DA4DTE: DEVELOPING A DIGITAL ASSISTANT FOR SATELLITE DATA ARCHIVES

Marco Corsi[1], *Giorgio Pasquali*[1], *Chiara Pratola*[1], *Simone Tilia*[1], *Sergios-Anestis Kefalidis*[2], *Konstantinos Plas*[2], *Mariangela Pollali*[2], *Eleni Tsalapati*[2], *Myrto Tsokanaridou*[2], *Manolis Koubarakis*[2], *Kai Norman Clasen*[3,4], *Leonard Hackel*[3,4], *Jakob Hackstein*[3,4], *Gencer Sumbul*[3], *Begüm Demir*[3,4] *and Nicolas Longépé*[5],

[1]e-GEOS S.p.A., Italy
[2]Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece
[3]Technische Universität Berlin, Germany
[4]BIFOLD - Berlin Institute for the Foundations of Learning and Data, Germany
[5]Φ-lab, ESA ESRIN, Frascati, Italy

ABSTRACT

We present the project DA4DTE, a project funded by the European Space Agency to create a digital assistant for satellite data archives. The digital assistant offers four search engines for satellite images (search by image, search by caption, visual question answering and knowledge graph question answering) which are orchestrated by a task interpreter in order to answer complex requests for users looking for Earth observation data.

Index Terms— digital assistant, search by image, search by caption, visual question answering, knowledge graph question answering

1. INTRODUCTION

With recent advances in natural language processing, conversational AI and knowledge graphs which are based on large language models such as BERT [1] and the GPT family [2], large North American companies have developed digital assistants (e.g., Alexa from Amazon, Cortana from Microsoft and Siri from Apple) and chatbots such ChatGPT from OpenAI, Bard from Google and Claude from Anthropic. These systems are able to answer questions and engage in dialogue on topics ranging from the weather, sports etc., to more serious topics such as programming, mathematics, medicine etc. Some of these systems, through appropriate interfaces, can also carry out tasks in the real world such as ordering products from electronic shops.

In comparison with the above state of the art, there is currently *no* satellite data provider that offers a digital assistant to guide users in finding the Earth observation data that they are interested in. This is a functionality that is urgently needed so that the wealth of Earth observation data made available by programs such as Copernicus and Landsat, are easily accessible to expert and non-expert users (e.g., journalists that want to find the latest satellite image covering an area affected by an environmental catastrophe). To fill this gap, the European Space Agency issued the call for tender AO/1-11055/21/I-EF-"Demonstrator Precursor Digital Assistant Interface For Digital Twin Earth" on February 7, 2022. The tender was won by our consortium led by the Italian company eGEOS, with the National and Kapodistrian University of Athens and the TU Berlin as subcontractors. The funded project is called "DA4DTE: Demonstrator Precursor Digital Assistant Interface For Digital Twin Earth". It started in October 2022 and will end in March 2024 (duration 18 months).

DA4DTE will develop *the first digital assistant for satellite image archives* and demonstrate it in three use cases (vessel detection, water bodies dynamics, training dataset construction). DA4DTE builds on recent work of the academic project partners on deep learning techniques for satellite images, search engines for satellite images [3], visual question answering [4], question answering over knowledge graphs and linked geospatial data [5, 6, 7], and question answering engines for satellite data archives [8].

This paper presents the current status of the project DA4DTE and discusses the work to be carried out until the end of the project.

2. THE ARCHITECTURE OF THE DIGITAL ASSISTANT

The digital assistant high-level architecture is structured in three main components:

- *API*: It is an entry point both for user queries and user interface. It is responsible for the activation of the back-

This work was supported by ESA project DA4DTE (contract no. 4000139212/22/I-EF). Eleni Tsalapati has received funding from the EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie GA No 101032307.

end components that work to return meaningful results to natural language queries focused on the use cases.

- *Content-Based Engines:* The content-based engines are the functionalities directly used by the API for query and search. They are at the heart of the digital assistant solution and are described in the next section.
- *Catalogue:* It is the component used to store metadata about the different resources that will be used by the API, e.g., it will host metadata for Earth Observation (EO) data and auxiliary data. EO data include SpatioTemporal Assets Catalogue (STAC)[1] metadata that contain references to the data stored by the engines (e.g., metadata of images where certain features are present). Auxiliary data include datasets used as a reference for semantic purposes (land cover, geospatial features, etc.) extracted from input images.

The architecture of the digital assistant is thus divided into five main logical sub-components, as shown in Figure 1. Each component is responsible for a specific function:

1. *Digital Assistant User Interface*. It is the main entry point to interact with the system and it is based on the digital assistant API. It performs the search and visualizes the results on a web user interface.
2. *Digital Assistant API*. It exposes a user endpoint to perform a query using natural language and returns answers by providing meaningful results in the form of natural language messages and structured data/metadata.
3. *Digital Assistant Catalogue*. The Catalogue is composed of the STAC Database, used to store metadata, and hash tables, used by the Image-based engines (described below) to compute the search and retrieval tasks. It can be queried using STAC API. In this way, the digital assistant can return standard-based metadata content that can be displayed in the user interface (e.g., STAC metadata is a GeoJSON file that can easily be displayed on a mapping interface).
4. *Image-based Engines*. There are two engines of this kind: 'search by image' and 'search by text' engine.
5. *Question Answering Engines*. There are two engines of this kind: the knowledge graph question answering engine and the visual question answering engines.

3. THE TASK INTERPRETER AND THE SEARCH ENGINES

In this section we briefly present the functionality of the four search engines that are used by the digital assistant and, then, we describe the task interpreter that orchestrates these back-end engines.

[1] https://stacspec.org/en

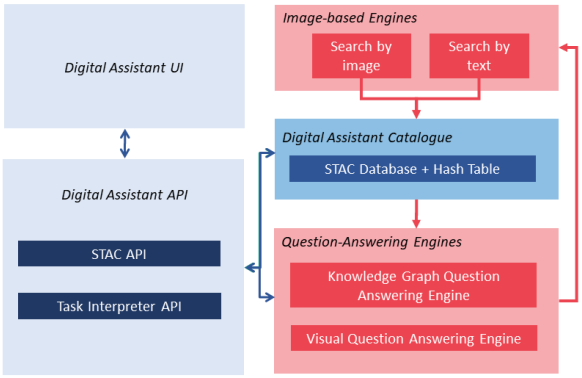

Fig. 1. Digital assistant system architecture

Search by image. A "search by image" engine takes a query image and computes the similarity function between the query image and all archive images to find the most similar images to the query in a scalable way. This is achieved based on two main steps: i) the image description step, which characterizes the spatial and spectral information content of remote sensing (RS) images; and ii) the image retrieval step, which evaluates the similarity among the considered hash codes and then retrieves images similar to a query image in the order of similarity. During DA4DTE, we develop a "search by image" engine based on two self-supervised methods: 1) deep unsupervised cross-modal contrastive hashing (DUCH) [9]; and 2) cross-modal masked autoencoder (CM-MAE) [10, 11] methods. For both methods, the image description step is composed of two modules: 1) feature extraction module, which learns deep feature representations of RS images by exploiting visual transformers (ViT) [12]; and 2) deep hashing module, which learns to map image representations into hash codes. The first module of the DUCH method is based on contrastive self-supervised image representation learning, while that of the CM-MAE method is based on unsupervised masked image modelling. The second module of each method employs a hashing subnetwork with binarization loss functions. Our engine has both the single-modal (also known as uni-modal) and cross-modal content-based image retrieval capability due to the consideration of the modality-specific encoders.

Search by text. A 'search by text' (caption) engine takes a text sentence to search for images, achieving cross-modal text-image retrieval. During DA4DTE, we develop a "search by text" engine by adapting the above-mentioned self-supervised DUCH and CM-MAE methods to be operational on text based queries. To this end, we apply some adaptation. The feature extraction module of each method is adapted to extract feature representations of image-text pairs by exploiting

bidirectional transformers (e.g., BERT [1]) as text-specific encoders together with ViTs as image-specific encoders. The second module of each method is adapted to learn cross-modal binary hash codes for image and text modalities by simultaneously preserving semantic discrimination and modality-invariance in an end-to-end manner.

Visual question answering. Visual question answering (VQA) enables users to ask questions about the content of RS images in a free-form manner, extracting valuable information. In this context, an efficient and accurate VQA model called LiT-4-RSVQA [4], based on a lightweight transformer architecture, has been developed by the TU Berlin and will be used for DA4DTE. The LiT-4-RSVQA architecture comprises several key components: i) a lightweight text encoder module, ii) a lightweight image encoder module, iii) a fusion module, and iv) a classification module. To assess the performance of the VQA model, we will use data from the DA4DTE use cases.

Knowledge graph question answering. The Knowledge Graph Question Answering Engine (KGQA) accepts questions in natural language (English) that ask for satellite images satisfying certain criteria and returns links to such datasets. The questions can refer to image metadata but also to geographic entities (e.g., the Loch Ness Lake or the city of Munich) both of which are included in the target knowledge graph (containing geospatial data from GADM[2] and OpenStreetMap, hashcodes and image metadata). In this way, the users can request things like "Show me all Sentinel-2 images from February 2021 that show rivers in the Emilia Romagna region and have cloud cover less than 10%". KGQA initially processes the input question and then translates it to a GeoSPARQL query, which is executed over a Strabon [13] endpoint returning the final results. For question processing, REL [14] is used for named entity recognition and disambiguation. For the identification of classes of objects (e.g., river, region), geospatial relations (e.g., in), properties (e.g., area of a region), and metadata (e.g., cloud cover), a rule-based approach based on constituency and dependency parsing is employed. A GeoSPARQL query is, then, generated by combining neural network techniques with a dynamic template-based approach. Subsequently, this query is rewritten by GoST [3] to utilize materialized relations.

The Task Interpreter. The main target of the Task Interpreter is to enable the digital assistant to act as an integrated whole instead of a number of individual engines, as far as being the intermediary between the user and them. For each user request, the Task Interpreter selects one or more of the backend engines to activate, mainly based on text similarity (utilizing pre-trained BERT transformers) with requests that can be answered from each of them. For more particular characteristics of engines' functionalities, other techniques are used, e.g., named entity recognition is performed with SpaCy[4]. If the user does not intend to query any of the engines, the task interpreter presents a conversational functionality, which is performed via chat models [15, 16]. Additional capabilities offered by the Task Interpreter include asking for clarifications on ambiguous user requests, enhancing backend engine responses before presenting them to the user, and keeping track of conversations in a memory cell.

[2] https://gadm.org/

[3] https://github.com/AI-team-UoA/GoST

[4] https://spacy.io/

4. USE CASES

In this section we present the three use cases that will demonstrate the value of the digital assistant.

UC1: Vessel detection. It can be very useful for users that want to monitor illegal activities (e.g. illegal fishing in restricted areas) or for maritime surveillance. Moreover, it could be also useful to detect suspicious vessels, e.g., comparing vessels detected in satellite image with the AIS data. The goal of the use case is to make accessible and easy to use Sentinel-1 and Sentinel-2 data, especially focusing the maritime domain, for both expert and non-expert users.

UC2: Water bodies dynamics. The digital assistant can help non-expert users to easily retrieve satellite data containing water bodies or flood events and, if requested, also to extract them, allowing their localisation. Moreover, the digital assistant is useful also to expert users, such as SatCen and operators of Copernicus EMS programme, to help them retrieve faster satellite data with flood events from the catalogue and extract floods in a very simple way, helping their work on creating maps.

UC3: Training dataset construction for AI models. The digital assistant makes the dataset retrieval simpler, allowing the user to specify what type of dataset they want to generate through a simple query, e.g., "Construct a dataset of Sentinel-2 for the last three months with water bodies": thus, the digital assistant will retrieve all the patches coming from Sentinel-1 or Sentinel-2 data containing the specific class requested. Furthermore, the digital assistant makes it possible to construct a dataset starting from an example image or images, thus creating a dataset visually or semantically similar to the input image/images.

5. THE DIGITAL ASSISTANT IN ACTION

The digital assistant can handle different types of requests on EO data. A user's request expressed in natural language, can activate different engines. The request can be made using either the UI or the API and, similarly, the results can be downloaded from both of them. In this section, the operation of the digital assistant is shown in a real-case scenario (see workflow in Figure 2), which is described below in detail:

1. The user poses the following request to the digital assistant: "Show me all images with ships near the port of Genoa".

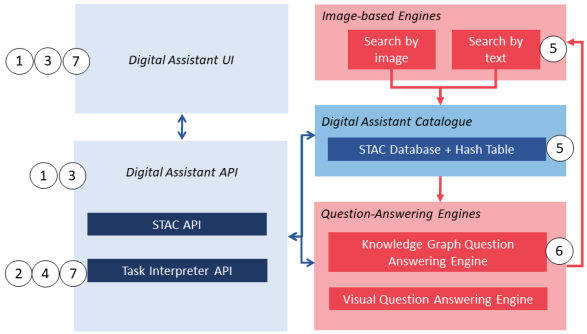

Fig. 2. An example query and its workflow through the digital assistant architecture

2. The API forwards the query to the Task Interpreter which asks for disambiguation: "Please repeat the request by replacing "near" with a specific distance".
3. The user restates: "Show me all images with ships, within 100 km from the port of Genoa".
4. The API forwards the clarified request to the Task Interpreter, which decides which engines should be activated and parses the sentence into two separate tasks to be performed by the 'search by text' and KGQA engines, respectively: (i) Retrieve all images containing vessels. (ii) On the retrieved images, find all those that are within 100 km of the port of Genoa.
5. The 'search by text' engine gives the hash codes of the images containing vessels to KGQA.
6. The KGQA engine filters these images, based on the distance from the geographical object of "the port of Genoa".
7. All images within 100 km of the port of Genoa that contain vessels are displayed on the Digital Assistant UI and/or can be downloaded via the Digital Assistant API.

Similar to this example, other types of requests can be fulfilled by activating the VQA engine or the 'search by image' engine, e.g., in queries where a specific question about an image is expressed or where a search for images similar to a given example is required. In addition, it is also possible to directly extract features from the searched images by asking the digital assistant to count or extract objects, such as vessels or water bodies.

6. SUMMARY

We presented project DA4DTE, funded by ESA, which develops a digital assistant for satellite image archives and applies it to three use cases.

REFERENCES

[1] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NACCL*, 2019.

[2] OpenAI, "Gpt-4 technical report," 2023.

[3] A. K. Aksoy et al., "Satellite image search in AgoraEO," *PVLDB*, vol. 15, no. 12, 2022.

[4] L. Hackel et al., "LiT-4-RSVQA: Lightweight transformer-based visual question answering in remote sensing," in *IEEE IGARSS*, 2023.

[5] Punjani, D. et al., "The question answering system geoqa2," in *2nd International Workshop on Geospatial Knowledge Graphs and GeoAI: Methods, Models, and Resources*, 2023.

[6] Kefalidis, S.-A. et al., "Benchmarking geospatial question answering engines using the dataset GeoQuestions1089," in *ISWC*, 2023.

[7] M. Koubarakis, Ed., *Geospatial data science: a hands-on approach based on geospatial technologies*. ACM Books, 2023.

[8] D. Punjani et al., "EarthQA: a question answering engine for Earth observation data archives," in *IEEE IGARSS*, 2023.

[9] G. Mikriukov et al., "Unsupervised contrastive hashing for cross-modal retrieval in remote sensing," in *ICASSP*, 2022.

[10] K. He et al., "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.

[11] G. Kwon et al., "Masked vision and language modeling for multi-modal representation learning," in *ICLR*, 2023.

[12] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[13] K. Kyzirakos et al., "Strabon: A semantic geospatial DBMS," in *ISWC*, 2012.

[14] J. M. van Hulst et al., "REL: an entity linker standing on the shoulders of giants," in *SIGIR*, 2020.

[15] Stephen Roller et al., "Recipes for building an open-domain chatbot," in *EACL*, 2021.

[16] Y. Zhang et al., "DIALOGPT : Large-scale generative pre-training for conversational response generation," in *ACL*, 2020.