MS6. System and module-level architecture development

Version 0.4

Documentation Information

Contract Number	2018-EU-IA-0104
Project Website	www.saintgeorgeonabike.eu
Contratual Deadline	31/08/2020
Nature	Report
Author	Maria-Cristina Marinescu (BSC)
Contributors	Artem Reshetnikov (BSC), Mónica Marrero (EF), Antoine Isaac (EF), Joaquim Moré (BSC)
Reviewer	José Eduardo Cejudo Grano de Oro (EF), Albin Larsson (EF), Ariadna Lobo (BSC)
Keywords	Object detection, caption classifier, search API, transfer learning, language model, semantic meta-data

Co-financed by yhe Connecting Europe Facility of the European Union

Change Log

Version	Author	Description Change
V0.1	Maria Cristina Marinescu (BSC)	First iteration
V0.2	José Eduardo Cejudo Grano de Oro (EF)	Reviewed
V0.3	Albin Larsson (EF)	Reviewed
V0.4	Ariadna Lobo (BSC)	Reviewed

Table of Contents

1. Introduction

The goal of Saint George on a Bike is to provide rich information about European cultural heritage artifacts. The document *MS3 User requirements and use case definition* is at its second iteration and sets up to identify the use cases and user requirements that will give concrete shape to the project's goal. The MS6 milestone is also at its first iteration, and it reports on the system and module-level architecture for the technical solutions we implemented up until now to address the General service for enriching collections (use case 3.1), the service that produces the enriched metadata. As described in the requirements section of document MS3, there is more than one type of output that may be generated, depending highly on the type of input available. The levels of semantic output that we currently contemplate are the following:

- Semantic resources in form of tags coming from existing vocabularies
- Textual tags
- Textual captions
- Semantic graphs which contain both entities and relationships

It is important to say that these are options for output types. While we consider all of them, we do not yet know whether it is viable to generate them all, as they may involve very different techniques.

At this point in the project, we have designed and implemented several solutions that can generate textual tags or captions. Just as we explain in the MS3 document, we are in the process of identifying which controlled vocabulary we can choose the semantic tags from. The obvious option is the Europeana Entity Collection tags, but we are also considering related sources such as DBpedia, Wikidata, or more specific vocabularies used by Europeana providers. Finally, generating metadata in the form of a semantic graph is a more complex task, which requires detecting not only the objects represented in a painting, but also their relationships. In their simpler form, these can be prepositions that relate subjects to objects; in their more complex form, they are verbs. We have not reached the point in the project where we are ready to tackle this problem; during the next year we will evaluate the viability of such an implementation.

During the past year we have implemented four techniques (see section 2) that all contribute to the generation of textual tags and captions, and aim to support our main use case *3.1 General service for enriching collections.* We have additionally implemented a classifier whose purpose is to help extract relevant captions for paintings, in a simple canonical form that supports the automated application of natural language processing techniques. This technique does not reflect a use case in itself, but rather is an attempt to test whether we could generate useful input data for algorithms that produce data useful for one of the main cases.

In the rest of this document we explain the system and module-level architecture for each of these five techniques, as well as their implementation. Sections 2.1 to 2.4 explain the generation of SGoaB enrichments. They also discuss briefly the iconographic level of the objects that the different methods successfully target. The final global architecture will reflect the choices in terms of implementing the rest of the use cases: 3.2 Ingesting results from general enrichment service into Europeana, 3.3. Search based on enrichment, 3.4 Populate a crowdsourcing tool with candidate enrichments, and 3.5 Upload in data sharing platforms.

2. Generating textual tags and captions

For the time being, our objectives were the generation of object labels and image captions, two of the four levels of semantic output that the project is considering as possible enrichment metadata. The techniques we have implemented target:

1. Basic level classes ("person", "bird", etc.)

2. Higher level concepts ("knight"), possibly derived from combinations of lower concepts (e.g: "Man" + "Spear" + "Horse" => "Knight")
3. Named entities, also via combinations (e.g: "Man" + "Trident" => "Neptunus")
4. Concepts, possibly reifying some of the above objects ("Piéta", "seascape", "crucifixion")
5. Imaginary places ("Hell") and persons. This category applies to some of the previous levels.

The rest of this section explains the architecture and high-level implementation of the techniques we have implemented. All of them are subject to further improvement over the course of the project.

2.1 Improving object detection based on time context

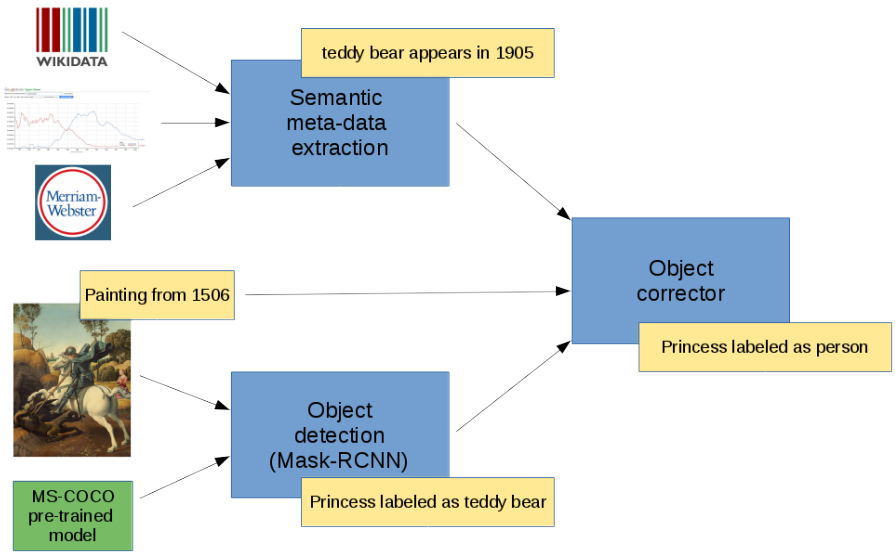

Figure 1. Improving object detection by placing them in the correct time context.

Figure 1 illustrates the architecture for the object detection module that relies on time information extracted from external sources to place detected objects in time. The detected objects are originally labeled with classes from the COCO dataset. The time of the first use of the word is the minimum information that the algorithm requires to filter anachronic terms. We identified three sources that contain information for overlapping subsets of concepts: Wikidata, Google Ngram viewer, Merriam-Webster dictionary approach. Some of them provide more precise information on the probability of use of a word during time, rather than just the year or period of first use. In the end, we chose the dictionary approach, in which we can control the correct meaning of the word, and take into account the historical context of the word.

This information helps detect anachronic concepts and either remove them, or replace them with the next most probable objects that fits the time period. For instance, the image of the princess in Raphael's painting on Saint George and the dragon is detected as a teddy bear. Given that teddy bears were first mentioned at the beginning of the 20th century, and the painting is from the 16th century, the algorithm detects a mismatch. This triggers the object correction step, which in this case, chooses the next most probable label for the princess´ bounding box, namely *person*. The output of this tool is a set of textual tags, concretely the names of classes used by the MS COCO dataset. It thus generates basic level classes, as far as the “Iconographic level” requirements define in section 4.3. of MS3.

2.2 Training object detection with iconographic classes

Object detection is a base step for several tasks including caption generation and search. There are plenty of pretrained models (VGG-16, VGG-32, ResNet, etc.) based on different datasets which can be used in object detection. However, object detection in cultural heritage has its own limitations. These models are usually trained with datasets whose object classes have no symbolic and iconographic dimension. However, when describing paintings, classes cannot be basic and broad-brush. For example, a bishop, Virgin Mary, or Saint George cannot be referred to as just a person when the painting contains object classes that identify them. Even a simple task such as recognizing animals and people can easily convert into a complex task if we'd like to know if the animal is a superbeing (dragon, minotauro, etc.), or what is the occupation of a person. That is why we decided to train our own transfer learning model, which is able to detect classes with focus on cultural heritage. The detected objects are therefore labeled with our own class names.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point for computer vision or natural language processing tasks - given the vast compute and time resources required to develop neural network models for these problems and the huge gains it provides when applied to related problems.

Our implementation uses the Mask-RCNN (Kaiming et al. (2017)) model based on the pre-trained weights of the MS COCO dataset, as a starting point for the transfer model. The training set consists of more than 4000 manually labeled examples with annotations (source of image, file path, bounding box information, class names) in VOC Pascal XML format. We defined 53 classes based on a careful selection process that first eliminates anachronic classes from the COCO dataset, and then sets to detect the most common objects present in paintings, to further filter this set. A painting class corresponds to a category in a painting collection. The painting collection we have taken as a reference is the *Wikimedia Commons* collection of paintings labeled by the regular expression "*Paintings of…"* (https://commons.wikimedia.org/wiki/Category:Paintings_of_people).

The next step extends the set of classes. *Wikimedia Commons* categories and subcategories are very useful to discern new painting classes when querying about basic classes. For example, if we query about paintings of people we find the subcategory *angels_with_humans*. In this case, *humans* is a general reference that covers the basic classes *people, men, women* and *angel* is a new painting class because Wikimedia has the category *Paintings of people with angels.* Starting from the filtered COCO dataset, new classes are added that are related via Wikimedia categories and subcategories. Among the possible classes derived from Wikimedia categories we have chosen a sample with iconographic and symbolic meanings, supernatural and metamorphosed animals (*swan* in Leda's paintings, *cow* in the rape of Europa) and *devils*. Apart from dragons, other fantastic animals are *unicorn*, *centaur, Minotaur*. We also consider classes that help to identify people that have a social role. The process of class selection has more detailed steps - explained at length in the following evolving document that explains the criteria we used to chose the training classes: https://docs.google.com/document/d/1Cj1BRraczYga3I9QimKDhOreNvQLh6MGl5VxjOFTj98/edit#heading=h.7f4jdyfsca0w.

2.3 Improving object detection based on language model

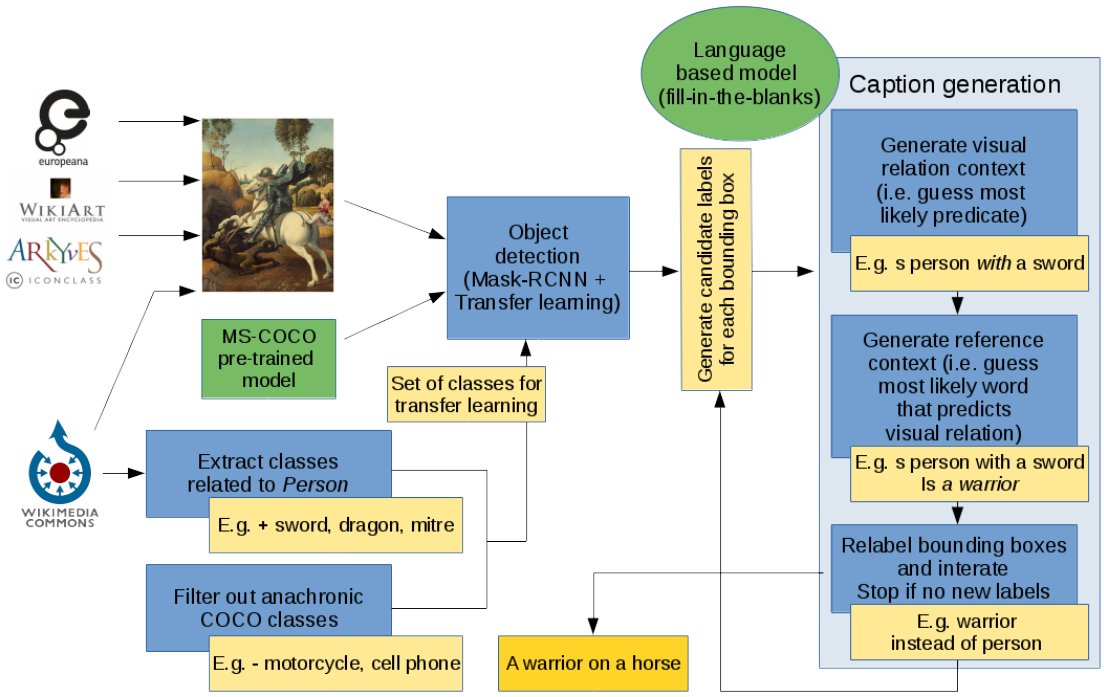

Figure 2. Caption generation using a language model.

Figure 2 illustrates the caption generation technique that we have implemented, which is based on a language model (using BERT). The input to this model is the image representation of the painting containing a set of bounding boxes, one for each object class. The output is a set of statements that explain the visual relationships between the classes in the bounding boxes. In these texts, the classes corresponding to the bounding boxes are referred to with more specific denominations according to their visual relationships. The set of texts are different captions describing the same painting and they are ready to be evaluated manually or with semi-automatic methods.

We use the detection network Mask-RCNN to identify bounding boxes in a painting and generate candidate labels for each of them. We apply transfer learning to train the object detection model, starting from weights provided by a MS COCO pre-trained model. Our training set contains around 4000 paintings in VOC Pascal format, and was labeled manually. The object detection model is trained to recognize classes of two provenances: those MS COCO classes that are not anachronic, and subcategories of, or related to, the representations of people in the Wikimedia Commons catalog. These cover iconographic classes (e.g. the Annunciation), symbolic classes (e.g. key of heaven for St. Peter), as well as imaginary beings, occupations (e.g. monks, knights), etc.

The goal of caption generation is to refine the references to the main object(s) among all salient objects of a painting. We consider that the main object is the one whose bounding box intersects the largest number of other bounding boxes. For each object whose bounding box overlaps with the main object's, the algorithm first generates a set of sentences that describe the possible visual relations between the two. These texts are generated by using a language model to guess missing words that mask the possible relations between the two objects, and they are called visual relation contexts. We then use the language model again to generate the most appropriate completions that specializes the original object class in the visual relation context. For instance, a person carrying a cross is Jesus, while a person with a crown becomes a queen or a king. Only those pairs that are predicted with high accuracy by the language model, will generate a visual relation context. Each of the visual relation contexts that pass this filter are then placed in a reference context to refine the main object and generate the captions.

The output of this tool is a set of textual captions and can generate basic level classes, higher-level concepts, and named entities, as far as the “Iconographic level” requirements define in section 4.3. of MS3.

2.4 Caption generation based on attention mechanism

Traditional approaches for caption generation are based on encoder/decoder architectures of deep learning models. The encoder part is normally implemented as a pre-trained model such as VGG-16, Resnet, etc with a single fully connected layer. All of these models are trained on data sets of photographs of objects and situations from the real world. This is a problem for us given that paintings can represent symbolic cultural images or events, and imaginary beings, which do not appear in pictures of real life. While currently we are using the pre-trained COCO model as the encoder, this is the reason we decided that, in the next step, we will train our own object detection model by using the MaskRCNN architecture and a transfer learning approach which include 53 classes (the same as in section 2.2).

The decoder part is implemented as a Recurrent Neural Network (RNN) with attention mechanism (GRU or LSTM). The encoder output and the decoder output from the previous iteration are both passed to the decoder as input. When the RNN is generating a new word, the attention mechanism is focusing on the specific part of the image, so the decoder only uses specific parts of the image based on its previous training. This approach is efficient only if the encoder can correctly detect some features that enable labeling objects with names that can help the decoder make the correlation between specific areas of the image (identified by object labels) and caption words. Detecting specific features (angel, monk, sword, Christ) is a key point of making the proper correlation on part of the decoder, which re-enforces our decision to train our own object detection model.

2.5 Caption classifier

The caption classifier is a tool whose purpose is to extract phrases from descriptions associated with paintings, which are likely to be descriptions of what is going on in the images. The idea is to create an aligned painting / caption dataset, which may then be used for training to generate captions. The classifier is used to filter out sentences in painting descriptions that do not refer to the content depicted in the image. This is a way of reusing full descriptions from museums, Europeana, and Wikimedia datasets in order to get training data to improve the automatic generation of enriched data.

This is not a trivial task. Most paintings don´t have any description associated with them, but others do; these descriptions may be collected from Wikipedia or any other art encyclopedia or museum site. The problem is that these descriptions generally include text that has to do with the historical context, the life of the author, the medium or style, and so on. Little of it is dedicated to the actual scene depicted. Even when this is the case, very often the language that is used is in the style or art professionals, that is, complex from a literary point of view. This is a big challenge for Natural Language Processing tools, which work best over simple canonical statements.

We take as the canonical example (for the training set) the MS COCO captions, which were manually generated. As examples of bad captions we take text from art Wikipedia pages that mostly describe the life of painters and some drawing techniques. The dataset of 4000 captions is balanced - 2000 (randomly extracted) good captions from COCO, 2000 (manually selected) bad captions from Wikipedia.

To test the classifier, we used again a mix of two datasets. We downloaded about 40K images with captions from Europeana (between the 12th and the 19th century). Using (Google) machine translation, we translated all of them into English. After applying caption classification, we were

able to extract about 2000 (correct) good pairs of image/caption[1]. While the percentage is small, it is very difficult to obtain good captions at all, so the 2K pairs are extremely relevant for us. The second test has been made with data from the "Web Gallery of Art". We downloaded all their images (around 20K items) and were able to generate another about 2000 good pairs of image/caption. Figure 3 illustrates this process.

The implementation uses a standard sequential architecture of Neural Networks, specifically a Long Short Term Memory (LSTM) architecture.

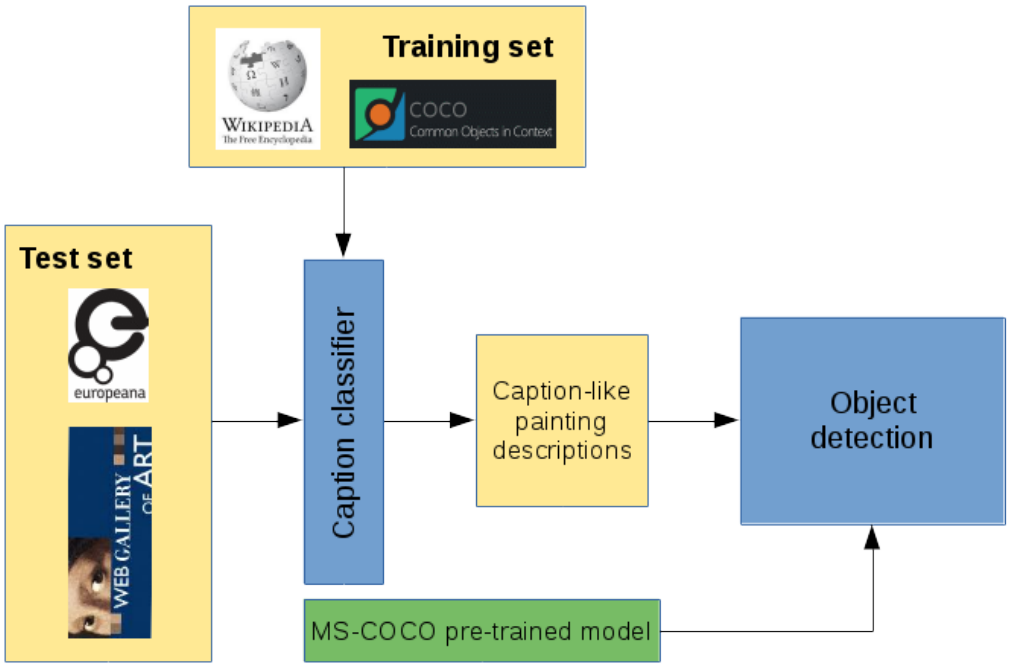

Figure 3. Training, testing, and use of the caption classifier.

3. Deploying enrichments to serve use cases

Below we illustrate the alternative ways in which the SGoaB rich metadata may be incorporated in the higher-level architecture. Deploying these enrichments addresses use cases 3.2 Ingesting results from general enrichment service into Europeana and 3.3. Search based on enrichment.

We will seek to develop the project's semantic search functionality on top of Europeana's APIs as much as possible. Europeana already hosts two search APIs. Europeana's "basic" Search API only accesses "regular" metadata provided by Europeana's institutional data providers and only a small portion of possible annotation data (i.e. transcriptions of textual documents), while the Annotation API accesses all other annotations, possibly including some more structured enrichments that could be produced by the project's service.

The following figure illustrates the current Europeana configuration for its metadata and annotation repositories and APIs.

[1] These 2000 good captions were deemed correct following a quick review of all of them by the authors, which was confirmed by a careful evaluation of a subset of 300.

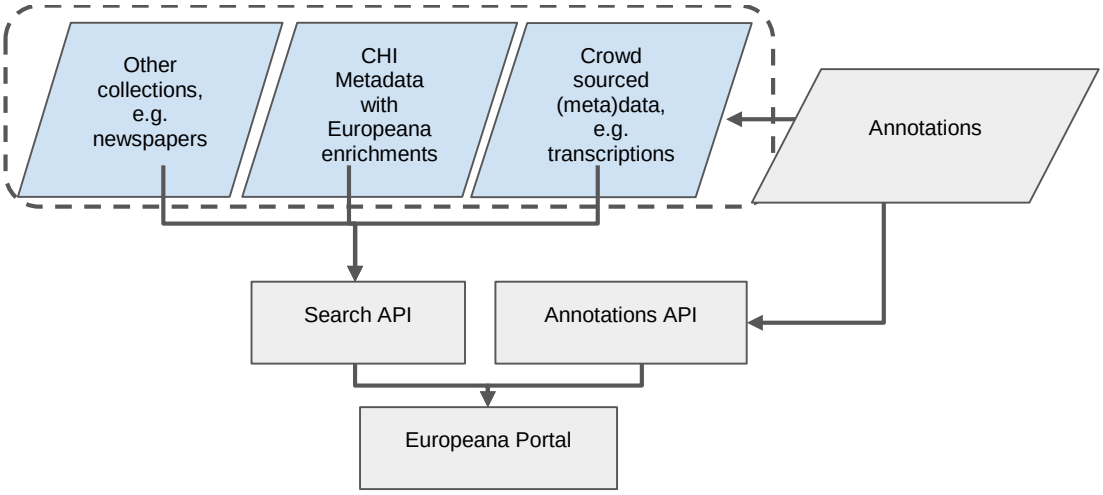

Figure 4. Europeana current configuration.

Figure 5 shows how the project could build upon Europeana's main APIs and portal, after loading the Europeana annotations database with the results of the SGoaB enrichment tools. Using Europeana's APIs and portal would save the need of creating a project search service as a separate stack, but it would require them to be adapted to exploit annotations that represent our project's enrichments (currently the Search API only exploits transcriptions, which do not include the enrichments our project would create).

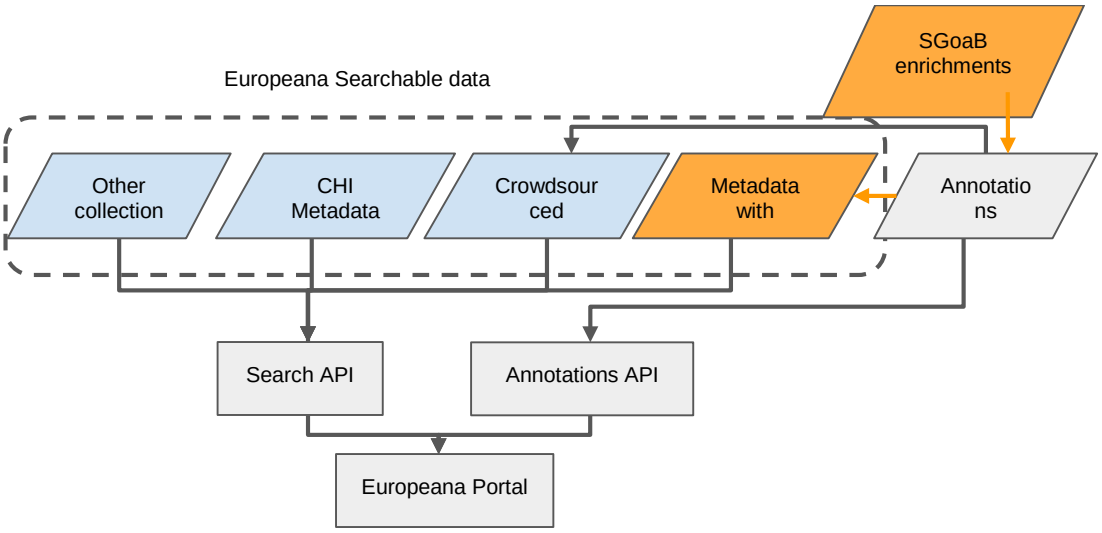

Figure 5. Building on Eurpeana's functionality to ingest and search SGoaB enrichment.

Should this prove impossible, the project would have to build its own Search API and Search UI, while the SGoaB enrichment may still become part of the annotations, as shown below:

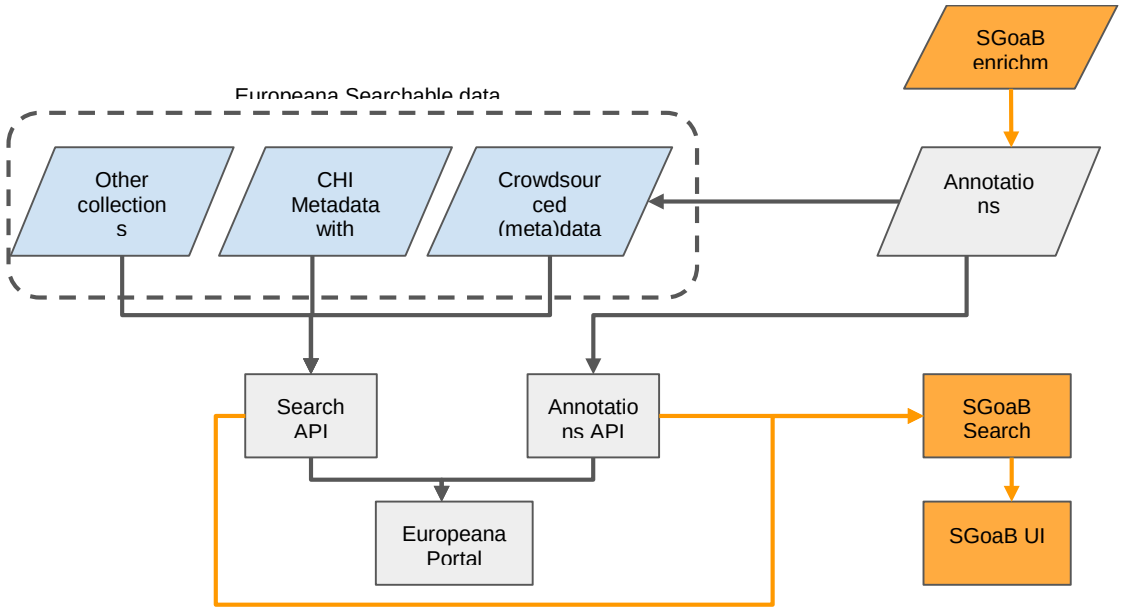

Figure 6. Building our own search API and portal; Europeana-stored enrichments.

The next uncertainty concerns the use of Europeana's Annotation API. It may happen that enrichments of specific shape or quality cannot be loaded in the Annotations API, or not easily exploitable from there, in which case, the project-specific Search API would have to rely on its own enrichment storage and access layer, as shown in Figure 7:

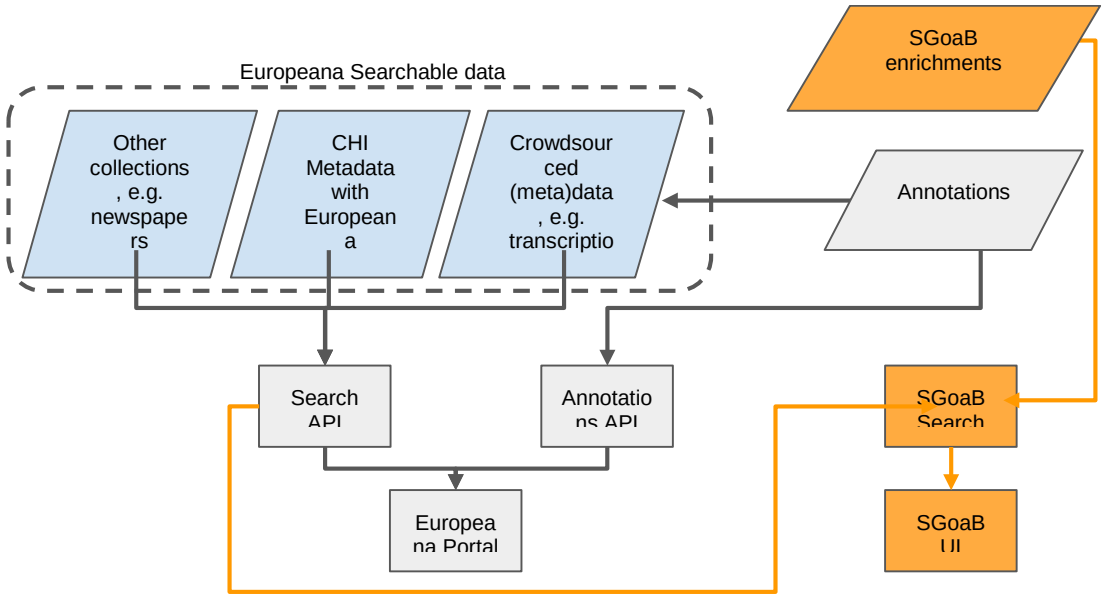

Figure 7. Building our own search API and portal; local enrichments.

4. Future steps

This is the first iteration of the MS6 document and it refers mostly to use case 3.1 General Service for Enriching Collections. As we explained in section 2, we are currently generating enrichments in the form of textual tags and captions. Future work is thus structured around several directions:

Improve the current methods for textual tag and caption generation

Increase training dataset size for object detection to about 10K pictures. This is the set that includes bounding box information and NOT the image/caption pairs dataset.

Use our own trained model (described in 2.2) as an encoder for 2.4 to improve the caption generation using the attention mechanism.

Increase training dataset size for caption generation with relevant canonical captions that can be effectively analyzed via Natural Language Processing techniques (go from currently 4K to about 10K pairs).

Look into the drifting meanings of a word, or homonymic meanings of words, to be able to deal with different meanings over (potentially) distincts time intervals.

Test other language models besides BERT.

Test other approaches to caption classification, such as fitting a language model over the image/caption dataset.

Address the rest of the use cases: Ingestion of results into Europeana, Search API, Crowdsourcing campaign for validation of enrichments, and uploading in data sharing platforms. The first two cases (3.2 and 3.3. In the MS3 document) are tightly linked to bullet point 3 below.

Section 3 explains the enrichment deployment alternatives for Europeana, Uploading the enrichments in data sharing platforms depends on the protocols and data formats that these sharing platforms are using. We are currently starting to look at this issue as part of the MS7 document (Compliance with the Metadata Quality Assurance).

This leaves the Crowdsourcing use case. Europeana is currently preparing a first campaign which will initially target object detection. Concretely, the images will come from the Wikimedia category of Animals. In the future we plan to organize campaigns for more categories, the detection of all the objects in an image, but also with different objectives, between them action labeling and evaluation of captions.

Implement one of the two options for the Search API and portal, either by using the existing Europeana site, or by local hosting at BSC

With regard to the semantic level of the output meta-data, Implement semantic tag extraction and evaluate the viability of generating semantic graphs. This decision will be informed by the ability to generate good metadata about basic and higher level actions (see section 4.3 "Iconographic level" requirements, in MS3). This task is also connected to the language model-based object detection mechanism (2.3). For robustness reasons, there needs to be a way to evaluate the visual relation contexts for images that don't necessarily describe most probable situations, e.g. *A man killing a horse* will never have higher language probability than *A man riding a horse*, although this may be exactly what the painting is showing.

List of Figures