DOI: 10.5281/zenodo.7359654
DRAFT

Triple

Transforming Research through Innovative Practices for Linked Interdisciplinary Exploration

[SEPTEMBER 2022]

Advancing Open Scholarship

D2.5 – REPORT ON DATA ENRICHMENT

Version 1.0 – Final
PUBLIC

H2020-INFRAEOSC-2019
Grant Agreement 863420

DRAFT

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863420

Disclaimer- "The content of this publication is the sole responsibility of the TRIPLE consortium and can in no way be taken to reflect the views of the European Commission. The European Commission is not responsible for any use that may be made of the information it contains."

This deliverable is licensed under a Creative Commons Attribution 4.0 International License

D2.5 – Report on data enrichment

Project Acronym:	**TRIPLE**
Project Name:	**Transforming Research through Innovative Practices for Linked Interdisciplinary Exploration**
Grant Agreement No:	**863420**
Start Date:	**1/10/2019**
End Date:	**31/03/2023**
Contributing WP	**WP2**
WP Leader:	**IBL-PAN**
Deliverable identifier	**D2.5**
Contractual Delivery Date: 02/2020	**Actual Delivery Date: 10/2022**
Nature: Report	**Version: 1.0 Final**
Dissemination level	**PU**

DRAFT

Revision History

Version	Created/Modifier	Comments
0.0	Luca De Santis (Net7)	Index structure
0.1	Luca De Santis (Net7)	First complete version
0.2	Gert Breitfuss (Know-Center), Haris Georgiadis (EKT)	Revisions
0.3	Luca De Santis (Net7)	Applied suggested changes, added the “Conclusions”, “Publication Date normalization” and “Authors’ names normalization” chapters
1.0	Luca De Santis (Net7)	Final Version

Table of Contents

D2.5 – Report on data enrichment 0

1. Data enrichment and normalisation in the TRIPLE project 5

2. Publications and authors processing 7

2.1. Homogenization of content from data sources 7

2.1.1. OAI-PMH DC mapping 10

2.1.2. OAI-PMH EDM mapping 12

2.1.3. OpenAIRE mapping 14

2.1.4. Isidore mapping 15

2.2. Data normalisation for publications 17

2.2.1. Publication Date normalisation 19

2.2.2. Language normalisation 19

2.2.3. Keywords normalisation 20

2.2.4. Document Types normalisation 22

2.2.5. Licences and Access Rights normalisation 24

2.2.6. Authors' names normalisation 26

2.3. Language recognition and translation 28

2.4. Classification and automatic annotation 31

2.4.1. Classify service enrichment 32

2.4.2. Annotate service enrichment 34

2.5. Identification of duplicate publications 35

2.6. Authors disambiguation 37

3. Projects processing 39

4. Conclusions 44

5. References 45

DRAFT

List of figures

Figure 1 - Publications data flow 6
Figure 2 - Projects data flow 6
Figure 3 - Taxonomies codes as keywords before normalisation 21
Figure 4 - Accuracy of language detection using Apache Tika or Lingua 28
Figure 5 - Comparing language metadata with language detection done with Apache Tika. The case of Isidore 29
Figure 6 - Comparing language metadata with language detection done with Apache Tika. The case of Isidore 30
Figure 7 - How Google Scholar deals with duplicates 36
Figure 8 - An example of an unclaimed profile in GoTriple 39
Figure 9 - Keywords of a CORDIS project 44

List of tables

Table 1 - TRIPLE Publications Data Model 8
Table 2 - OAI-PMH DC mapping 10
Table 3 - OAI-PMH EDM mapping 12
Table 4 - OpenAIRE mapping 14
Table 5 - Isidore mapping 15
Table 6 - TRIPLE Document Types vocabulary 22
Table 7 - TRIPLE Licences vocabulary 24
Table 8 - TRIPLE Access Rights vocabulary 26
Table 9 - TRIPLE Categories codification 32
Table 10 - Expansion of the TRIPLE data model 36
Table 11 - Fields of the Profiles Elasticsearch index for the management of publications' authors 38
Table 12 - TRIPLE Projects Data Model 40
Table 13 - Mapping from CORDIS project taxonomy to the TRIPLE Disciplines 41

DRAFT

Acronyms

AKA	Also Known As
BASE	Bielefeld Academic Search Engine
CEF	Connecting Europe Facility
COAR	Confederation of Open Access Repositories
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DCTerms	DCMI Metadata Terms
DOI	Digital Object Identifier
EDM	Europeana Data Model
EuroSciVoc	European Science Vocabulary
JSON	JavaScript Object Notation
MORESS	Mapping of Research in European Social Sciences and Humanities
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
SCRE	Semantic Content Retrieval Engine
SSH	Social Sciences and Humanities
URL	Uniform Resource Locator
XML	Extensible Markup Language

DRAFT

Publishable Summary

In this deliverable, the strategies for data enrichment in TRIPLE are presented. Through the Core Pipeline, named SCRE, metadata regarding publications and projects for the Social Sciences and Humanities are automatically harvested, mapped in the TRIPLE data model, curated, enriched and finally saved in the GoTriple platform's indexes.

The document starts by presenting the ways SCRE imports publications metadata from OAI-PMH endpoints, OpenAIRE and Isidore data dumps. This reflects the strategies for integrating content which was planned in the project. On the one hand, OAI-PMH is a well-known and established standard for content harvesting: many data providers, especially those of small dimension, support it, facilitating therefore their onboarding in GoTriple. The support for OpenAIRE and Isidore, on the other hand, responds to the wish to also harvest data from large aggregators, a strategy that allowed GoTriple to quickly present a significant amount of publications in its index (more than 4 million at the time of writing).

Then the normalisation strategies applied to the acquired metadata are described. By analysing the first batches of acquired data, it has been decided to define the rules to normalise and clean the attributes for the following metadata: publication date, language codes, keywords, document types, licences, access rights and authors' names. In the document, the definition of controlled vocabularies for some of these attributes is also presented.

Then enrichment services are explained, including language recognition, translation, automatic classification and annotation.

The services to detect duplicate publications and to disambiguate authors are also discussed, followed by the presentation of the acquisition and processing of project metadata

Some final remarks on the data enrichment process, including the difficulties that have been faced and solved, conclude the document.

DRAFT

1. Data enrichment and normalisation in the TRIPLE project

TRIPLE's goal is to implement a multilingual discovery platform for the Social Sciences and Humanities (SSH) domains: its main outcome is the GoTriple website, where users can look for and find information about publications, projects, authors and researchers' profiles by using the platform's search engine.

All the platform's data are automatically harvested from various sources, thanks to the metadata ingestion and curation pipeline, named SCRE, whose technical architecture will be fully described in deliverable D4.4 [1]. Basically, SCRE processes data by using a pipeline approach: it consists in fact of several specialised services, built by using the Apache Camel [2] technology, each dedicated to implement a particular feature. Roughly speaking we can distinguish amongst:

- *Connectors*: the components which retrieve metadata about publications and projects from specific data sources
- *Processors*, which curate or enrich the original metadata, according to the logic described herein
- *Persisters*, which finally saves the enriched metadata in the platform indexes.

In fact, we can imagine the data enrichment and normalisation of metadata applied in SCRE as a "data flow", starting from the retrieval of the single information (a publication or a project description) by a connector, the enrichment of its metadata while passing through every single processor and finally the memorisation of the final result in the platform indexes, implemented with the Elasticsearch search engine [3], via a persister.

The SCRE architecture is actually more complicated than this, but for the purpose of the present deliverable we simplified the description

The data flows for publications and projects processing are presented in the two diagrams that follow.

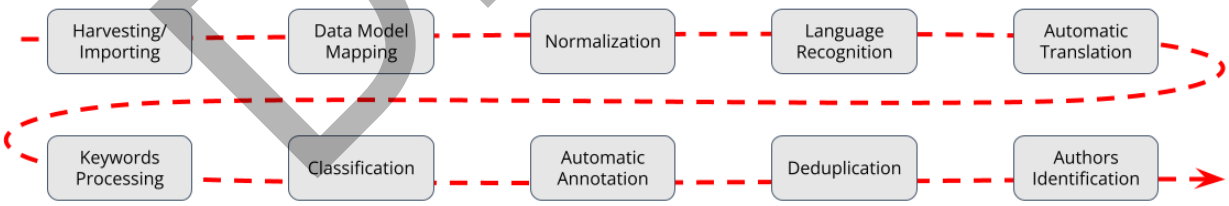


Figure 1 - Publications data flow

Figure 2 - Projects data flow

DRAFT

In the following chapters the normalisation and enrichment logic devised in TRIPLE is presented, by distinguishing the strategies applied for publications and projects. While the goal of persisters is simply to save the metadata into the specialised Elasticsearch indexes, the actual adaptation, curation and enrichment of the original data are performed by connectors and processors, whose core logic is therefore described in the chapters that follow.

2. Publications and authors processing

2.1. Homogenization of content from data sources

As said publications metadata are acquired from various sources by using dedicated SCRE's connectors. Connectors must:

- Access data from external sources or local file dumps
- Retrieve every piece of information, in this case the metadata of a single publication
- Identifying the salient metadata and mapping them to the TRIPLE data model's schema.

In TRIPLE the following connectors have been implemented:

- OAI-PMH
- OpenAIRE
- Isidore.

OAI-PMH [4] is the most used connector since a significant number of aggregators and data providers offer a dedicated endpoint to allow third parties to harvest their repositories via this standard protocol. Technically speaking, an OAI-PMH endpoint is a collection of web services, accessible on the web via the HTTP(S) protocol. Normally providers arrange their data in sets, "an optional construct for grouping items for the purpose of selective harvesting" [4].

Information is returned in XML format by using several data models: the most common is Dublin Core (DC), whose mapping has been implemented in TRIPLE together with the Europeana Data Model (EDM).

To retrieve data from a specific OAI-PMH provider, it is necessary to specify in SCRE the following parameters: URL of the endpoint, the supported data model (DC or EDM) and the name of a single set to retrieve. The connector periodically queries every OAI-PMH endpoint defined in the platform in search of new content: it exploits in fact the incremental harvesting mechanism of the protocol to only request updated content, that is those published (or modified) after the last harvesting of that specific repository.

OpenAIRE [5] is "a network of Open Access repositories, archives and journals that support Open Access policies" [6]. OpenAIRE provides TRIPLE with data dumps, consisting of a collection

DRAFT

of files in their specific JSON format [7], aggregated and compressed in a single .zip file, corresponding to a selection of their SSH related publications.

These dumps are periodically and automatically produced by OpenAIRE through a query of their indexes done by using specific search keywords describing the SSH domain. The corresponding dump is then published on Zenodo, where it can be downloaded and transferred to the local file system of SCRE.

The OpenAIRE connector, therefore, scans the local file system where an OpenAIRE dump is located: the only parameter to specify for the connector is therefore the path of the dump that must be imported by SCRE. In this case, SCRE always performs a full import: all the files included in the dump will be therefore processed by SCRE.

Isidore [8] is the SSH discovery platform developed by Huma-Num since 2009. It can be considered as the prototype and the main inspiration for the GoTriple platform and for the TRIPLE project in general. As in the case of OpenAIRE, the Isidore publications metadata are made available as files in the local file system of SCRE, formatted in XML and organised in a hierarchy of directories. Again, also for Isidore the harvesting consists of a full import of all the files stored in the path of the file system, path that is taken by SCRE as the only parameter for performing the acquisition.

Even if they are not currently used in TRIPLE, it is worth mentioning the two other experimental connectors that SCRE offers. The first allows the retrieval and processing of web feeds published in the RSS and ATOM syndication protocols. The second allows crawling web pages in order to extract salient data from them. These two connectors have not been used in TRIPLE since the metadata that they can extract are very limited, basically title, url, abstract, publication date (plus full-text and main image). Also, they are not specialised for publications but might be more generically exploited to retrieve any possible content published as a web page. While not employed at the moment, their use can be reconsidered in the future if the need arises to harvest information from providers who only publish data as “ordinary” web pages, not supporting OAI-PMH or metadata dump exports.

Regardless of the specific data source it connects to, every connector processes publications metadata one at the time. It loops the results of a query (an OAI-PMH call or the scanning of the local file system for OpenAIRE and Isidore) to retrieve the information of a single publication, which is then mapped in the corresponding fields of the TRIPLE data model. The TRIPLE data model is illustrated in Table 1 .

Field	Description	Class
Identifier	List of the document’s original identifiers	schema:identifier
Doi	It includes the valid DOI of the document, if existing.	schema:identifier
Title	Title of the document	schema:headline

DRAFT

Abstract	Abstract or description of the document	schema:abstract
Author	Authors of the document: list of authors' names and identifiers (of the Profile Elasticsearch index).	schema:author
Contributor	List of contributors	schema:contributor
Document type	Type of the document	schema:additionalType
Keywords	List of producer's keywords	schema:keywords
Subject (MORESS categories)	List of MORESS Categories (TRIPLE's disciplines)	sioc:topic
TRIPLE thesaurus	TRIPLE thesaurus entries	schema:knowsAbout
Language	Language of the content	schema:inLanguage
Publication date	Date of publication or creation	schema:datePublished
Publisher	List of publishers	schema:publisher
Aggregator	Aggregator of the document (eg: Isidore)	schema:provider
Primary producer	Primary producer of the publication (eg: HALSHS)	schema:producer
License	A license or a type of license	Schema:license
Access Rights	Information on access status	schema:conditionsOfAccess
URL of full document	URL of the full text version of the article	schema:url
URL of the landing page	URL of the landing page of the document.	schema:mainEntityOfPage
Sources information	Source (free text) from: dc:source, dcterms:source (e.g. journal issue)	schema:mentions
Sources URL	Source (HTTP) from: dc:source, dcterms:source (e.g. URL from a publishing platform)	schema:isBasedOnURL
Spatial location of dataset	Content's spatial location of collection (list)	schema:spatialCoverage
Temporal period of dataset	Content's temporal period of collection (list)	schema:temporalCoverage
Format	File format (multiple values)	schema:encodingFormat

DRAFT

Funding reference	List of projects' identifiers (of the Project Elasticsearch index).	schema:funder

Table 1 - TRIPLE Publications Data Model

The mapping rules of each connector vary according to the nature of the metadata schema of the data source. It is practically impossible to have a full correspondence of the publications metadata with the TRIPLE data model and this is particularly true with "poorest" descriptive formats such as OAI-PMH DC. What follows are the mapping rules implemented for each connector implemented in TRIPLE.

It is important to point out that to identify a single element, for example the date of the publication, several elements in the original metadata are analysed and searched for. This is due to the fact that very often data providers don't fully follow the standards (e.g. OAI-PMH DC) either for limitation in their expressivity or for different interpretations of them. For example, it might happen that expanded data models are used as opposed to those advocated by the standard (e.g. DCTerms [9] instead of the "core" DC vocabulary). This forces the SCRE connectors to apply several strategies to recognise the metadata element to map in the corresponding field of the TRIPLE data model, as it will be evident in the rules described in the chapters that follow.

For example, for OAI-PMH DC, whose model is not very rich, normally publishers use the same element, dc:rights, to indicate both the access rights (conditionOfAccess) and the licence of the document. It is therefore the Data Normalisation processor, described in chapter 2.2, that takes care of distinguishing between these two cases.

It is also worth noting the rules applied to discriminate the URLs of a document, in particular the possible full text of the publication ("url" element) and its landing page, which is a normal web page which presents the article ("mainEntityOfPage" element). In the absence of specific elements to distinguish these two cases, we simply resort to recognise if specific elements of the original metadata (e.g. dc:source or dcterms:relation for OAI-PMH DC) contain a valid URL, starting with "http", and, in case, if it ends or not with ".pdf.

DRAFT

2.1.1. OAI-PMH DC mapping

We present here the mapping rules from OAI-PMH DC metadata to the TRIPLE data model.

XML element	TRIPLE Data Model element
identifier (header)	id (attribute of document element)
dc:title, dcterms:title	headline
dcterms:identifier, dc:identifier	identifier; if it is a valid DOI, it is also stored in the "doi" element

ONLY if the element doesn't start with "http"	
dcterms:identifier, dc:identifier dcterms:relation, dc:relation, dc:source, dcterms:source ONLY if the element starts with "http" and ends with ".pdf"	url
dcterms:identifier, dc:identifier Only the first element is taken. If null the elements dc:source, dcterms:source are considered. In any case ONLY if the element starts with "http" and doesn't end with ".pdf"	mainEntityOfPage
dcterms:language, dc:language	inLanguage
dcterms:rights, dc:rights, dcterms:license	conditionOfAccess
dcterms:license, dc:rights, dcterms:rights	license
dcterms:publisher, dc:publisher	publisher
dcterms:date, dc:date, dcterms:issued, dcterms:created, dcterms:available	date_published
dc:type, dcterms:type	additionalType
dc:subject , dcterms:subject Only keywords that contain the xml:lang attribute or that have no attribute are considered (see chapter 2.2 for an explanation of this choice)	keywords
dc:description, dcterms:description, dcterms:abstract	abstract
xml:lang of the abstract element (see previous row) if present.	lang attribute of the abstract element
dc:creator, dcterms:creator	author

DRAFT

dc:source, dcterms:source ONLY if the element doesn't start with "http"	mentions
dc:source, dcterms:source ONLY if the element starts with "http"	isBasedOnURL
dcterms:temporal If empty we take dc:coverage, dcterms:coverage are taken, only if they start with a number	temporalCoverage
dcterms:spatial If null the elements dc:coverage, dcterms:coverage are taken, only if they DON'T start with a number	spatialCoverage
dc:format, dcterms:format	encodingFormat
dcterms:contributor, dc:contributor	contributor

Table 2 - OAI-PMH DC mapping

2.1.2. OAI-PMH EDM mapping

We present here the mapping rules from OAI-PMH EDM metadata to the TRIPLE data model.

XML element	TRIPLE Data Model element
identifier (header)	id (attribute of document element)
emd:isShownAt If null the elements dc:identifier, dc:source are considered. In any case ONLY if the element starts with "http" and doesn't end with ".pdf"	mainEntityOfPage
dc:title	headline
dc:identifier ONLY if the element doesn't start with "http"	identifier; if it is a valid DOI, it is also stored in the "doi" element
edm:isShownBy If null the elements dc:identifier, dc:relation,	url

DRAFT

edm:isShownAt, edm:hasView ONLY if they start with “http” and ends with “.pdf”	
dc:language	inLanguage
odrl:inheritFrom, edm:rights, dc:rights	conditionsOfAccess
odrl:inheritFrom, edm:rights, dc:rights	license
dc:publisher	publisher
dcterms:created	datePublished
dc:type	additionalType
dc:subject	keywords
dc:description	abstract
xml:lang of the abstract element (see previous row) if present.	lang attribute of the abstract element
dc:creator	author
dc:source ONLY if the element doesn’t with “http”	mentions
dc:source ONLY if the element starts with “http”	isBasedOnURL
dcterms:temporal If null the element dc:coverage is taken, only if it starts with a number	temporalCoverage
dcterms:spatial If null the element dc:coverage is taken, only if it DOESN’T start with a number	spatialCoverage
dc:format, dcterms:hasFormat	encodingFormat
dc:contributor	contributor

Table 3 - OAI-PMH EDM mapping

DRAFT

2.1.3. OpenAIRE mapping

We present here the mapping rules from OpenAIRE metadata to the TRIPLE data model.

JSON element	TRIPLE Data Model element
ResultPid/value id	identifier ; if it is a valid DOI, it is also stored in the “doi” element
id	id
maintitle	headline
instance/url ONLY if the element starts with “http” and ends with “.pdf”	url
language/code	inLanguage
originalId, instance/url, collectedfrom/key, instance/collectedfrom/key, instance/hostedby/key ONLY if the element starts with “http” For instance/url it mustn’t also end with “.pdf”	isBasedOnURL
originalId, instance/url, collectedfrom/key, instance/collectedfrom/key, instance/hostedby/key, ONLY if the element doesn’t start with “http”	mentions
subjects/subject/value ONLY if subjects/subject/scheme=keyword and the element doesn’t contain “[“	keywords
instance/type	additionalType
description	abstract

DRAFT

bestaccessright/label	conditionsOfAccess
instance/license	license
instance/url ONLY if the element starts with “http” and doesn’t end with “.pdf”	mainEntityOfPage
publicationdate	datePublished
format	encodingFormat
publisher	publisher
contributor	contributor
author/fullname	authors
coverage ONLY if it starts with a number	temporalCoverage
coverage ONLY if it DOESN’T start with a number	spatialCoverage

Table 4 - OpenAIRE mapping

2.1.4. Isidore mapping

We present here the mapping rules from Isidore metadata to the TRIPLE data model.

XML element	TRIPLE Data Model element
/isidore[@uri]	id (attribute of document element)
url ONLY if the element starts with “http” and doesn’t end with “.pdf”	mainEntityOfPage
title	title
/isidore[@uri] && ore/similar	identifier; if it is a valid DOI, it is also stored in the “doi” element
//ore/aggregates[@crawl='true'] If no //ore/aggregates[@crawl='true']	url

DRAFT

take /isidore/url ONLY if the element starts with “http” and ends with “.pdf”	
items[@type='ISIDORE_LANG']/item/prefLabel[@xml:lang='en']	inLanguage
dcterms:license, dc:rights accessRights, dcterms:accessRights	conditionOfAccess
dcterms:license, dc:rights accessRights, dcterms:accessRights	license
publishers/publisher	publisher
date/NormalizedDate	datePublished
//items/ISIDORE_TYPE_FACET/preflabel[@xml:lang = “en]	additionalType
subjects/subject only subjects containing the xml:lang attribute (We take only those keywords containing “en” as language, as advised by Isidore personnel)	keywords
xml:lang (part of the element in the previous row)	lang (attribute of keywords. For what has been said before, we will always have “en” here)
abstract	abstract
xml:lang (part of the element in the previous row)	lang (attribute of abstract element)
enrichedCreators/creator	author
dc:source, dcterms:source ONLY if the element doesn’t start with “http”	mentions
dc:source, dcterms:source ONLY if the element starts with “http”	isBasedOnURL
coverages/coverage	temporalCoverage

DRAFT

ONLY if it starts with a number	
coverages/coverage ONLY if it DOESN'T start with a number	spatialCoverage
dc:format	encodingFormat
/isidore/source_info/sourceName	producer
contributors/contributor	contributor

Table 5 - Isidore mapping

2.2. Data normalisation for publications

Harvesting data from different sources in TRIPLE poses a problem in terms of both quality and diversity of the retrieved metadata. The data normalisation components aim at producing a consistent, and possibly improved, output by processing the original metadata acquired by the connectors.

The normalisation rules described herein are the result of the analysis performed on real data retrieved from different sources[1] in a previous phase of the project. We have in fact encountered several problems including:

- impossible dates of publication, e.g. "0" or "512".
- missing or wrong association with languages in the textual elements "title" and "abstract"
- inconsistent use of labels and codes to indicate the language (e.g. "en", "en_US", "eng" always for the English language)
- use of free textual descriptions to specify the type of the publication and, especially, the licence and access rights
- use of the same metadata element to indicate both licences and access rights
- Items and codes of the providers' taxonomies together with free-form keywords (see chapter 2.2.3).

The metadata elements that it has been decided to curate and normalise in TRIPLE are:

- publication date (date_published)
- the language of the document (in_language) and the "lang" attribute of titles (headline), abstracts and keywords
- keywords
- the document type (additional_type)
- the licence

[1] Including DOAJ, OpenAIRE, EKT and a subset of the Isidore dataset for a total of around 1.6 million documents.

DRAFT

- the access rights (conditions_of_access).

The data normalisation process in SCRE is based on these guidelines:

- removing duplicates when they appear
- cleaning textual strings, by trimming leading and trailing spaces and removing all the HTML codes in them
- defining a controlled vocabulary for some normalised elements
- defining for each element a set of rules to determine the right value to associate with its normalised counterpart
- always maintaining the original metadata received, which means that normalised values are copied in separate elements of the final TRIPLE Publications index on Elasticsearch.

The latter point is very important: this guarantees the possibility to access the original metadata, while the normalised elements allow for a better presentation and more effective filtering of search results in GoTriple.

For this reason, the TRIPLE Data Model presented in 2.1 has been expanded to include the following elements:

- original_publicationDate
- original_languages
- original_documentTypes
- original_license
- original_conditionsOf Access
- the original_lang attribute for headline (title), abstract and keywords
- discarded_keywords (see chapter 2.2.3).

What follows are the normalisation rules that have been implemented in the SCRE processors.

2.2.1. Publication Date normalisation

It has been decided to maintain as valid dates only those after 1.700. The result is then normalised in the ISO 8601 [10] format, e.g.

- yyyy
- yyyy-mm
- yyyy-mm-dd.

In any case, the original metadata is stored in the *original_date_published* field of the Elasticsearch index. Moreover, an extra field, *date_facets*, is also added to expand the publication date in the DateTime format: this way YYYY here becomes YYYY-01-01 and YYYY-mm becomes YYYY-mm-01. This field has been added to enable grouping by year in the facet filters of the search results page of GoTriple.

DRAFT

2.2.2. Language normalisation

Its purpose is to identify and correctly represent the language of a publication and of its textual elements (title, abstract, keywords).

It has been decided to create a controlled vocabulary containing TRIPLE's 11 languages (Croatian, English, French, German, Greek, Italian, Polish, Portuguese, Slovenian, Spanish and Ukrainian), other most common languages (e.g. Arabic, Dutch, Swedish…) and two special labels, "other" and "undefined".

Language elements are identified through a series of pattern matching rules. When the language specified in the metadata element is not in the controlled vocabulary the result is set to "other" while when the language is missing from the element it is considered "undefined".

All language elements are formatted in the ISO-639-1 [11] two-characters notation, the format used by the language recognition library and by the automatic translation service, which will be presented in the following chapters.

What follows are the 26 entries of the language vocabulary used in the normalisation process. For each admitted language the corresponding ISO-639-1 code is included.

- Croatian (hr)
- Catalan (ca)
- English (en)
- French (fr)
- German (de)
- Greek (el)
- Italian (it)
- Polish (pl)
- Portuguese (pt)
- Spanish (Castilian) (es)
- Slovenian (sl)
- Serbian (sr)
- Ukrainian (uk)
- Hungarian (hu)
- Dutch (nl)
- Russian (ru)
- Hebrew (he)
- Swedish (sv)
- Danish (da)
- Finnish (fi)
- Norwegian (no)
- Albanian (sq)
- Turkish (tr)
- Arabic (ar)

DRAFT

- other
- undefined.

2.2.3. Keywords normalisation

Almost all publications have assigned a certain set of keywords, normally chosen by authors, that describe the content of the article. Analysing the first datasets harvested in TRIPLE, various issues regarding keywords were noticed. Some were easy to fix, for example removing duplicates or trimming the blank spaces before and after every string.

It was also decided to normalise the language attribute associated with them if present (the "lang" attribute) by using the controlled vocabulary described in the previous chapter. At the same time it was decided, contrary to what has been implemented for titles/headlines and abstracts (see next chapters), not to use the automatic language recognition service, on the one hand, because this service might produce inaccurate results for short strings, on the other hand, we noticed that sometimes terms in multiple languages appear together in the same keyword element.

Another issue, raising a long discussion amongst TRIPLE WP2 partners, is how to deal with keywords that are actually codes or labels of taxonomies used by data providers. This issue was raised since it is necessary to present data in a clean way, without keywords that might look meaningless or confusing to the final user. An example is shown in the image that follows: as it can be seen, strings like "J" and "JZ2-6530" are meaningless to the final user as they are in fact codes of taxonomies entries used by the data provider of the article.

Figure 3 - Taxonomies codes as keywords before normalisation

By analysing several data sources, we noticed certain patterns that identify the keywords to maintain and those to remove, which depend on the various data sources. Basically, the keywords which refer to a taxonomy include the vocabulary name in the xsi:type attribute of the element: once identified we can simply skip them. This is how the normalisation rule is applied to the various data sources.

OAI-PMH

We accept as keywords the dc:subject elements ONLY if they do not contain any other attribute besides xml:lang.

DRAFT

Examples:

```
<dc:subject>Livestock</dc:subject>  - ACCEPTED
<dc:subject xml:lang="sl-SI">slovenski jezik</dc:subject> - ACCEPTED
<dc:subject xsi:type="dcterms:LCC">Agriculture (General)</dc:subject> - REJECTED
```

OpenAIRE

Keywords are taken from subjects/subject/value elements only if:

- the corresponding subject/scheme = keyword
- the values doesn't contain the "[" character.

Examples:

```
"subject":{"scheme":"keyword","value":"marl soils"}} - ACCEPTED
"subject":{"scheme":"MAG","value":"Geology"}} - REJECTED
"subject":{"scheme":"keyword","value":"[SHS.EDU]Humanities and Social Sciences/Education"}} - REJECTED
```

Isidore

Keywords are taken from subjects/subject only if the subject contains the xml:lang attribute.

Examples:

```
<subject xml:lang="fr">sélection à l'entrée</subject> - ACCEPTED
<subject>[SHS.SCIPO]Humanities and Social Sciences/Political science</subject> - REJECTED
```

It is important to point out that this removed data is not lost but it is maintained in the discarded_keywords element of the Elasticsearch GoTriple Publications index. Here the whole element of a discarded keyword is stored as a string (e.g. '<dc:subject xsi:type="dcterms:LCC">S1-972</dc:subject>').

Finally every single keyword in the GoTriple Elasticsearch index is stored as:

```
{
    text: "the keyword",
    lang: original language normalised in the ISO-639-1 format,
    original_lang: the language (if present) of the original metadata,
}
```

Other normalisation rules that were discussed but not applied include: changing the case of the strings; removing strings shorter than a certain length; automatically separating terms in strings containing commas.

DRAFT

2.2.4. Document Types normalisation

In the original metadata, document types are specified as strings in various forms: the most common situations have been identified and pattern matching rules have been developed to perform the normalisation to the controlled vocabulary. The latter has been created after having analysed the metadata received in the first GoTriple index: we tried to include the most common situations by also considering the list of types that the Isidore platform uses and the entries of the COAR Resource Types vocabulary [12]. The table that follows shows the list of TRIPLE document types together with its COAR correspondence.

TRIPLE Document Type	Codification	COAR Resource Type
Article	typ_article	Journal Article: https://vocabularies.coar-repositories.org/resource_types/c_6501/
Bibliography	typ_bibliography	Bibliography: https://vocabularies.coar-repositories.org/resource_types/c_86bc/
Blog post	typ_blog-post	Blog post: https://vocabularies.coar-repositories.org/resource_types/c_6947/
Book	typ_book	Book: https://vocabularies.coar-repositories.org/resource_types/c_2f33/
Conference	typ_conference	Conference Output: https://vocabularies.coar-repositories.org/resource_types/c_c94f/
Dataset	typ_dataset	Dataset: https://vocabularies.coar-repositories.org/resource_types/c_ddb1/
Image	typ_image	Image: https://vocabularies.coar-repositories.org/resource_types/c_c513/
Learning object	typ_learning-object	Learning object: https://vocabularies.coar-repositories.org/resource_types/c_e059/

DRAFT

Manuscript	typ_manuscript	Manuscript: https://vocabularies.coar-repositories.org/resource_types/c_0040/
Report	typ_report	Report: https://vocabularies.coar-repositories.org/resource_types/c_93fc/
Periodical	typ_periodical	Other periodical: https://vocabularies.coar-repositories.org/resource_types/QX5C-AR31/
Preprint	typ_preprint	Preprint: https://vocabularies.coar-repositories.org/resource_types/c_816b/
Review	typ_review	Review: https://vocabularies.coar-repositories.org/resource_types/c_efa0/
Software	typ_software	Software: https://vocabularies.coar-repositories.org/resource_types/c_5ce6/
Text	typ_text	Text: https://vocabularies.coar-repositories.org/resource_types/c_18cf/
Thesis	typ_thesis	Thesis: https://vocabularies.coar-repositories.org/resource_types/c_46ec/
Map	typ_map	Map: https://vocabularies.coar-repositories.org/resource_types/c_12cd/
Other Assigned only if there is no other possible association	other	Other: https://vocabularies.coar-repositories.org/resource_types/c_1843/

DRAFT

with a document type		
Undefined Assigned if the corresponding original metadata is missing	undefined	-

Table 6 - TRIPLE Document Types vocabulary

2.2.5. Licences and Access Rights normalisation

For creating the licences and access rights vocabularies, presented in the two tables below, the starting point was again the analysis of the most common metadata in the first GoTriple index. This also allowed us to define the pattern matching rules to normalise the metadata received as input.

TRIPLE License	Codification
CAIRN	lic_cairn
Creative Commons Various spelling and acronyms, e.g. CC0, CCBY,…, and full Creative Commons URLs as well..	lic_creative-commons
Open source Various spelling and licence names, e.g. Apache, GPL, BSD, MIT	lic_open-source
Clarin Pub	lic_clarin-pub
Microsoft Public Licence	lic_ms-pl
Microsoft Reciprocal Licence	lic_ms-rl
Open Data Various spelling including DbL, Open Data Commons, Open Database Licence	lic_open-data ()
Meta-Share Various spelling including:	lic_meta-share

DRAFT

META-SHARE No Redistribution, META-SHARE NonCommercial NoRedistribution, META-SHARE Commercial No Redistribution For a Fee, META-SHARE Noncommercial No Redistribution For a Fee	
CLARIN-ACA Including CLARIN ACA-NC	lic_clarin-aca
CLARIN-RES Including CLARIN RES-NC	lic_clarin-res
ELRA licences	lic_elra
Other Assigned only if there is no other possible association with a licence	other
Undefined Assigned if the corresponding original metadata is missing	undefined

Table 7 - TRIPLE Licences vocabulary

TRIPLE Access Right (conditionsOfAccess)	Codification
All Rights Reserved Including multiple variants and spelling: ©, (c), tous droits réservés, derechos de autor, copyright, c.	acr_all-rights-reserved
Open Access Including variants, e.g. accès libre	acr_open-access
Closed Access	acr_closed-access
Restricted access or use	acr_restricted-access-or-use
Public Domain Including variants, e.g. domaine publique	acr_public-domain
Free Access	acr_free-access

DRAFT

Other Assigned only if there is no other possible association with an access right	other
Undefined Assigned if the corresponding original metadata is missing	undefined

Table 8 - TRIPLE Access Rights vocabulary

2.2.6. Authors' names normalisation

As already said, some harvesting protocols (e.g. OAI-PMH with DC format) have a limited semantic: in particular authors are represented as a flat list. This makes it impossible to easily manage situations in which an author's name is presented with multiple spellings. For example the same Greek author might be indicated as such:

```
<dc:creator xml:lang="en">Kapanidis, Nikolaos</dc:creator>
<dc:creator xml:lang="el">Καπανίδης, Νίκος</dc:creator>
```

It is evident that the risk of creating duplicates is high, especially because the ASCII character transliteration might lead to different ways of spelling the same name.

This is the case for example for the previous Greek name, *Νίκος*, which can be correctly transliterated both as *Nikos* and *Nikolaos*.

Since TRIPLE has multilingualism as a main goal, a special normalisation rule is needed.

The one implemented in the SCRE pipeline checks if authors have the *xml:lang* attribute set and if it is different for some of them. If this is the case, we proceed with the following steps:

- all authors' names are extracted and converted to 7-bit ASCII by using the JUnidecode Java library [13].
- duplicates are identified and removed
- the authors with the *xml:lang* attribute equal to "en" (English) are taken. If no "en" lang element is present, we take the authors in the main language of the publication; otherwise we simply select randomly the authors with the same language attribute.
- the names of the previous steps are selected as the authors of the publication
- the remaining authors are stored in the "discarded_authors" multivalue field of the Publications index of Elasticsearch. Since this field is indexed, a search for a Greek name, for example, can return the correct result.

According to this rule, the following cases would be managed as such.

```
<dc:creator xml:lang="en">Kapanidis, Nikolaos</dc:creator>
<dc:creator xml:lang="el">Καπανίδης, Νίκος</dc:creator>
```

leads to:

DRAFT

- author: Kapanidis, Nikolaos
- discarded_authors: Καπανίδης, Νίκος

while:

```
<dc:creator xml:lang="en">Forschungsgruppe Wahlen, Mannheim</dc:creator>
<dc:creator xml:lang="de">Forschungsgruppe Wahlen, Mannheim</dc:creator>
```

leads to:

- author: Forschungsgruppe Wahlen, Mannheim

The discarded authors will be stored in the front-end Elasticsearch index as:

```
"discarded_authors": [
  {
    "lang": "el",
    "value": "Καπανίδης, Νίκος"
  },
  {
    "lang": "fr",
    "value": "Kapanidis, Nikos"
  },
  …
]
```

2.3. Language recognition and translation

The need to use a language recognition processor in SCRE is raised from the fact that some publications miss the language attribute in their textual descriptions, that is the title/headline and the abstract.

It was decided to use Apache Tika [14], a very popular Java toolkit for processing textual documents, with plenty of features including language recognition, after having confronted its accuracy with another quite well-known Java library, Lingua [15].

To decide which library to use, their behaviour was analysed with a sample of about 350 titles and 440 abstracts extracted from documents harvested in the first phase of the TRIPLE project, by considering two cases:

1) Language identification using the original text;
2) Language identification using a normalised version of the text (no punctuation, lower case letters, etc.);

The results obtained are shown in the diagrams below which show on the one hand how Tika consistently performs better than Lingua and on the other that normalising the text doesn't

DRAFT

bring any improvement. Therefore Tika is the service used for the language normalisation processor in SCRE on plain titles and abstracts.

Figure 4 - Accuracy of language detection using Apache Tika or Lingua

From an initial analysis of the processed data we often noticed differences between the language identified by Tika and the language specified as an attribute of the title and abstract elements (not to be confused with the language specified in the <language> field which refers to the content of the article).

A further analysis was then made in order to decide whether it is more convenient to always use Apache Tika regardless of the presence or absence of the lang attribute of titles and abstracts, or to use Apache Tika only when this attribute is not present.

The results of this analysis on, respectively, samples of 477 documents from Isidore and 500 documents from EKT, are shown in the diagrams that follow.

DRAFT

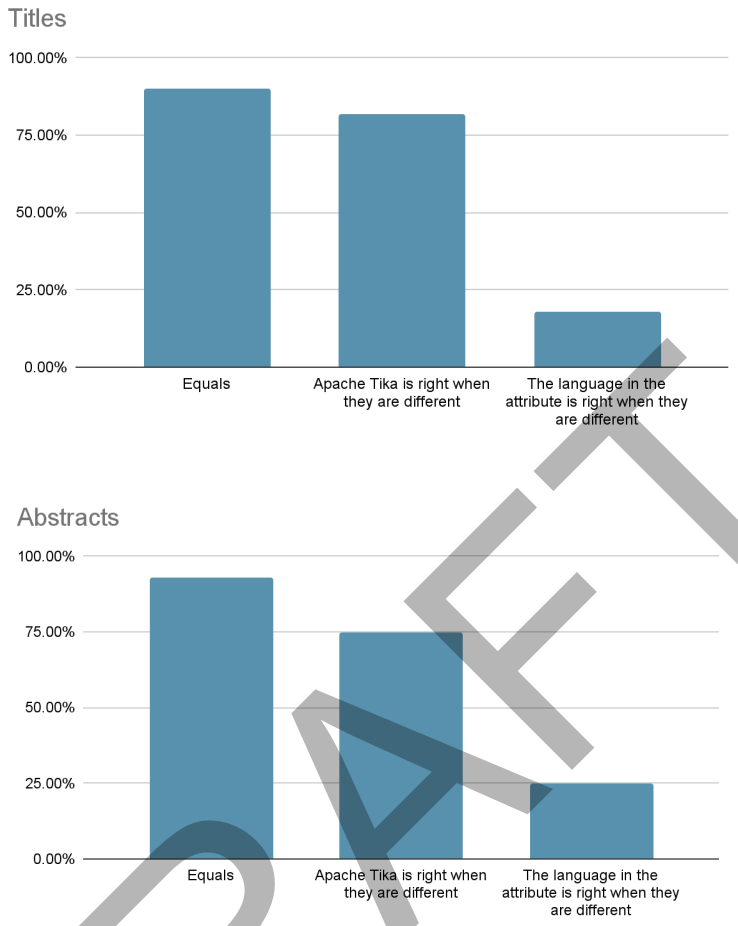
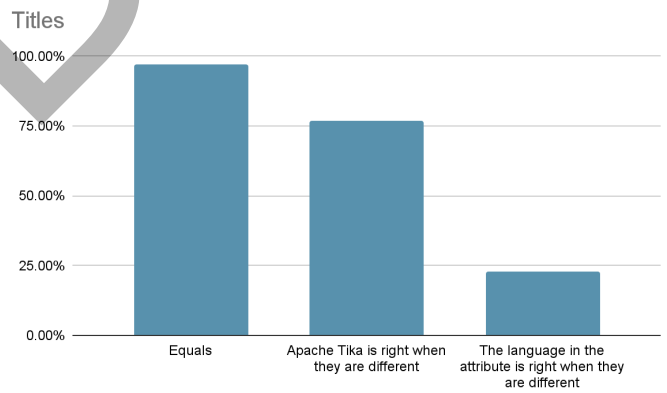


Figure 5 - Comparing language metadata with language detection done with Apache Tika. The case of Isidore



DRAFT

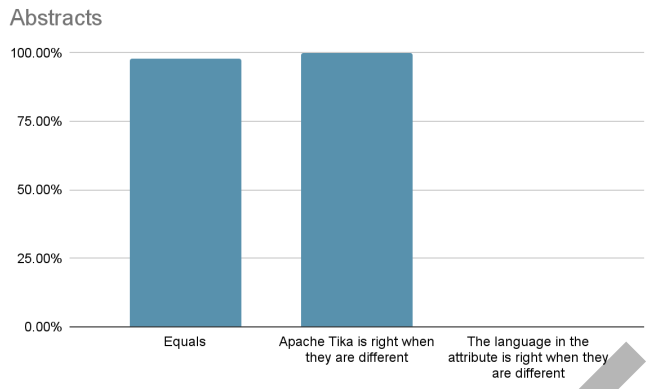


Figure 6 - Comparing language metadata with language detection done with Apache Tika. The case of EKT

This analysis proved that it is always better to detect the language of titles and abstracts through Tika, even when they originally have the "lang" attribute. The latter, as in the case of keywords, is not lost, since the structure to memorise these elements in the GoTriple Elasticsearch index is:

```
{
      text: text,
      lang: language automatically identified by Tika in the ISO-639-1 format
      original_lang: the language (if present) in the original metadata
      translated: boolean (see below)
}
```

The normalisation of titles and abstracts includes the following phases:

- removing all HTML code in them
- removing duplicates
- trimming or trailing blank spaces
- the language is always recognised via Tika and normalised with the vocabulary shown above
- texts in the same language are concatenated to form a single text.

  Example:

  - "lang":"en", "text":"example of English text number 1"

  - "lang":"en", "text":"example of English text number 2"

  Result: "lang":"en", "text":"example of English text number 1 example of English text number 2"

DRAFT

In order to guarantee the widest fruition of GoTriple multilingual data, it has been decided to always provide an English translation for these texts if it is missing from the original metadata. For this purpose the eTranslation service has been used. This service has been selected as one of the third party applications integrated in GoTriple (see [16]): it is a service operated by CEF Digital [17], a public European infrastructure which provides free tools, support and funding for building digital services. It provides good quality machine translation for all the official languages of the EU as well as Ukrainian, Chinese, Russian and Turkish.

After Tika has assessed the language of the title and the abstract and no English versions are available, the eTranslation API is invoked. The title and the abstract in the main language, that is specified in the "in_language" attribute of the publication, are merged by using the following string as a separator:

```
[notranslate]####[/notranslate]
```

This way it is possible to have the translation of both metadata without the need to do separate calls to the API for each one of them.

In order to avoid unnecessary calls to the service, we verify that the available text is not too short, otherwise it cannot be translated properly by eTranslation. After some analysis on previous data imported in TRIPLE, it has been decided to only translate text longer than nine characters: if the abstract is missing or shorter than that, only the title is sent for translation (if its length >= 9).

2.4. Classification and automatic annotation

It consists of two independent services that expose APIs that, given a text, return:

- for classification, one or more "discipline", that is the 27 MORESS categories that have been selected in the TRIPLE project as representatives of the SSH domain
- for automatic annotation, one or more concepts from the TRIPLE Vocabulary [18], a set of over 3.300 SSH entities created in the TRIPLE project.

Both vocabularies are multilingual and support the 11 official main languages of TRIPLE.

The SCRE pipeline merges the headline and the abstract into a single string of text, considering the main language of the document. In case the latter is not amongst those supported by these services, the English translation, original or obtained through eTranslation, is used. Using English is therefore always the backup strategy for classifying and annotating a publication.

It can happen anyway that sometimes it is impossible to assess the language: not always the Tika service (see chapter 2.2.2) is able to identify the idiom of the text, either because the headline and the abstract are uncommon (e.g. they are very short, the headline is not a significant text and the abstract is empty, etc) but also when the language is not supported by Tika (rarely). In this case the text is not sent to the classify and annotate services.

DRAFT

The technical details about the inner mechanism of these services will be described in deliverable D4.4 [1]. We limit here to present the kind of enrichment that they provide and the logic by which the SCRE service will exploit them.

2.4.1. Classify service enrichment

At the time of writing this service supports nine languages: English, French, Spanish, German, Greek, Croatian, Italian, Polish and Portuguese.

It takes as parameters:

- the language of the text, codified in ISO-639-1
- the threshold
- the text to be classified.

By using a machine learning algorithm, the service tries to recognise possible categories to assign to the text, each with a precision score. If the threshold parameter is set, only the categories with a higher score are taken into account. In any case, the service returns only up to two categories.

The categories of the publication are stored in the subject (topic) element of the Elasticsearch Publications index by using the codification shown in the following table.

Category	Codification
Archaeology and Prehistory	archeo
Architecture and Space Management	archi
Art and Art History	art
Biological Anthropology	anthro-bio
Classical Studies	class
Communication Sciences	info
Cultural Heritage and Museology	museo
Demography	demo
Economies and Finances	eco
Education	edu
Environmental studies	envir

DRAFT

Gender Studies	genre
Geography	geo
History	hist
History, Philosophy and Sociology of Sciences	hisphilso
Law	droit
Linguistics	lang
Literature	litt
Management	manag
Methods and Statistics	stat
Musicology and Performing Arts	musiq
Philosophy	phil
Political Science	scipo
Psychology	psy
Religions	relig
Social Anthropology and Ethnology	anthro-se
Sociology	socio

Table 9 - TRIPLE Categories codification

DRAFT

2.4.2. Annotate service enrichment

It is the service in charge of recognizing keywords from the TRIPLE vocabulary in publication. At present, the service is still in beta, supports only the English language and is in the process of being integrated into the SCRE enrichment pipeline.

The service accepts as parameters:

- the language of the text, codified in ISO-639-1
- the text to be annotated.

The response is very rich (see D6.6 [19] for the detailed API description): the service in fact has been developed in order to be very generic and expressive, and therefore potentially reusable in many other contexts beyond the TRIPLE project.

For our present use case, only the "pref_label" and the "uri" elements of the response are taken. For each category its label in the various languages can be found in the pref_label element, whose structure is in fact:

```
"pref_label": [
  {
    "lang": "fr",
    "value": "Anthropologie"
  },
  {
    "lang": "en",
    "value": "Anthropology"
  }
]
```

The "uri", as the name implies, is the "dereferencable" identifier of the category in the TRIPLE vocabulary, e.g.

```
"uri": "http://semantics.gr/authorities/SSH-LCSH/sh85005581"
```

The topic element of the Elasticsearch index will be therefore a multivalue field, in which for each category we will have a structure like the following:

```
knows_about: [
     { //for each keyword
        "uri": "xyz",
        "labels": [
            {
                "label": "in Italian",
                 "lang": "it"
            },
           {
                 "label": "in English",
                 "lang": "en"
           },
           {...}
```

DRAFT

```
            ],
        },
        {...},
    ]
```

2.5. Identification of duplicate publications

The first datasets harvested in TRIPLE showed a small but not negligible percentage of duplicated publications. This happened especially because in the TRIPLE project large aggregators have been integrated, which quite often harvest documents from the same SSH sources.

Identifying duplicates was not an easy task because of the differences normally found in metadata describing the same publications. For example, for a single document we could have different identifiers, in particular different DOIs or authors written in alternative ways (e.g. “Francesca Di Donato, “Di Donato, F.”, “Donato, Francesca Di”, …).

The implemented algorithm, which is also the result of a Bachelor Thesis [20] at the University of Pisa done under Net7’ supervision, tries to identify duplicates by confronting:

- the DOI if present
- the title: we select the title in the main language of the document. Then its text is normalised through: punctuation removal; transformation from Unicode to ASCII, in order to facilitate the comparison of the strings; putting all letters in lowercase; trimming blank characters at the beginning and at the end
- the year of publication
- the number of authors.

As far as the latter point is concerned, it was decided to compare the number of authors and not their names, to avoid false negatives due to different ways of writing or misspelling.

The actual deduplication algorithm will be described in D4.4. Here it is important to describe how the duplicated publications are represented in the GoTriple Elasticsearch Publications index.

The starting point was thinking about the representation of duplicates in the results of a search. An interesting approach is the one used by Google Scholar, which is shown below. Duplicates are in fact grouped as a single document, a “cluster” which shows that there are more versions available in the index for that publication. In the main search interface therefore a cluster appears together with “normal” documents. For each cluster, the number of the duplicates associated with it is shown: by clicking this number it is possible to “expand” and see all the duplicates that can be accessed independently.

DRAFT

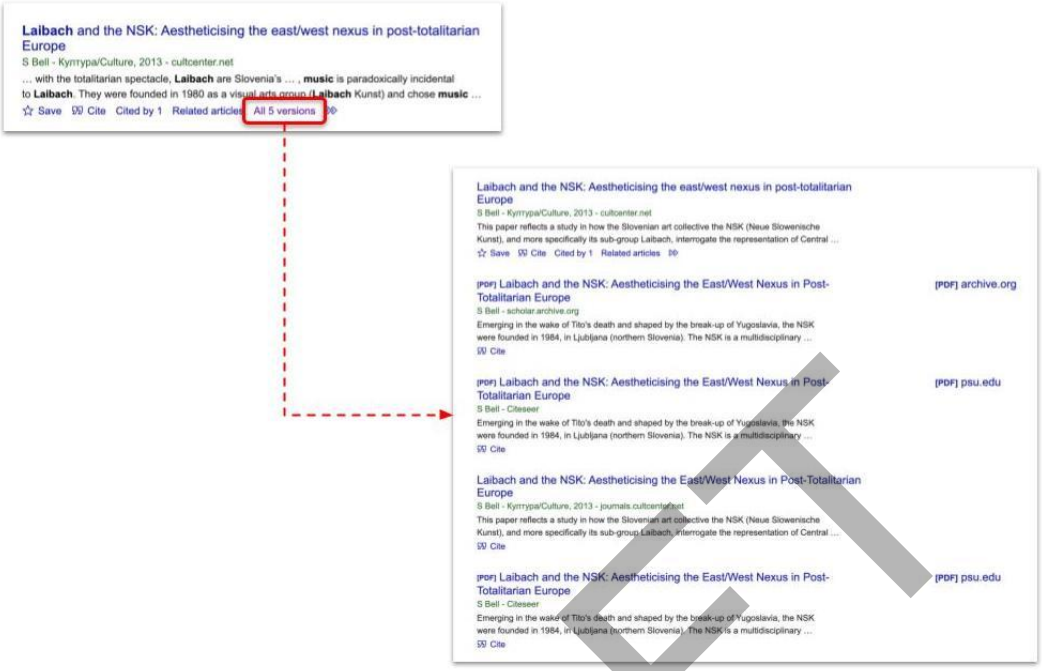


Figure 7 - How Google Scholar deals with duplicates

Technically speaking a cluster is not a real document but a representation of the union of all the duplicate documents it represents. Since the original TRIPLE data model, presented in chapter 2.1, doesn't take into account this concept, it was decided to expand it by introducing the notion of "cluster" documents. The extra elements aptly added to the model are presented and described in the table below.

New elements in the TRIPLE Data Model	Description
cluster_id	The identifier of the cluster. This field is not null only for clusters and duplicated documents.
is_cluster	A boolean field, TRUE for clusters, NULL otherwise
is_duplicate	A boolean field, TRUE for duplicate documents, NULL otherwise
cluster_children_count	Only available for clusters: it stores the number of duplicate documents the cluster represents

Table 10 - Expansion of the TRIPLE data model

DRAFT

Since clusters are an aggregation of multiple documents, some of its fields are built by merging the values of the single duplicates: think about fields as identifier, publisher, aggregator, URL of the landing page or URL of the full document.

After a user searches for a keyword we expect that (s)he receives both “normal” documents and clusters on the same page, as Google Scholar does. Therefore we had to change the search API by introducing an extra condition to filter out duplicates (“is_duplicate” NOT TRUE) in the main search but at the same time to retrieve all the information that allow to understand if the result is a cluster (“is_cluster” TRUE) and to present it in the proper way (“cluster_id” and “cluster_children_count” to show how many duplicates are there).

Then by clicking on the cluster the user is presented with the list of the duplicates, obtained by calling the search API by specifying this time to only retrieve duplicates (“is_duplicate” TRUE) belonging to the same cluster, indicated by the “cluster_id”.

2.6. Authors disambiguation

Processing publications implies at the same time to manage “Authors”. In fact the Profiles index is by default automatically populated by extracting authors from the publications metadata. This strategy, which seems quite simple and straight-forward, poses in reality a lot of problems, including:

- it is not easy to recognise “real persons”: sometimes in fact we have as authors things like “Department of Computer Science” or “ACM Conference 2021”;
- a single person might be spelled in a different way, e.g. “Suzanne Dumouchel”, “Dumouchel, Suzanne”, “Dumouchel, S.”;
- there might be homonyms among authors.

The disambiguation procedure aims at solving the latest two points: it has been inspired by some previous experiences ([21], [22]) and it is based on a set of rules which take into account, given a publication:

- the name of every single author and its possible variants (e.g. “Dumouchel, Suzanne” -> “Suzanne Dumouchel”, “Dumouchel, S.”)
- the year of the publication
- the publisher, which often represents the Research Centre the author is associated with
- the keywords of the publication.

By confronting all these elements, through a heuristic, the procedure tries to determine, given an author if it consists of a duplicate of an existing one. If this is the case, the new one is an alias, or an “AKA” as we say, of the former and the links between their publications are updated accordingly.

As seen above, the procedure tries to assess if the author is the same not only using the name as a criteria. For example, we might have two homonyms, publishing articles in a different

DRAFT

timespan which are about different topics and therefore they are recognised as different persons. At the same time, a variant of a name (e.g. “Dumouchel, Suzanne”, “Dumouchel, S.”) might identify the same author if the other criterias match, for example, if the publications are all in the same temporal range and several of the keywords match.

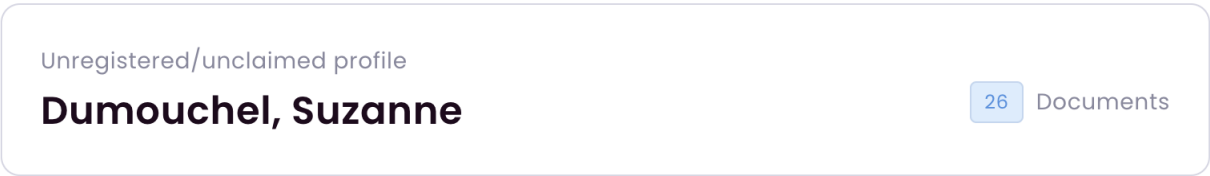
This disambiguation can be therefore considered as an enrichment of the original metadata produced in the TRIPLE project.

The Elasticsearch schema of the Profile index, which takes into account duplicates, is presented and explained in the table that follows.

Elasticsearch field	Description
id	the identificator of the author on the Elasticsearch index, which is automatically calculated. It is based on the normalised name of the author (e.g. spaces are replaced by “_”, accented letters are replaced by their unaccented correspondents, etc) merged with a random string of 21 characters created by the Nano ID library[2]. For example: “César De Santis" can have as id “cesar_de_santis_l6FfvrDgWBlxpw8Ve9U5I”
fullname	The complete name of the author as received in the publication
AKA	This field is only present for duplicates: it contains the id of the author recognised as “original”
author_of	The list of the documents IDs in the Publications Elasticsearch index which are attributed to the author

Table 11 - Fields of the Profiles Elasticsearch index for the management of publications’ authors

Finally, it is important to point out that this portion of the Profiles index schema only refers to automatically assessed authors, those that in the GoTriple search results are presented as “Unregistered/unclaimed” profiles (see image below).



[2] Nano ID [23] is a library for generating random IDs which have a tiny probability of generating duplicates.

DRAFT

Figure 8 - An example of an unclaimed profile in GoTriple

The management of registered users of the GoTriple platform, whose attributes are stored in the same index, is not considered herein since it doesn't represent a case of data enrichment.

3. Projects processing

Project data has limited support in GoTriple, since it is not easy to find data sources to harvest SSH related initiatives. It has been decided therefore to import data only from:

- CORDIS: data about EU funded projects, related to the SSH domain, under the FP7 [24], H2020 [25] and Horizon Europe [26] research work programmes
- OPERAS Crowdfunding projects [27], implemented as a dedicated channel of the WeMakeIt platform
- OPERAS' COESO projects [28]: citizen science projects managed through the VERA platform, currently under development and not yet operational.

At the time of writing only projects from CORDIS have been imported in TRIPLE. The strategy decided in this case has been generalised and will be applied also as soon as it is possible to harvest data from the other two sources.

In particular, instead of developing a dedicated connector for each data source, a generic project file format import service has been implemented.

It is configured by simply specifying the directory to import data from and the frequency of the import, which in the case of the CORDIS project is very low, as these datasets either regard closed work programmes (FP7, H2020) or are not updated very frequently (normally once a month).

We assume therefore that all input files will have a definite JSON format which reflects the TRIPLE Projects data model presented in the table that follows. If a file doesn't conform to the schema it will be discarded and the corresponding error logged.

Field	Description	Class
Identifier	Official identifier of the project	schema:identifier
Name	Name of the project	schema:name
Alternate name	Acronym or other name(s) of the project	schema:alternateName
Description	Project description and objectives	schema:description
Start date	Start date of the project	schema:startDate
End date	End date of the project	schema:endDate

DRAFT

Organization	Coordinating entity (eg: CNRS-HN)	schema:organization
Funder	Funder of the project (eg: European Commission)	schema:funder
Funding type	Type of grant (eg: H2020)	schema:fundingScheme
Crowdfunding information	Crowdfunding or agency (can be empty)	schema:sponsor
Keywords		schema:keywords
Subject (MORESS categories)	MORESS	sioc:topic
TRIPLE thesaurus	TRIPLE thesaurus	schema:knowsAbout
URL of the project		schema:URL

Table 12 - TRIPLE Projects Data Model

In the case of CORDIS, an ad-hoc procedure has been implemented to produce the data JSON file for the import in SCRE.

It starts by retrieving the specific CORDIS dataset (for FP7, H2020 and Horizon Europe projects) in zip format: then it expands it, analyses the data files in Excel format (.xlsx) to filter only those regarding SSH projects and finally produces the JSON file in the right format for SCRE to import.

Not all the files in a CORDIS dump are processed: we only take in consideration:

- euroSciVoc.xlsx, which contains the classification of the projects according to the European Science Vocabulary (EuroSciVoc) [29]. This file is analysed first in order to obtain the identifiers of only those projects regarding the SSH domain.
- project.xlsx, which contains the description of the project
- organization.xlsx to retrieve the information about the Coordinating Entity.

The "funder" for CORDIS is always the European Commission so the value of this field is inserted directly as a constant, while for "funding type" the name of the specific workprogramme is taken.

A mapping between the EuroSciVoc entries of the "Humanities" and "Social Sciences" domains and the TRIPLE Disciplines has been created, where for one entry of EuroSciVoc there is one, and sometimes two or three, corresponding TRIPLE Disciplines. The mapping is presented in the table that follows.

CORDIS EuroSciVoc	Triple Discipline	Extra TRIPLE discipline
humanities/arts/architectural design	Architecture and space management	Art and art history
humanities/arts/art history	Art and art history	

DRAFT

humanities/arts/modern and contemporary art	Art and art history	
humanities/arts/musicology	Musicology and performing arts	Art and art history
humanities/arts/performing arts	Musicology and performing arts	Art and art history
humanities/arts/visual arts	Art and art history	
humanities/history and archaeology/archaeology	Archaeology and Prehistory	History, Classical studies
humanities/history and archaeology/history	History	
humanities/history and archeology/history/prehistory	Archaeology and Prehistory	History, Classical studies
humanities/languages and literature/general language studies	Linguistics	
humanities/languages and literature/linguistics	Linguistics	
humanities/languages and literature/literature studies	Literature	
humanities/other humanities/library sciences	Communication sciences	
humanities/philosophy, ethics and religion/ethics	Philosophy	Religions
humanities/philosophy, ethics and religion/philosophy	Philosophy	
humanities/philosophy, ethics and religion/religions	Religions	
social sciences/economic and business/business and management	Management	Economies and finances
social sciences/economic and business/economics	Economies and finances	
social sciences/educational sciences/didactics	Education	
social sciences/educational sciences/inclusive education	Education	
social sciences/educational sciences/pedagogy	Education	
social sciences/educational sciences/special education	Education	
social sciences/law/admirality law	Law	

DRAFT

social sciences/law/constitutional law	Law	
social sciences/law/criminology	Law	
social sciences/law/human rights	Law	
social sciences/law/international law	Law	
social sciences/law/law enforcement	Law	
social sciences/law/penology	Law	
social sciences/media and comunications/graphic design	Communication sciences	
social sciences/media and comunications/journalism	Communication sciences	
other social sciences	Sociology	
social sciences/political sciences/government systems	Political science	
social sciences/political sciences/political communication	Political science	
social sciences/political sciences/political policies	Political science	
social sciences/political sciences/political transitions	Political science	
social sciences/political sciences/public administration	Political science	
social sciences/psychology/behavioural psychology	Psychology	
social sciences/psychology/cognitive psychology	Psychology	
social sciences/psychology/developmental psychology	Psychology	
social sciences/psychology/ergonomics	Psychology	
social sciences/psychology/phycholinguistics	Psychology	
social sciences/psychology/psychotherapy	Psychology	
social sciences/psychology/social psychology	Psychology	Sociology
social sciences/social geography/cultural and economic geography	Geography	
social sciences/social geography/transport	Geography	
social sciences/social geography/urban studies	Geography	

DRAFT

social sciences/sociology/anthropology	Social Anthropology and ethnology	Sociology
social sciences/sociology/demography	Demography	Sociology
social sciences/sociology/family studies	Sociology	
social sciences/sociology/gender studies	Gender studies	Sociology
social sciences/sociology/globalization	Sociology	
social sciences/sociology/governance	Sociology	
social sciences/sociology/ideologies	Sociology	
social sciences/sociology/industrial relations	Sociology	
social sciences/sociology/social issues	Sociology	
social sciences/sociology/social work	Sociology	

Table 13 - Mapping from CORDIS project taxonomy to the TRIPLE Disciplines

Finally, each element of the EuroSciVoc category "hierarchy" is taken as a keyword of the project, together with the name of the EU call where the project has been funded (e.g. "FP7-PEOPLE-IEF-2008").

For example, a CORDIS project classified as "social sciences/psychology/psychotherapy", funded under the "ERC-2016-STG" call, has keywords "social sciences", "psychology", "psychotherapy" and "ERC-2016-STG".

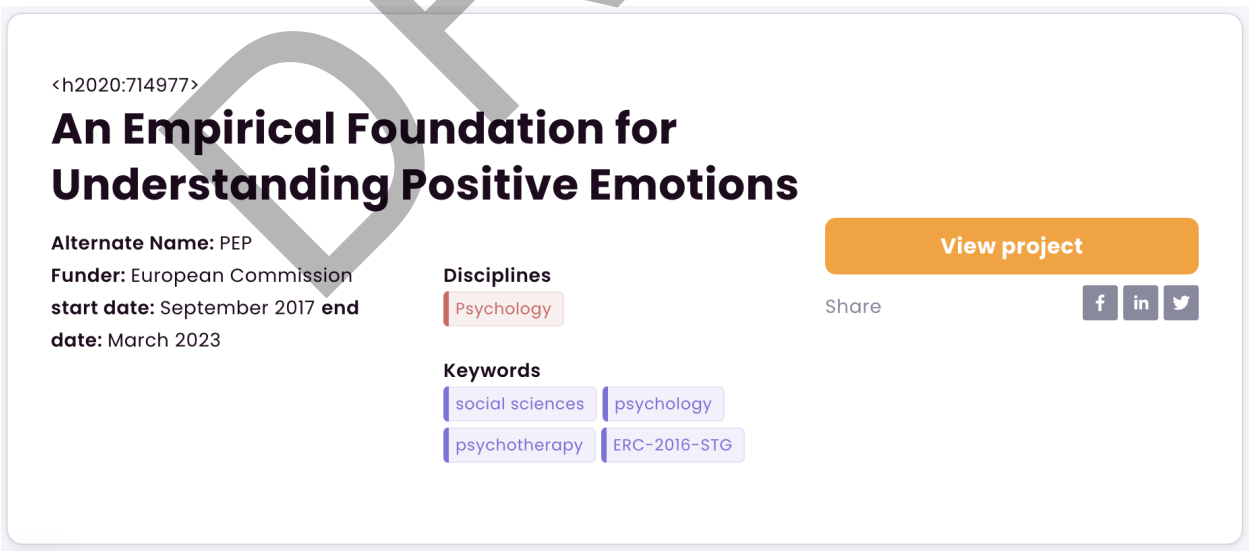


Figure 9 - Keywords of a CORDIS project

DRAFT

4. Conclusions

The strategies for data cleaning and enrichment in TRIPLE that have been presented herein are the result of an iterative process.

Metadata retrieved by the SCRE pipeline in the initial phases of development of GoTriple have been analysed and difficult cases identified. Every proposed solution has been evaluated in collaboration with WP2 colleagues, whose experience in similar contexts proved strategic to assess the quality of the proposed choices or led to alternative and more effective approaches.

The data normalisation process in fact proved anything but straight-forward: several problems have been faced as long as new data sources were added to the platform. Issues within data quality of the original metadata, mismatch with the TRIPLE data model, the use of free textual strings for structural attributes, custom extensions of standards by data providers, are just some of the difficulties that have been met and to which a possible solution has been proposed and implemented in SCRE, as it is described in the previous chapters.

At the time of writing GoTriple has ingested more than 4 million publications and 21.000 projects: data acquisition will continue in these ending months of the project and, also, beyond its conclusion. Discussions with data providers, being large aggregators like BASE [30] and Europeana [31] or small repositories alike, are ongoing and will lead to a significant growth of data in GoTriple.

The result of the strategies described herein will be checked when ingesting new data and, in case, the needed corrections will be applied, in order to always guarantee the possible best quality of the information that GoTriple can offer to the users.

DRAFT

5. References

[1] D4.4 - Technical and User Documentation for the TRIPLE system

[2] Apache Camel – https://camel.apache.org/

[3] Elasticsearch - https://www.elastic.co/elasticsearch/

[4] OAI-PMH Protocol - http://www.openarchives.org/OAI/openarchivesprotocol.html

[5] OpenAIRE - https://www.openaire.eu/

[6] Wikipedia description of OpenAIRE - https://en.wikipedia.org/wiki/Framework_Programmes_for_Research_and_Technological_Development#OpenAIRE

[7] Baglioni, M. et al: "OpenAIRE Research Graph: Json schemas of the dump" https://zenodo.org/record/5799514/#.YwRkK-xBw40

[8] Isidore - https://isidore.science/

[9] DCMI Metadata Terms - https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[10] ISO 8601 - https://www.iso.org/standard/70907.html

[11] ISO 639-1 - https://www.iso.org/standard/22109.html

[12] COAR Resource Types Vocabulary - https://vocabularies.coar-repositories.org/resource_types/

[13] JUnidecode - https://github.com/gcardone/junidecode

[14] Apache Tika - https://tika.apache.org/

[15] Lingua, language detection library - https://github.com/pemistahl/lingua

[16] TRIPLE Deliverable "D5.1 - Report on Third-Party Applications Integration" - https://zenodo.org/record/5702399

[17] CEF Digital - https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home

[18] TRIPLE Vocabulary: an SSH multilingual vocabulary based in LCSH - https://www.semantics.gr/authorities/vocabularies/SSH-LCSH/?language=en

[19] D6.6 - TRIPLE_D6.6 APIs development-RP3

[20] M. Velardita. "Sviluppo di un servizio di de-duplicazione di pubblicazioni scientifiche", Bachelor Thesis, Computer Science Department of University of Pisa, 2022

[21] Staša Milojević. "Accuracy of simple, initials-based methods for author name disambiguation", Journal of Informetrics vol. 7, n. 4, 2013

[22] D'Angelo, C.A., van Eck, N.J. "Collecting large-scale publication data at the level of individual researchers: A practical proposal for author name disambiguation", Scientometrics, 2020

[23] Nano ID - https://zelark.github.io/nano-id-cc/

[24] CORDIS - EU research projects under FP7 (2007-2013) - https://data.europa.eu/data/datasets/cordisfp7projects?locale=en

DRAFT

[25] CORDIS - EU research projects under Horizon 2020 (2014-2020) - https://data.europa.eu/data/datasets/cordish2020projects?locale=en
[26] CORDIS - EU research projects under HORIZON EUROPE - https://data.europa.eu/data/datasets/cordis-eu-research-projects-under-horizon-europe-2021-2027?locale=en
[27] OPERAS Crowdfunding projects - https://wemakeit.com/channels/operas
[28] COESO Project - https://coeso.hypotheses.org/
[29] EuroSciVoc - https://op.europa.eu/en/web/eu-vocabularies/euroscivoc
[30] BASE - https://www.base-search.net/
[31] Europeana - https://www.europeana.eu/

DRAFT