



*Research Lifecycle Management technologies for  
Earth Science Communities and Copernicus users in EOSC*

## Deliverable D3.2v1

### Design, implementation and deployment of text mining and enrichment services Phase 1

Grant agreement number	101017501
Start date of the project	Reliance
Duration of the project	24 months
Type of Action	Research and Innovation action
Coordinator	PSNC

Due date of delivery	31/12/2021
Actual date of delivery	07/01/2022
Work package	WP3
Type of deliverable	Report
Dissemination level	Pub
Responsible	Expert.ai
Reviewer	Oscar Corcho (UPM)
Version	1.0



This project has received funding from the European research infrastructures (including e-Infrastructures) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017501

---

## List of authors, contributors and reviewers

Name	Role	Organization
Raul Ortega	Author	Expert.ai
Andres Garcia-Silva	Author	Expert.ai
Jose Manuel Gomez-Perez	Author	Expert.ai

---

## History of changes

Version	Date	Change	Authors	Organization
0.1	11.11.2021	Table of Content	Raul Ortega	Expert.ai
0.2	15.12.2021	Executive Summary	Andres Garcia-Silva	Expert.ai
0.2.1	16.12.2021	Introduction	Andres Garcia-Silva	Expert.ai
0.2.3	20.12.2021	Sections 2 and 3	Raul Ortega	Expert.ai
0.2.4	21.12.2021	Section 4	Raul Ortega	Expert.ai
0.2.5	22.12.2021	Section 5	Raul Ortega	Expert.ai
0.2.6	23.12.2021	Document revision	Andres Garcia-Silva	Expert.ai
0.3	24.12.2021	Conclusions	Andres Garcia-Silva	Expert.ai

## Glossary

Acronym	Explanation
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
FoR	Field Of Research
NLP	Natural Language Processing
NL API	Natural Language API
LM	Language Model
OpenAIRE	Open Access Infrastructure for Research in Europe
REST	Representational State Transfer
ROHub	Research Object Hub
SQUAD	Stanford Question Answering Dataset

## Table of Contents

<b>1</b>	<b>Executive Summary .....</b>	<b>7</b>
<b>2</b>	<b>Introduction.....</b>	<b>8</b>
2.1	Scope.....	9
2.2	Audience .....	9
2.3	Structure .....	9
<b>3</b>	<b>Semantic annotations and information extraction: Enrichment API .....</b>	<b>9</b>
3.1	Document Enrichment .....	10
3.2	Research Object Enrichment.....	12
3.3	Future add-ons: structured information from publications.....	13
<b>4</b>	<b>Enhanced content-based retrieval: Search and Recommendation API.....</b>	<b>14</b>
4.1	Search API.....	14
4.2	Research Object Dashboard.....	15
4.3	Recommendation API .....	15
4.4	Collaboration Spheres.....	17
<b>5</b>	<b>Extended analytics services in support of the scientific enterprise .....</b>	<b>20</b>
5.1	Influence Networks .....	20
5.2	Novelty Score .....	20
5.3	Reading Comprehension .....	21
5.4	Enrichment Dashboard .....	21
<b>6</b>	<b>Conclusions and future work .....</b>	<b>21</b>
<b>7</b>	<b>References.....</b>	<b>22</b>
	<b>Appendix.....</b>	<b>22</b>
A.	Document Enrichment API documentation .....	22
B.	Solr query examples and documentation .....	23
B.1	Single term search.....	23
B.2	Facet search .....	25
C.	Recommendation API documentation.....	27

---

## List of Figures

Figure 1 Landing page for the Text Mining and Enrichment services in Reliance.....	8
Figure 2. Document Enrichment process. ....	11
Figure 3. Research Object Enrichment .....	13
Figure 4. Content-based recommendation process .....	16
Figure 5. General screenshot of the Collaboration Spheres .....	17
Figure 6. Screenshot of the Navigation panel of the Collaboration Spheres .....	18
Figure 7. Screenshot of a Summary Card of the Collaboration Spheres .....	18
Figure 8. Screenshot of the spheres panel of the Collaboration Spheres. ....	19
Figure 9. Screenshot of the login and tutorial panel of the Collaboration Spheres.....	20

## 1 Executive Summary

This deliverable reports the progress in the design and development of the first version of the text mining and enrichment services for Research Objects and scientific documents in RELIANCE. In the early stage of the project, we gathered a corpus of scientific documents in the domains of interest for RELIANCE user communities that was used to customize the knowledge graph underlying the text mining and enrichment services. Once the knowledge graph was adapted to the scientific vocabulary used by RELIANCE users, we were ready to start the development of the text mining and enrichment services.

Among such services is the semantic enrichment service of Research Objects and Scientific documents. The enrichment service processes the text from documents or Research objects and yields semantic metadata describing the text content. The semantic metadata is a synthesis of the text and includes concepts, lemmas, multi-word expressions, and topics. In addition, we are working on the generation of a Field of Research metadata that we plan to integrate in the enrichment service. To generate this type of metadata, we trained a classifier by fine-tuning a pre-trained RoBERTa language model on the corpus that we gathered for RELIANCE users where scientific papers are tagged with Field of Research categories (e.g., Geology, Oceanography, or Atmospheric Sciences). The enrichment service for documents is already onboarded in EOSC as a RESTful service, and the service that enriches Research Objects is integrated in ROHub.

Next, we developed information retrieval tools exploiting the semantic metadata added to Research Objects along with their text content. First, a faceted search engine where users can search the Research Objects collection in ROHub using the traditional keywords, but also facets for each of the types of the semantic metadata. A faceted search engine allows complex queries and retrieving Research Objects more precisely. Second, a recommendation engine that suggests potentially relevant research objects based on the content of a selection of other research objects or users. If the reference is a research object, the recommendation engine uses the semantic metadata of the research object to recommend similar research objects. If the reference is a user, the recommendation engine first aggregates all the semantic metadata of the research objects owned by such user and then uses the semantic metadata to suggest relevant research objects. The search engine and the recommendation system are being onboarded in EOSC as RESTful services.

During this stage of the project, we lay the foundations for the development in the next phase of the extended set of analytics services where we plan to address the Influence Network Extraction, the Novelty Score for Research Objects, Support to Reading Comprehension, and Text Mining and Enrichment Dashboard. While the design and development of such services will be covered in depth in the second version of this deliverable, in this version we have included a brief description of each of them. In the second version of this deliverable, we plan to increase the metadata produced by the enrichment service including titles, abstracts, authors, citations, and data cube references.

Finally, in the second phase of the project we will work on the user interfaces for the recommendation system, the search engine, the visualization of the semantic enrichment results for research objects, and a prototype of a browser plugin that allows scientists to ingest scientific publications into their bibliographic research objects as they browse the internet searching for relevant literature for their research.

## 2 Introduction

Research Objects aggregate heterogeneous resources associated with a particular research activity and metadata relevant to understand and interpret their content using semantic annotations that are user and machine readable. However, most of the semantic annotations in a research object describe its structure and the type of resources it aggregates, and few metadata is concerned with the actual content of the research object (Gomez-Perez, Palma, & Garcia-Silva, 2017). Without metadata about the content the vision of automatic or at least assisted processing of scientific information using research objects is unfeasible. The lack of content annotations limits research objects potential outreach and diffusion of scientific outcomes, ultimately hampering their reuse by other researchers. Resources in Research Objects are multimodal containing a mix of text, data, images and code, and the automatic processing of multimodal information is still an open research problem. In this document we describe the design and development of text mining services to address the lack of content annotations in research objects, and information retrieval tools to boost the research object findability. For dissemination and documentation purposes we have designed a web page<sup>1</sup> describing the text mining and enrichment services that we are developing in RELIANCE (see Figure 1).

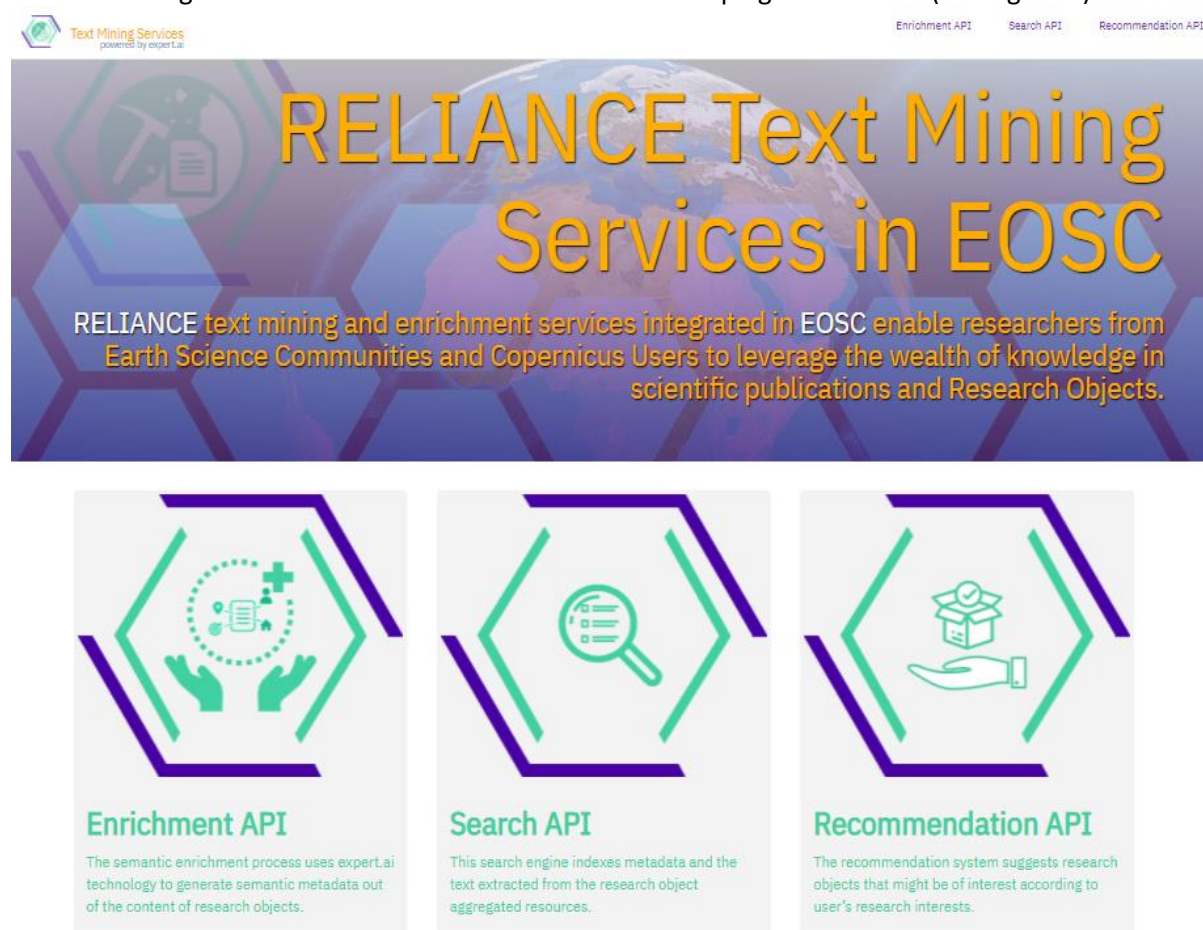


Figure 1 Landing page for the Text Mining and Enrichment services in Reliance

We start by describing the semantic enrichment service that processes the research object and generates metadata that synthetizes the textual content of the resources it aggregates. This service can be used also to process single documents including plain text, PDF, Word, and PowerPoint files. Thanks to the analysis of the corpus of scientific texts relevant to RELIANCE users that was gathered at the early stage of the project, the enrichment service has been customized by integrating such

<sup>1</sup> <https://reliance.expertcustomers.ai/>



specialized vocabulary in its underlying knowledge graph. The enrichment service has been integrated in ROHub so that public Research Objects are enriched automatically, and the version that enriches single documents has been onboarded on EOSC as RESTful service<sup>2</sup>.

Next, we focus on the information retrieval tools that we have developed for Research Objects. First, we describe the search engine that tap into the semantic annotations generated by the enrichment service to support faceted search of the research object collection in ROHub. Facets support complex queries and enable more accurate searches. Second, we describe the recommendation system that suggests potentially relevant research objects in ROHub. This recommendation system also leverages the semantic annotations to generate the recommendations based on the content of the research objects. We have implemented RESTful services for the search engine and recommendation system. Both services are being onboarded in EOSC<sup>3</sup>. We are currently working on the Collaboration Spheres web application where users can interact with the recommender system using a drag and drop interface to explore the ROHUB research object collection.

Finally, we briefly present design considerations of the group of extended set of analytics services developed in task 3.4. In the second year of the project, we plan to develop fully this set of services.

## **2.1 Scope**

This deliverable covers the description of the status of the Enrichment, Search and Recommendation services. We also present the next steps to improve these services and a first introduction of the experimental analytics that will be developed in further phases of the project.

## **2.2 Audience**

This document is intended for the RELIANCE Consortium and related scientific communities. In addition, researchers interested in the use of the RELIANCE text mining services find valuable information about such services in this document.

## **2.3 Structure**

The rest of the document is structured as follows. The enrichment service is described in Section 3. The Search and Recommendation services are presented in Section 4. We present some design considerations about the extended analytics services, that will be addressed in the second year of the project, in section 5. Finally in Section 6 we present the conclusions and future work.

# **3 Semantic annotations and information extraction: Enrichment API**

Without annotations about the content of Research Objects, information retrieval tools are limited to work only with the text that can be extracted from the aggregated resources and metadata. Text is unstructured data that is hard to deal with by machines. Conventional search engines, such as Solr<sup>4</sup> and Elastic Search<sup>5</sup>, use inverted indexes that allow searching the document collection using keywords (Baeza-Yates & Ribeiro-Neto, 2011). Drawbacks of keyword-based search include low recall, since synonyms of the keyword are not considered in the search, and low accuracy when ambiguous keywords are used to retrieve documents without considering the user intended meaning of the keyword.

Recently, information retrieval systems are using natural language processing (NLP) techniques to improve the search experience. While there have been impressive advances in the NLP field using

---

<sup>2</sup> <https://marketplace.eosc-portal.eu/services/enrichment-api>

<sup>3</sup> The status in EOSC onboarding process of the Search service and the Recommendation service is Approved.

<sup>4</sup> <https://solr.apache.org/>

<sup>5</sup> <https://www.elastic.co/elasticsearch/>

neural networks (Devlin, Chang, Lee, & Toutanova, 2019) in information retrieval systems (Khattab & Zaharia, 2020), these systems still act as black boxes which are difficult to interpret and offer no explanations for their predictions. In RELIANCE we opted for a knowledge-based approach to NLP where the system predictions are backed up by a knowledge graph where linguistic information is explicitly modeled. We use NLP to process the text that is available in Research Objects and generate metadata that summarizes the content in the Research Object. Such metadata can be visualized by users and injected in search engines and recommendation systems.

In RELIANCE we design and develop an Enrichment service that generates semantic annotations from documents and Research Objects. The Enrichment API comprises two modules: the Document Enrichment service and the Research Object Enrichment service. The former extracts semantic annotations from documents such as PDF, Word, PowerPoint, and plain text files. The latter processes the files in a Research Object and generates a set of annotations as a summary of the content. In the following we describe in the detail such services.

### 3.1 Document Enrichment

The Document Enrichment uses expert.ai Natural Language API<sup>6</sup> (NL API) to process the text extracted from documents. The NL API is a cloud-based software service providing a comprehensive set of natural language understanding capabilities. These capabilities include linguistic tasks such as part-of-speech, morphological analysis, lemmatizations, syntactic analysis, and semantic analysis. In addition, the NL API can perform sentiment analysis, text classification and the extraction of key phrases, named entities, and relations.

The NL API relies on the **Sensigrafo** knowledge graph that encodes general-purpose linguistic knowledge. In Sensigrafo each node is a group of words sharing a meaning, i.e., a concept, and these nodes are connected among them by linguistic relationships such as meronymy, hypernymy, and hyponymy to name just a few. Each node in the graph contains a set of attributes, such as grammar type, definition, domain, and frequency, that define the characteristics of words and concepts. The NL API uses the Sensigrafo to disambiguate the meaning of words considering the context. Disambiguated words are assigned a node in the Sensigrafo.

As part of the work in T3.1 (see RELIANCE deliverable D3.1), we customized the Sensigrafo knowledge graph by adapting it to domains of interests for RELIANCE user communities. The custom Sensigrafo extends the general-purpose Sensigrafo with new domain-specific terms obtained from a text corpus of scientific papers.

The Document Enrichment service is a RESTful API which receives a file with text as an input and generates relevant annotations following the workflow described in Figure 2. The service only accepts plain text files, Word documents, PDF files, or Power Point files. In further phases of the project, we will assess including more extensions such as Jupyter notebooks, readme files or data cubes text content. When a request for the enrichment of a document is issued, the service extracts the raw text from the document, using Apache PDFBox<sup>7</sup> in the case of PDF files and Apache POI<sup>8</sup> in the case of Word and Power Point files. Then, the enrichment service makes a request to the NL API which returns the semantic metadata generated from the text. Each metadata piece is associated with a score indicating the importance of that entity over the entire text. The metadata generated by the enrichment service is the following:

- **Domains:** Fields of knowledge mentioned in the document based on its main concepts.

---

<sup>6</sup> <http://expert.ai>

<sup>7</sup> <https://pdfbox.apache.org/>

<sup>8</sup> <https://poi.apache.org/>

- **Concepts:** Most frequent concepts mentioned in the text which are modeled in the Sensigrafo. We plan to link each Sensigrafo concept with the corresponding Wordnet Synset (Miller, 1995).
- **Expressions:** Most relevant phrases and multiword expressions found in the text.
- **People:** People names or aliases found in the text.
- **Places:** Places names or aliases found in the text.
- **Organizations:** Organization names or aliases found in the text.

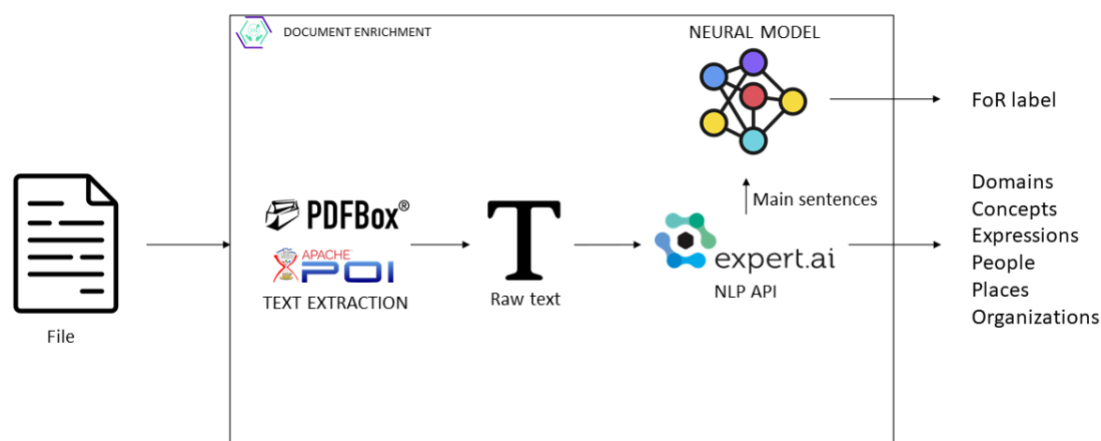


Figure 2. Document Enrichment process.

In addition to the semantic metadata from NL API, we are currently working on integrating in the Document Enrichment a module to generate metadata about the fields of research identified in the document. To generate such metadata, we leverage the RELIANCE text corpus that we gathered in T3.1 from SciGraph<sup>9</sup>. In SciGraph each scientific publication is annotated with a category in the **Fields of Research (FoR) taxonomy**<sup>10</sup>. Such annotations were extracted and integrated in the RELIANCE text corpus<sup>11</sup>.

We train a classifier using the FoR annotations in the RELIANCE text corpus. Our dataset contains 209319 publications described by title and abstract. We split the dataset to use 80% of the data for training and 20% for testing. We use a pretrained transformer (Vaswani, et al., 2017), fine-tuned on the task of categorizing the articles in the FoR categories. We train a RoBERTa large model (Liu, et al., 2019) for 4 epochs using a batch size of 2 and a learning rate of 1e05, and cross entropy as loss function. Neural Language models like RoBERTa are limited to input text of less than 512 tokens. Since the raw texts from the documents may be larger than that, we use the NL API to get the main sentences in the text to feed the model with a text that fit in the 512 tokens limit.

In Table 1 we present the evaluation results of a preliminary classifier that we have trained in a multiclass classification task where each document is assigned a Field of Research. The evaluation metrics are the standard metrics including Precision, Recall and F1. The weighted average of the three metrics of 0.97 is very promising, indicating that it is possible to assign Field of Research categories to documents. Nevertheless, to make this classifier useful we need to turn it into a multilabel classifier where each document is assigned one or more Fields of Research. We plan to train such classifier in

<sup>9</sup> SciGraph is Springer Nature knowledge graph of scientific publications: <https://scigraph.springernature.com/>

<sup>10</sup> Fields of Research taxonomy is available at: <https://www.arc.gov.au/grants/grant-application/classification-codes-rfcd-seo-and-anzsis-codes>

<sup>11</sup> <https://zenodo.org/record/4721343#.Yc2msWjMJPY>

the next project phase. We also plan to release the classification model and data used to train it in Zenodo.

Table 1. Evaluation Results of the Field of Research Classifier and train and test splits distribution

Field Of Research	Precision	Recall	F1-Score	Train set	Test set
Geochemistry	0.81	0.84	0.77	8587	2188
Geology	0.91	0.91	0.91	53221	13219
Oceanography	0.84	0.87	0.85	15493	3900
Atmospheric Sciences	0.87	0.90	0.88	19572	4785
Geophysics	0.87	0.89	0.88	10212	2615
Physical geography and environmental geophysics	0.86	0.86	0.86	21423	5383
Other Earth Sciences	0.84	0.73	0.78	948	243
<b>Earth Sciences</b>	<b>0.98</b>	<b>0.98</b>	<b>0.98</b>	<b>129506</b>	<b>32333</b>
Environmental science and management	0.89	0.85	0.87	17832	4307
Ecological Applications	0.63	0.45	0.53	725	184
Soil Sciences	0.93	0.96	0.94	19392	5040
<b>Environmental Sciences</b>	<b>0.93</b>	<b>0.92</b>	<b>0.93</b>	<b>37949</b>	<b>9531</b>
<b>Total</b>	<b>0.97</b>	<b>0.97</b>	<b>0.97</b>	<b>167455</b>	<b>41864</b>

The enrichment service generates the metadata in json format by default. However other serializations such as turtle are supported. The Document Enrichment service is available in the following url: <https://reliance.expertcustomers.ai/eosc/enrichment>. A service request needs to be issued following the Document Enrichment API documentation section A of the Appendix.

### 3.2 Research Object Enrichment

The Research Object Enrichment service, depicted Figure 3, receives the Research Object id (w3id) as input argument, and returns the semantic metadata in turtle format. At high level view, the service first extracts the resources in the Research Object. Then the resources are filtered considering the resource type and sent to the Document enrichment service to get the semantic annotations. Since each resource is described with a set of annotations we need to aggregate them to generate a unique set of annotations representative for the Research Object.

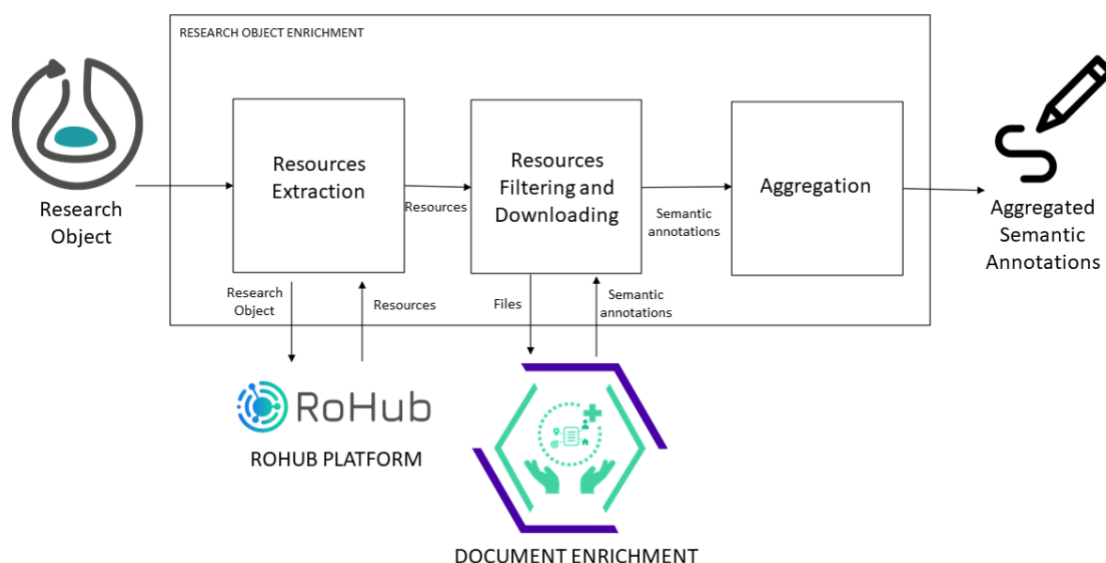


Figure 3. Research Object Enrichment

To extract the Research Object resources we use the ROHub API<sup>12</sup>. First, we query the ROHub API using the research object id for the resources in that research object. Then, we filter the resources returned by the platform keeping resources of the following resource types:

- Document
- BibliographicResource
- Conclusions
- Hypothesis
- ResearchQuestion
- ResultsPresentation
- Paper
- File

Next, the service downloads the resources that have passed the previous filter, and checks whether the associated file type is accepted by the Document Enrichment service (Word documents, Power Point files, PDF files or plain text files). Once the curated list of resources is ready, and their associated files are downloaded, the service issues requests to the Document Enrichment service to get the metadata. As a result, we have a list of annotations per each resource in the Research Object. To generate the final set of annotations for the Research Object we first create a list per each annotation type, i.e., Lemma, Concept, Expression, People, Location, and Organization, where we add the annotations, and corresponding scores, of each resource. When adding an annotation, if it is already in the list we keep only one annotation and add up the scores of the existing annotation and the new one. Then we sort the annotation lists in descending order according to the score and select the top k annotations<sup>13</sup> of each type as representative of the Research Object.

### 3.3 Future add-ons: structured information from publications

In the next phases of the project, we aim at increasing the metadata that we can extract from documents. We study using heuristics and neural models to extract information such as the title, the

<sup>12</sup> <https://api.rohub.org/api/>

<sup>13</sup> The Research Object Enrichment service returns at most 20 annotations per annotation type.

abstract or the citations of an article or a research document. We are considering using PDFFigures2<sup>14</sup> to extract also visual metadata from this type of resources, such as figures and tables. In addition, we plan to work on the extraction of references to Data Cubes or to software repositories. An interesting line of work that we are considering is to use frameworks to annotate source code encapsulated or reference in Research objects and add such annotations in the Enrichment process.

## 4 Enhanced content-based retrieval: Search and Recommendation API

Semantic annotations can be exploited to improve the findability of specific Research Objects within a large repository. Moreover, this metadata can help to find relations between documents which otherwise would remain hidden. The Search and Recommendation APIs use the annotations generated by the Enrichment API to enhance the retrieval of relevant Research Objects as we see in detail in the following.

### 4.1 Search API

Standard search technology based on user-generated metadata and keywords has limitations regarding the search results and the complexity of user searches. When users search using ambiguous keywords, search engines just retrieve documents containing the keywords disregarding their meaning, thus retrieving documents potentially not relevant for the user. In addition, keyword-based search engines miss documents that contain synonyms of the keywords and morphological variations such as verb conjugations or even plurals, hence producing incomplete results<sup>15</sup>. By leveraging the semantic metadata generated by the Research Object Enrichment Service, a Search Engine can deal effectively with ambiguous words and synonyms. Each concept identifies uniquely the sense in which words are used, and contains the synonyms used to refer to them. Lemmas are the canonical form of words and thus can be used to tackle morphological variations.

The Search API relies on a document index<sup>16</sup> that supports efficient indexing and retrieval of documents. Every time a Research Object is created in ROHub, a request to the Enrichment Service is issued to generate the semantic metadata, that is then stored in the document index. To manage the content and support search on the index we currently use the Solr<sup>17</sup> platform. The combination of the document index and Solr enables faceted queries that can untap the potential of the annotations, easing the exploration of Research Objects based on their content.

To populate the index, each Research Object is represented as a Solr document. A document in Solr must be uniquely identified so we use the Research Object Id. The user-generated metadata from Research Objects hosted in ROHub, along with the metadata generated from the Enrichment API, are used as the document fields. Table 2 shows the mapping between the annotations and the fields of the Solr schema. All the fields can be used to retrieve matching documents. They are all retrievable in a query result and the ones which are associated with a semantic annotation can adopt multiple values. The 'Content' field aggregates all the other fields including the document id and it is the primary search source when no other field names are specified in the query.

---

<sup>14</sup> <http://pdffigures2.allenai.org/>

<sup>15</sup> Some Search Engines supports synonyms by allowing users to enter them by hand in special files. See for example the Solr Synonym Filter at: [https://solr.apache.org/guide/6\\_6/filter-descriptions.html#FilterDescriptions-SynonymFilter](https://solr.apache.org/guide/6_6/filter-descriptions.html#FilterDescriptions-SynonymFilter)

<sup>16</sup> We use a Lucene index: <https://lucene.apache.org/>

<sup>17</sup> <https://solr.apache.org/>

Table 2. Partial view of the field structure of the Solr Index, with their mapping to the semantic annotations from the Enrichment API

Field	Type	Semantic annotation	Multivalued	Indexed	Stored
id	string			X	X
title	text_general			X	X
autocomplete	string	All	X	X	X
description	text_general			X	X
creator	text_general			X	X
author	string			X	X
source_ro	string			X	X
sketch	string			X	X
domains	string	Domains	X	X	X
concepts	string	Concepts	X	X	X
compound_terms	string	Expressions	X	X	X
people	string	People	X	X	X
place	string	Places	X	X	X
organization	string	Organizations	X	X	X
content	text_general			X	X
_version_	long			X	

The Search service is available in the following url: <https://reliance.expertcustomers.ai/solr/ROHub/>. The service supports standard Solr queries. In section B of the Appendix we present the documentation of the search service and show some examples of basic and faceted queries, along with the response from Solr. All the requests must be issued with basic credentials, which are documented also in Solr query examples and documentation.

## 4.2 Research Object Dashboard

For the next project phase, we plan to develop a Research Object dashboard where users can search and explore public Research Objects in ROHub, using an interactive user interface where metadata added by the enrichment service is visualized and at the same time used to drill down and filter the research objects. The dashboard is a web application full of widgets supporting the visualization of time-series, histograms, bar and pie charts, tag clouds, tables, and documents, to name some of the supported widgets. The dashboard also includes a query box where users can pose queries against the RELIANCE Search engine using either single term queries or faceted queries.

To implement the dashboard, we have evaluated two open-source technologies: Banana<sup>18</sup> and Kibana<sup>19</sup>. Banana is integrated with Solr, the base platform of the Search Engine, while Kibana works with an Elastic search engine. Nevertheless, Banana is based on Kibana version 3.0, an old version given that Kibana latest version is 8.0. In addition, the Kibana repository in GitHub shows an active developer community while the latest update in Banana repository is from 2019. Thus, we think that the best option is to use Kibana even though this decision implies to migrate our current Search platform from Solr to Elastic Search. Both search platform Solr and Elastic Search support faceted search so the user functionality would not suffer any change.

## 4.3 Recommendation API

While a search engine is used when the user knows exactly what to look for, a recommendation system is the right tool to explore the information in large repositories when the user only has some

<sup>18</sup> <https://github.com/LucidWorks/banana/>

<sup>19</sup> <https://github.com/elastic/kibana>



clues about the content of interest. The goal of the recommendation API is to suggest Research Objects that can be of interest to researchers, based on their associated Research Objects and their collaborations. Thus, the social dimension of the researchers plays a crucial role in this service. Along with the content-based annotations from the Enrichment API and the enhanced search engine from the Search API, the social network that emerges from research collaborations on different Research Objects, add value to the recommendation of Research Objects of interest to the user. Unlike a simple content-based approach, the Recommendation API puts the user at the core of the suggestion, using her network as part of the features that the system uses to recommend new Research Objects.

The Recommendation service suggests a list of Research Objects related to a set of Research Objects and/or Users that defines the recommendation context. The service relies on the document Index built for the Search API. Thus, it only accepts Research Objects that are enriched and indexed, and users who owns at least one indexed research object. The API returns a list of Research Objects, sorted by relevance to the recommendation context and the user profile.

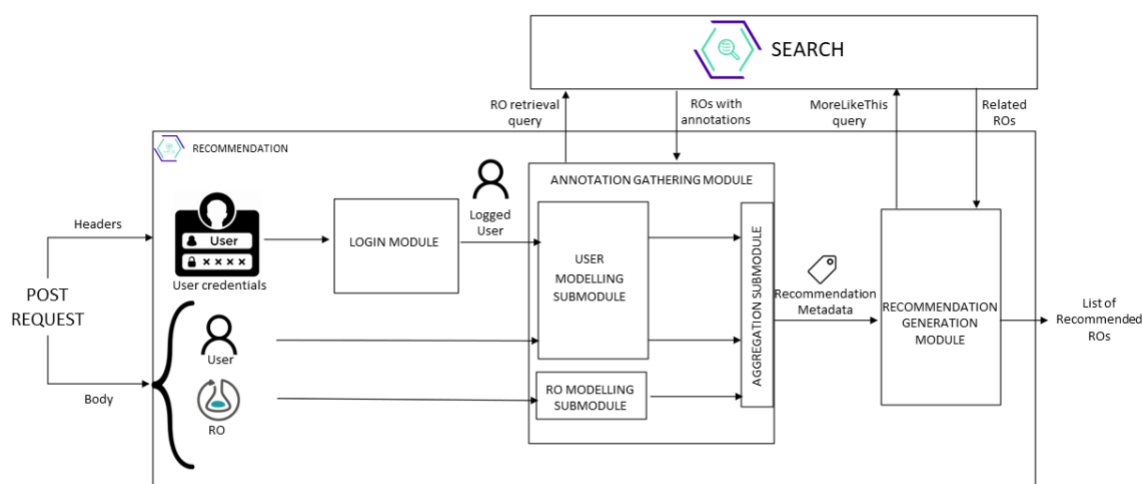


Figure 4. Content-based recommendation process

Figure 4 describes the recommendation service. Once a user has logged in through basic authentication, the Recommendation API initializes the user modelling module. This module uses the Search API to look for Research Objects owned by the user within the document Index and aggregates the semantic metadata of each one of them in one single object that we call Logged User metadata. Next, the service checks if the entities in the recommendation context that are specified in the body of the request are either Research Objects or users, otherwise it returns an error. In the case of recommendation contexts containing users, the service initializes again the user modelling module, to extract the annotations from research objects owned by such users. Once we have gathered the annotations for the entities in the recommendation context, the Aggregation module generates the Recommendation metadata, using a mix of the annotations of the recommendation context and the Logged User metadata. Then, such Recommendation metadata is used to send a query to the Search API, which returns similar Research Objects. Lastly, the Recommendation API will return the selected list of Research Objects, sorted by their similarity value, in json format. This list represents the custom recommendation within the user context for that query.

The Recommendation service generates a list of ROs URIs in JSON format, and it is available in: <https://reliance.expertcustomers.ai/spheresbackend/services/jsonservices/api/>. The service documentation is available in section C of the Appendix.



#### 4.4 Collaboration Spheres

### Research Objects

search a Research Object...

My Research Objects	Collaborators
Collaborations	Related
All Research Objects	All Scientists
Factors determining the diffusion of ...	gorgio.castellan@bo.ismar.cnr.it
SIOS's Earth Observation (EO). Remote...	annefou@geo.uio.no
Amplified ozone pollution in cities	service-account-generation-service
The effects of COVID-19 induced lockd...	annefou@geo.uio.no
COVID-19 lockdown measures reveal hum...	strahma@man.poznan.pl
Nitrogen dioxide reductions from sat...	dario.schellano@ingv.it
Analysis of Air Quality during the CO...	federica.foglia@ismar.cnr.it

More information:

**Impact of the Covid-19 Lockdown on Air quality over Europe**

on: 2021-12-19 21:18:33 2318942 by: annefou@geo.uio.no

**Main topics:** lockdown, March, data

**Area of knowledge:** ecology, medicine, meteorology

**Description:** The COVID-19 pandemic has led to significant reductions in economic activity, especially during lockdowns. Several studies has shown that the concentration of nitrogen dioxide and particulate matter levels have reduced during lockdown events. Reductions in transportation sector emissions are most likely largely responsible for the NO2 anomalies. In this study, we analyze the impact of lockdown events on the air quality using data from Copernicus Atmosphere Monitoring Service over Europe and at selected locations.

### Scientists

search a scientist/specialist...


Figure 5 shows a screenshot of the Collaboration Spheres web application. The user interface presents a navigation panel at the left, and the Collaboration Spheres at the right. The left panel introduces the list of Research Objects and Scientists that can be included in the recommendation context, and a summary card which shows information about Research Objects or users selected in the navigation panel or the collaboration spheres. The entities on the left panel can be dragged to the Collaboration Spheres centre, forming the context to drive the recommender process. Then, the application displays the list of recommended research objects all over its concentric circles. Also, in the upper right part of the screen, there is a tutorial and the login panel.



Figure 6. Screenshot of the Navigation panel of the Collaboration Spheres

The navigation panel depicted in Figure 6 is split in two columns. The one on the left is for enriched Research Objects, and the other on the right is for users that own at least one of those enriched Research Objects. Any entity listed on this panel is draggable into the Spheres and clicking on them displays the related information down below on the summary card. Both columns have a search bar which facilitates to find specific Research Objects or Scientist using its title or username. Moreover, clicking on “All Research Objects” or “All Scientists” displays the entire available lists for both categories, filtered following the terms of the search bar if used.

#### More information:



### Impact of the Covid-19 Lockdown on Air quality over Europe

on: 2021-12-19 21:18:33.231894Z by: [annefou@geo.uio.no](mailto:annefou@geo.uio.no)

**Main topics:** lockdown, March, data

**Areas of knowledge:** ecology, medicine, meteorology

**Description:** The COVID-19 pandemic has led to significant reductions in economic activity, especially during lockdowns. Several studies has shown that the concentration of nitrogen dioxide and particulate matter levels have reduced during lockdown events. Reductions in transportation sector emissions are most likely largely responsible for the NO2 anomalies. In this study, we analyze the impact of lockdown events on the air quality using data from Copernicus Atmosphere Monitoring Service over Europe and at selected locations.

Figure 7. Screenshot of a Summary Card of the Collaboration Spheres

The summary card depicted in Figure 7 presents a preview of the information regarding a Research Object or a user. It is available for any Research Object or user, both in the navigation panel or the Spheres, just by clicking on them. It shows the title or the username, and three main topics and areas

of knowledge of a Research Object or User workspace. Besides, in the case of Research Objects, the summary information includes their description, creation date, and sketch image, if it is available.

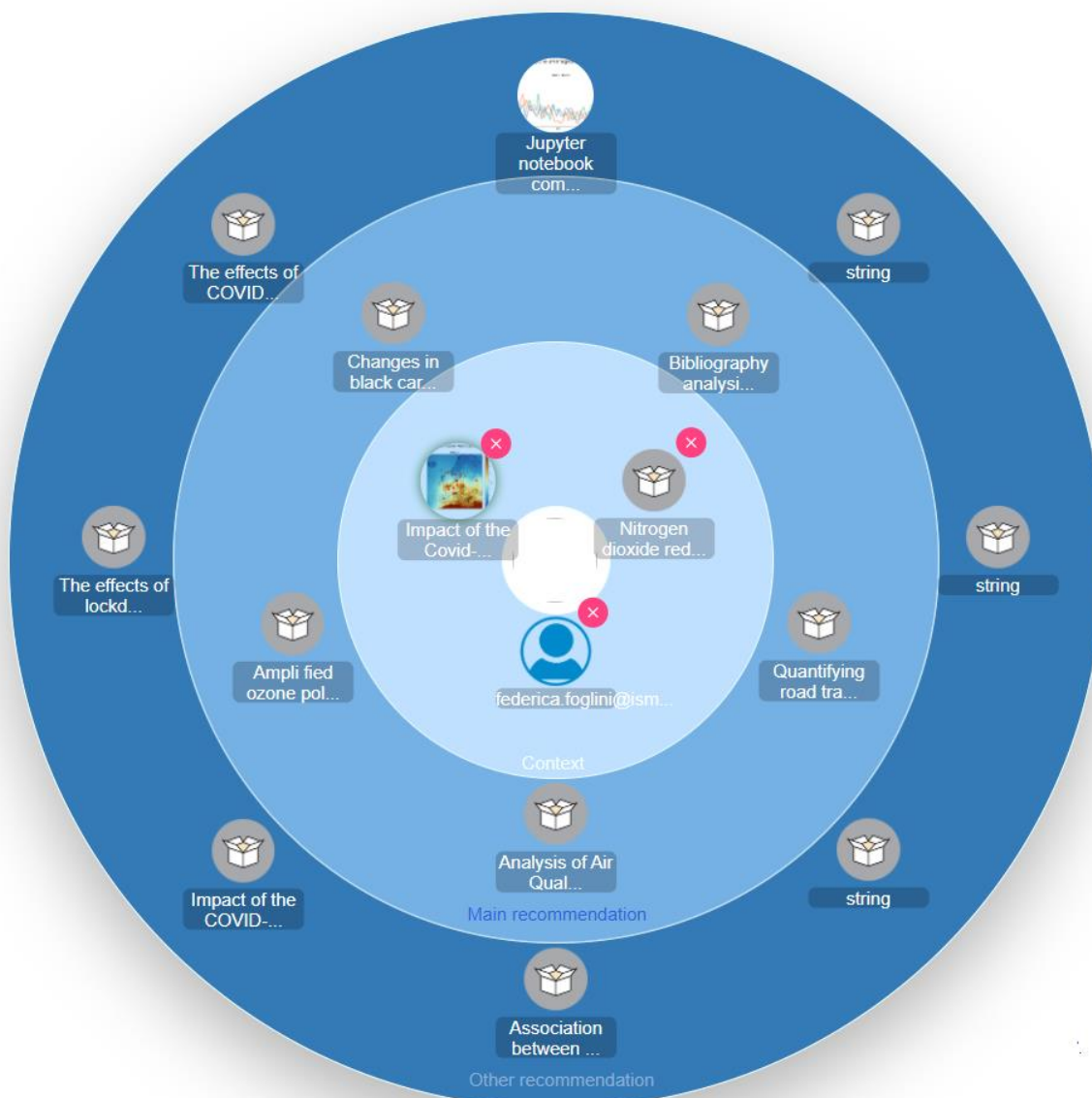


Figure 8. Screenshot of the spheres panel of the Collaboration Spheres.

The spheres panel in Figure 8 is the main component of the web application since it defines the recommendation context and displays the recommended items. It consists in four co-centric spheres. The two smallest are used to define the recommendation context which serves as the query, and the two outer spheres are placeholders for the recommendation results. The suggested Research Objects are placed considering their importance for the recommendation: the closer they are located to the central sphere, the more relevant are for the query.

The user can drag up to three research objects or scientists from the navigation panel and drop them into the second sphere to define the context of the recommendation that will be used to call to the Recommendation API. After the recommendation has been defined the system displays the recommendation.

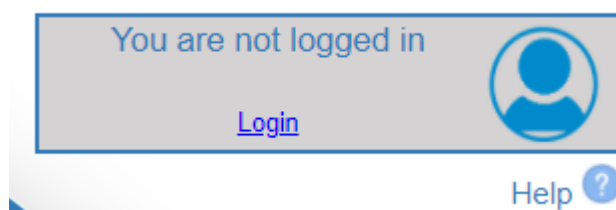


Figure 9. Screenshot of the login and tutorial panel of the Collaboration Spheres.

As mentioned in the Recommendation API section, the integration with the new authentication service of ROHub in RELIANCE is work in progress. Thus, the grey tabs in Figure 6 denote that those sections aren't available yet. However, once the authentication integration is ready, they shall show the Research Objects owned by the logged user, the collaborations with other colleagues and other related Research Objects. Furthermore, if a user is logged in the application, the most inner circle of the Collaboration Spheres will show it as the centre of the recommendation.

Figure 9 shows how the login panel currently looks like. When the ROHub authentication is integrated in the Collaboration Spheres, clicking on Login will send the user to the ROHub Login portal, and once logged in, the user will return to the full version of the Collaboration Spheres. Furthermore, in Figure 9 there is a "Help" button, which display a brief tutorial to facilitate the use of the application for beginners.

## 5 Extended analytics services in support of the scientific enterprise

In this section we briefly describe the set of extended analytics services, including some design considerations, that will be developed on the second year of the project. The services in this category are experimental since the underlying tasks and techniques are still subject of research activities.

### 5.1 Influence Networks

In addition to the Recommendation service, we plan to develop a tool to help users exploring the research object collection that visualizes the influence network of an author. The influence network is a directed graph where the analyzed author X is modeled as a node in the center, with input links from nodes representing other authors who have influenced author X, and output links to nodes representing other authors that have been influenced by the author X. This influence network can be used then to explore the research objects and scientific papers of authors in the influence graph. Each author in this graph is marked with a score indicating the strength of the influence. The influence score can be measured in terms of co-authored research objects and citations between research papers, but also by considering for example the research objects or publications working on the same datasets or data cubes.

To build the influence network we need to extract from scientific publications metadata such as Title, Author, Abstract, and Citations, Datasets, and Datacubes. Thus, in the next phase of the project we will enhance the Document Enrichment service to extract such metadata. We also plan to explore an experimental line of work to extract scientific claims from publications and use such claims to identify more influence relations between authors.

### 5.2 Novelty Score

Our goal is to add novelty score as metadata to Research Objects that indicates how novel it is considering existing research objects in ROHub, and scientific papers in OpenAIRE. We think that the novelty score is a useful indicator for example to rank the Search engine results or the suggested Research Objects by the Recommendation system. To calculate the novelty score we plan to use

semantic similarity between Research Objects and Scientific publications. Semantic similarity attempts to compare two different texts by comparing the meaning of words and sentences. Hence a keyword-based representation of the text is not enough to perform this task.

The semantic similarity can be calculated using dense vector representations generated by neural language models like BERT (Devlin, Chang, Lee, & Toutanova, 2019) or RoBERTa (Liu, et al., 2019), known as contextualized word embeddings, that then can be compared using the cosine similarity. Other approaches that have produced good results in the state of the art are to fine-tune the language models on the semantic similarity task using the Semantic Textual STS dataset<sup>20</sup> and to use probabilistic topic modelling (Badenes-Olmedo, Redondo-García, & Corcho, 2019) to represent and compare the documents.

### **5.3 Reading Comprehension**

To help users checking whether they have understood the content of a research object or a scientific paper we plan to generate a quiz about such content. The quiz comprises a set of questions that are generated automatically, and the corresponding answers extracted from the text content of the Research object or the scientific publication.

To generate the quiz, we resort to natural language processing tasks such as question generation and question answering where language models have shown very good results. Thus, for the question generation we plan to use a T5 model (Raffel, et al., 2020) fine-tuned on SQUAD following an answer aware approach where the model is trained to generate questions from examples containing a paragraph, a question, and the answer. Another option is to use a BART model (Lewis, et al., 2020) fine-tuned also on SQUAD following an answer agnostic approach where the model is trained to generate the question given a paragraph and some example questions. To answer the questions generated by either T5 or BART we plan to use a RoBERTa model (Liu, et al., 2019) fine-tuned on the task of extractive question answering using SQUAD as dataset.

### **5.4 Enrichment Dashboard**

Currently the results of the Enrichment Service are visualized in ROHub as part of the metadata for each Research Object. However, there is not a visualization that aggregates the metadata associated with all the Research Objects owned by a user. Thus, we plan to develop an enrichment dashboard that provides researchers with a simple and easy way to visualize in one glance the information extracted from scientific text and enrichment services. The dashboard is an appealing user interface comprising different widgets to display the metadata according to the type of information. In a similar way to the search dashboard described in 4.2 we plan to use Kibana as the technology framework to develop the dashboard.

## **6 Conclusions and future work**

In this deliverable we describe the main results achieved so far on the design and development of the RELIANCE text mining and enrichment services. These services are customized to the vocabulary used by RELIANCE user communities thanks to the text corpus of scientific communications gathered and reported in D3.1, considering their research interests. Such text corpus was analysed to enrich the Sensigrafo knowledge graph that encodes linguistic information and is used by the text analytics engine used in the project to analyse and disambiguate text. Upon the text analytics engine we develop a semantic enrichment service of Research Objects and single document to add metadata summarizing the text content. The Enrichment service of Research Objects is integrated in ROHub so

---

<sup>20</sup> STS benchmark available at [http://ixa2.si.ehu.es/stswiki/index.php/Main\\_Page](http://ixa2.si.ehu.es/stswiki/index.php/Main_Page)

that every public research object is automatically enriched with semantic metadata, and the enrichment service for single documents is onboarded in EOSC.

We develop a Search Engine that taps into the metadata added by the semantic enrichment enabling more complex queries and generating more complete and accurate results. To complement the Search service where users are required to know what they are looking for, we develop a Recommendation system where Research Objects are suggested considering the research objects owned by the user or other research objects that are of interest for the user. The recommendation is content-based and leverages the metadata added to Research Objects by the enrichment service to find related research objects. Both services are onboarded in EOSC. In addition, we present a first version of the Collaboration Spheres, a web application acting as the user interface for the recommendation system. We are working towards the integration of the Collaboration Spheres with the new authentication service of ROHub developed for RELIANCE and expect to finish the integration in the second phase of the project.

In this deliverable we briefly describe and include some design considerations about the extended analytic services which are going to be fully addressed in the second phase of the project. In addition, in the second year of the project we plan to integrate new visualization tools for the enrichment results, and the Search Engine. We also plan to develop a browser plugin to create bibliographic research objects as scientists browse the internet for relevant literature, where they can visualize the annotations added by the semantic enrichment Service.

## 7 References

There are no sources in the current document.

# Appendix

## A. Document Enrichment API documentation

### Request

Method	URL
POST	<a href="https://reliance.expertcustomers.ai/eosc/enrichment">https://reliance.expertcustomers.ai/eosc/enrichment</a>

Type	Params	Values
POST	file	file

### **file**

**file** must be a binary file (docx, doc, pptx, pdf or txt), sent with all client requests.

### Response

Status	Response
--------	----------

200	<pre>{   "filename": &lt;filename&gt;,   "lang": &lt;language&gt;,   "domains": [&lt;domain&gt;: &lt;value&gt;],   "concepts": [&lt;concept&gt;: &lt;value&gt;],   "expressions": [&lt;expression&gt;: &lt;value&gt;],   "people": [&lt;person&gt;: &lt;value&gt;],   "places": [&lt;place&gt;: &lt;value&gt;],   "org": [&lt;organization&gt;: &lt;value&gt;] }</pre> <p> filename (string)  language (string)  domain (string)  value (integer)  concept (string)  expression (string)  person (string)  place (string)  organization (string) </p>
400	{"error": "Missing file"}
401	{"error": "Incorrect file. Please use only doc, docx, pptx, pdf or txt"}
500	{"error": "Something went wrong. Please try again later."}

## B. Solr query examples and documentation

### B.1 Single term search

#### Request

Method	URL
GET	https://reliance.expertcustomers.ai/solr/ROHub/select

Type	Params	Values
GET	q	query
HEADER	authorization	User: standard_user Password: standard_user



### query

**query** must be a string term, e.g.:

<https://reliance.expertcustomers.ai/solr/ROHub/select?q=volcano>

### Response

Status	Response
200	<pre> {   "response_header":   {     "status": 0,     "QTime": 0,     "params":     {       [         "q": &lt;query&gt;       ]     }   }   "response":   {     "numFound": &lt;numFound&gt;,     "start": 0,     "numFoundExact": true,     "docs":     [       {         "id": &lt;id&gt;,         "title": &lt;title&gt;,         "autocomplete": [&lt;autocomplete&gt;],         "description": &lt;description&gt;,         "creator": &lt;creator&gt;,         "author": &lt;author&gt;,         "source_ro": &lt;source_ro&gt;,         "sketch": &lt;sketch&gt;,         "organization": [&lt;organization&gt;],         "place": [&lt;place&gt;],         "concepts": [&lt;concept&gt;],         "domains": [&lt;domain&gt;],         "content": &lt;content&gt;,         "compound_terms": [&lt;compound_terms&gt;],         "_version_": &lt;_version_&gt;       }     ]   } } </pre> <p> <b>query</b> (string)  <b>numFound</b> (integer) </p>



	<pre> id (string) title (string) autocomplete (string) description (string) creator (string) author (string) source_ro (string) sketch (string) organization (string) place (string) concept (string) domain (string) content (string) compound_terms (string) _version_ (integer) </pre>
404	<pre>{"error": "Unauthorized"}</pre>

## B.2 Facet search

### Request

Method	URL
GET	<a href="https://reliance.expertcustomers.ai/solr/ROHub/select">https://reliance.expertcustomers.ai/solr/ROHub/select</a>

Type	Params	Values
GET	q	query
GET	fq	facet_query
HEADER	authorization	User: standard_user Password: standard_user

### facet\_query

**facet\_query** must be a key-value entity. Key must be a field of the Solr schema. Value must be a string term, e.g.:

[https://reliance.expertcustomers.ai/solr/ROHub/select?fq=concepts:volcano&q=\\*:\\*](https://reliance.expertcustomers.ai/solr/ROHub/select?fq=concepts:volcano&q=*:*)

### Response

Status	Response
200	{

```

"response_header":
{
    "status": 0,
    "QTime": 0,
    "params":
    {
        [
            "q": <query>,
            "fq": <facet_query>
        ]
    }
}
"response":
{
    "numFound": <numFound>,
    "start": 0,
    "numFoundExact": true,
    "docs":
    [
        {
            "id": <id>,
            "title": <title>,
            "autocomplete": [<autocomplete>],
            "description": <description>,
            "creator": <creator>,
            "author": <author>,
            "source_ro": <source_ro>,
            "sketch": <sketch>,
            "organization": [<organization>],
            "place": [<place>],
            "concepts": [<concept>],
            "domains": [<domain>],
            "content": <content>,
            "compound_terms": [<compound_terms>],
            "_version_": <_version_>
        }
    ]
}
}

query (string)
facet_query (string)
numFound (integer)
id (string)
title (string)
autocomplete (string)
description (string)
creator (string)
author (string)
source_ro (string)

```

	<b>sketch</b> ( <b>string</b> ) <b>organization</b> ( <b>string</b> ) <b>place</b> ( <b>string</b> ) <b>concept</b> ( <b>string</b> ) <b>domain</b> ( <b>string</b> ) <b>content</b> ( <b>string</b> ) <b>compound_terms</b> ( <b>string</b> ) <b>_version_</b> ( <b>integer</b> )
<b>404</b>	<b>{</b> "error": "Unauthorized" <b>}</b>

## C. Recommendation API documentation

### Request

Method	URL
<b>POST</b>	https://reliance.expertcustomers.ai/spheresbackend/services/jsonservices/api/

Type	Params	Values
POST	ros	<b>ros</b>
POST	scientists	<b>scientists</b>

### **ros**

**ros** must be an array of strings. The strings must be research objects URIs that are available in the Search API. It can be empty.

### **scientists**

**scientists** must be an array of strings. The strings must be user URIs that are available in the Search API. It can be empty.

### Response

Status	Response
<b>200</b>	<b>{</b> "results": [ <b>&lt;result&gt;</b> ], <b>}</b>  <b>result</b> ( <b>string</b> )
<b>400</b>	<b>{</b> "error": "Wrong URI in context" <b>}</b> <b>{</b> "error": "Empty context. Please add 1 to 3 elements to the

---

	query"}
500	{"error": "Something went wrong. Please try again later."}