EOPEN

opEn interOperable Platform for unified access and analysis of Earth observatioN data

H2020-776019

Semantic reasoning for decision making

Dissemination level:	Public
Contractual date of delivery:	Month 33, 31/07/2020
Actual date of delivery:	Month 33, 30/07/2020
Work package:	WP5 Semantic representation and the EOPEN ontology
Task:	T5.1 The EOPEN ontology T5.3 Reasoning for decision support
Type:	Demonstrator
Approval Status:	Approved
Version:	1.0
Number of pages:	55
Filename:	D5.2-Semantic reasoning for decision making_2020-07-30_v1.0.docx

Abstract

This deliverable reports the progress that has been made in semantic mapping and population of data that are pertinent to EOPEN project. Retrieval and reasoning techniques are also described. At the beginning of the document the updates on the user requirements and ontology requirements document are defined. Ontologies related to agriculture and earth observation, querying and reasoning standards and interlinking systems are presented. A comparison between graph and relational databases is also available in the literature review. The ontologies and mapping according to the different types of data are also presented. Reasoning, localisation, semantic querying and enrichment techniques and results are displayed. At the end of this document, an ontology validation example is presented, containing information from all individual EOPEN components, the mappings and the way that semantic querying service retrieves the results.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 776019

History

Version	Date	Reason	Revised by	Approved by
0.0	18/06/2020	ToC creation	Georgios Meditskos (CERTH)	Ilias Gialampoukidis (CERTH)
0.2	22/06/2020	Updates on User requirements and Ontology requirement specification document	Stelios Andreadis (CERTH)	Georgios Meditskos (CERTH)
0.3	26/06/2020	Study on the relevant literature	Ioannis-Omiros Kouloglou (CERTH)	Georgios Meditskos (CERTH)
0.5	04/06/2020	Ontology, querying and reasoning methodology	Stelios Andreadis (CERTH)	Georgios Meditskos (CERTH)
0.6	06/07/2020	Ontology validation example	Ioannis-Omiros Kouloglou (CERTH)	Georgios Meditskos (CERTH)
0.7	24/07/2020	Pre-final draft sent for internal review	Georgios Meditskos (CERTH)	Ilias Gialampoukidis (CERTH)
0.8	28/07/2020	Internal review	Thanasis Mavropoulos (CERTH)	Gabriella Scarpino (SERCO)
1.0	30/07/2020	Updated document after review Ready for submission	Georgios Meditskos (CERTH)	Ilias Gialampoukidis (CERTH) Stefanos Vrochidis (CERTH)

Author list

Organisation	Name	Contact Information
CERTH	Georgios Meditskos	gmedisk@iti.gr
CERTH	Stelios Andreadis	andreadisst@iti.gr
CERTH	Ioannis-Omiros Kouloglou	kouloglou@iti.gr
CERTH	Ilias Gialampoukidis	heliasgj@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr

Executive Summary

In this deliverable we describe the progress that has been made in issues related to ontologies, semantic mapping, reasoning and querying. The main goal of this deliverable is to present the updates in the EOPEN ontology (T5.1) and reasoning for decision support (T5.3) since the previous deliverable (M19).

The present deliverable describes the mapping, reasoning and querying mechanisms that have been developed in EOPEN to meet the requirements that have been defined by the users. Different types of non-EO data which are coming from social media information analysis (such as Topics, Events, Tweets) and EO data (Flood Maps) are represented using a semantic format. All the above-mentioned data come from different components of EOPEN. Data are further enriched using the localisation component and an interconnection with Babelfy, BabelNet and WordNet platforms to associate terms extracted from social media with hypernyms found in WordNet. Geospatial semantic queries are executed to retrieve data that correspond to specific location and time periods. Such information is useful for the users of EOPEN, who can access it using a GUI, since it may assist in decision-making issues.

This work comprises the last version of the system which was firstly introduced in D5.1.

Abbreviations and Acronyms

CQ	Competency Questions
DL	Description Logic
GDB	Graph databases
KB	Knowledge Base
LSTM	Long Short-Term Memory
OGC	Open Geospatial Consortium
ORSD	Ontology Requirements Specification Document
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	RDF Schema
SHACL	Shapes Constraint Language
SPIN	SPARQL Inferencing Notation
SWRL	Semantic Web Rule Language
WGS	World Geodetic System

Table of Contents

1 INTRODUCTION

The scope of WP5 is creating the appropriate semantic knowledge structures to map the information coming from different components of the EOPEN project. The constantly updated Knowledge Base takes advantage of the data coming from:

- Topics which is a result of knowledge analysis extracted from social media
- Events which is a result of knowledge analysis extracted from social media
- Tweets extracted from social media
- Flood maps which is an analysis result from Earth Observation data
- Information coming from Babelfy, BabelNet and WordNet
- Location- and organisation-type entities found in tweets and retrieved via named entity recognition techniques

The architecture of WP5 is presented in Figure 1. Different EOPEN components (i.e. topics, events, etc) provide their results into the "Semantic Mapping" service. Data are represented using a semantic format and saved into the Knowledge Base. A "Semantic Querying" service is also available, which receives a specific time period and area of interest and returns the results that correspond to the defined criteria. Results are presented via a GUI, to provide to the end user an easy way to access the data and take the appropriate decisions using the extracted knowledge.

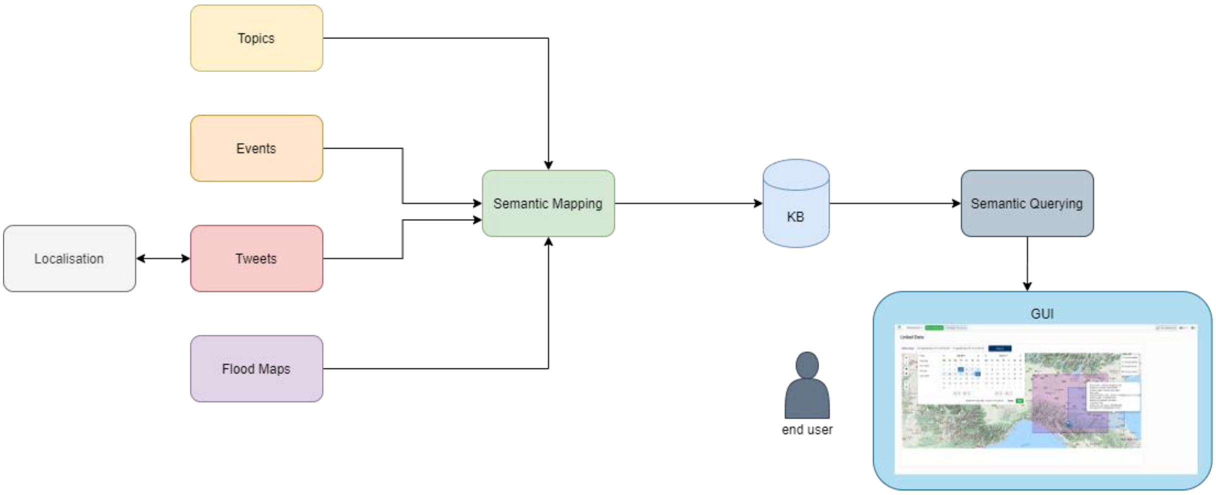

Figure 1. EOPEN semantic components architecture (T5.1 and T5.3)

1.1 Information integration

Information coming from heterogeneous components of EOPEN is integrated using the geolocation as an interconnection point. In most of the data mentioned above (namely events, tweets, flood maps) location information is available. Such information is either extracted from sentinel images (e.g. flood maps), or the localisation module (e.g. tweets).

Combining the location with the timestamp that is available on each of these cases we perform semantic queries and create intelligent interconnections between cross-database data which are useful for the end users. In such way, we offer the complete knowledge to the users, containing results of different components' analyses, who want to be aware of environment-related issues under a specific area and time period.

More specifically, as described in PUC1 users who are responsible for environmental and civil protection issues, want to be aware of emergencies, like for instance, floods. The Knowledge Base contains multiple information coming from social media text analysis and sentinel image surveillance and analysis. Using and combining such information, the semantic retrieval system retrieves information related to flood events to help end users decide whether there is an emergency situation happening and actions need to be taken. As information is combined from many different sources, the accuracy of detected flood events is high.

Accordingly, for PUC2 the main information related to food security are coming from twitter metadata analysis. Events that appear to have a big change in issues related to food security show that the users, which are science centres and institutes, possibly need to take an action under the food security domain. For PUC3, the most appealing information has to do with the changes over Snow and Flood events over time. The users, who are environmental agencies and managers, can be aware of such changes using the timestamp that is provided and make predictions over climate and environmental changes in future periods.

1.2 RDF Repositories

The main disadvantages (Jaiswal, 2013) of using graph databases are associated with the lack in security and their maturity level, which is not as high as relational databases, since the technology is much more recent. Their main advantage (Jaiswal, 2013) is their flexibility. The schema can easily change to adapt to the changing needs of the users and the nature of data. Another advantage (Jaiswal, 2013) is that the execution time of queries in graph databases takes much less time than it does in relational databases. This is extremely significant in terms of EOPEN, as geospatial queries take more time to be computed and results need to be retrieved in the shortest possible time, mostly in emergency cases of a natural catastrophe.

More specifically, the functionalities described in section 1.1are available due to the usage of semantic technologies and graph databases. These technologies are more oriented in data relationships. Complex relationships between data are represented and easily retrieved using semantic queries to meet the needs of the users. Response time is extremely low, taking into consideration that geospatial queries are in most cases time-consuming.

The rest of this document is organised as follows: Section 2 describes all the updates that have been made to the user requirements and the Ontology Requirement Specification Document since D5.1 “The EOPEN ontology and semantic reasoning support”. Section 3 gives a literature review on issues related to the ontologies, querying and reasoning and interlinking. A comparison between graph and relational databases is also presented in the same section. Section 4 describes the way we re-use the Web annotation model to map the results that are pertinent to EOPEN. Section 5 describes the reasoning, querying and decision support framework providing the methodology, datasets, parameters and results where applicable. In the end, section 6 gives an overall example of mapping results related to a specific user story and receiving knowledge for the users under a specific area of interest and time period. In section 7, a conclusion of this document is presented.

2 UPDATED USER REQUIREMENTS RELEVANT TO ONTOLOGIES AND REASONING AND ONTOLOGY REQUIREMENT SPECIFICATION DOCUMENT

Updated user requirements

In this section we present the ways that we support various user requirements as these have been described on D2.2 “User requirements”. As such requirements have also been described in D5.1 “The EOPEN ontology and semantic reasoning support”, we only focus on updates that have been made on this document since D5.1. A brief description of the user requirements which are relevant to the WP5 representation and reasoning framework is presented on Table 1.

Table 1. Updated user requirements relevant to WP5 representation and reasoning framework

User Requirement ID (D2.2)	Description	WP5 Relevance / Dependency
PUC1_GA1	Must be provided with capabilities for = data dissemination and integration of EO data, weather information and relevant social media text and images.	• Support data mapping coming from social media and EO data • Support integration of two Knowledge Bases using a service to assist in decision making issues
PUC1_GA4	Must be enabled to merge different administrative database and formats in a unique platform with all data shared.	• Support integration of two Knowledge Bases using a service to assist in decision making issues
PUC1_GA7	Must be provided with an interactive archive of each event; all data from social network communities and from satellites should be stored in a specific database to provide a history of each event.	• Support the ability to save event data in the KB
PUC3_GC2	Should be provided with easy access and management of datasets.	• Support retrieving semantic data using a service to assist in decision making issues and generate notifications if needed
PUC3_GC3	Must be provided with data integration capabilities.	• Support integration of two Knowledge Bases using a service to assist in decision making issues
PUC3_GC7	Must be enabled to browse historical observations.	• Support retrieving semantic data of a specific time period using a service to assist in decision making issues

OSRD

In this section we present the updates that have been made in the OSRD which was initially presented in D5.1 “The EOPEN ontology and semantic reasoning support”. The OSRD provides a description of the ontological framework of EOPEN. Competency questions is the section that has been mostly updated. The other sections are also presented, with small adjustments, in terms of completeness.

EOPEN ORSD	
1	**Purpose**
	The representation framework of EOPEN has as a purpose to offer the appropriate ontological structures and vocabularies to provide a semantic representation model that captures the results that are pertinent to the EOPEN analysis modules. The ontological framework provides the annotation model that is responsible for issues related to data modelling, reasoning and integration, such as: • Structures to capture the metadata of non-EO data and more specifically social media related information i.e. topics, events, and EO-data i.e. flood maps, etc. • A well-defined data model which assists on defining annotation and assertion, sharing and reusing between different EOPEN modules and also in external platforms.
2	**Scope**
	The ontology developed in EOPEN needs to formally capture: • Social media information extracted from twitter and the analysis of the results. • Information related to localisation, coming from the location extracted from events, tweets and flood maps. • Information related to twitter topics which comes after the analysis of the extracted information. • Information related to the events which comes after the analysis of social media extracted information. • Information related to flood maps which comes after the EO data analysis. The EOPEN ontologies have been designed in a way to enhance knowledge sharing and promote interoperability and reusability. Extensibility and modularity is provided using a pattern-based approach.
3	**Implementation Language**
	OWL 2 (Grau et al., 2008), a language proposed by W3C, has been used for the implementation of the ontology and knowledge representation in the Semantic Web.
4	**Intended End-Users**

	The system that has been developed in EOPEN has been designed to provide the available knowledge to the different types of end users using the authoring tools: • ***PUC1: Flood risk assessment and prevention*** **Firefighters and administration offices:** Firefighters who want to be aware of real-time events, such as floods, and administration offices that are responsible for environmental and civil protection issues. • ***PUC2: EO datasets to monitor Food Security in South Korea*** **Science Centres and Institutes:** Science centres and institutes who want to be aware of issues related to food security i.e. rice production or overproduction estimations. • ***PUC3: Monitoring Climate Change through Earth Observation*** **Reindeer researchers and government agencies:** Reindeer researchers and government agencies who want to be aware of climate and environment-related issues to select the most appropriate environment for reindeer herding and handle repair-related issues i.e. rail, road etc.
5	**Ontology Requirements**
	Non-functional requirements
	NFR1. Existing vocabularies, ontologies and standards should be adopted in order to formulate the ontology
	Functional requirements: Groups of competency questions
	In the table below we present a set of Competency Questions (CQ) as these have been defined by the user requirements and Pilot Use Case scenarios. Other technical partners also contributed to the definition of the questions. In section 6, a full-case simulation example is presented covering the requirements that have been defined for the generation of the annotation models. **<u>Tweets</u>** **CQ1** Which is the timestamp of the tweet? **CQ2** Which is the use case of the tweet? **CQ3** In what language does the tweet refer to? **CQ4** Which is the location of the tweet? **<u>Events</u>** **CQ5** What is the timestamp of the event? **CQ6** Which is the use case of the extracted event? **CQ7** Which is the language of the extracted event? **CQ8** Which is the score of the extracted event? **CQ9** What is the change in the extracted event? **CQ10** What is the location of the event? **CQ11** Which keywords are related to the event? **CQ12** Which is the level 1 hypernym WordNet identifier for each keyword? **CQ13** Which is the level 2 hypernym WordNet identifier for each keyword?

	CQ14 Which is the level 3 hypernym WordNet identifier for each keyword? **<u>Topics</u>** **CQ15** Which is the timestamp of the topic detection? **CQ16** Which use case is the topic related to? **CQ17** Which is the language of the topic? **CQ18** How many topics have been detected? **CQ19** Which is the identifier of each topic? **CQ20** Which labels are related to the topic? **CQ21** How many labels does a specific topic contain? **CQ22** Which is the score of each label? **CQ23** Which tweets are related to the topic? **CQ24** How many tweets are related with the topic? **CQ25** Which tweets are top ranked? **CQ26** How many top ranked tweets does a specific topic contain? **<u>Flood map</u>** **CQ27** Which is the flood map path? **CQ28** Which is the sensing date of the flood map? **CQ29** From which satellite does the image come from? **CQ30** Is the corresponding area flooded? **CQ31** What is the total area of the image in square meters? **CQ32** What is the total flooded area found in the image in square meters? **CQ33** What is the flood percentage? **CQ34** What is the name of the location? **CQ35** What are the polygon coordinates of the flooded area? **<u>Semantic retrieval</u>** **CQ36** Which events correspond to a given time period and polygon area? **CQ37** With what information are the events found in this area related to? **CQ38** Which tweets correspond to a given time period and polygon area? **CQ39** With what information are the tweets found in this area related to? **CQ40** Which flood maps correspond to a given time period and polygon area? **CQ41** What information does the flood map instance of a specific area and time period contain?

3 STATE OF THE ART

3.1 Ontologies

3.1.1 Agriculture

Creating an appropriate ontology in the agricultural domain is a complex procedure, as it consists of many concepts and relationships (Song, Wang, Ying, Yang, & Zhang, 2012). The ontology divides the knowledge into three categories: the object of labour, the means of labour and the production process as shown in Figure 2. The object of labour contains all the relationships that exist for the specific object, the means of labour contain information related to the materials, while the production process contains information related to the agricultural production cycle stages.

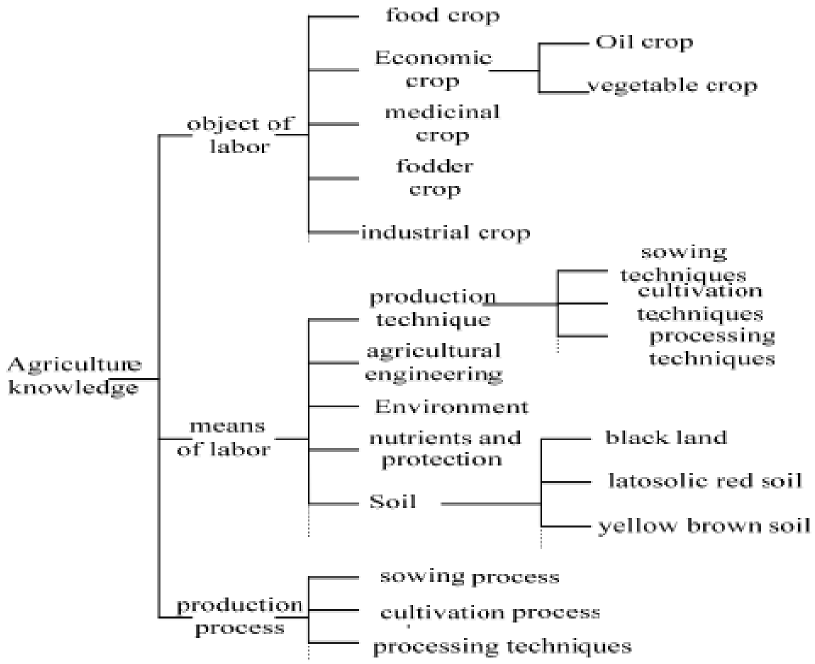

Figure 2. Agricultural knowledge system (Song et al., 2012)

More specifically, the suggested model contains information such as cultivation and processing practices, storage, pests control, genetic attributes etc. Such information is described using the corresponding classes as shown in Figure 3.

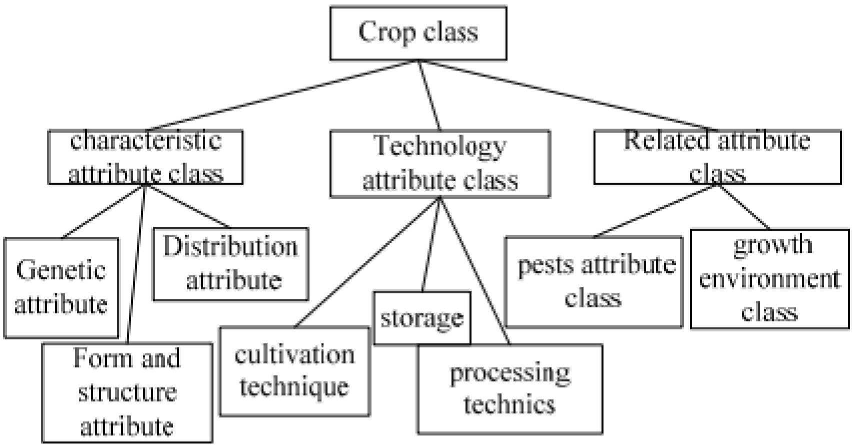

Figure 3. Crop attribute classes (Song et al., 2012)

An ontology that offers knowledge representation for vegetable crops cultivation under greenhouse environments is OntoCrop (Maliappis, 2009). More specifically, the ontology represents information such as crop types physiology, common cultivation practices and pest control. Plants are characterised by properties such as growth stage, name, infected part, type of infection, information about disorders. Semantic rules are also implemented to extract useful conclusions according to i.e. symptoms.
An ontology that is oriented in the internet of things aspect is AgOnt (Agriculture ontology for the purpose of agriculture internet of things) (Hu, Wang, She, & Wang, 2011). The ontology is lightweight, as it does not take into account complex procedures such as food processing or agriculture activities. On the contrary, great importance is given to the environment of the products. Agricultural products are related with product, phase, seeding procedures, location, physical conditions, and temporal information. The relationship between the abovementioned objects is shown in Figure 4. The ontology aims in supporting healthy food management.

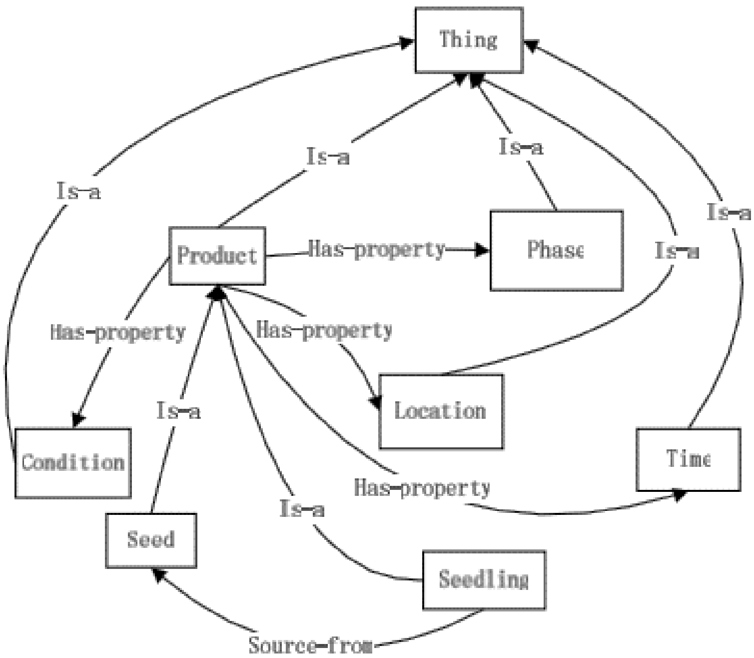

Figure 4. The top-level ontology of AgOnt (Hu et al., 2011)

A uniform representation of concepts extraction and text classification results is described in (Su et al., 2012). Concepts are matched in the ontology classes, comparing concepts that are either identical, or have a similar meaning. Irrelevant terms are removed from the ontology. An algorithm that calculates concept similarity weights is used for this reason. The agricultural ontology directory offers many different types of products such as agricultural, fishery, livestock, planting and agricultural material.

3.1.2 Earth Observation

In literature, many ontologies have been introduced to represent earth observation data. Most of them target the environmental sector and more specifically environmental monitoring. An ontology to represent data under the hydrological monitoring domain is presented in (Wang, Wang, & Chen, 2017). Such ontologies can help in environmental protection and water resources management issues. Events, sensors and observations are the main ontology classes (Figure 5). Observations are divided into many subcategories like meteorological and water quality, sensors are divided to physical and chemical, while on the other hand events describe any change in hydrological cycle.

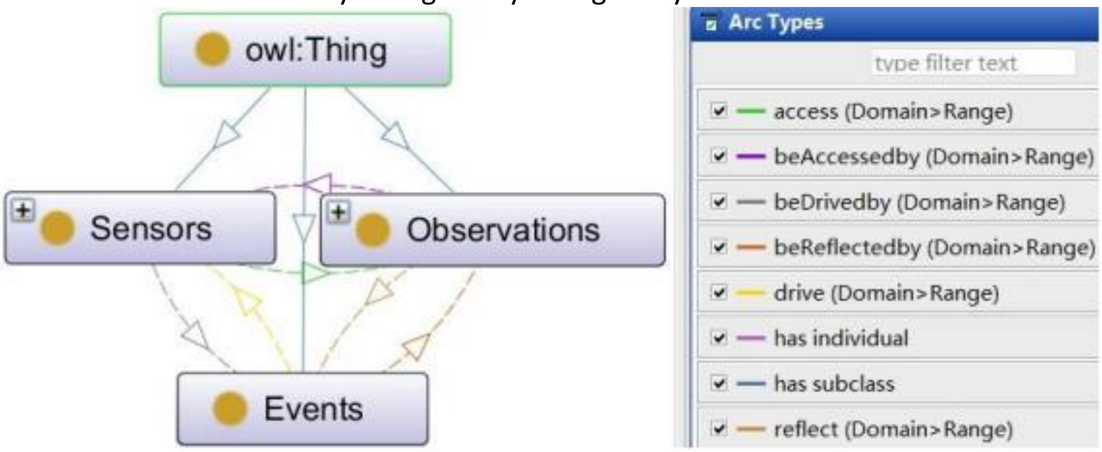

Figure 5. The main classes of hydrological ontology (Wang et al., 2017)

The Modular Environmental Monitoring Ontology (MEMOn)(Masmoudi et al., 2019) is based on other ontologies, namely the Basic Formal Ontology (BFO), the ENVironment Ontology (ENVO), the Semantic Sensor Network ontology (SSN) and the Common Core Ontologies (CCO). The ontology offers eight main modules (Figure 6) covering more aspects than the abovementioned ontology to represent emergency situations e.g. earthquakes, floods, that exist under the hydrological monitoring domain. MEMOn also provides the structures to represent physical conditions (disaster), spatiotemporal information (geolocation and time) and environmental features (procedure and material).

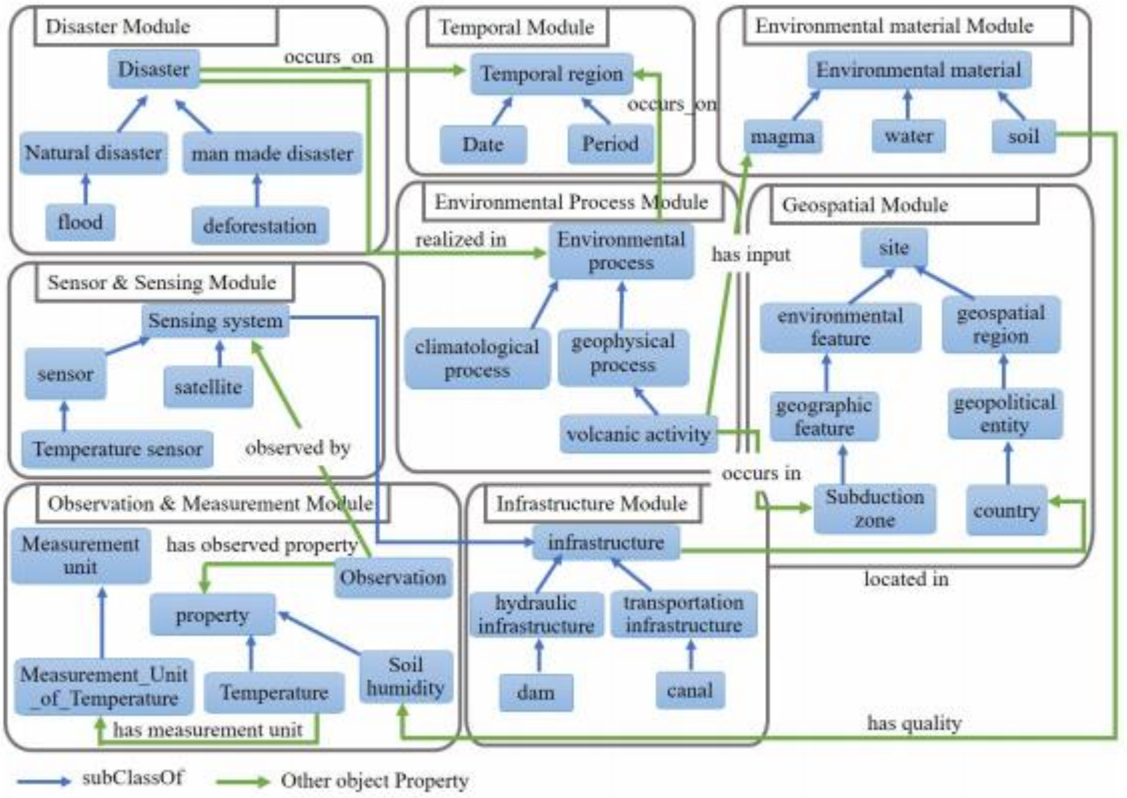

Figure 6. MEMOn ontology modules overview (Masmoudi et al., 2019)

3.2 Querying and reasoning

3.2.1 SWRL: Semantic Web Rule Language

Combining RuleML (Binary/Unary Datalog) with OWL ontology language (Lite and DL) led to the creation of SWRL (Horrocks et al., 2004). Expressivity is increased as Horn rules can be combined with ontology knowledge bases. SWRL is an OWL extension that uses rules to define the ontologies. SWRL does not allow denial definitions, which are defined using the `owl:complementOf` structure of OWL. SWRL builds beyond OWL though to extend the expressivity of the ontology as it offers ontology definitions which are impossible to be represented using the OWL syntax. On the other hand, the increased expressivity of SWRL is associated with the undecidability of reasoning, meaning that there do not exist any reasoning systems to handle the full expressivity of SWRL rules. To avoid such problems, there have been defined some SWRL variables which are responsible for limiting the expressivity. Safe Rules are implemented in most of the reasoning systems following the pattern that all variables which are found in the head, should also be part of the body, given that variables can refer to existing ontology objects.

3.2.2 Restrictions and rules: SPIN and SHACL standards

The web modelling languages have been designed to offer the appropriate structures to represent the static structure of the data. OWL and RDF Schema (RDFS) are used in order to define properties, classes and the relationship between entities, while SKOS is used to describe vocabulary concepts and hierarchies. All the above-mentioned languages define the axiomatic definitions of the data structures, but do not offer the description of the general computational behaviour of objects.

Languages that are oriented to the objects offer mechanisms to define the behaviour of the objects as they describe methods and classes which are associated with using class members. The methods that are oriented to the objects usually describe how changing one attribute affects the other attributes. Constraints of the methods are in most cases designed to ensure that the state of the object is within the limitations that the designer of the class has determined.

The following section describes the SPIN and SHACL standards and the way they are used.

SPARQL Inferencing Notation (SPIN)

In SPIN (Knublauch, Hendler, & Idehen, 2011) standard, concepts coming from languages that are oriented to the objects, rule-based systems and querying languages are combined to describe an object's behaviour on the web. A key function of SPIN is that class definitions are connected to SPARQL queries in order to synthesize the expected behaviour of the classes. SPARQL provides well-formed semantics in queries between the RDF data and this is the main reason why it is highly used in graph stores and RDF search engines. In addition, sufficient expressivity is provided in data and queries computation. SPARQL queries are expressed through the use of RDF triples that are based on SPIN syntax, in order to ease maintenance and storage. In that way, SPIN offers a connection between the SPARQL queries and the RDF resources as well as SPARQL queries reuse and sharing. The first level of SPIN standard is an RDF vocabulary for SPARQL.

SPIN also offers the ability to adapt to inference rules i.e. INSERT/DELETE queries, SPARQL CONSTRUCT etc. and restriction checks i.e. CONSTRUCT, ASK queries as it combines SPARQL and RDF syntax. SPIN, apart from the rules that are designed to generate new RDF statements using RDF commands also provides constructors, which are a type of inference rules that offer resources initialisation with default values. The constraints of SPIN are used to define the conditions that the class members should meet.

It is possible to share the rules, the restrictions and the class definitions of SPIN via the web as SPIN is based on RDF syntax. Maintaining rules that have a local scope is an easy task when rules are attached to classes.

The second level of the SPIN framework is related with the properties that are defined to apply restrictions and rules into the classes. Providing this functionality makes SPIN a well-known model to the users. SPIN is totally integrated in the Semantic Web, offering a useful standard on Internet resources that need to be integrated from a variety of sources.

In SPIN, SPARQL queries can be reused taking into account the mechanisms that are provided which describe SPARQL queries that use predefined variables. Higher level modelling languages can specify the templates to characterise variable values using the bound variables.

In addition, SPIN supports a post-modelling functionality able to define SPARQL functions based on SPARQL expressions and also integrate SPARQL query templates that could be reused. Modelling languages with executable declarative semantics are defined via a mechanism of SPIN standard. A library of usually needed modelling templates is also provided, based on this mechanism and the available templates and functions which is described as the third level of the SPIN standard (Figure 7).

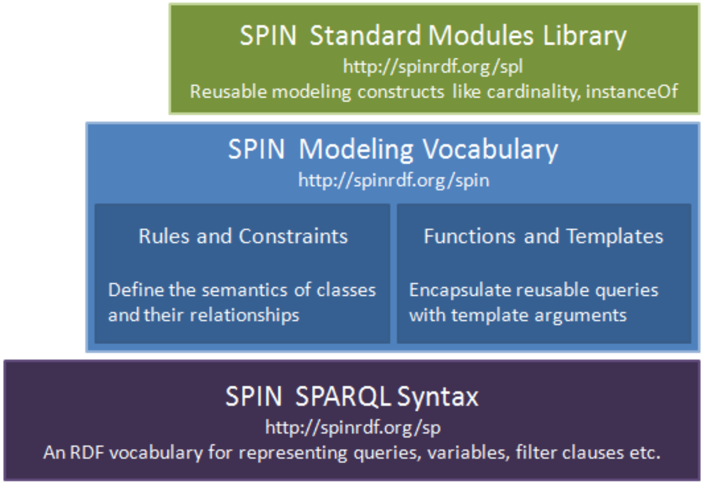

Figure 7. The three stages of SPIN[1] standard

SPIN also supports the definition of custom SPARQL functions, using the concept of standards. The functions and their arguments are determined using the RDF vocabulary. Using their URIs, SPIN functions can be exchanged over the web while SPARQL processors can search dynamically and on-demand method definitions, like Web service calls. Each time a SPIN function is running, the query set for the operation is computed. This method allows complex SPARQL queries construction using a combination of other SPARQL query templates.

SPIN's contribution in the semantic web is very important mostly for applications. The reason behind that, is that SPIN is based on a SPARQL standard that is more application-oriented. The principles that SPIN follows, assist in processing, understanding and interacting between interlinked data sources for both humans and machines. Recognising SPIN as a W3C recommendation will facilitate developing semantic web-based applications that are portable and practical.

Shapes Constraint Language (SHACL)

A standard that has been developed to define structural constraints on RDF charts is SHACL[2]. The standard has been developed by W3C RDF Data Shapes team in 2014, published in October 2015 and proposed in June 2017. SHACL comes from a combination of SPIN, OSLC and ShEx resource schemes. The first plan was to incorporate different validation approaches creating a unified language which was later defeated as ShEx and SHACL are totally different approaches.

SHACL consists of two components: (a) kernel that uses RDF vocabulary to define the common variables and shapes and (b) SHACL-SPARQL that offers a mechanism which extends SPARQL. There are two SHACL extensions available, one for supporting complex features such as expressions and rules and one, named SHACL-Javascript, for applying

[1] http://spinrdf.org/spin-architecture.html

[2] https://www.w3.org/TR/shacl/

Javascript restrictions. SHACL uses shape structures to organise variables and information and RDF is used on the other hand in order to compile it.

3.3 Graph and relational databases

Relational databases

Relational databases[3] are databases that are responsible for saving data points which are connected. Such types of databases are based on the architecture of the relational model, using which, data are represented into table formats. Each row of the table is a record that is associated with a unique ID (named key). The columns of the table correspond to the different attributes that data offer. Each record usually has a value for each attribute and in that way relationships are created between different data points.

Relational databases offer many benefits:

- **Data consistency**
  Consistency is maintained between instances, namely applications and database copies. For data in different time periods, new instances are created even for the same user.
- **Commitment and Atomicity**
  Relational databases follow strict policies for commitment as rules are defined in detailed level. Changes in the databases are permanent. Such databases create commitments when they can commit for all parts. This related-to-commitment capability is called atomicity. Data in the database are accurate and follow the specified rules, policies and regulations.
- **Stored Procedures and Relational Databases**
  Since accessing the data requires many repetitive actions that need code to access the database, functions are required. These functions can be stored procedures that can be accessed using a simple application call.
- **Database Locking and Concurrency**
  When multiple users try to change data at the same time over a relational database, conflicts may arise. In such cases, locking and concurrency techniques are used to avoid those conflicts. Locking techniques prevent different users from accessing data at the time they are updated, while concurrency manages the activity when multiple users query the database at the same time.

Graph Databases

Graph databases[4] (GDB) are databases, which are based on the graph theory, and use graph structures to apply semantic queries over edges, nodes and properties to represent and store data. Nodes may be instances or entities and are equivalent to a row of a relational database. Edges are the relationships between different nodes and are represented using the lines that connect different nodes. The edges may be directed or not directed. In such a way data are linked together, while in many cases data can be retrieved using one operation. Properties are information that are related to nodes. An example of using nodes, edges and properties in a graph model is shown in Figure 8. Querying in graph databases is a fast procedure as data are perpetually stored. Data can be visualised using a graph format.

[3] https://www.oracle.com/database/what-is-a-relational-database/

[4] https://en.wikipedia.org/wiki/Graph_database

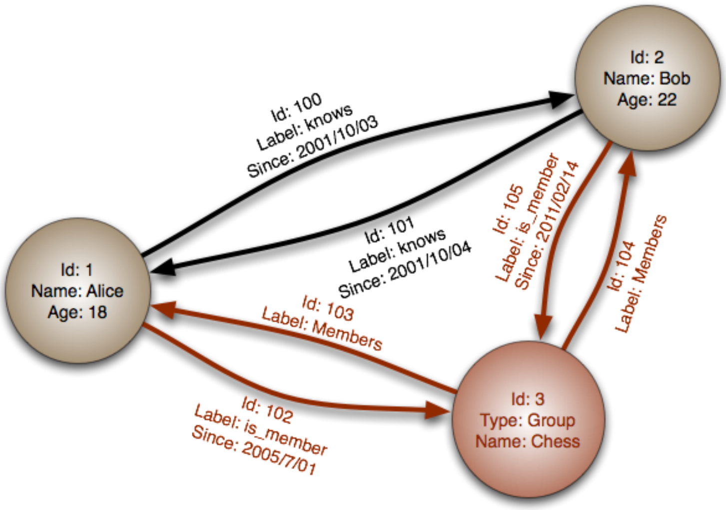

Figure 8. Example of graph representation using nodes, edges and properties

Graph databases are NoSQL databases designed to address the limitations that are related to relational databases. Graph model offers explicit connections between the nodes of data, while relational and NoSQL database models link data using implicit connections. Retrieval in graph databases is a simple and fast procedure, even in complex hierarchical structures that are difficult to model using relational databases and this is because of the design of graph databases.

Comparison

A comparison between relational and graph databases is presented in the table below. In most cases, graph databases are used when data relationships exist in the core of the requirements.

Table 2. Comparison between relational and graph databases[5]

	Relational Database	Graph Database
Relationships	Relationships between entities exist as keys of within the dataset	Relationships between entities are represented using tables
Query performance	The query performance is reduced when size of the dataset is increased	The query performance is degraded when the number of relationships is increased

[5] https://www.sqlshack.com/understanding-benefits-of-graph-databases-over-relational-databases-through-self-joins-in-sql-server/

Adding new relationships	Introducing new relationships is harder as it requires changes in the definition of an existing table	Adding new relationships is easy

3.4 Interlinking

As linked data wealth is constantly increasing, there comes the need of interconnecting data coming from heterogeneous sources in order to exploit this wealth. In (Zhu et al., 2017), a system is presented which receives data from heterogeneous sources (namely earth observation, meteorological and health) and represents the data using an RDF format. The similarity between the different datasets is calculated to achieve data interlinking focusing on eight main characteristics.

A system that provides integration between different earth observation data to offer data management is presented in (Yang & Li, 2016). Data are sent from various data sources to the system. For the data concerning China multiple components have been developed in order to offer adaptations between different interfaces. For international data the GEO DAB agent is used. The result is China GEO Data Center, a web portal that shares earth observation data using satellite images of China.

PREDICAT (Masmoudi et al., 2019) is a system that focuses on natural catastrophes prediction. The system receives a large amount of data coming from citizens using many different web data sources (data collection layer) and stores them into different data structures (big data layer). Appropriate services have been developed to provide accessibility of data (service layer). PREDICAT uses different ontologies to semantically represent the data that are pertinent to the system (semantic layer). The system overcomes data heterogeneity and provides a common structure of interconnected objects containing spatiotemporal information (data integration layer), while a reasoner and a decision maker are implemented to provide the appropriate responses to the user (data processing layer). In the end, the system applies semantic machine learning and prediction techniques to predict natural catastrophes and support decision making of such issues (application layer). Users can have access to queries and results using a web interface (user interface layer).

In CANDELA project[6], a module has been developed that offers semantic search functionality into EO images and other associated metadata. The module gathers datasets that contain Sentinel images and information correlated to them, utilising also open datasets. An ontology is developed according to the needs of the project based on existing ontologies, namely GeoSPARQL, OWL-Time, SOSA, DCAT and PROV-O. The framework integrates the related entities and provides a semantic searching mechanism, giving the opportunity to the users to retrieve the images that fulfil the criteria that are set on each search. The following table (Table 3) creates a comparison between the two projects (EOPEN and CANDELA) in a semantic manner.

[6] http://candela-h2020.eu/

Table 3. Comparison between EOPEN and CANDELA projects

	EOPEN - GraphDB	EOPEN – Strabon	CANDELA
Input data format	JSON	GeoTIFF	CSV, JSON/GeoJSON, GeoTIFF, Shape files
Metadata type	EO data (satellite images) and non-EO (tweets)	EO derived products	EO data (satellite images) Open Data available as raster files (land cover, land use) or vectorial geometry (territorial units) Data resulting from EO image analysis (change, land cover annotation) Contextual data
Derived data	Tweet locations, topics		
Semantic language	RDF	RDF	OWL
Semantic format	Turtle	N3, N-Triples, Turtle	N-triples, Turtle
Vocabularies that the ontology relies on	Web Annotation, Basic Geo, GeoSPARQL	EOP, DCAT, GMD, GML, Atom	GeoSPARQL, OWL-Time, SOSA, DCAT, PROV-O
Procedure	▪ Data collection ▪ Ontology development ▪ Data integration ▪ Semantic reasoning	▪ Dataset/metadata selection ▪ Mapping rules development ▪ Data conversion and integration	▪ Dataset selection ▪ Ontology development ▪ Data integration ▪ Semantic search
Triple store	GraphDB	Strabon	Strabon
Graph structure (one/several)	One graph		One graph
Service language	Java	Python	Python
Semantic search/querying	SPARQL	SPARQL, GeoSPARQL	SPARQL, GeoSPARQL
Use of Docker	Yes	Yes	Yes

4 THE EOPEN ONTOLOGIES

4.1 EOPEN Annotation Model

As described in D5.1, a Web Annotation Data Model is used to map EOPEN annotation data. The Web Annotation Data model is a generic model that offers representation of annotations as a set of linked resources. Each annotation contains a target and a body that is strongly attached to the target. In the EOPEN project each annotation type is treated in a different way. More information about that is depicted in the table below:

Table 4. Web Annotation Data Model in EOPEN.

Component	Annotation name	Type	Associated entity
Topics	Cluster_Annotation	Target	Collection instance
		Body	ClusterBody instance
Events	Event_Annotation	Target	Event_Target instance
		Body	Topic instance
Tweets	Tweet_Annotation	Target	TweetTarget instance
		Body	TweetBody instance
Flood Map	ChangeDetectionAnnotation	Target	ChangeDetectionTarget instance
		Body	ChangeDetectionBody instance

4.2 EOPEN population

In this section we present the way we use the Web Annotation Data model in order to fulfil the mapping needs of EOPEN. Each subsection describes the way we map the results of different EOPEN components (namely topics, events, tweets, flood maps). An overview of the annotation classes that are supported in this project is shown in the figure below.

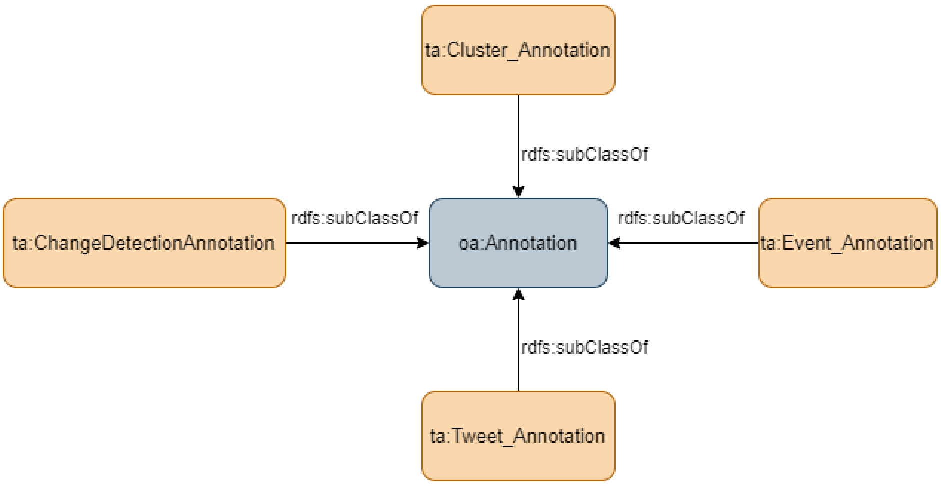

Figure 9. The four annotation classes in EOPEN

4.2.1 Topics

Topic annotations are mapped using a `ta:Cluster_Annotation` class instance. The instance allows a connection establishment between a topic collection (target) and a cluster body (body). The first contains all the topic-related information (i.e. topics, labels, associated WordNet entities, tweets, etc.), while the latter contains more generic information like use case, language and timestamp. The schema of what has been described is shown in Figure 10 while a mapping example is presented in section 6.1. The circular grey components show the data properties and the existence of values in specific properties which may be of type String, Integer, etc.

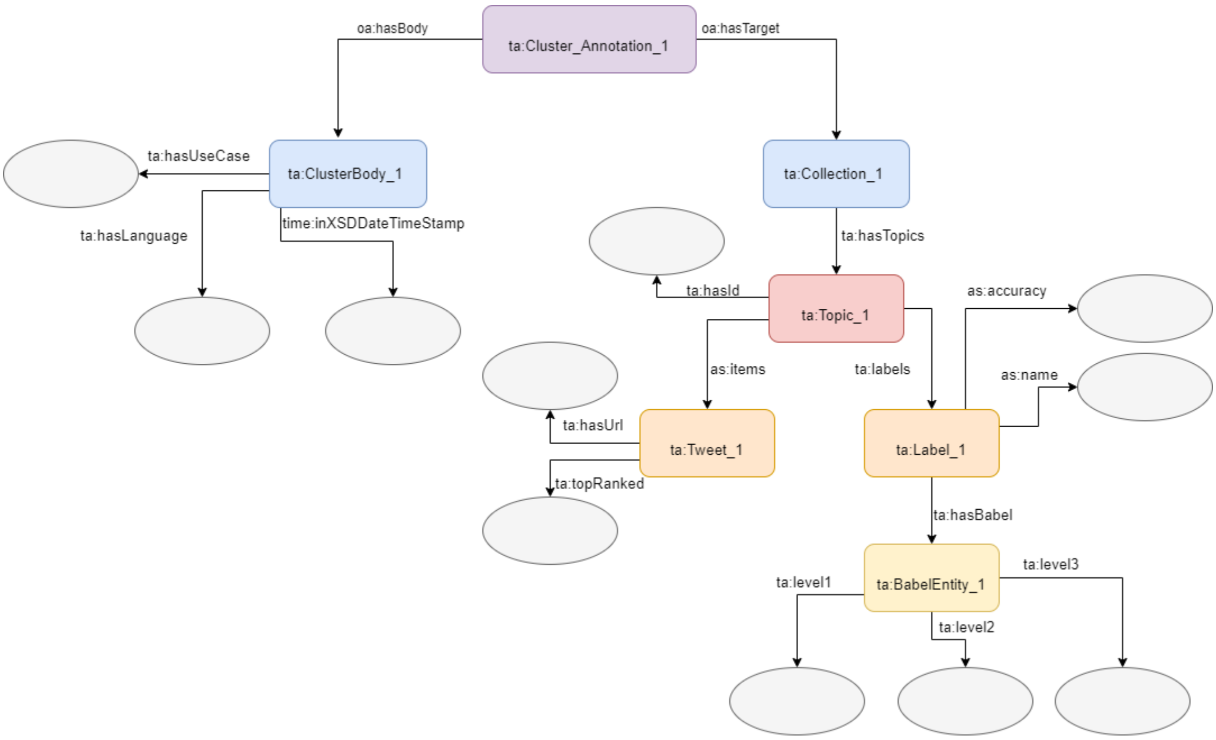

Figure 10. Population schema for Topic detection results

4.2.2 Events

Event annotations are mapped using a `ta:Event_Annotation` class instance. The instance allows a connection establishment between an event target (target) and a topic (body). The first contains more generic information like language, score, rate, etc., while the latter contains all the topic-related information (i.e. topics, labels, associated WordNet entities, etc.). The schema of what has been described is shown in Figure 11 while a mapping example is presented in section 6.2. The circular grey components show the data properties and the existence of values in specific properties which may be of type String, Integer, etc.

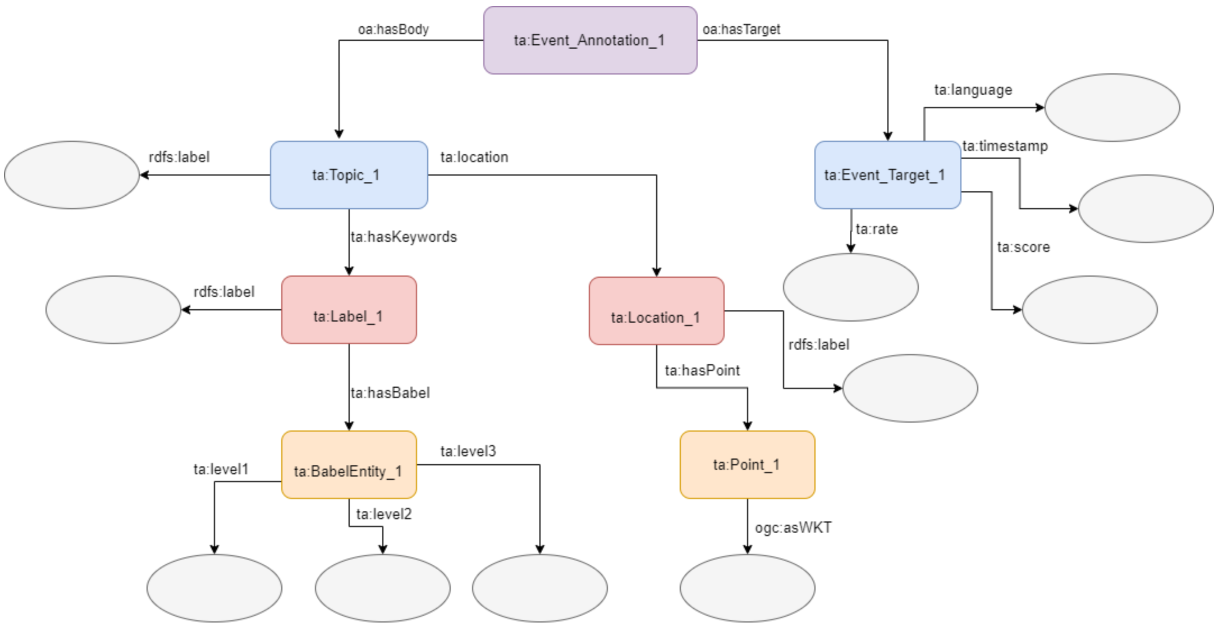

Figure 11. Population schema for Event detection results

4.2.3 Tweets

Tweet annotations are mapped using a `ta:Tweet_Annotation` class instance. The instance allows a connection establishment between a tweet target (target) and a tweet body (body). The first contains all the tweet-related information (i.e. tweet id, location, point etc.) while the latter contains more generic information like use case, language and timestamp. The schema of what has been described is shown in Figure 12 while a mapping example is presented in section 6.3. The circular grey components show the data properties and the existence of values in specific properties which may be of type String, Integer, etc.

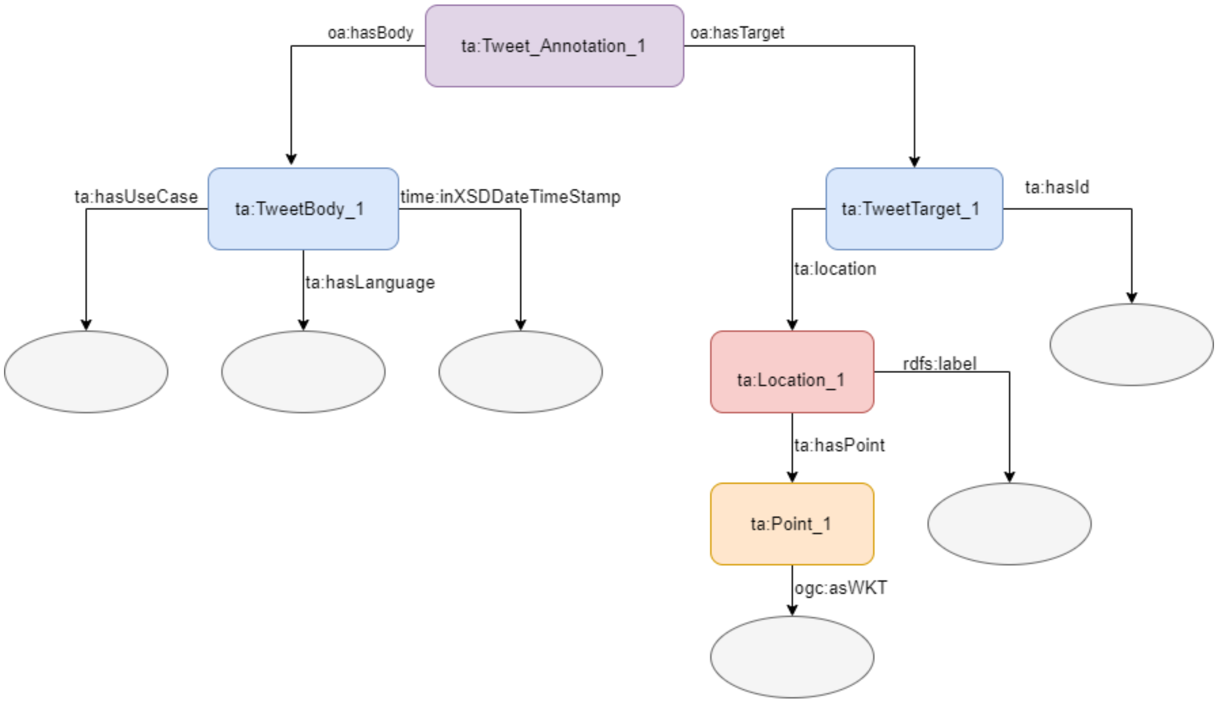

Figure 12. Population schema for Tweets

4.2.4 Flood map

Topic annotations are mapped using a `ta:ChangeDetectionAnnotation` class instance. The instance allows a connection establishment between a change detection target (target) and a change detection body (body). The first contains all flooded area information (i.e. percentage, area, flood polygon, etc.) while the latter contains more generic information like date, flood map file, the sentinel that the image comes from, etc. The schema of what has been described is shown in Figure 13 while a mapping example is presented in section 6.2. The circular grey components show the data properties and the existence of values in specific properties which may be of type String, Integer, etc.

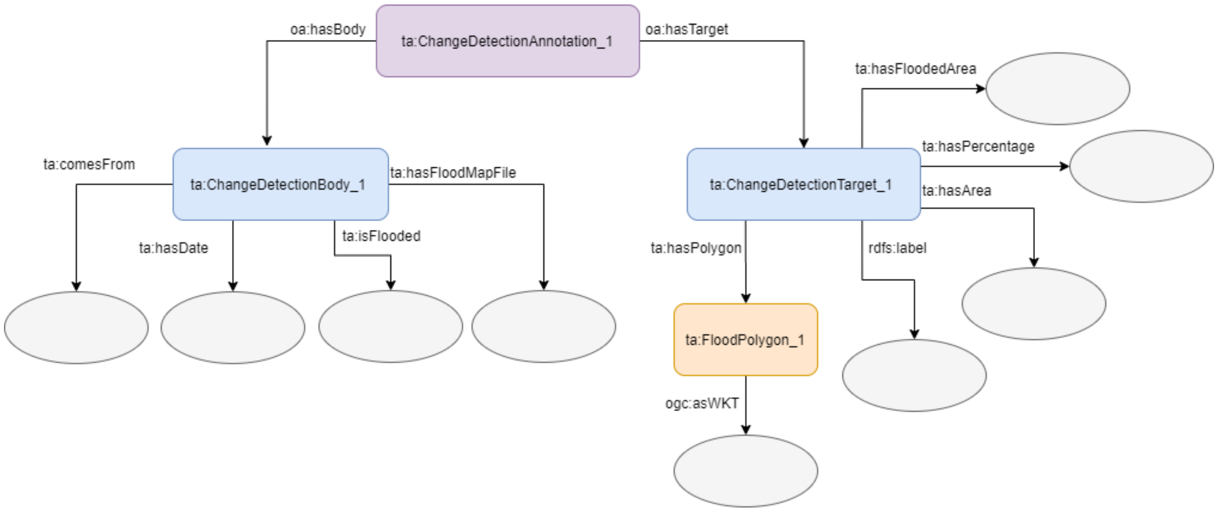

Figure 13. Population schema for Flood map results

4.3 Representing Location

In D5.1 we described how we mapped the geospatial results using the World Geodetic System (WGS) standard. Since the needs of the project have changed, in the final version we now use the GeoSPARQL vocabulary in order to map the geospatial data. The reason behind this selection, is that GeoSPARQL can be used to easily query GraphDB and apply many geospatial queries, something that is not feasible when using the WGS standard. GeoSPARQL offers a wide list of functions to easily query semantic geospatial data.

In EOPEN case, we use the `ogc:Geometry` class to map the different geometries that are given by the EOPEN components. The property `ogc:asWKT` is used in order to connect a location instance with specific geometry coordinates. As shown in Table 5, two schemas are supported: Points and Multipolygons.

Table 5. Using GeoSPARQL vocabulary to map different geometry types

Geometry type	Semantic object
Point	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> POINT(10.311302241097014 44.23624963837918)^^ogc:wktLiteral" .
Multipolygon	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((10.311302241097014 44.23624963837918, 12.270048101694844 44.23624963837918, 12.270048101694844 45.21575843334017, 10.311302241097014 45.21575843334017, 10.311302241097014 44.23624963837918)))^^ogc:wktLiteral" .

5 REASONING, QUERYING AND DECISION SUPPORT

5.1 Localisation

Within this section we report on additions, changes and improvements that were made to the localisation framework, in order to comply with the needs of the EOPEN use cases, as defined in D2.2. The objective of the service is to retrieve location and organisation entities found in user tweets, calculate the proper coordinates and pinpoint them on a map. Supported languages are dependent of the project's use cases and include English, Italian and Finnish.

5.1.1 Methodology

The adopted methodology, as reported in D5.1, is based on machine learning approaches and while in principle remains the same as before, it has been updated to reflect the latest developments in the field. While the chosen models still rely on the Long Short-Term Memory (LSTM)-based architecture of D5.1, various implementations have been tested (BiLSTM-CRF, BiGRU-CRF, BiLSTM-CNNs-CRF, BiLSTM-CNN) to determine the best possible outcome.

BiLSTM-CRF (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016): The previously used approach relied on a biLSTM for the character embeddings step and a word2vec-based step for word representation. Concatenation is managed via dropout, while the decoding phase is handled by a CRF layer.

BiGRU-CRF (Peters, Ammar, Bhagavatula, & Power, 2017): In this configuration, for the encoding phase the single LSTM network is replaced by two bidirectional GRUs. The decoding once again is handled by a CRF layer.

BiLSTM-CNNs-CRF (Ma & Hovy, 2016): A BiLSTM is responsible for the encoding step, a CNN is used to extract character level features, while again a CRF is utilised for the decoding phase.

BiLSTM-CNN (Chiu & Nichols, 2016): This model is the only one that does not rely on a CRF layer for the decoding phase, opting instead for a softmax layer. The encoding follows the same directions as the previous systems, by employing a LSTM solution.

Regarding the English tests, apart from testing the abovementioned models, another update to the process involved the use of ELMo contectual embeddings. These changes proved sufficient enough to achieve state-of-the-art NER results, improving the previous efforts by almost 1.5%. With regard to the Italian methodology, there have been updates to both the utilised model and the dataset. The latter involved enrichment of the previously used data with extra annotation samples and the relevant updates will be presented in section 5.1.2 Concerning the DNN model, the same applies as for the English test, where different approaches were tested in their standard default configuration and with the addition of the BERT methodology. By utilising different linguistic resources, which proved to be more adequate for the task, an amelioration of almost 3% was achieved in the resulting scores. The Finnish models are a new addition to the EOPEN pipeline, having been implemented after D5.1. Again, all four models were tested using different embedding options (besides character and word representations, BERT embeddings were evaluated). According to initial tests the ELMo embeddings for Italian and Finnish did not ameliorate the respective BERT scores, so relevant, extensive testing was not pursued.

Since the linguistic data that the presented systems attempts to handle, originate from twitter, there were frequent occurrences of special characters like "#" and "@". These are not usually encountered in prose texts (like the ones forming the training dataset), hence may pose problems in the correct recognition of named entities. To counter this issue, these special character were replaced by a comma during text pre-processing. Other alternatives, that were explored but did not work optimally, included the total omission of these characters and their replacement by a full stop.

The localisation steps after NER has been performed, that lead to the placement of a map pin, did not require any updates since the original implementation was functional and efficient.

5.1.2 Datasets

The datasets that form part of the evaluation procedure are reported in detail in D5.1. The English dataset did not receive any updates, since the updated methodology provided results which were deemed satisfactory. However, since the same did not apply to the Italian language, certain updates were performed to the respective methodology, with the main differentiation being the addition of more annotated sentences (albeit, with only LOC/ORG annotations) to the utilised dataset (Evalita2009), as described below. Likewise, concerning the Finnish use case, the DIGITODAY dataset served as a basis, on top of which more annotated sentences of our own manual annotation efforts were added.

The task of manual annotation was assigned to PUC1 and PUC3 leaders, i.e. AAWA and FMI, who were requested to mark words of tweet texts as locations (LOC) or organisations (ORG), or leave unmarked in case the words referred to other entities. To assist their effort, an online tool has been created for user-friendly annotating (Figure 14). The user is able to switch languages and get paginated lists of tweets, in sets of 600. As stated in the annex on the left of the tool, by clicking once on a word, it is labelled as location and is coloured green. By clicking twice, the word is labelled as organisation and is coloured red. If the user wishes to cancel the annotation, clicking the word thrice removes the label and is coloured back to black.

The tool offered an easy and fast way to collect annotations, resulting in a valuable dataset of 6,000 labelled tweets in Italian and 6,000 in Finnish. Details of the added annotated material are shared in Table 6. Almost two thirds of the Italian tweets and more than one third of the Finnish tweets were found to contain at least one location, leading to circa 6,000 and 3,500 entities annotated as locations respectively. The tweets that included at least one organisation were relatively fewer for both languages, but still a considerable amount of organisation labels were produced.

Table 6. Results of the annotation

Language	Tweets	Containing LOC	Containing ORG	LOC entities	ORG entities
Italian	6,000	3,716	692	6,157	864
Finnish	6,000	2,199	580	3,530	799

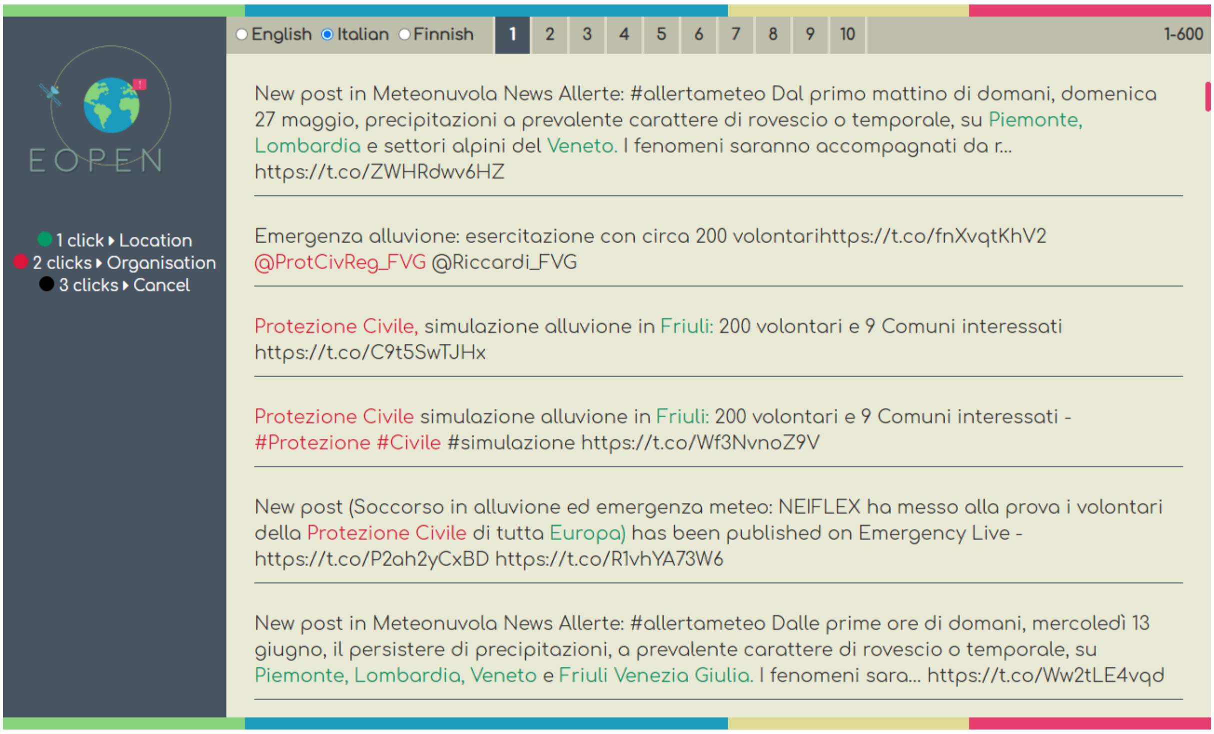

Figure 14. Online tool for annotating locations and organisations

5.1.3 Network parameters and training

The hyperparameters used in the case of Finnish are the same as the ones already reported in D5.1 for English and Italian. Having tested different values, no substantial improvements have been observed with hyperparameter tuning for each approach, which indicates that the best settings have been already determined by the creators of these approaches, as found in the respective papers.

5.1.4 Results

The evaluation of the work done in the localisation task focuses not only in F1-scores, as was the case with D5.1, but also in the time the model requires to handle a relatively short collection of tweets (10 user tweets). This is an important factor when deciding which approach will be implemented in the final system, since analysis needs to be completed swiftly and in real time. The limited number of tweets that were used during evaluation is indicative of an actual use case scenario, where small batches need to be annotated each time. Another factor worth considering when deciding on the most appropriate model is the individual LOC and ORG scores, instead of the overall F1-score for all classes. In some cases, models that presented the best overall F1-score where underperforming in the specific classes; the relative decision was influenced by the performance of individual scores.

English use case: Results achieved in the previous version of the localisation tool were already close to the state-of-the-art for English NER. To improve on those, consequent testing was performed with the addition of BERT and ELMo embeddings to the model pipeline, with the respective scores of the best performing models being visible in Table 7.

Table 7. Performance evaluation of the proposed system (EN) vs. the baseline system and state-of-the-art approaches

System (CoNLL2003)	Precision (%)	Recall (%)	F1-score (%)	Runtime (secs)
Our system (v1)	90.95	90.94	90.97	(not reported)
Our system (BiLSTM-CNN-CRF)	89.92	91.27	90.59	8
Our system BiLSTM-CNN-CRF + ELMo	91.94	92.90	**92.42**	28
Our system BiLSTM-CNN-CRF + BERT	91.27	92.03	91.65	34
Our system (BiLSTM-CRF)	90.58	91.08	90.83	8
Our system BiLSTM-CRF + ELMo	91.69	92.99	92.33	28
Our system BiLSTM-CRF + BERT	91.45	92.39	91.91	34
Our system (BiGRU-CRF)	89.77	90.69	90.22	8
Our system (BiLSTM-CNN)	88.48	90.53	89.49	**6**
Best-scoring shared task system: Florian et al., 2003	88.99	88.54	88.76	(not reported)
Baevski, A. et al. 2019	(not reported)	(not reported)	93.5	(not reported)

Both ELMo and BERT embeddings added extra efficacy to the pipeline and improved previously reported results. During tests the best scores were consistently achieved with the use of the ELMo embedings. In what follows, the ten sentences that were used to evaluate the model's performance are presented with coloured annotation, using the best, non-ELMo/BERT, model. With green font the entities that were correctly identified and with red the ones that were either not annotated at all, or a wrong annotation was assigned (e.g. PER instead of LOC).

Relevant passage (EN – BidLSTM-CNN-CRF):

1. Matteotti square is flooded. #underwater #flooding
2. The sewers are flooded. #Vicenza #flooding
3. #Bacchiglione #flooding #Vicenza The river has overflowed.
4. The levees are cracked at Angeli bridge.
5. Houston fears climate change will cause catastrophic flooding
6. Could see heavy rain and local flooding from storms on Monday in New Jersey ...
7. How quick-thinking mother saved family from Grenfell fire by flooding her flat
8. Flying in over the snow covered fields of Finland was quite magical!

9. current weather in Tampere: light shower snow, -4°C, 92% humidity, wind 3kmh
10. Not only one, many snowploughs coming to the rescue. #oslo

Italian use case: Having already updated the Italian dataset with additional annotation examples (as presented in D5.1), we performed a plethora of evaluation tests to determine which model best serves the needs of EOPEN. The evaluation findings indicate that while approaches that leverage BERT (Polignano, Basile, de Gemmis, Semeraro, & Basile, 2019) did not achieve great success rates in the established dataset, they performed better in the short collection of tweets. This is likely due to the specific implementation of BERT, that was pre-trained on a vocabulary of tweets, instead of prose text. Additionally, the execution time that was required to exploit the extra resources was prohibitive in a real world scenario; the BERT-based models necessitated a considerable amount of extra time (+25 seconds), in comparison to the non-BERT ones, just to load the embeddings in system memory, and thus the specific methodology was not explored any further. Consequently, while we report in Table 8 the results for both BERT and non-BERT models, for the actual EOPEN service we adopt the best non-BERT approach.

Table 8. Performance evaluation of the proposed system (IT) vs. the baseline system and state-of-the-art approaches

System (EVALITA2009)	Precision (%)	Recall (%)	F1-score (%)	Runtime (secs)
Our system (v1)	75.49	75.60	75.37	(not reported)
Our system 2-class (BiLSTM-CNN-CRF)	74.46	75.04	74.75	17
Our system (BiLSTM-CNN-CRF)	78.03	79.48	**78.75**	**14**
Our system (BiLSTM-CRF)	80.94	76.54	78.68	15
Our system (BiGRU-CRF)	74.79	78.21	76.46	17
Our system (BiLSTM-CNN)	73.23	77.41	75.26	15
Our system (BiLSTM-CNN-CRF + BERT)	72.15	74.02	73.07	46
Our system (BiLSTM-CRF + BERT)	73.27	75.47	74.36	41
Our system (BiGRU-CRF + BERT)	71.06	71.55	71.30	40
Our system (BiLSTM-CNN + BERT)	65.72	72.94	69.14	25
DNN: (Basile, Semeraro, & Cassotti, 2017)	82.86	81.82	82.34	(not reported)
Best-scoring shared task system: FBK_ZanoliPianta(Zanoli,	84.07	80.02	82.00	(not reported)

Pianta, & Giuliano, 2009)				
Rerank model: (Nguyen & Moschitti, 2012)	85.99	82.73	84.33	(not reported)

Additionally, tests were also performed with a modified version of the dataset, where the LOC and GPE classes were unified into one, the ORG one remained the same and the PER class was completely omitted. The 2-class accumulated results did not present significant alterations to the ones of the full dataset, with 2% improvement in the ORG class (69.4% 2-class vs 67.7% full) and 2% deterioration in the LOC joint class (79.88% 2-class vs 81.95% full). However, recognition results were significantly ameliorated when testing the model on the ten example tweets, with previously misclassified entities now receiving the correct annotation.

Again, as was the case with the English language, ten example tweets are presented to demonstrate the tool’s progress towards efficient location recognition in Italian. Both normal and 2-class annotation results are illustrated:

Relevant passage (IT – BiLSTM-CNN-CRF):

1. Ventennale dell’#alluvione di #**Sarno**, cosa è cambiato?
2. Dicono che **Genova** è solo rossoblù e fanno il tifo per l’alluvione
3. esiste tifoseria più ritardata del **Napoli**?
4. Presentazione il sistema di #allertameteo della #**ProtezioneCivile** della città di #**Gorizia**
5. Situazione di forte #allertameteo ieri in #**Spagna** per la #grandine.
6. #**Siracusa**, allagamento nel seminterrato dell'ospedale "Rizza"
7. Siamo al 21° anniversario dell'alluvione a **Stazzema**: una targa in ricordo
8. #documentario sull'alluvione #**Firenze** al @AquaFilmFestiva 2017
9. Maltempo, disastro in **Veneto**: dopo il super caldo, ecco l’alluvione.
10. **Ponte Milvio** fa acqua: ancora un allagamento in **via Prati della Farnesina**... #news #**Roma**

Relevant passage (IT – BiLSTM-CNN-CRF with 2-class annotation):

1. Ventennale dell’#alluvione di #**Sarno**, cosa è cambiato?
2. Dicono che **Genova** è solo rossoblù e fanno il tifo per l’alluvione
3. esiste tifoseria più ritardata del **Napoli**?
4. Presentazione il sistema di #allertameteo della #**ProtezioneCivile** della città di #**Gorizia**
5. Situazione di forte #allertameteo ieri in #**Spagna** per la #grandine.
6. #**Siracusa**, allagamento nel seminterrato dell'ospedale "Rizza"
7. Siamo al 21° anniversario dell'alluvione a **Stazzema**: una targa in ricordo
8. #documentario sull'alluvione #**Firenze** al @AquaFilmFestiva 2017
9. Maltempo, disastro in **Veneto**: dopo il super caldo, ecco l’alluvione.
10. **Ponte Milvio** fa acqua: ancora un allagamento in **via Prati della Farnesina**... #news #**Roma**

Finnish use case: An addition to the EOPEN localisation pipeline that is first being reported in the present document, is support for the Finnish language. Currently the fully-fledged implementation is operational, while in D5.1 there was no Finnish-related development done because of time constraints. In Table 9 the reported scores with each of the tested approaches are aligned with the evaluation of the English use case and confirm the conclusions drawn there; the BiLSTM-CNN-CRF model performs equally to the BiLSTM-CRF one, with marginal differences between the two, while the BERT scores provide the best overall performance but at the expense of supplementary operational time.

Table 9. Performance evaluation of the proposed system (FI) vs. the baseline system and state-of-the-art approaches

System (DIGITODAY)	Precision (%)	Recall (%)	F1-score (%)	Runtime (secs)
Our system (BiLSTM-CNN-CRF)	84.14	86.32	85.21	20
Our system (BiLSTM-CRF)	84.77	85.57	85.17	**8**
Our system (BiGRU-CRF)	85.54	86.37	85.95	22
Our system (BiLSTM-CNN)	82.68	84.62	83.64	21
Our system (BiLSTM-CNN-CRF + BERT)	90.42	89.33	89.87	35
Our system (BiLSTM-CRF + BERT)	90.06	90.27	**90.16**	41
Our system (BiGRU-CRF + BERT)	90.70	89.40	90.05	41
Our system (BiLSTM-CNN + BERT)	88.59	89.86	89.22	37
FiNER(Ruokolainen, Kauppinen, Silfverberg, & Lindén, 2020)	90.79	80.25	85.20	(not reported)
FinBERT cased(Virtanen et al., 2019)	91.30	93.52	92.40	(not reported)

In what follows, we present the ten example tweets that were used to assess the efficiency of the adopted model towards Finnish location recognition. The model of choice is a plain BiLSTM-CNN-CRF, without the use of extra resources/embeddings, such as ELMo or BERT.

Relevant passage (FI – BiLSTM-CNN-CRF):

1. Onnistuttu vangitsemaan ohikiitävä hetki, jolloin maassa on lunta. #**Helsinki** #joulu
2. **Turussa** on vain jäätä ja jotain lumen tapaista..#**Tampere** #**Turku**
3. **Oulu** puolilta päivin. Ei oikein kuvassa näy, mutta lunta totta vieköön tuiskusi.
4. Ja siitä, että voin viettää joulunpyhät paikassa, jossa on lunta! #**Jyväskylä** #joulu
5. #**Lahti**, lumi ja ladut yllättivät tänään myönteisesti.
6. #ilmastonmutos vetää henkeä? #**Pori** ssakin lunta!
7. #**Kuopio** **'ssa** sataa lunta ❄siis lisää lunta, entisten kinosten päälle, hautaudutaanko kokonaan lumeen
8. Alkaa olla kivasti lunta #**Vaasa**
9. Lumi riittää jo meidän pihaan! #**Joensuu** #lumi #sää

10. On kyllä kauniin näköistä kun ulkotuli on sulattanut lunta ympäriltään #ulkotuli #talvi #**Lappeenranta**

5.1.5 Conclusions

According to the presented results, along all three languages, it is evident that the application of additional linguistic resources, such as contextual embeddings, favours the selected model, rendering it capable of achieving better recognition results. However, there is a toll in computation efficiency, since the reported runtimes are increased respectively. Hence, to manage an almost real time processing of tweets, the EOPEN service needs to be based on a fast biLSTM implementation. The biLSTM-CNN-CRF approach was the most appropriate candidate for the selected task in all languages, combining great F1 results with manageable processing time.

5.2 Semantic enrichment and mapping

5.2.1 Methodology

For the semantic enrichment, we followed the methodology that we have described in detail in D5.1, which creates a useful interconnection between terms using Babelfy, BabelNet and WordNet technologies taking advantage of their available APIs. More specifically, each term is associated with Babelfy information that correspond to this term using the available Babelfy Java API. From the Babelfy information, apart from all other text-related information, we also extract the BabelNet URL for the entity containing the maximum global score. This URL is used to connect the BabelNet Linked data interface and crawl the WordNet synset identifier. In the end, we use this identifier to extract hypernyms of a three-level range from WordNet.

The new functionality of this procedure is associated with mapping the results into semantic format. Apart from the labels that are found in topics, we have extended this functionality to also enrich the keywords that are extracted from events using the three-level hypernyms of WordNet. In the ontology, we have generated a new class named BabelEntity. For each label, we create a new instance that belongs to the `ta:BabelEntity` class. The instance contains three different properties, each one corresponding to a different level of hypernym that is found in WordNet. For instance, `ta:level1` corresponds to a first level hypernym, `ta:level2` to a second level, etc as shown in Figure 15.

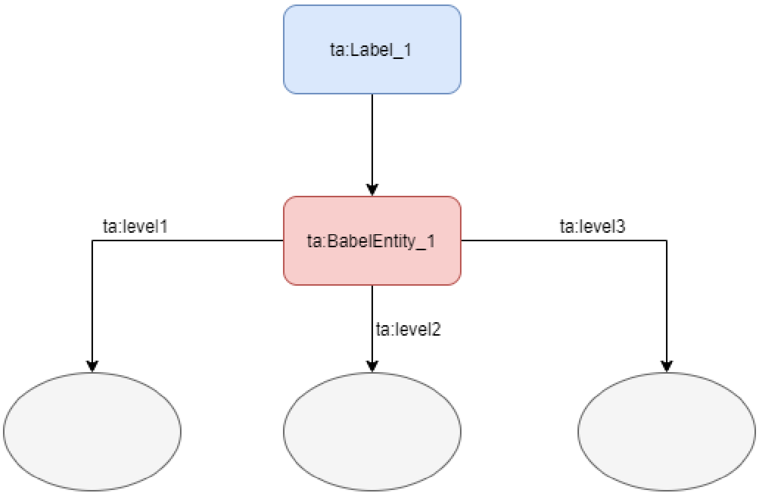

Figure 15. Example of mapping the entities extracted from WordNet

More information about the way that the results are mapped, including examples and execution time, are described in section 5.2.2

5.2.2 Results

A mapping example of a label found in topics and the results of semantic enrichment is shown in the table below. Results are saved in the Knowledge Base.

Table 10. Semantic mapping example for the three-level WordNet results

```
ta:Label_1 ta:hasBabel ta:BabelEntity_1 .


ta:BabelEntity_1 a ta:BabelEntity;
  ta:level1 "07958392-n";
  ta:level2 "00031563-n";
  ta:level3 "00002137-n" .
```

As shown in the next table, the execution time is strongly associated with the entity number that exists in the data. The execution time describes the time that it takes to transform the entities into semantic format, communicate with Babelfy, BabelNet and WordNet and populate the Knowledge Base.

Table 11. Semantic enrichment module execution time according to the entities number

Entities number	Execution time
1	3.13 seconds
5	10.61 seconds
10	27.44 seconds

5.3 EOPEN semantic querying to assist in decision making issues and generate notifications to the end user

5.3.1 Methodology

In order to assist in decision making issues we have developed a service which takes as input polygon coordinates and a specific time period and returns all the data that exist in the Knowledge Base and correspond to the given place and time period. In this way, different data inputs are combined, using as a connection point the geolocation. This functionality creates a useful interconnection between the two different Knowledge Bases which exist in the EOPEN project. The result contains knowledge extracted from all different components which is useful in users of all PUCs i.e. detect flood events and flood map results in a specific area (PUC1), etc., in order to take the appropriate decisions on time and prevent emergency situations. To achieve that, the service runs multiple SPARQL queries as presented below.

Events

The following SPARQL queries are used to retrieve the event data. The query (Table 12) retrieves all event annotation instances that correspond to the specific time period and location, while query (Table 13) retrieves all event-related information like location, language, rate, score etc. The SPARQL FILTER functionality is used to select the data that correspond to a specific time period and distance from a polygon, which is calculated using the `geof:distance` of GeoSPARQL. The latter, calculates the distance between a polygon and a point. In the specific case, the query checks whether the polygon contains the point.

Table 12. Semantic query to retrieve event annotation instances of a specific time period and location

```
PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ta: <https://eopen-project.eu/ontologies/tweet-annotations#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX time: <https://www.w3.org/2006/time#>
select distinct ?r where {
    ?r oa:hasBody ?b.
    ?b rdfs:label ?event.
    ?b ta:location ?a.
    ?a ta:hasPoint ?s.
    ?a rdfs:label ?lbl.
    ?s ogc:asWKT ?o .
    ?r oa:hasTarget ?t.
    ?t ta:language ?lang.
    ?t ta:rate ?rate.
    ?t ta:score ?score.
    ?t ta:timestamp ?ts.

BIND(geof:distance(?o, '''<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
            Polygon ((10.311302241097014 44.23624963837918, 12.270048101694844
44.23624963837918,12.270048101694844 45.21575843334017,10.311302241097014
```

```
45.21575843334017,10.311302241097014
44.23624963837918))'''^^geo:wktLiteral, uom:metre) AS ?distance)
    FILTER(?distance="0.0"^^xsd:double)
   FILTER(xsd:double(?ts)>="1574009907000"^^xsd:double && xsd:double(?ts)<="1588608753524"^^xsd:double)
}
```

Table 13. Semantic query to retrieve all event-related information

```
PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ta: <https://eopen-project.eu/ontologies/tweet-annotations#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX time: <https://www.w3.org/2006/time#>
select * where {
 BIND (<https://eopen-project.eu/ontologies/tweet-annotations#Event_Annotation_dfc11cd9013c4daaacb69b98758944b8> AS ?r)

    ?r oa:hasBody ?b.
    ?b rdfs:label ?event.
    ?b ta:location ?a.
    ?a ta:hasPoint ?s.
    ?a rdfs:label ?lbl.
    ?s ogc:asWKT ?o .
    ?r oa:hasTarget ?t.
    ?t ta:language ?lang.
    ?t ta:rate ?rate.
    ?t ta:score ?score.
    ?t ta:timestamp ?ts.

BIND(geof:distance(?o, '''<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
            Polygon ((10.311302241097014 44.23624963837918, 12.270048101694844
44.23624963837918,12.270048101694844 45.21575843334017,10.311302241097014
45.21575843334017,10.311302241097014
44.23624963837918))'''^^geo:wktLiteral, uom:metre) AS ?distance)
FILTER(?distance="0.0"^^xsd:double)
FILTER(xsd:double(?ts)>="1574009907000"^^xsd:double && xsd:double(?ts)<="1588608753524"^^xsd:double)
}
```

Tweets

The following SPARQL query is used to retrieve the tweet data. The query retrieves information like location, use case, language, id etc. In order to achieve that the SPARQL FILTER mechanism and `geof:distance` of GeoSPARQL are used as described in the previous query. In this case, the query calculates the distance between a point and a polygon.

Table 14. Semantic query to retrieve information related to the tweets

```
PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ta: <https://eopen-project.eu/ontologies/tweet-annotations#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX time: <https://www.w3.org/2006/time#>

select * where {
    ?r oa:hasBody ?b.
    ?r oa:hasTarget ?target.
    ?target ta:location ?a.
    ?target ta:hasId ?id.
    ?a ta:hasPoint ?s.
    ?a rdfs:label ?lbl.
    ?s ogc:asWKT ?o .
    ?b ta:hasUseCase ?usecase.
    ?b ta:hasLanguage ?lang.
    ?b time:inXSDDateTimeStamp ?ts.BIND(geof:distance(?o, '''<http://www.opengis.n
et/def/crs/OGC/1.3/CRS84>
            Polygon ((10.311302241097014 44.23624963837918, 12.270048101694844
44.23624963837918,12.270048101694844 45.21575843334017,10.311302241097014
45.21575843334017,10.311302241097014
44.23624963837918))'''^^geo:wktLiteral, uom:metre) AS ?distance)
    FILTER(?distance="0.0"^^xsd:double)
   FILTER(xsd:double(?ts)>="1574009907000"^^xsd:double && xsd:double(?ts)<="158860
8753524"^^xsd:double)
}
```

Flood map

The following SPARQL query is used to retrieve the flood map data. The query retrieves information like flooded area, percentage, flood map file, polygon coordinates etc., by filtering the data using the SPARQL FILTER mechanism and `geof:distance` of GeoSPARQL are used. The latter, calculates the distance between two multipolygons and in this case checks whether there is an overlap between the two polygons.

Table 15. Semantic query to retrieve information of flood maps

```
PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ta: <https://eopen-project.eu/ontologies/tweet-annotations#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX time: <https://www.w3.org/2006/time#>

select * where {
    ?fl oa:hasTarget ?a.
    ?a ta:hasPolygon ?s.
    ?a rdfs:label ?lbl.
    ?a ta:hasFloodedArea ?flooded_area.
    ?a ta:hasPercentage ?perc.
    ?a ta:hasArea ?area.
    ?s ogc:asWKT ?o .

    ?fl oa:hasBody ?b.
```

```
    ?b ta:comesFrom ?sent.
    ?b ta:hasDate ?ts.
    ?b ta:hasFloodMapFile ?flood_map.
    ?b ta:isFlooded ?flooded.


BIND(geof:distance("<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON
(((10.311302241097014 44.23624963837918, 12.270048101694844
44.23624963837918,12.270048101694844 45.21575843334017,10.311302241097014
45.21575843334017,10.311302241097014
44.23624963837918)))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>, ?o) as ?
distance)

FILTER(?distance="0.0"^^xsd:double)
FILTER(xsd:double(?ts)>="1574009907000"^^xsd:double && xsd:double(?ts)<="158860875
3524"^^xsd:double)
}
```

5.3.2 Results

In this section we present the results of the semantic mapping and querying services. The data that we used to run the experiments are presented in section 6. Table 16 shows the execution time of the mapping service according to the type of input data. The service receives data in JSON format, transforms them into semantic format and saves the results into a Knowledge Base. It is worth noting that in cases like tweet or flood map data the execution time seems to be extremely low, while in cases like topics or events the execution time is significantly higher. The reason behind this is that in cases of topics and events, the semantic enrichment component is also running, to enrich the keywords with WordNet synset information on a three-level range. The procedure described in section 5.2is the one that takes the most time. The execution time of such cases depends on the number of keywords that are given on each JSON.

Table 16. Mapping and saving execution time per data input type

Semantic mapping	Execution Time
Topics	14.40 seconds
Events	19.68 seconds
Tweet	1.76 seconds
Flood map	0.867 seconds

The following table shows the execution time of the service that retrieves the data that are found inside a specific polygon at a specific time period. The service creates a useful interconnection between all the data that are pertinent to the EOPEN project. The execution time is associated with the number of data that exist in the KB and correspond to the specific time period and polygon. In Table 17 we used the data that are presented in section 6.

Table 17. Execution time of semantic querying service

Service	Execution time
Semantic Querying (Retrieve Inside Polygon)	10.59 seconds

6 ONTOLOGY VALIDATION

In this section we present an end-to-end example of the EOPEN annotation model that is used in order to map the different components' results. In each section, we present an example of what the component receives as an input and what is running on back-end. The JSON input is transformed into an RDF format and saved in the triplestore. The services further support the user stories. The following example is related to flood risk assessment which is the purpose of PUC1. The results for events, tweets and flood map are associated with an area in Italy. Topics are also presented in terms of completeness.

6.1 Topics

For the topics that are detected using social media data, the mapping service receives a JSON as the following. The JSON contains information such as general topic information (i.e. use case, language, timestamp), label-related information (text and score) and tweet-related information (i.e. tweet id, top ranked tweets).

Table 18. Topics detection output


```
{
    "timestamp": 1588676261547,
    "usecase": "Floods",
    "language": "English",
    "topics": [
        {
            "id": "2",
            "labels": [
                {
                    "text": "countries",
                    "score": 8.7219454e-11
                },
                { … }
            ],
            "tweets": [
                "1257382158961868802",
                "1257441384728059905"
            ],
            "top_ranked_tweets": [
                "1257441384728059905"
            ]
        }
    ]
}
```


In order to save topic detection data, a post request is needed in https://proto2.eopen.spaceapplications.com/TwitterMapping/converter/topics containing the above JSON as a body. The service creates the appropriate mapping, as described in section 4.2.1`ta:Cluster_Annotation` resource is generated that is linked with the target of the annotation, i.e. the topic `Collection` and the body (`ta:ClusterBody_275b175e06f64edab55bbdefa1de9add`).

Table 19. RDF mapping for Topic detection results

```
@prefix as:    <http://www.w3.org/ns/activitystreams#> .
@prefix oa:    <http://www.w3.org/ns/oa#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix time:  <https://www.w3.org/2006/time#> .
@prefix ta:    <https://eopen-project.eu/ontologies/tweet-annotations#> .

ta:Cluster_Annotation_275b175e06f64edab55bbdefa1de9add
        a              oa:Annotation ;
        oa:hasBody     ta:ClusterBody_275b175e06f64edab55bbdefa1de9add ;
        oa:hasTarget   ta:Collection_275b175e06f64edab55bbdefa1de9add .

ta:BabelEntity_275b175e06f64edab55bbdefa1de9add7
        a           ta:BabelEntity ;
        ta:level1   "07958392-n" ;
        ta:level2   "00031563-n" ;
        ta:level3   "00002137-n" .

ta:Tweet_275b175e06f64edab55bbdefa1de9add2
        a              ta:Tweet ;
        ta:hasUrl      <https://twitter.com/1257441384728059905> ;
        ta:topRanked   true .

ta:Topic_275b175e06f64edab55bbdefa1de9add0
        a           ta:Topic ;
        as:items    ta:Tweet_275b175e06f64edab55bbdefa1de9add1 ,
ta:Tweet_275b175e06f64edab55bbdefa1de9add2 ;
        ta:hasId    "2" ;
        ta:labels   ta:Label_275b175e06f64edab55bbdefa1de9add7.

ta:Tweet_275b175e06f64edab55bbdefa1de9add1
        a              ta:Tweet ;
        ta:hasUrl      <https://twitter.com/1257382158961868802> ;
        ta:topRanked   false .

ta:Collection_275b175e06f64edab55bbdefa1de9add
        a              as:Collection ;
        ta:hasTopics   ta:Topic_275b175e06f64edab55bbdefa1de9add0 .

ta:ClusterBody_275b175e06f64edab55bbdefa1de9add
        a                          ta:ClusterBody ;
        ta:hasLanguage             "English" ;
        ta:hasUseCase              "Floods" ;
        time:inXSDDateTimeStamp    "1588676261547" .

ta:Label_275b175e06f64edab55bbdefa1de9add7
        a             ta:Label ;
        as:accuracy   "8.7219454E-11"^^xsd:float ;
        as:name       "countries" ;
        ta:hasBabel   ta:BabelEntity_275b175e06f64edab55bbdefa1de9add7 .
```

6.2 Events

For the events that are detected using social media data, the mapping service receives a JSON as the following. The JSON contains information such as general topic information (i.e. use case, language, timestamp, score, change), keyword-related information (i.e. text) and location-related information (i.e. location, point).

Table 20. Event detection output


```
{
    "timestamp": 1588608753523,
    "usecase": "Floods",
    "language": "English",
    "score": 0.8,
    "change": -0.96,
    "location": "Lucca",
    "point": {
        "x": 10.311302,
        "y": 44.236248
    },
    "keywords": [
        "spring",
        "flooding"
    ]
}
```


In order to save event detection data, a post request is needed in https://proto2.eopen.spaceapplications.com/TwitterMapping/converter/events containing the above JSON as a body. The service creates the appropriate mapping, as described in section 4.2.2`ta:Event_Annotation` resource is generated that is linked with the target of the annotation, i.e. the `Event Target` and the body (`Topic_dfc11cd9013c4daaacb69b98758944b8`). Each generated event belongs to the type specified in the use case, for instance the event below is of type EventSnow.

Table 21. RDF mapping for Event detection results

```
@prefix oa:    <http://www.w3.org/ns/oa#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix ogc:   <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ta:    <https://eopen-project.eu/ontologies/tweet-annotations#> .

ta:Location_dfc11cd9013c4daaacb69b98758944b8
        a             ta:Location ;
        rdfs:label    "Lucca" ;
        ta:hasPoint   ta:Point_dfc11cd9013c4daaacb69b98758944b8 .


ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b83
        a           ta:BabelEntity ;
        ta:level1   "00935783-v" ;
        ta:level2   "00938019-v" ;
        ta:level3   "null" .

ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b89
        a           ta:BabelEntity ;
        ta:level1   "00217578-v" ;
        ta:level2   "01210571-v" ;
        ta:level3   "01335412-v" .


ta:Point_dfc11cd9013c4daaacb69b98758944b8
        a           ogc:Geometry ;
        ogc:asWKT   "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
```

```
POINT(10.311302 44.236248)^^ogc:wktLiteral" .

ta:Event_Annotation_dfc11cd9013c4daaacb69b98758944b8
        a              ta:EventSnow ;
        oa:hasBody     ta:Topic_dfc11cd9013c4daaacb69b98758944b8 ;
        oa:hasTarget   ta:Event_Target_dfc11cd9013c4daaacb69b98758944b8 .

ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b82
        a           ta:BabelEntity ;
        ta:level1   "00255558-v" ;
        ta:level2   "00255184-v" ;
        ta:level3   "00255753-v" .

ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b88
        a           ta:BabelEntity ;
        ta:level1   "13540166-n" ;
        ta:level2   "00029976-n" ;
        ta:level3   "00001930-n" .

ta:Label_dfc11cd9013c4daaacb69b98758944b83
        a             ta:Label ;
        rdfs:label    "spring" ;
        ta:hasBabel   ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b83 .

ta:Label_dfc11cd9013c4daaacb69b98758944b89
        a             ta:Label ;
        rdfs:label    "flooding" ;
        ta:hasBabel   ta:BabelEntity_dfc11cd9013c4daaacb69b98758944b89 .


ta:Event_Target_dfc11cd9013c4daaacb69b98758944b8
        a              ta:EventTarget ;
        ta:language    "English" ;
        ta:rate        "-0.96" ;
        ta:score       "0.8" ;
        ta:timestamp   "1588608753523" .


ta:Topic_dfc11cd9013c4daaacb69b98758944b8
        a                ta:Topic ;
        rdfs:label       "Snow" ;
        ta:hasKeywords   ta:Label_dfc11cd9013c4daaacb69b98758944b89 ,
ta:Label_dfc11cd9013c4daaacb69b98758944b83 ;
        ta:location      ta:Location_dfc11cd9013c4daaacb69b98758944b8 .
```

6.3 Tweets

For the tweets that are extracted from social media data, the mapping service receives a JSON as the following. The JSON contains information such as general information (i.e. id, use case, language, timestamp) and location-related information (i.e. location, point).

Table 22. Single tweet output


```
{
    "id": "875637434934939648",
    "timestamp": 1588608753523,
    "usecase": "Floods",
```

```
    "language": "English",
    "location": "Lucca",
    "point": {
        "x": 10.311302,
        "y": 44.236248
    }
}
```


In order to save tweet data, a post request is needed in https://proto2.eopen.spaceapplications.com/TwitterMapping/converter/tweet containing the above JSON as a body. The service creates the appropriate mapping, as described in section 4.2.3`ta:Tweet_Annotation` resource is generated that is linked with the target of the annotation, i.e. the `TweetTarget` and the body (`ta:TweetBody_6b30b60df74740e09e3919b02daa3bfa`).

Table 23. RDF mapping for single tweet results

```
@prefix oa:    <http://www.w3.org/ns/oa#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix ogc:   <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time:  <https://www.w3.org/2006/time#> .
@prefix ta:    <https://eopen-project.eu/ontologies/tweet-annotations#> .

ta:Tweet_Annotation_6b30b60df74740e09e3919b02daa3bfa
        a              oa:Annotation ;
        oa:hasBody     ta:TweetBody_6b30b60df74740e09e3919b02daa3bfa ;
        oa:hasTarget   ta:TweetTarget_6b30b60df74740e09e3919b02daa3bfa .

ta:Location_6b30b60df74740e09e3919b02daa3bfa
        a             ta:Location ;
        rdfs:label    "Lucca" ;
        ta:hasPoint   ta:Point_6b30b60df74740e09e3919b02daa3bfa .

ta:Point_6b30b60df74740e09e3919b02daa3bfa
        a           ogc:Geometry ;
        ogc:asWKT   "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
POINT(10.311302 44.236248)^^ogc:wktLiteral" .

ta:TweetTarget_6b30b60df74740e09e3919b02daa3bfa
        a             ta:TweetTarget ;
        ta:hasId      "875637434934939648" ;
        ta:location   ta:Location_6b30b60df74740e09e3919b02daa3bfa .

ta:TweetBody_6b30b60df74740e09e3919b02daa3bfa
        a                        ta:TweetBody ;
        ta:hasLanguage           "English" ;
        ta:hasUseCase            "Snow" ;
        time:inXSDDateTimeStamp  "1588608753523" .
```

6.4 Flood map

For the flood map results that are extracted from sentinel images, the mapping service receives a JSON as the following. The JSON contains information such as general flood map information (i.e. sensing date, flood map, flooded area in square meters, flood percent, etc.) and location-related information (i.e. name, area, polygon coordinates).

Table 24. Flood map output


```
{
    "flood_map": {
        "flood_map":
"S1B_IW_GRDH_1SDV_20191117T165802_20191117T165827_018970_023C98_6CFA_flood_
map.tif",
        "sensing_date": "1574009907000",
        "satellite_constellation": "Sentinel-1",
        "is_flooded": true,
        "whole_area_sq_meters": 39675624900,
        "flooded_sq_meters": 2365000000,
        "flood_percent": 0.05960839,
        "area_geo_info": {
            "type": "Feature",
            "properties": {
                "name": "Vicenza wide region",
                "area": 39675624900
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            10.311302,
                            44.236248
                        ],
                        [
                            12.270048,
                            44.236248
                        ],
                        [
                            12.270048,
                            45.21576
                        ],
                        [
                            10.311302,
                            45.21576
                        ],
                        [
                            10.311302,
                            44.236248
                        ]
                    ]
                ]
            }
        }
    }
}
```


In order to save flood map data, a post request is needed in https://proto2.eopen.spaceapplications.com/TwitterMapping/converter/flood_map containing the above JSON as a body. The service creates the appropriate mapping, as described in section 4.2.4`ta:ChangeDetectionAnnotation` resource is generated that is linked with the target of the annotation, i.e. the `ChangeDetectionTarget` and the body (`ta:ChangeDetectionBody_fce5e2a1fbef470884f8d1ab26426cb9`).

Table 25. RDF mapping for flood map results

```
@prefix oa:    <http://www.w3.org/ns/oa#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix ogc:   <http://www.opengis.net/ont/geosparql#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ta:    <https://eopen-project.eu/ontologies/tweet-annotations#> .

ta:ChangeDetectionBody_fce5e2a1fbef470884f8d1ab26426cb9
        a                   ta:ChangeDetectionBody ;
        ta:comesFrom        "Sentinel-1" ;
        ta:hasDate          "1574009907000" ;
        ta:hasFloodMapFile
"S1B_IW_GRDH_1SDV_20191117T165802_20191117T165827_018970_023C98_6CFA_flood_map.tif
" ;
        ta:isFlooded        "true" .

ta:ChangeDetectionAnnotation_fce5e2a1fbef470884f8d1ab26426cb9
        a              oa:Annotation ;
        oa:hasBody     ta:ChangeDetectionBody_fce5e2a1fbef470884f8d1ab26426cb9 ;
        oa:hasTarget   ta:ChangeDetectionTarget_fce5e2a1fbef470884f8d1ab26426cb9 .

ta:ChangeDetectionTarget_fce5e2a1fbef470884f8d1ab26426cb9
        a                  ta:ChangeDetectionTarget ;
        rdfs:label         "Vicenza wide region" ;
        ta:hasArea         "39675624900" ;
        ta:hasFloodedArea  "2365000000" ;
        ta:hasPercentage   "0.0596083869116325" ;
        ta:hasPolygon      ta:FloodPolygon_fce5e2a1fbef470884f8d1ab26426cb9 .

ta:FloodPolygon_fce5e2a1fbef470884f8d1ab26426cb9
        a           ogc:Geometry ;
        ogc:asWKT   "<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON
(((10.311302 44.236248, 12.270048 44.236248, 12.270048 45.21576, 10.311302
45.21576, 10.311302
44.236248)))^^<http://www.opengis.net/ont/geosparql#wktLiteral>" .
```

6.5 Semantic querying and generate notifications to the end user

In terms of associating all the data that are pertinent to EOPEN, which are stored into different triplestores, we have implemented a service that receives a JSON (Table 26) containing a geometry and two dates (start_date, end_date) corresponding to the time period under observation. To access the service a POST request is required in https://10.6.1.70:8088/RetrieveInsidePolygon/.

Table 26. Example input of the semantic retrieval service


```
{
    "geometry": "POLYGON((10.311302 44.236248, 12.270048
44.236248,12.270048 45.21576, 10.311302 45.21576, 10.311302 44.236248))",
    "start_date": "2019-06-18T04:58:27.000Z",
    "end_date": "2020-07-15T04:12:33.524Z"
}
```

The service runs multiple SPARQL queries, as presented in section 5.3.1and detects the data that correspond to the given time period and area. Data are composed of different components results (namely events, tweets and flood maps). Results are returned using a GEOJSON format and contain all data of the abovementioned services that fulfil the criteria of location and time (Table 27).

Table 27. Example results of semantic retrieval service


```
{
    "features": [
        {
            "geometry": {
                "coordinates": [
                    "10.311302",
                    "44.236248"
                ],
                "geometry_name": "Point_a48eded66ea34195b51e533c7acfbcb0",
                "type": "Point"
            },
            "id": "a48eded66ea34195b51e533c7acfbcb0",
            "type": "Feature",
            "properties": {
                "feature_type": "has_events",
                "score": "0.8",
                "usecase": "Floods",
                "keywords": [
                    "spring",
                    "flooding"
                ],
                "change": "-0.96",
                "language": "English",
                "location": "Lucca",
                "timestamp": "1588608753523"
            }
        },
        {
            "geometry": {
                "coordinates": [
                    "10.311302",
                    "44.236248"
                ],
                "geometry_name": "Point_625b344a17b841cc8280661a7696d89d",
                "type": "Point"
            },
            "id": "Point_625b344a17b841cc8280661a7696d89d",
            "type": "Feature",
            "properties": {
                "feature_type": "has_tweets",
                "usecase": "Floods",
                "language": "English",
                "location": "Lucca",
                "id": "875637434934939648",
                "timestamp": "1588608753523"
            }
        },
        {
            "geometry": {
                "coordinates": [
```

```
                        [⊟
                            "10.311302",
                            "44.236248"
                        ],
                        [⊟
                            "12.270048",
                            "44.236248"
                        ],
                        [⊟
                            "12.270048",
                            "45.21576"
                        ],
                        [⊟
                            "10.311302",
                            "45.21576"
                        ],
                        [⊟
                            "10.311302",
                            "44.236248"
                        ]
                    ],
                    "geometry_name":
"FloodPolygon_e0073fb5e6d24a9daf66b73d06173239",
                    "type": "Polygon"
                },
                "id": "e0073fb5e6d24a9daf66b73d06173239",
                "type": "Feature",
                "properties": {⊟
                    "flooded_sq_meters": "2365000000",
                    "feature_type": "contains_flooded_areas",
                    "location_name": "Vicenza wide region",
                    "flood_map":
"S1B_IW_GRDH_1SDV_20191117T165802_20191117T165827_018970_023C98_6CFA_flood_
map.tif",
                    "sensing_date": "1574009907000",
                    "satellite_constellation": "Sentinel-1",
                    "is_flooded": "true",
                    "whole_area_sq_meters": "39675624900",
                    "flood_percent": "0.05960839"
                }
            }
        ],
        "type": "FeatureCollection"
}
```


In the end, the users can access that unified knowledge that comes from the analysis of tweets and satellite images using a GUI. Users can select a specific area of interest, which can be either a polygon or a bounding box, and define a specific time period that they want to investigate. The result is a map containing all the information (events, tweets and flood maps) that correspond to the given criteria (bounding box and timeframe) (Figure 16).

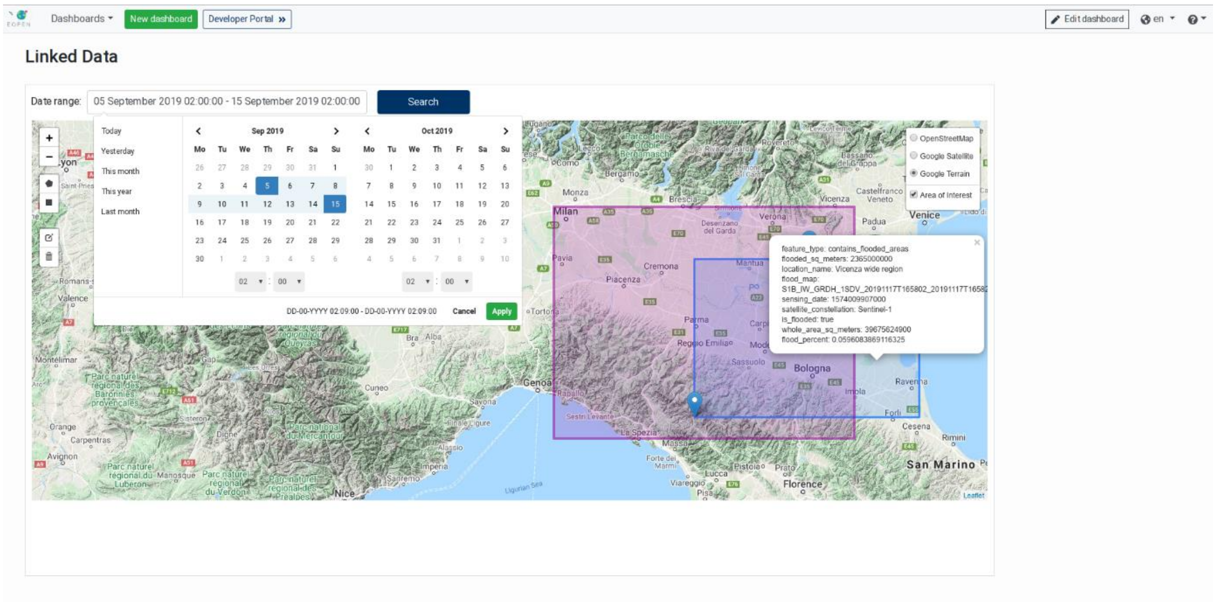

Figure 16. Results of semantic querying service as integrated in the GUI

7 CONCLUSIONS

In this document we described all the updates that have been made since D5.1 “The EOPEN ontology and semantic reasoning support”. The updates on the user requirements and Ontology Requirement Specification Document (OSRD) are presented at the beginning of this document. State-of-the-art technologies have also been presented including the ontologies related to agriculture and earth observation, some querying and reasoning standards, architecture of systems associated with interlinking and a comparison between relational and graph databases.

In the rest of this deliverable the main scope is describing the updates on EOPEN ontology (T5.1) and reasoning for decision support (T5.3). Updates have been made in ontologies where the mapping has been extended to represent more types of data and extra properties. A semantic retrieval service applies geospatial semantic queries in the data that have been saved in the Knowledge Base to extract the data that correspond to a specific place and time. At the end of this document, a full-case ontology validation example is presented related to PUC1. The example presents the representation model that is formed for each component’s results and the semantic retrieval service’s results given a specific time period and geographic coordinates. All data are flood-related. Results are presented to the end users, using a GUI, to assist in decision-making issues related to the environmental monitoring.

REFERENCES

Basile, P., Semeraro, G., & Cassotti, P. (2017). Bi-directional LSTM-CNNs-CRF for Italian sequence labeling. *CEUR Workshop Proceedings*. https://doi.org/10.4000/books.aaccademia.2339

Chiu, J. P. C., & Nichols, E. (2016). Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*. https://doi.org/10.1162/tacl_a_00104

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, *6*(4), 309–322. https://doi.org/10.1016/j.websem.2008.05.001

Horrocks, I., Patel-schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL : A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission 21*.

Hu, S., Wang, H., She, C., & Wang, J. (2011). AgOnt: Ontology for agriculture internet of things. *IFIP Advances in Information and Communication Technology*. https://doi.org/10.1007/978-3-642-18333-1_18

Jaiswal, G. (2013). Comparative analysis of Relational and Graph databases. *IOSR Journal of Engineering*. https://doi.org/10.9790/3021-03822527

Knublauch, H., Hendler, J. A., & Idehen, K. (2011). SPIN - Overview and Motivation W3C: Member Submission 22 February 2011.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*. https://doi.org/10.18653/v1/n16-1030

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*. https://doi.org/10.18653/v1/p16-1101

Maliappis, M. T. (2009). Applying an agricultural ontology to web-based applications. *International Journal of Metadata, Semantics and Ontologies*. https://doi.org/10.1504/IJMSO.2009.026261

Masmoudi, M., Taktak, H., Lamine, S. B. A. Ben, Boukadi, K., Karray, M. H., Zghal, H. B., … Guegan, C. G. (2019). PREDICAT: A Semantic Service-Oriented Platform for Data Interoperability and Linking in Earth Observation and Disaster Prediction. *Proceedings - IEEE 11th International Conference on Service-Oriented Computing and Applications, SOCA 2018*. https://doi.org/10.1109/SOCA.2018.00035

Nguyen, T.-V. T., & Moschitti, A. (2012). Structural reranking models for named entity recognition. *Intelligenza Artificiale*. https://doi.org/10.3233/ia-120038

Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*. https://doi.org/10.18653/v1/P17-1161

Polignano, M., Basile, P., de Gemmis, M., Semeraro, G., & Basile, V. (2019). AlBERTo: Italian BERT language understanding model for NLP challenging tasks based on tweets. *CEUR Workshop Proceedings*.

Ruokolainen, T., Kauppinen, P., Silfverberg, M., & Lindén, K. (2020). A Finnish news corpus for named entity recognition. *Language Resources and Evaluation*. https://doi.org/10.1007/s10579-019-09471-7

Song, G., Wang, M., Ying, X., Yang, R., & Zhang, B. (2012). Study on Precision Agriculture Knowledge Presentation with Ontology. *AASRI Procedia*. https://doi.org/10.1016/j.aasri.2012.11.116

Su, Y. R., Wang, R. J., Chen, P., Wei, Y. Y., Li, C. X., & Hu, Y. M. (2012). Agricultural Ontology Based Feature Optimization for Agricultural Text Clustering. *Journal of Integrative Agriculture*. https://doi.org/10.1016/S2095-3119(12)60064-1

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., … Pyysalo, S. (2019). Multilingual is not enough: BERT for Finnish. *ArXiv*, *abs/1912.07076*.

Wang, C., Wang, W., & Chen, N. (2017). Building an ontology for hydrologic monitoring. *International Geoscience and Remote Sensing Symposium (IGARSS)*. https://doi.org/10.1109/IGARSS.2017.8128433

Yang, J., & Li, G. (2016). Earth Observation data integration and opening system. *International Geoscience and Remote Sensing Symposium (IGARSS)*. https://doi.org/10.1109/IGARSS.2016.7730428

Zanoli, R., Pianta, E., & Giuliano, C. (2009). Named Entity Recognition through Redundancy Driven Classifiers. *Proc. of Evalita*.

Zhu, Y., Zhu, A. X., Song, J., Yang, J., Feng, M., Sun, K., … Zhao, H. (2017). Multidimensional and quantitative interlinking approach for Linked Geospatial Data. *International Journal of Digital Earth*. https://doi.org/10.1080/17538947.2016.1266041

A Appendix

A.1. Software and tools

The final version of the semantic mapping service that has been developed in terms of EOPEN project is available on https://gitlab.com/rousi.maria1/twittermapping. The service receives a JSON file, converts it to RDF and saved the results in GraphDB triplestore. Properties files are provided to connect to the Babelfy API. The final version of the service for querying and retrieving metadata that are associated with a specific time period and location is available on https://gitlab.com/rousi.maria1/RetrieveInsidePolygon. A dockerfile is available on both services since they have been integrated with docker.