Aiteam

NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS

From Pixels to Answers: Leveraging Pythia for Earth Observation Intelligence

Despina-Athanasia Pantazi
Sergios-Anestis Kefalidis
Kostas Plas

Satellite event of Big Data from Space 2025

Overview

- Introduction
- Satellite Event Overview
- Technologies Overview
  - Part 1: From unstructured EO data to RDF with Python
  - Part 2: Storing and Querying RDF data with GraphDB
  - Part 3: Natural Language Interface by employing Pythia
  - Part 4: Visualizing geospatial results with Leaflet.js
  - Part 5: Query execution optimization with JedAI-Spatial and GoST

Introduction

Introduction

AIteam
NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS

CORE EXPERTISE / TECHNOLOGY DOMAIN

- ✓ Artificial Intelligence
- ✓ Earth Observation
- ✓ Question Answering for Knowledge Graphs
- ✓ Artificial Intelligence for the Public Sector

RESEARCH ACTIVITIES / PRODUCTS / SERVICES

- ✓ Semantic Technologies for satellite data
- ✓ Deep Learning
- ✓ Natural Language Processing
- ✓ Databases
- ✓ AI and the Law

Supervisor
Prof. Manolis Koubarakis

Established in: 2012

ORGANISATION TYPE
Academic

STAFF: 15

ADDRESS:
University Campus, Ilisia
Athens, 157 84

Email: koubarak@di.uoa.gr
Web: https://ai.di.uoa.gr

Introduction

- **MAJOR SPACE-RELATED PROJECTS**

✓ FAIR2Adapt

✓ AI4Copernicus

✓ ExtremeEarth

✓ DA4DTE

- **MAIN PARTNERS**

✓ European space agency

✓ National observatory of Athens

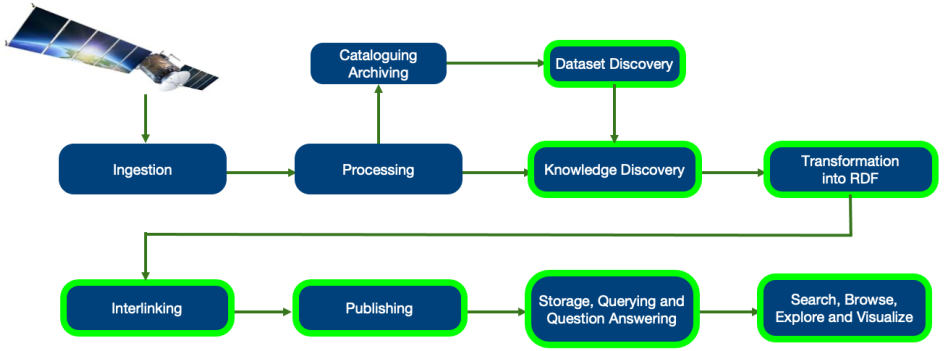
✓ European center for medium-range weather forecasts

- **MAIN CUSTOMERS**

✓ European space agency

✓ Users in European projects

AIteam
NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS



eosc | FAIR2Adapt

AI4 copernicus

AI on Demand

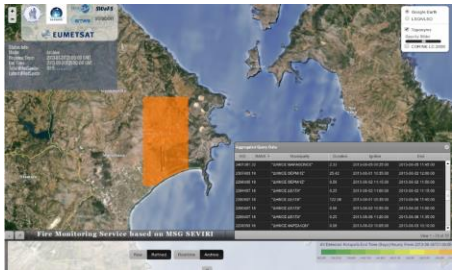
EXTREME EARTH

Introduction

AIteam
NATIONAL & KAPODISTRIAN
UNIVERSITY OF ATHENS

- **STRATEGIC PRIORITIES**

✓ Development of knowledge graphs, question answering engines and chatbots for the space domain.



- **LONG TERM VISION**

✓ Applying modern Artificial Intelligence techniques to Earth Observation

✓ Implementation of applications with societal and economic value

Satellite Event Overview

Satellite Event Overview

- Recent advances in satellite technology have enabled frequent, high-resolution, and global-scale monitoring of the Earth. This explosion in the availability of **Earth Observation (EO) data** presents opportunities to support critical societal challenges.
    - Artificial Intelligence, knowledge graphs, and semantic technologies are emerging as powerful enablers for bridging the gap between **raw EO data** and **actionable insights**.

- This tutorial will walk participants through the full pipeline of **building intelligent EO services**—from ingesting raw satellite data to creating responsive, natural language-powered applications, using technologies and tools that are being applied to research projects like FAIR2Adapt.

- The event is hands-on, enabling participants to learn by doing, while also offering a blueprint that can be replicated across other regions and domains.

Satellite Event Overview

Schedule: A half-day tutorial (3 hours):

Time	Session
09:00 - 09:30	**Introduction** Welcome and introduction by the presenters. Presentation of tools and challenges to be demonstrated.
09:30 - 10:30	**From Earth Observation Data to the Triple Store** Part 1: From Unstructured EO Data to RDF with Python Part 2: Storing and Querying RDF Data with GraphDB
10:30 - 11:00	**Coffee Break**
11:00 - 11:40	**Pulling down the SPARQL barrier for RDF access** Part 3: Developing a Natural Language Interface
11:40 - 12:00	**Visualization of results** Part 4: Visualizing Geospatial Results with Leaflet.js
12:00 - 12:20	**Optimizing for realtime performance** Part 5: Query execution optimization with JedAI-Spatial and GoST
12:20 - 12:30	**Summary and discussion**

Satellite Event Overview

- The tutorial will cover tools and methodologies to **transform** raw EO data into semi-structured RDF data which can be easily and efficiently queried by both novice and expert users.
- Although the tutorial will be grounded in a use case focused on Hamburg, Germany, the methodologies and tools presented are generalizable to EO scenarios across domains and geographies.
- You can follow along hands-on during the tutorial sessions and execute the code in real-time.
- Runnable code checkpoints will be provided to ensure that participants can easily set up their environments and follow the tutorial without any issues.

Technologies Overview

Part 1: From unstructured EO data to RDF with Python

- Starting point: take raw Earth Observation datasets (satellite images with accompanying metadata) and converting them into the RDF format using a simple ontology.
  - OpenStreetMap (OSM) data and satellite images from Sentinel-1 and Sentinel-2, provided by the project DA4DTE and used in the project FAIR2Adapt.
- This transformation is performed using **toposkg_lib**, a custom semantic parsing library developed by our group, capable of extracting structured knowledge from unstructured EO data.
- **toposkg_lib** is part of the **Topos framework** and its tool chain for handling common tasks in knowledge graph construction and interlinking.
  - The tools are made available via a Python library, a desktop application, and a chatbot implemented through LLM function calling.

toposkg_lib: https://pypi.org/project/toposkg/0.1.2/

Part 1: From unstructured EO data to RDF with Python

- **Topos framework** was also used to develop **ToposKG**, a comprehensive geospatial knowledge graph that integrates:
  - topologically consistent administrative division data.
  - general purpose geospatial information.
  - natural resource data including lakes, rivers, forests and mountains.

Natural Resource	Data Source	Number of Entities	Disk Footprint
Mountains	GMBA	5,629	201MB
Water Bodies	OSM	20,426,694	34GB
Seas	MarineRegions	101	339MB
Forests	OSM	15,449,268	23G
Administrative	GAUL	48,316	4.2GB
Administrative	OSM	1,583,535	25.3GB
POIs	OSM	43,885,054	14GB

Part 1: From unstructured EO data to RDF with Python

- The **Resource Description Framework (RDF)** is data model for representing information (especially **metadata**) about **resources** in the Web.

- **RDF** is intended for situations in which information about Web resources needs to be **processed by applications**, rather that being only displayed by people.

- **RDF** is based on the idea of identifying resources using **Web identifiers** and describing **resources** in terms of simple **properties** and property **values**.
  - To identify resources, RDF uses **Uniform Resource Identifiers (URIs)** and **URI references (URIrefs)**.
  - The **resources** being described have **properties** which have **values**.

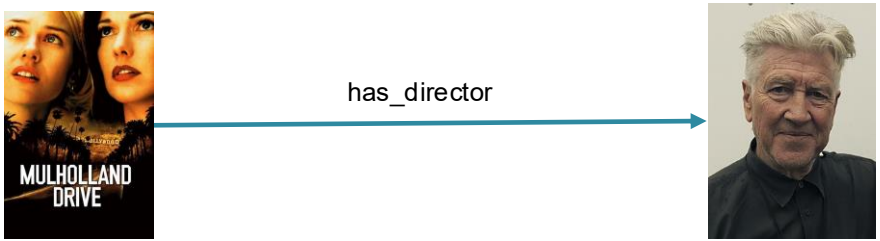
https://www.w3.org/RDF/

Part 1: From unstructured EO data to RDF with Python

Terminology:

- The part that identifies the thing the statement is about is called the **subject**.
- The part that identifies the property or characteristic of the subject that the statement specifies is called the **predicate**.
- The part that identifies the value of that property is called the **object**.

Example: How do I represent this fact in RDF?



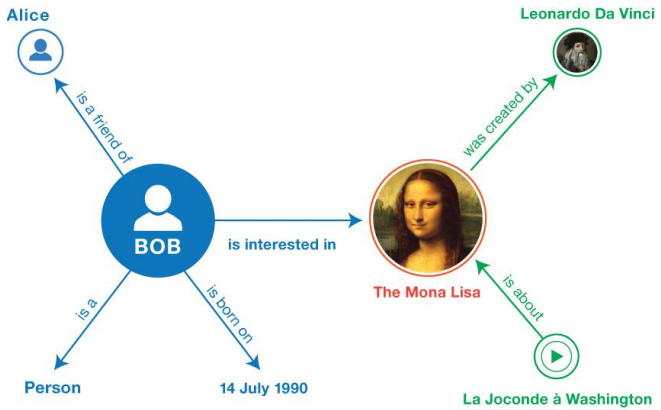
Subject: Mulholland_Drive_film **Predicate:** has_director **Object:** David_Lynch

Part 1: From unstructured EO data to RDF with Python

- **RDF graphs:** RDF models statements by **nodes** and **edges** in a **graph**.

- In the RDF graph notation, a **statement** is represented by:
  - a node for the **subject**
  - a node for the **object**
  - an edge for the **predicate**, directed from the subject node to the object node.

- Example:



Part 1: From unstructured EO data to RDF with Python

- Being a powerful and expressive framework for representing data, **RDF** is used for building **knowledge graphs** – richly interlinked, interoperable and flexible information structures.

- The heart of the **knowledge graph** is a knowledge model – a collection of interlinked descriptions of concepts, entities, relationships and events where:

    - Descriptions have formal semantics that allow both people and computers to process them in an efficient and unambiguous manner.

    - Descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities related to it.

    - Diverse data is connected and described by semantic metadata according to the knowledge model.

Part 2: Storing and Querying RDF data with GraphDB

- The newly-formed RDF data, along with geospatial data concerning administrative divisions and natural features, is loaded into **GraphDB**.
- In addition to the built-in GeoSPARQL support, we present a plug-in for supporting **stSPARQL**.
- These extensions allow the representation and querying of geometries (points, polygons, etc.) using geospatial functions, making **GraphDB** ideal for storing earth observation data with spatial dimensions.

GraphDB: https://graphdb.ontotext.com

Part 2: Storing and Querying RDF data with GraphDB

- Ontotext **GraphDB** is a highly efficient, scalable and robust graph database with RDF and SPARQL support.
  - ✓ It is one of the few triplestores that can perform real-time semantic inferencing at scale.

- **stRDF** is an extension of RDF for the representation of geospatial information that changes over time. **stSPARQL** is the query language for **stRDF.**
  - ✓ Spatial data types are introduced.
  - ✓ Geospatial information is represented using spatial literals of these datatypes.
  - ✓ OGC standards Well-known text (WKT) and Geography Markup Language (GML) are used for the serialization of spatial literals.

Manolis Koubarakis, Kostis Kyzirakos: **Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL**.ESWC (1) 2010: 425-439. http://dx.doi.org/10.1007/978-3-642-13486-9_29.

Part 2: Storing and Querying RDF data with GraphDB

Example: A **strRDF** triple derived from **GeoNames** that represents information about the Greek town 'Olympia' including an approximation of its geometry.

- **GeoNames** (www.geonames.org/) is a geographical database that covers all countries with >11M placenames. Data is available for download free of charge.

```
PREFIX geonames: <http://geonames.org/>
PREFIX noa: <http://noa.gr/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>

geonames:264637 geonames:name "Olympia";
        rdf:type noa:Town;
        strdf:hasgeometry "POLYGON((21.5 18.5, 23.5 18.5, 23.5
21, 21.5 21, 21.5
18.5))<http://www.opengis.net/def/crs/EPSG/0/4326>"^^strdf:WKT.
```

Part 2: Storing and Querying RDF data with GraphDB

- **Example** (cont'd): Three **strRDF** triples produced from the processing chain of the National Observatory of Athens that represent burnt areas.

```
PREFIX noa: <http://noa.gr/>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>


noa:BA1 a noa:BurntArea;
strdf:hasgeometry "POLYGON((20 20,20 22,22 22,22 20,20 20))"^^strdf:WKT.

noa:BA2 a noa:BurntArea;
strdf:hasgeometry "POLYGON((23 18, 24 19, 23 19, 23 18))"^^strdf:WKT.

noa:BA3 a noa:BurntArea.
strdf:hasgeometry "POLYGON((20 15, 21 15, 21 16, 20 15))"^^strdf:WKT.
```

Part 2: Storing and Querying RDF data with GraphDB

- **Example** (cont'd): Demonstration of the functionality of stSPARQL by posing the query: "Return the names of towns that have been affected by fires".

```
SELECT ?name WHERE {
?town rdf:type noa:Town;
      geonames:name ?name;
      strdf:hasGeometry ?townGeom.
?ba rdf:type noa:BurntArea;
    strdf:hasGeometry ?baGeom.
FILTER(strdf:overlap(?townGeom ?baGeom))}
```

➢ **Result:**

?name
"Olympia"

Part 3: Natural Language Interface by employing Pythia

- Now that the RDF data is ready, participants will integrate **Pythia**, a geospatially-aware question-answering engine that:
  - Is designed to function as a plug-and-play question answering solution for general-purpose and specialized knowledge graphs
  - Does not require fine-tuning, making it a viable solution for use cases that lack training datasets.
- To achieve this, **Pythia** introduces a novel combination of agentic reasoning and bi-directional graph search to identify and explore connections between nodes relevant to the questions.
- Extensive experiments show that **Pythia** achieves state-of-the-art performance on multiple datasets and knowledge graphs, along with transferability and the ability to handle geospatial queries.

Pythia: https://github.com/SKefalidis/Pythia

Part 3: Natural Language Interface

- One more tool is **TerraQ**, a Text-to-SPARQL system which provides a natural language interface for RDF stores.
  - It does so by translating natural language questions into semantically equivalent SPARQL queries.

- **TerraQ** is a question-answering engine that simplifies interaction with EO data, by removing the need for users to understand SPARQL. As a result, EO data is made more easily accessible to users, while maintaining accuracy and computational efficiency
  - Interacts using only natural language, no expert knowledge is required.
  - **Sentinel mission images**: Access to 1.5M Sentinel-1 and Sentinel-2 images of Europe.
  - Built using a range of industry standard and custom built tools to ensure a rapid execution.
  - Open source.

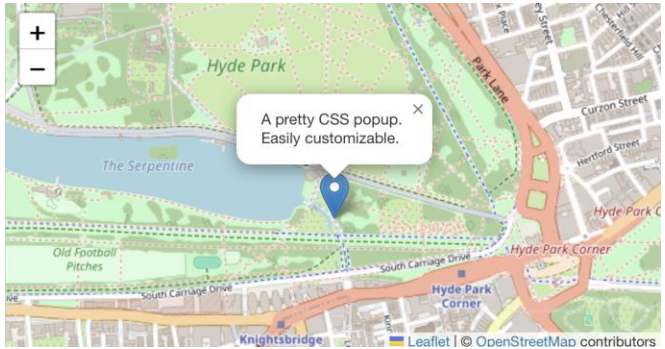
- BiDS Presentation: A multi-agent system to orchestrate interactions with Digital Twins of Earth October 1st 11:45 AM.

TerraQ: https://terraq.di.uoa.gr

Part 4: Visualizing geospatial results with Leaflet.js

- Until now, results are presented in a raw, CSV-like format. When dealing with satellite image data, it is important to be able to see the images and their visualizations on a map.

- We show how such functionality can be implemented using **Leaflet.js**:
  - It is an open-source JavaScript library for interactive maps.
  - It works efficiently across all major desktop and mobile platforms.
  - It can be extended with plugins and it has a well-documented API.



Part 5: Query execution optimization with JedAI-spatial & GoST

- For the last part of our tutorial we showcase the complexity of geospatial calculations when using large polygons. This results in long response times, making our system frustrating to use.

- To deal with this we employ geospatial interlinking and query rewriting to speed-up query execution.

- Tools: **JedAI-spatial**, **GoST**

Part 5: Query execution optimization with JedAI-spatial & GoST

- **JedAI-spatial** is an open-source system for computing topological relations according to the DE9IM model between datasets with geometric entities.

- **GoST - GeoSPARQL to SPARQL Transpiler** is a transpiler converting GeoSPARQL queries to SPARQL queries using geospatial materialized relations.

Marios Papamichalopoulos, George Papadakis, George Mandilaras, Maria Siampou, Nikos Mamoulis, Manolis Koubarakis. **Three-dimensional Geospatial Interlinking with JedAI-spatial**. Journal of Web Semantics, Volume 81, July 2024. https://doi.org/10.1016/j.websem.2024.100817

Thank you!

Find us in our web page: https://ai.di.uoa.gr/
Or follow us on twitter @AITeamUoA

Aiteam
NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS