Strengthening European integration through the analysis of conflict discourses

Revisiting the Past, Anticipating the Future

15 September 2019

# RePAST Deliverable D6.8

## RePAST Data Platform

*Constantinos Djouvas, Fernando Mendez and Vasiliki Triga*

*(with the contribution of country leaders)*

*Cyprus University of Technology*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769252

*Project information*

**Grant agreement no**: 769252

**Acronym**: RePAST

**Title**: Strengthening European integration through the analysis of conflict discourses: revisiting the past, anticipating the future

**Start date**: May 2018

**Duration**: 36 months

**Website**: www.repast.eu

*Deliverable information*

**Deliverable number and name**: D6.8 RePAST Data Platform

**Work Package**: WP6

**Lead Beneficiary**: CUT

**Version**: 1.6

**Authors**: Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (with the contribution of country leaders)

**Submission due month**: July 2019

**Actual submission date**: 15 September 2019

**Dissemination level**: Websites, patents filling, etc.

**Dissemination level**: Public

**Status**: Submitted

| Document history | | | | | |
|---|---|---|---|---|---|
| **Version** | **Date** | **Author(s) / Organisation** | **Status** | **Description** | **Distribution** |
| 1.1 | 08/07/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | draft | First draft for review | RePAST cloud folder |
| 1.2 | 15/07/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | 2nd draft | Second draft | RePAST cloud folder |
| 1.3 | 16/07/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | Final draft for review | Final draft for review | RePAST cloud folder |
| 1.4 | 18/07/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | Draft after comments by three internal reviewers | New draft for review | RePAST cloud folder |
| 1.5 | 29/07/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | Second draft after reiterations among the authors | Final draft for review before submission | RePAST cloud folder |
| 1.6 | 14/09/2019 | Constantinos Djouvas, Fernando Mendez and Vasiliki Triga (CUT) | Final after review by the Coordinator and the EE | Final version | RePAST cloud folder |

Peer reviewed by:

| **Partner/Body** | **Reviewer** |
|---|---|
| RePAST Consortium | Dimitra L. Milioni (Coordinator), CUT |
| RePAST Consortium | Irene Martín Cortés, UAM |
| RePAST Consortium | Eugenia Siapera |
| Ethics Expert / Data Protection Officer | Ljubica Pendaroska |

# Table of Contents

## Executive Summary

This report provides a technical description of the RePAST Data Platform. The latter has a three-fold layered structure based on: (1) A processing layer, which harmonises and augments input data from the various work packages into a common JSON structure. This layer is also the site for the application of textual analysis techniques, such as word-embedding algorithms that map document associations among the RePAST corpus. An in-house Python program is at the core of this layer. (2) A storage layer, which houses the multiple data structures and is indexed to facilitated seamless searches across different structures. This layer is underpinned by Elasticsearch technology that is both lean and highly scalable. (3) A presentation layer, which is where the end users interact with the RePAST Data Platform. It is designed to allow the end users to navigate across the different data structures and shape the way in which the material is presented. The visualisation is powered by Kibana technology. The report describes how the Data Platform was designed according to the above multilayered architecture, its implementation status and how technical challenges have been addressed.

## 1. Introduction

At the core of the RePAST project sits the so-called Data Platform, an online, open, highly interactive web application with multiple functionalities that aims at engaging a wide range of users, from the academic community and civil society organisations to the general public. The ultimate aim of the platform is to provide the opportunity to its users for an intergenerational, interethnic or inter-communal dialogue. This is to be achieved based on two main modules: a) interactive storytelling and b) textual analysis of contested pasts. More specifically, interactive storytelling is a concept that is based on creating a repository of data, which is drawn from two main sources: a) primary interview material from interviews conducted within WP2 (Oral history), collected material within WP3 (Media), WP4 (Arts and Culture) and WP5 (Political Discourses), and b) secondary material from existing databases, mainly Europeana Collections. These types of data are organized in the platform's database after being processed, 'tagged' and classified in thematic categories to allow its users to select, combine and compare narratives and perspectives around significant milestones of contested pasts as well as construct their own storytelling projects. The second module, namely the textual analysis of contested pasts, is based on an embedded-in-the-platform tool for textual analysis of data that derives from the RePAST research, such as focus groups transcripts (WP3 and WP5), political parties excerpts (WP5), for single- and multiple-case analysis to enable cross-country comparisons. The textual analysis tool is based on existing tools of text and content analysis using open source technologies. The functionality of this tool through the RePAST data platform enables interested researchers to conduct their own analysis on collective memory and conflict.

While the overall goals of the two modules are briefly outlined here, the technical aspects and the challenges involved are described in detail in the present report. Implementing such a platform requires different innovations mainly for dealing with the heterogeneity and structureless of the data, the lack of topic-specific tools for textual analysis, and the implementation of a highly interactive platform that presents all the different types of data in an aggregated and intuitive manner.

This report presents in detail the overall architecture of the Data Platform. In addition to the online aspects of the platform, i.e. the database and the web app, the individual modules implemented for the offline processing of the different datasets hosted at the platform are also presented. Furthermore, the report demonstrates the current status of the platform along with the steps that are next required for its completion.

## 2. Relation to work packages

Although this deliverable belongs to WP6 (Dissemination, Innovation and Policy Recommendations), it is closely related to and depends upon WPs 2 to 5. This is because WP2 through to WP5 address the various conflict discourses rooted in troubled pasts that RePAST investigates (Oral History, Media, Art and Culture, and Political Discourse respectively). Each of

these WPs produce data of different types and formats that is processed, analyzed, uploaded, and finally graphically presented on the platform.

As a result, close coordination is required for the correct and timely delivery of the various datasets to be hosted in the Data Platform. However, since the different materials will be delivered at different times, tools that completely automate the various procedures need to be implemented for processing and uploading the data to the platform. Besides, given that the Data Platform aims to be preserved and updated in the future and beyond the duration of RePAST, this kind of functionality is crucial. Also, finetuning the functionalities of the platform (e.g., creating common fields for mapping items across collections) as well as the appearance of the platform will be an iterative process that can only be completed once all the material is available for processing and uploading.

## 3. Architecture

The implementation of the RePAST Data Platform requires a multi-layer architecture. More specifically, a three-layer architecture consisting of a processing, a storage and a presentation layer is implemented (Figure 1). Bellow, we present the purpose, the functionality, and the components of each layer.

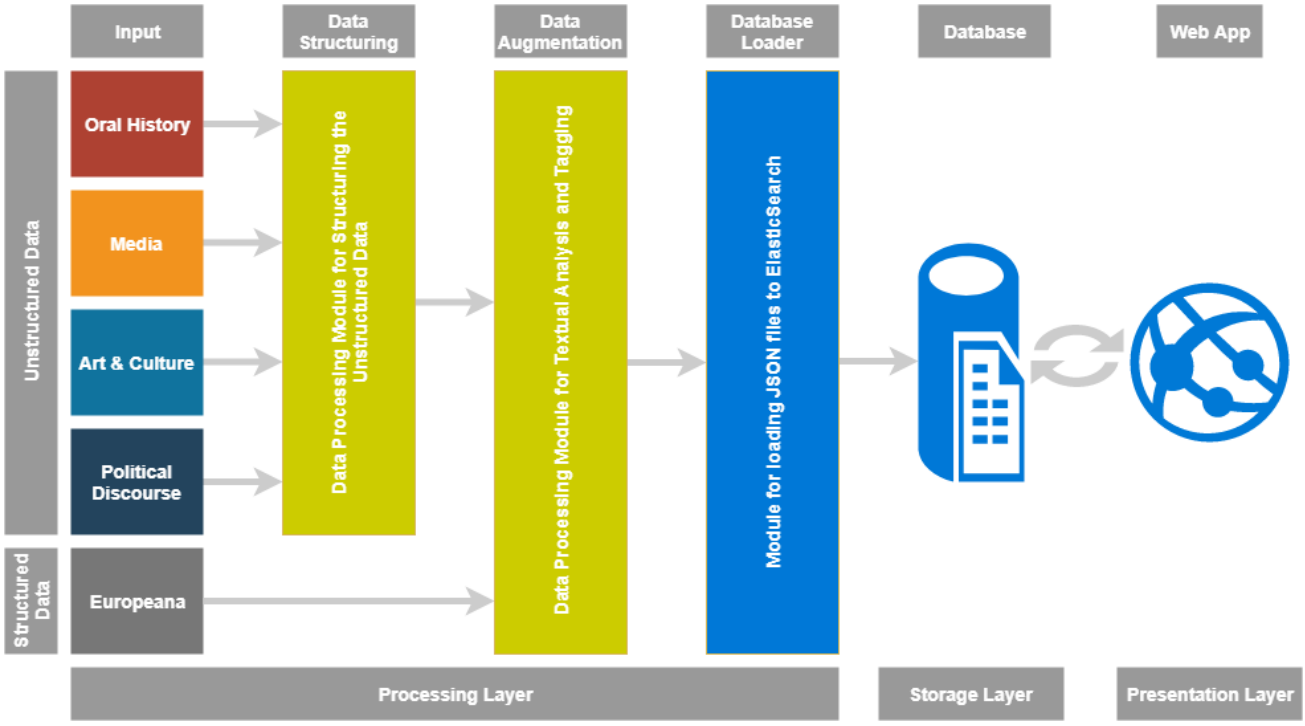



*Figure 1. RePAST Data Platform Architecture*

## 3.1 Processing Layer

As depicted in Figure 1, the data to be inputted to the platform consists of five distinct input materials. All these diverse materials will be stored into a database, which mandates well-defined structured datasets. For example, a word document does not have any inherited structure, since as far as a database is concerned, a word document is a single (maybe long) piece of text. On the contrary, CSV or JSON[1] formatted data are structured and thus they can be processed and directly loaded into a database.

The main purpose of the processing layer is the transformation of the different data derived from WPs 2 to 5 into a well-formatted and structured form, so that they can be processed and loaded into the database. First, input materials are split into two categories, structured and unstructured. Starting from the former, structured materials are extracted from the Europeana repository in JSON format. Europeana repository is an extremely rich database of cultural material in relation to important historical periods of the past for every European country. The goal is to identify and select solely the material concerning the periods of troubled pasts that is relevant for every case study. By filtering the Europeana material based on the actual period of the troubled past for every case study and by importing it into the RePAST data platform, any related search can be further enriched with this respective material. Thus, this data can be processed by the next module in the proposed pipeline, the Data Augmentation module, without any pre-processing. To be more precise, every Europeana 'object' (e.g. a photo from the 1974 events in Cyprus) can emerge and be accessed in a related search to a specific country and/or trouble past as additional material that the user can use along with the primary data derived from RePAST research.

The rest of materials (those derived from RePAST researchers - Oral History, Media, Art & Culture, and Political Discourse) are created by different research groups. Not surprisingly, they are collected and processed using different techniques and formats that exhibit high variability. Data with such characteristics are impossible to process in the absence of harmonisation. Thus, the first module of the Processing Layer deals with this issue, transforming all the different data into well-structured and consistent objects represented as JSON objects.

Having transformed all the data into well-formatted JSON objects, the next module of the Processing Layer, the Data Augmentation module, processes all the objects using a two-step process. First, all data entries are augmented with different labels, e.g., country, conflict type, etc. The purpose of this augmentation is the creation of common fields among data types that will facilitate both inter- (entries of the same data type) and intra- (entries across data types) dataset entry associations. The labelling process is incremental by starting using some general labels that create links between the various types of data and the case studies. Once more data are inserted

[1] JSON is an open-standard file format that uses human-readable text to represent data objects consisting of key–value pairs. One can think that as a flatten CSV file, were column names become the keys and corresponding row values become the values.

into the platform, the labels will be expanded with the goal of producing the largest possible number of common labels that will apply to all data. For example, a criterion of ordering data can be the basis of experience (direct-transmitted) of participants in the interviews of oral history. In addition, a series of other metadata will be added such as age, gender etc. Presently, since a small part of the data is inserted in the platform, the architecture that is implemented is designed in such a way that can accommodate any number of additional labels.

The second step of the process deals with the textual analysis, which is a functionality offered to the platform's users, both citizens and researchers. The analysis will be undertaken by embedded tools that would work offline with the purpose to identify similarities between the selected documents under analysis. For this purpose, we will utilize 'word embedding techniques' for analyzing and then associating (in terms of inter-document similarity) the different entries. Word embedding techniques create a multi-dimensional space where each word in a corpus represents one dimension in the multi-dimensional space – we refer to this representation as a model. Then, utilizing the model, each document in a corpus, based on the words it contains, is represented as a vector in the multi-dimensional space. Having such a representation the problem of calculating the distance among two documents is reduced to the problem of calculating the distance between two vectors in the multi-dimensional space. This can be done using different techniques (e.g., Euclidean Distance, Angles, etc.). In our case, and despite the fact that one can find a number of pretrained models online, we will utilize the Doc2VEC (Le, 2014) algorithm, in order to create a model specifically designed for the domain of the project, and thus increase its accuracy.

Utilizing the aforementioned model, one can calculate the distance among any pair of documents. In our case, however, we should be able to rank several documents using a single similarity metric and not just be able to find similarities among pairs of documents. To overcome this, we draw upon Rossiello work (Gaetano Rossiello, 2017) where first the centroid of the model is calculated (i.e., the 'average' document) and then the distance of each document from the centroid of the model is calculated. For example, if our corpus consists of ten documents, we will first create a model using Doc2VEC for representing the entire corpus, then calculate the 'average' document, and finally, for each document, calculate its distance from the 'average' document. This will provide a single distance value for each document (using as a pivot point the 'average' document) that can be used for identifying similarities among all the documents of the corpus.

The final module of this layer pipeline deals with database data loading, i.e., loads all the processed datasets into the platform's database. For this purpose, a tailormade program capable of processing the JSON objects produced by the Data Augmentation module will be implemented.

## 3.2 Storage Layer

All the data collected by the RePAST researchers and processed by the Processing Module will be loaded into a database. The database is designed to serve the analytical needs of the data platform, yet at the same time is in line with the internal database described in D3.2 for purposes of compatibility. Each data type should be loaded into a designated table and the database should support multiple indexes to facilitate unlimited searches across different tables.

### 3.3. Presentation Layer

The presentation layer is the layer that end-users will be utilizing for interacting with the system. It should facilitate interactive storytelling through an intuitive and highly interactive interface. To achieve this, the Presentation Layer will allow users to visually navigate through the augmented data, allowing them to shape the data in a way that can be convenient to them. To provide an example, in case a user wishes to find material that is related to a less known aspect of the country's troubled past he/she might be interested in (e.g. the role of women in the Cyprus problem), he/she will be able to perform a search in the material and filter it out by accessing information in data derived from primary sources, enriched by related visual material from the Europeana collection. This way the user can combine information at various levels (e.g. lay discourses, political parties, art, media) to be able to re-construct extended narratives on the topic at stake and in addition he/she might go beyond his/her initial goal and be able to compare the case (Cyprus) with other similar ones (if any) through the searches facilitated by the data platform functionalities.

## 4. Technical Challenges

The nature of the project poses some unique challenges that need to be solved for different aspects of the proposed architecture. More precisely, solutions to deal with the following issues have to be devised:

*1. Unstructured to structured data transformation:* Inevitably, given the interdisciplinary nature of the research team, data are collected using different techniques. This results not only in data in different formats (e.g., Word Documents, Excel Documents, etc.), but also in data that lack a common structure (e.g., Word Documents with arbitrary sequence of entries, or missing entries). This is highly problematic for both algorithms and databases. Thus, techniques that transform the unstructured data provided by the researchers into a harmonized form need to be developed. Such techniques involve a common labelling system (that was mentioned previously) while more technical details are provided in section 5 below.

*2. Highly heterogeneous materials collected across different WPs of the project (Oral History, Media, Art & Culture, and Political Discourse):* As mentioned previously, the challenge that needs to be addressed concerning the degree of heterogeneity of the data generated by RePAST WPs is to develop a harmonization technique by adding some common fields (e.g., country, conflict type, etc.). These fields will be identified by the RePAST research team so that they more accurately represent the association among data types. It is noteworthy that the data deriving from the various WPs are forwarded to the data platform after having been coded and analyzed based on the respective analytical techniques by each WP separately.

*3. Lack of topic-specific textual analysis tool:* Generally, state-of-the-art textual analysis tools are based on machine learning. However, such techniques require models trained using domain-specific data. In our case, models for the domain of troubled past do not exist, thus, models need to be trained in-house.

## 5. Design and Implementation

In section 3 we presented the overall architecture of the RePAST Data Platform. In this section, we present the technologies used for the implementation of the aforementioned architecture along with the design decisions made for overcoming the challenges mentioned in section 4.

Starting from the Processing Layer, different Python programs are implemented. The first module of the layer is responsible for transforming a set of input data to JSON objects (one for each entry of each data type) to the next module of the pipeline, the Data Augmentation module. Input data, as depicted in the architecture section, are split into five categories, four derived from the RePAST researchers, and one from the Europeana API. For the first four data categories (Oral History, Media, Art & Culture, and Political Discourse) the objective was to transform the unstructured input files into well-defined structured data. For this purpose, four parsers (one for each data category) are implemented in Python. Depending on the format of the input file (e.g., Word Document, Excel Document, etc.), a different approach is adopted. For the last data category, i.e. Europeana API, a Python program implementing the Europeana Entity API (Europeana, 2019) is implemented. Data extracted from Europeana API are stored locally in JSON format.

The second module in the pipeline, i.e. the Data Augmentation module, deals with the processing of JSON files, implementing a two-step process. First, it creates associations among entries by adding common fields. This functionality is implemented by designated Python programs, each one handling a different data type. Then, a single Python program extending the GENSIM (Radim Řehůřek, 2010) library performs the textual analysis on the data. The outcome of this analysis is then stored in each JSON object. Finally, after the completion of Data Augmentation module, the next and final step of the Processing Layer is to load the processed data into a database. This was also implemented using Python, and more precisely with a parameterized program capable of dealing with all the different data types, loading them into designated indices[2], one for each different data type.

The Storage Layer is implemented using Elasticsearch. One can think of Elasticsearch as a regular database, however Elasticsearch is a search engine (i.e., an information retrieval system designed to extract stored information though arbitrary queries). Elasticsearch is preferred over other, more traditional, databases in order to facilitate fast, reliable, and scalable arbitrary searches across different indexes.

Finally, the Visualization Layer is built using Kibana (Kibana, 2019). Kibana is a data visualization web application with extended visualization capabilities built on top of the data extracted from Elasticsearch. An example of the current implementation of the RePAST Data Platform visualization is depicted in Figures 2 and 3 below.

[2] Indices are logical units of Elasticsearch (the 'database' schema we are using) inside which data are stored.

***Figure 2. A map representation of a search based on data related to WP4 (Arts and Culture). The map shows the countries in which the respective search generated results.***

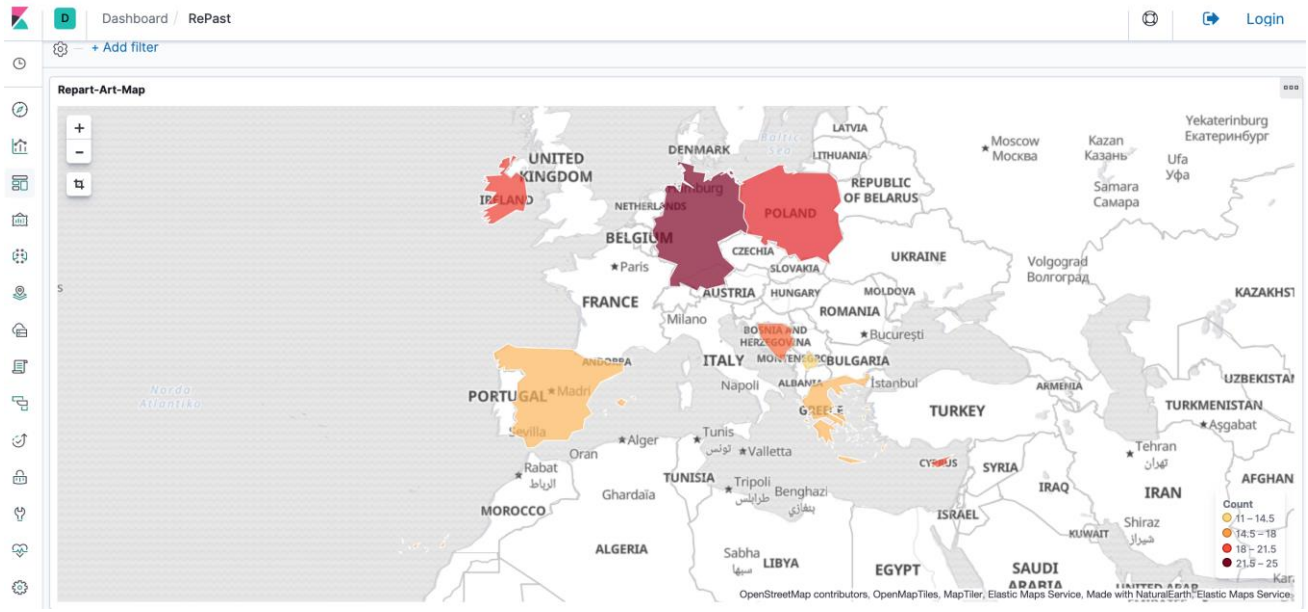
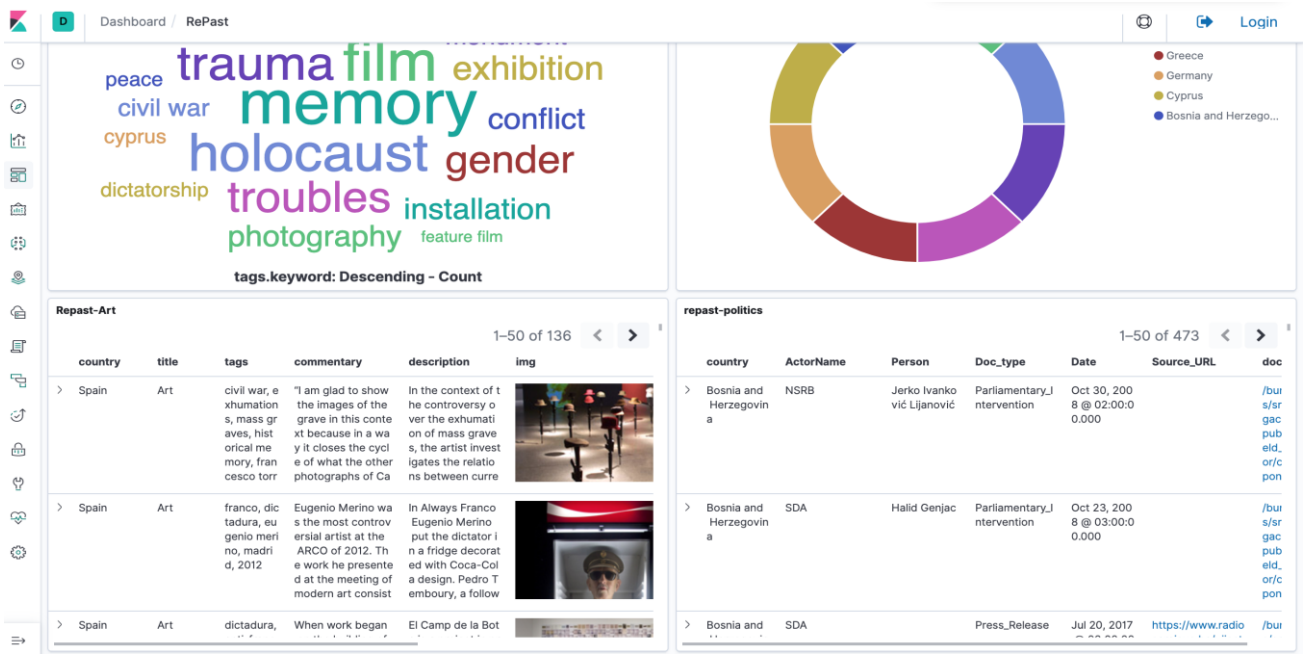



***Figure 3. The figure shows a series of data visualization results based on searches in relation to Arts and Political discourses such as a word cloud, a pie chart representing the proportion of data per country as well as synoptic tables in relation to art and political discourses related data including all the relevant details (e.g. source, url, date etc.).***



## 6. Next Steps

All the algorithms of the abovementioned architecture are implemented. However, the data collection process in some WPs is still ongoing at the time of writing this report. Thus, we are still missing various data to be processed and uploaded to the platform. Even though adding new data is a straightforward process, given that the tools automating the process have been designed and implemented, the new data will affect two modules of the proposed architecture. First, new data will change the model that was trained for textual analysis. To solve this, the model will be retrained, and all documents will be reclassified. Second, the new data might reveal some limitations regarding the embedded analytical applications. If this occurs, all necessary changes will be implemented to provide the best possible experience to the end users.

## References

*Elasticsearch*. (2019, 07 11). Retrieved from Elasticsearch: https://www.elastic.co/products/elasticsearch

*Europeana*. (2019, 07 11). Retrieved from https://pro.europeana.eu/resources/apis/entity

Gaetano Rossiello, P. B. (2017). Centroid-based Text Summarization through Compositionality of Word Embeddings. *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres* (pp. 12-21). Valencia, Spain: Association for Computational Linguistics.

*Kibana*. (2019, 07 11). Retrieved from Kibana: https://www.elastic.co/products/kibana

Le, Q. a. (2014). Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning* (pp. II-1188--II-1196). Beijing, China: JMLR.org.

Radim Řehůřek, P. S. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New*, (pp. 45 - 50). Valletta, MaltaELRA.

www.repast.eu