questionary

Questions Responses 5 Settings

How easy was to find the pipeline execution results?

5 responses

- yes, thanks to the tutorial
- Easy
- Medium, if you check it you find it but it's not intuitive at first sight
- 4 out of 5
- sasa

As TAO user, is the administration interface of interest for you (for adding and configuring containers, processing components etc.)?

5 responses

- yes, once the steps are configured and uploaded, the pipeline definition and execution is quite straightforward
- Cannot comment
- Yes
- no

Do you have any suggestions for improving the TAO (interface, functionalities etc.)?

4 responses

- Process to upload components in the library
- Explicit data input
- I know you are working on it, but the functionality to process several files at the same time. Also, it is a good option to work with parquet files.
- possibly add smart assistant in the future

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

8.2 ExpertAI Q&A

Do you find the waiting times for generating the expert system reasonable? (Considering that they depend on the length of the PDF)	How useful do you find the feature of referencing the original PDF passages for explainability purposes?	Do you consider the response generated by the LLM an added value compared to having only the relevant document passages?
Yes	4	Useful
Yes	5	Essential
Yes	5	Essential
Yes	5	Essential
Yes	3	Useful
Yes	4	Useful
Yes	4	Useful
Yes	5	Useful
Yes	4	Useful
Yes	5	Essential
Yes	5	Useful

In terms of accuracy, do you consider the responses received to be:	Did you generally find the correct answer within the first five retrieved passages?	Does the generated response summarize well the answer you were expecting?	What would you change to improve usability?
Correct	No	Yes	I had liked to have a more precise relationships between pieces of the generated answer and pieces of the contexts used to generate it
Correct	Yes	Yes	The UI/UX can be improved. For example, the default dashboard should be more intuitive. It is unclear how to get to the right window from the documentation. It is unclear what is "QA product manager" in this UI
Almost correct	Yes	Yes	Generate answer should be in default.
Almost correct	Yes	Yes	N.A.
Almost correct	Yes	Yes	NA
Correct	No	Yes	The "eye" icon is hard to find and doesn't correlate to the work / task being undertaken. Also hard to know that I can look at multiple files.
Correct	Yes	Yes	Responses UI should be cleaner, at least for non-experts
Correct	Yes	Yes	AI generated summary should reference the sources. Perhaps re-rank the results, so that they show up in the same order as how they appear in the summary.
Correct	Yes	Yes	I had liked to have more detailed information about support from the chunks to pieces of the answers.
Correct	Yes	Yes	Gen ai should be ON by default for more complex queries, have inline links referencing the chunk in the source document . Look at Google NotebookLM
Almost correct	Yes	Yes	Usability is already good.

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

8.3 Lion-Linker

1) How easy was it to install the library?	2) How easy was it to understand and use the library?	3) Did the library perform as expected for your tasks?
2	2	Yes
1	1	Yes
3	3	The tasks were not explained

4) How helpful did you find the documentation?	5) What improvements or features would you like to see in the library?
2	extension of possible functionalities, confidence for linking
3	Nothing special in this moment, I will see when using it.
2	more documentation and use of a configuration file (e.g., the model to use)

8.4 SemT-X (SemT-UI)

UEQ User Experience Questionnaire

The user test for SemTUI was conducted following the UEQ methodology (https://www.ueq-online.org/).

The UEQ questionnaire contains 6 scales with 26 items to evaluate the following aspects:

- Attractiveness: Overall impression of the product. Do users like or dislike the product?
- Perspicuity: Is it easy to get familiar with the product? Is it easy to learn how to use the product?
- Efficiency: Can users solve their tasks without unnecessary effort?
- Dependability: Does the user feel in control of the interaction?
- Stimulation: Is it exciting and motivating to use the product?
- Novelty: Is the product innovative and creative? Does the product catch the interest of users?

The UEQ does not produce an overall score for the user experience. Because of the construction of the questionnaire, it does make no sense to build such an overall score (for example by calculating the mean over all scales), since this value cannot be interpreted properly.

The scales of the UEQ are grouped into pragmatic quality (Perspicuity, Efficiency, Dependability) and hedonic quality (Stimulation, Originality). Pragmatic quality describes task related quality aspects, and hedonic quality the non-task related quality aspects.

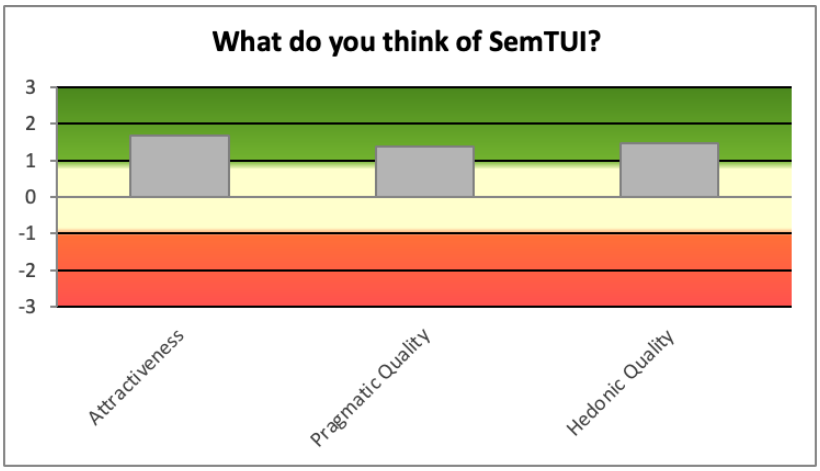
Values between -0.8and 0.8 represent a neutral evaluation of the corresponding scale, values > 0,8 represent a positive evaluation and values < -0,8 represent a negative evaluation.

The range of the scales is between -3 (horribly bad) and +3 (extremely good). But in real applications, in general, only values in a restricted range will be observed. It is due to the calculation of means over a range of different persons with different opinions and answer tendencies (for example the avoidance of extreme answer categories) extremely unlikely to observe values above +2 or below -2.

The questionnaire received 9 answers that led to the following result:

Pragmatic and Hedonic Quality	
Attractiveness	1.67
Pragmatic Quality	1.39
Hedonic Quality	1.46

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



8.5 Wikifier and StreamStory

Please specify your profile (technical developer, manager, business case leader, other)	Which tools did you try?	Have your expectations for text processing and data exploration tasks been met?	What was the biggest challenge for you and what did you learn?
Technical developer	Wikifier	4	Understanding the parameters
tecnhical developer	Wikifier	5	No particular challenges, apart from the postman call not recognising the api key
technical developer	Wikifier	5	
technical developer	Wikifier	5	
technical developer	Wikifier	5	Understand the criteria used to link and the information reported in the interface about the evidence supporting the decision
technical developer, researcher	Stream Story	4	Somewhat unclear what the different features are at first. After clicking around I was able to understand.
technical developer	Wikifier;Stream Story	4	
business case leader	Wikifier;Stream Story	4	The need for preprocessing and an awareness of how the product works in order to understand the visualisations
Business case leader	Stream Story	4	Model definition
technical dev	Stream Story	5	interpreting the node graph and its relationships
technical developer	Stream Story	4	need processing to meet the required input format. How to select attributes for the model needs to be playing around.

Have the tools been self-explanatory/well documented? [Wikifier]	Have the tools been self-explanatory/well documented? [Stream Story]	Have the tools been useful, will you gain some insights from using the tools? [Wikifier]	Have the tools been useful, will you gain some insights from using the tools? [Stream Story]
I understood the main functionalities of the tool and used several of them	I was not able to try the tool at all	useful for my profile	useful overall, but not for my profile
I could easily use the tool and tried all functionalities interesting for me	I was not able to try the tool at all	extremely useful and I could discover things I did not know	not useful at all
I could easily use the tool and tried all functionalities interesting for me	I was not able to try the tool at all	useful overall, but not for my profile	useful overall, but not for my profile
I could easily use the tool and tried all functionalities interesting for me	I was not able to try the tool at all	useful for my profile	not useful at all
I understood the main functionalities of the tool and used several of them	I was not able to try the tool at all	useful for my profile	useful overall, but not for my profile
I was not able to try the tool at all	I understood the main functionalities of the tool and used several of them	extremely useful and I could discover things I did not know	useful for my profile
I understood the main functionalities of the tool and used several of them	I understood the main functionalities of the tool and used several of them	useful for my profile	useful for my profile
I could easily use the tool and tried all functionalities interesting for me	I understood the main functionalities of the tool and used several of them	useful for my profile	useful overall, but not for my profile
I could easily use the tool and tried all functionalities interesting for me	I could easily use the tool and tried all functionalities interesting for me	useful for my profile	useful for my profile
I was not able to try the tool at all	I could easily use the tool and tried all functionalities interesting for me	not useful at all	useful for my profile
I was not able to try the tool at all	I understood the main functionalities of the tool and used several of them	partially useful	useful for my profile

Which visualizations/functionalities would you use in your work? [Wikifier]	Which visualizations/functionalities would you use in your work? [Stream Story]	What is the best feature of the specific tool that you have used? Please, mention the name of the
API;annotation functionalities	other	Wikifier, links to wikipedia concepts.
API;annotation functionalities;state history;explanation insights	other	The explainability of the results of the algorithm (wikifier)
API	API	Wikifier - not about feature but more for the API in order to integrate the tool
annotation functionalities	other	Wikifier: The feature that allows me to show more results or less according to the confidence
general interface;API;annotation functionalities	other	Wikifier list of candidates + support - I could somehow understand the rationale for the decision
other	general interface;state history;time component;explanation insights	Stream Story - the automatic analysis of the data I upload seems very useful. I also liked the ability to
general interface;API;annotation functionalities;state history;time component;explanation insights	general interface;API;annotation functionalities;state history;time component;explanation insights	API
general interface	general interface	Wikifier feature extraction
general interface	general interface;time component;explanation insights	Visual relations
other	API;annotation functionalities;state history;time component;explanation insights	streamstory
other	general interface;state history;explanation insights	streamstory: visualization

What is the key missing functionality from the specific tools that you have used? Please, mention the name of the tool.	Please check the boxes below Collected data
Wikifier, provide category of linked entity, to facilitate identification of a pre-set list of entities.	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
N/A	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
Wikifier - nothing specific	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
Wikifier: it would be interesting to understand better how the confidence scores are being computed	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
testing different thresholds on the decision	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
Stream Story - using multiple datasets together to create a single story	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
N.A.	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
Auto-preprocessing :)	I have read and understood the policy about to the processing of my personal data
Tutorial (theoretical guidelines) for model creation	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes
i Would add a label to make clear what the x and y axes represent in the model visualization graph	I have read and understood the policy about to the processing of my personal data;I consent to the processing and use of collected data for research purposes

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

8.6 Koala UI

1. Please select your profile	2. Have your expectations for entity linking ta	3. How challenging was it for you to use KOALA-UI?
Technical Developer, Researcher	4	3
Manager, Businesss Case Leader	4	3
Technical Developer	5	2
Researcher	5	1
Researcher	4	2
Technical Developer	4	2
Researcher	2	3
Technical Developer, Researcher	5	4
Technical Developer	5	2

4. Is the KOALA-UI self-explanatory and well documented?	5. Have the tools been useful? Will you gain some insights from using the tools?
Understood main functionalities	Useful for my profile
Understood main functionalities	Useful for my profile
Understood main functionalities	Useful for my profile
Understood main functionalities	Useful overall but not for my profile
Understood main functionalities	Useful for my profile
Used easily all functionalities	Useful overall but not for my profile
Understood main functionalities	Partially Useful
Understood main functionalities	Useful for my profile
Used easily all functionalities	Useful for my profile

6. Which visualizations/functionalities do you use in your work?	7. What is the best feature of KOALA-UI in your opinion?
General propriety interface, API, Annotation Functionalities, Customized Entity Linking Visualisation Dashboard	Intuitive user interface, easy to understand
API, Customized Entity Linking Visualisation Dashboard	Automatic feature detection
General propriety interface, API, Annotation Functionalities	Manually adding entities
API, Annotation Functionalities	visually appealing, functional
API	Automatic processing
General propriety interface, API	Colourful, easy to interact with the results and showing the linking progress
API, Annotation Functionalities, Customized Entity Linking Visualisation Dashboard	I like very much filters and sorting and the capability to handle large tables
General propriety interface, API, Annotation Functionalities, Customized Entity Linking Visualisation Dashboard	Entity-linking and visualisation for large tabular data.
API, Annotation Functionalities, Customized Entity Linking Visualisation Dashboard	Good UI

8. What is the key missing functionality from KOALA-UI in your opinion?	Please check the boxes below
Ability to extend with data from the reconciliation knowledge graph	I consent to the processing and use of collected data for research purposes
Stop bottom, process may take long. Easy access to support information, like short pop-up info.	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
adding entity by url	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
n/a	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
N.A.	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
Stop button for linking process. In the case of wrong prediction, list of 5 sometimes still doesn't contain the correct answer.	I have read and understood the policy about to the processing of my personal data
I think all main functionalities are there. My impression is that it's requires some conceptual polishing: - I would decouple the upload of a table (annotated or not) and the execution of linking. - I would stress the features for revision of large annotated tables first; which means you should be able to upload a table that has been annotated outside the app - Then do the revision; - I would use two different orthogonal classifications for NE/LIT/UNDEFINED and IGNORED.	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
Latency issues	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes
User interaction could be improved	I have read and understood the policy about to the processing of my personal data, I consent to the processing and use of collected data for research purposes

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