EnrichEuropeana+ First Release of HTR Services

Version 1.0

Documentation Information

Action Number	2020-EU-IA-0075
Project Website	https://pro.europeana.eu/project/enricheuropeana
Contractual Deadline	28.2.2022
Nature	Technical Report
Author	Philip Kahle
Contributors	Florian Krull
Reviewer	Sergiu Gordea, Frank Drauschke
Version	1.0
Date	28.2.2022

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

Contents

Introduction

EnrichEuropeana + (fully titled 'Enriching Europeana through citizen science and artificial intelligence - unlocking the 19th century') aims to enhance Transcribathon (www.transcribathon.eu) as a service for cultural heritage institutions.

Scope

The work presented in this document includes the progress achieved in **Task 2.1 Handwritten Text Recognition Services**.

Objectives

The main objectives of EnrichEuropeana+ are:

- To engage public users and professionals in enhancing the semantic and multilingual description of Cultural Heritage objects by continuing the development of Transcribathon.
- To increase accessibility of manuscripts related to historical events and societal transformations in Europe within the 19th Century through a new Citizen Science crowdsourcing campaign to stimulate user engagement for transcribing, translating, and adding semantic enrichments.
- To transform Transcribathon into a service used by Cultural Heritage Institutions to crowdsource the enrichment of cultural object descriptions and improve the multilingualism of metadata.

The main objectives of Task 2.1 are:

- To make available an easy to use web-service for integrating Transkribus Handwritten Text Recognition in Transcribathon
- To reduce the effort that is required to transcribe manuscripts in Transcribathon with recognition results coming from the Transkribus Platform
- To enhance the transcriptions produced in Transcribathon with layout information

Introduction

The transcription of manuscripts from scratch, as enabled in Transcribathon, is often a time-consuming task even for trained paleography experts. Handwritten text recognition (HTR) technology can crucially reduce the effort, especially when the engine was trained specifically on the source material - the user can then concentrate on proofreading and correcting the automatically recognized text.

The Transkribus Platform [1][2], which was developed in the scope of the EU projects *tranScriptorium* and *READ,* enables end-users to apply this technology in their transcription workflow: after the manuscripts have been uploaded to the platform, a user can apply various document image analysis tools, such as layout detection and handwritten text recognition, on his/her documents and manually take corrections between those steps. Once enough transcriptions have been created, a specialised handwriting model can be trained to improve the recognition accuracy on the material in focus. Eventually, finalised transcriptions can be exported in various formats.

In this task of the project, the goal was to expose the relevant functionality of Transkribus for integration with Transcribathon in an easy to use web-service. Handwriting models trained in the Transkribus platform can be applied to manuscripts in Transcribathon, without going through the complete document lifecycle that is required in the usual workflow described above.

Architecture Overview

In order to ease the task of integrating HTR functionality, a new API, the Transkribus Processing API, was implemented that hides most of the complexity of the standard Transkribus API from its users and focuses on processing single image files in as few requests as possible. E.g. images no longer need to be organised in documents and collections, they can be conveniently processed one after another.
The Transkribus backend services were specifically enhanced where necessary to cater for this type of workload. Also, the API is backed by a distinct set of worker machines which required an expansion of the infrastructure.
Furthermore, a Java client library was developed to showcase the operation of the API and quickstart the integration. Once the recognition result for an image has been retrieved and stored, the HTR web plugin, developed in task 3.1 (Milestone 2), can be leveraged to finalise the transcription. Figure 1 provides an overview of the interfacing and the message exchange between these system components. A detailed overview of the Transkribus processing API is provided in the following section.

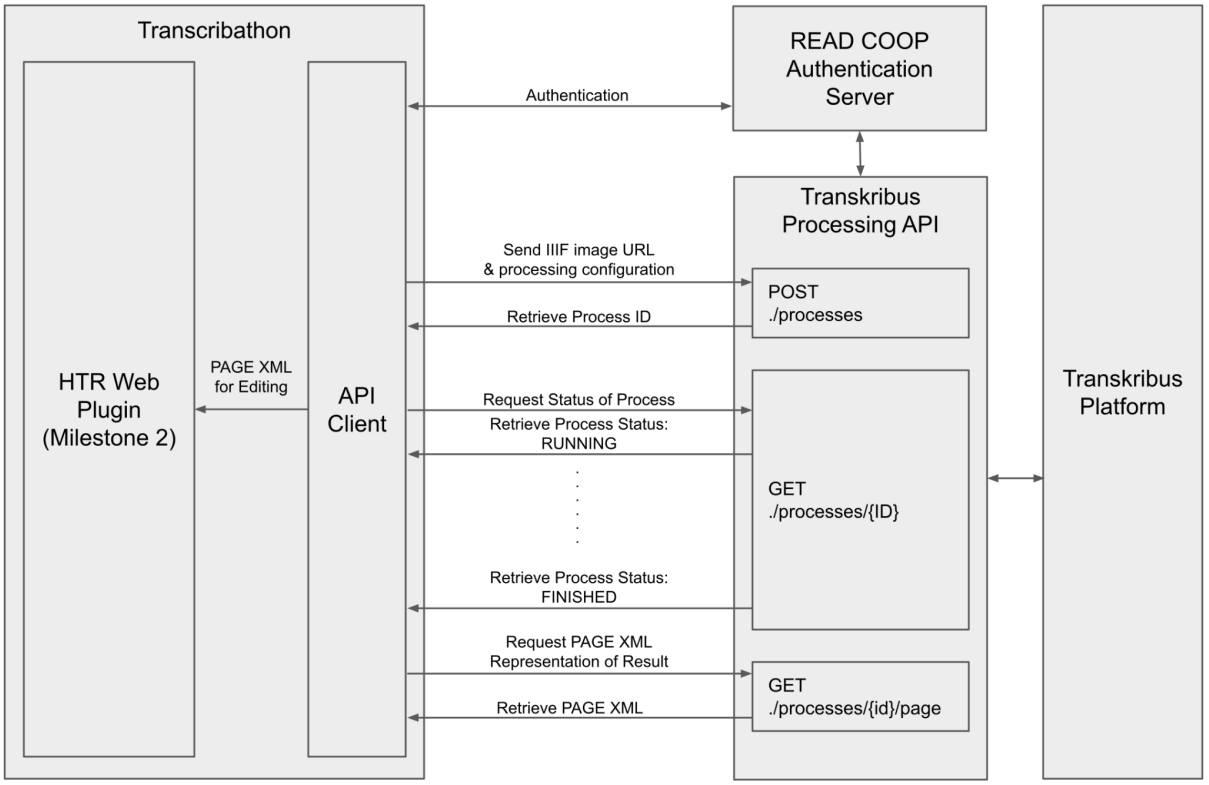

Figure 1: System Architecture Overview

Transkribus Processing API

The Transkribus Processing API exposes its functionality through a REST interface. The documentation is provided in an OpenAPI document[1] and visualised in a Swagger console[2] that also allows developers to test the respective HTTP requests.

Workflow

In order to access the API, the client application needs to authenticate with the READ COOP authentication server and retrieve an access token that is used to authorise requests in the API. Any Transkribus user account can be used to operate the API after it has been enabled for this service.

The processing of an image file is started with a POST request that needs to include the image information, a configuration section and optional pre-existing layout information in the content section.

In case the image is available online, a HTTP(S) URL can be included and the API's backend will retrieve the file from this location. Otherwise, the image data may be embedded in the request as a Base64-encoded character sequence.

The configuration section specifies the document image analysis tools to be applied on the image data. The current mode of operation requires the ID of an HTR model that is used for recognizing the text. The API allows reference to any of the HTR models trained by the user in Transkribus as well as all models published by READ COOP and the Transkribus community. To

[1] https://transkribus.eu/processing/v1/openapi.json
[2] https://transkribus.eu/processing/swagger

the date of this writing there are 87 models[3] publicly available, that have been trained on material from different eras and on a variety of script types and languages.

The Transkribus system will automatically run a region and text line detection on the image data. The actual text recognition will use this information and the text output will be associated with the detected layout elements.
In case the recognition shall be restricted to a specific area of interest in the image, an optional content section allows the API's user to send pre-existing layout information that will be picked up in the process.

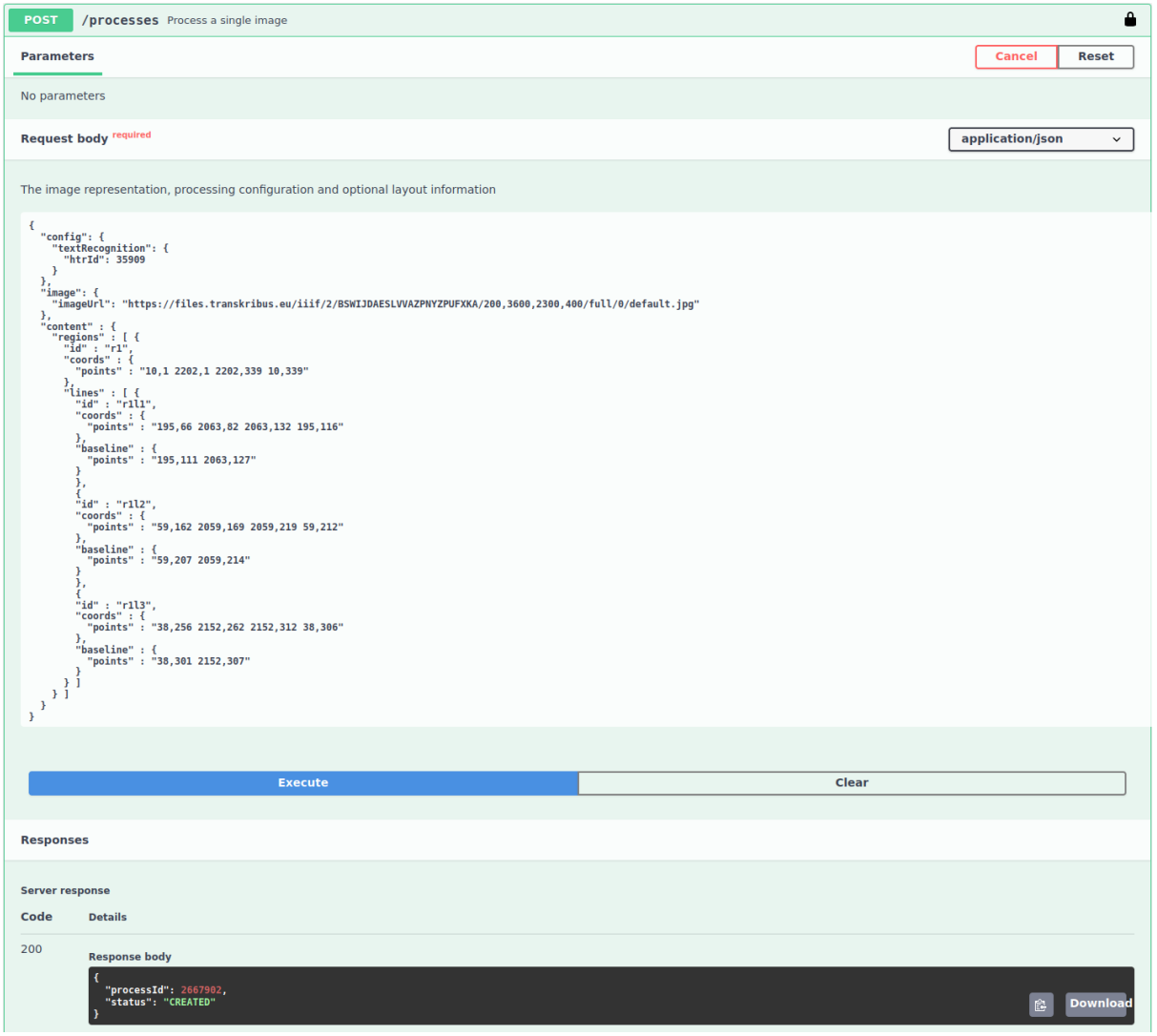

Figure 2: A request to trigger the layout detection and text recognition on the given image.

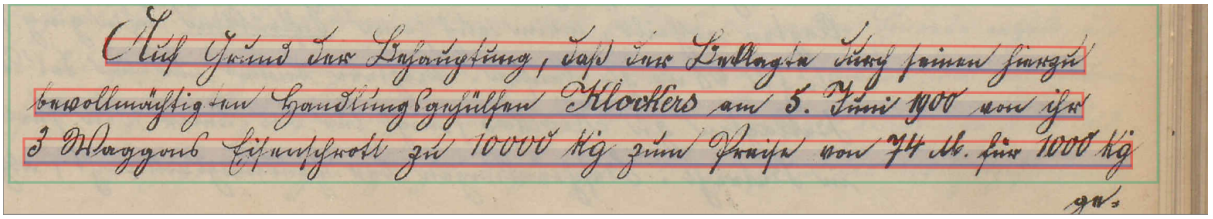
Figure 2a: Visualisation of the layout information provided with the request indicated in Figure 2

[3] https://readcoop.eu/transkribus/public-models/, accessed on 2 February, 2022

Figure 2 shows an example request in the Swagger Console that references an image available through Transcribathon's IIIF image server, a text recognition configuration and content section. The API will respond with an object containing a process ID and the processing status "CREATED".

The processing status can now be polled with GET requests using this process ID until the status switches to "FINISHED". This final response includes the detected layout and recognized text in a JSON representation, see content section in Figure 3.

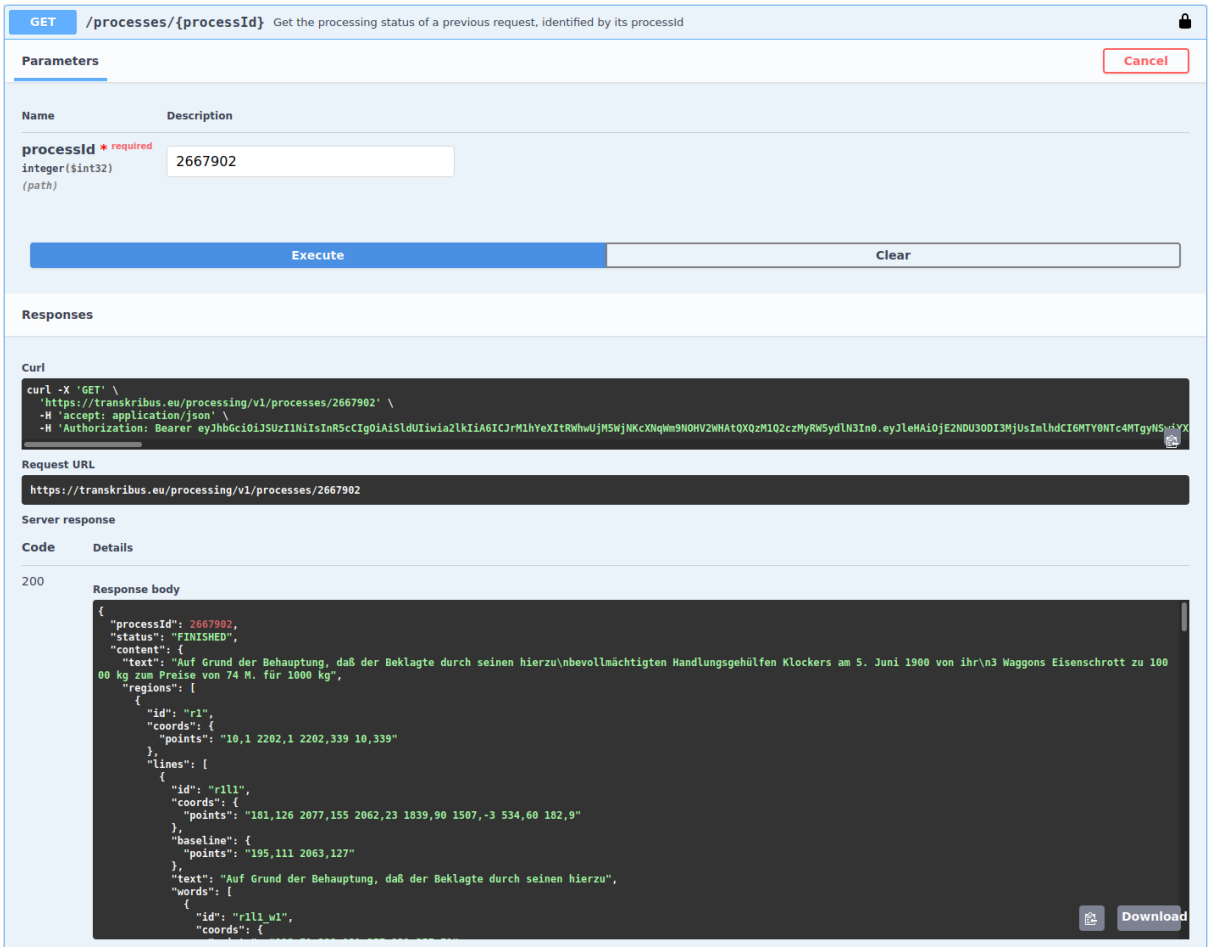

Figure 3: The request for retrieving the processing status and HTR text. The example shows the response for the process started in Figure 2, including the recognised text.

Alternative representations of HTR output are now available through specific endpoints including ALTO XML v4[4] and PAGE XML[5]. The HTR web plugin described in the Milestone 2 document uses the PAGE XML representation which is depicted in Figure 4. By using the same processing task identifier, the user is able to access the preferred serialization of HTR output.

[4] https://loc.gov/standards/alto/, accessed on 23 February, 2022
[5] https://github.com/PRImA-Research-Lab/PAGE-XML, accessed on 23 February, 2022

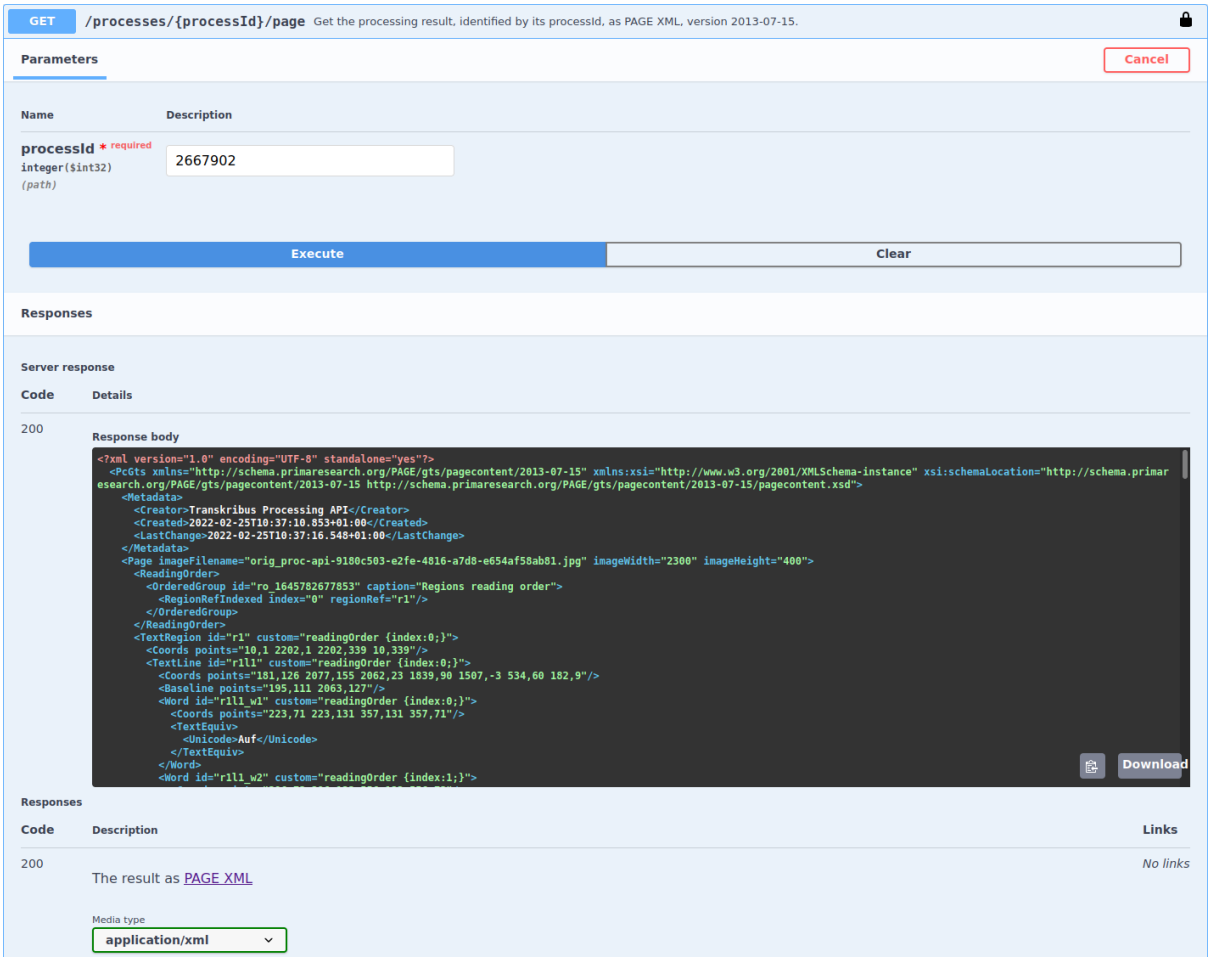

Figure 4: the GET request for retrieving the result in PAGE XML format

Java Client Library

An additional Java client library was developed to test and showcase the functionality of the API and to set a starting point for the integration in Transcribathon. This library implements the HTTP communication with the API and includes the required object definitions for integration in any Java application. Furthermore, it provides a simple batch workflow implementation for processing larger batches of images efficiently as well as a command line interface that runs the HTR service on images on the user's disk.
The client library is made available on github[6].

Transkribus Models API

Besides looking up the model's IDs at the READ COOP website or in the Transkribus client applications, programmatic access to the list of available models is enabled through the API of the Transkribus platform. A new Models API[7] was developed and released in February 2022. It allows simple browsing of models as well as faceted search on the basis of the properties of the models' metadata (e.g. name, creator, description, etc.) and the characteristics of the training materials (e.g. century, script type, language, etc.). Hence, the Transcribathon backend can

[6] https://github.com/EnrichEuropeana/htr-client
[7] https://transkribus.eu/TrpServer/rest/models/text

access the current list of models and present them to users of the tools integrating the HTR functionality.

Custom Model Training in Transkribus

Both the Processing API and the Models API allow access to all private models the authorised user can see in Transkribus besides the ones publicly available. Training a custom handwriting model can crucially decrease the error rates in recognition results and thereby reduce the effort needed to finalise transcriptions.

In order to create a new HTR model, the training data (i.e. images with manually created transcriptions) needs to be available in the Transkribus platform. This data could be either ingested from existing transcriptions (e.g. items that were previously transcribed in Transcribathon), or it can be manually provided by end users. For each image the training process requires the layout markup (text regions and lines) and the textual representation of each lines' content.

We are currently in the process of creating a training data set using the ration cards provided by State Archive Zagreb. Figure 5 presents the transcription editor of the Transkribus Expert Client, where the end users are able to manually transcribe the text highlighted in the specific regions of the document. Green rectangles represent the regions of interest that have been provided with the images, while the text lines detected by Transkribus are indicated with red rectangles.

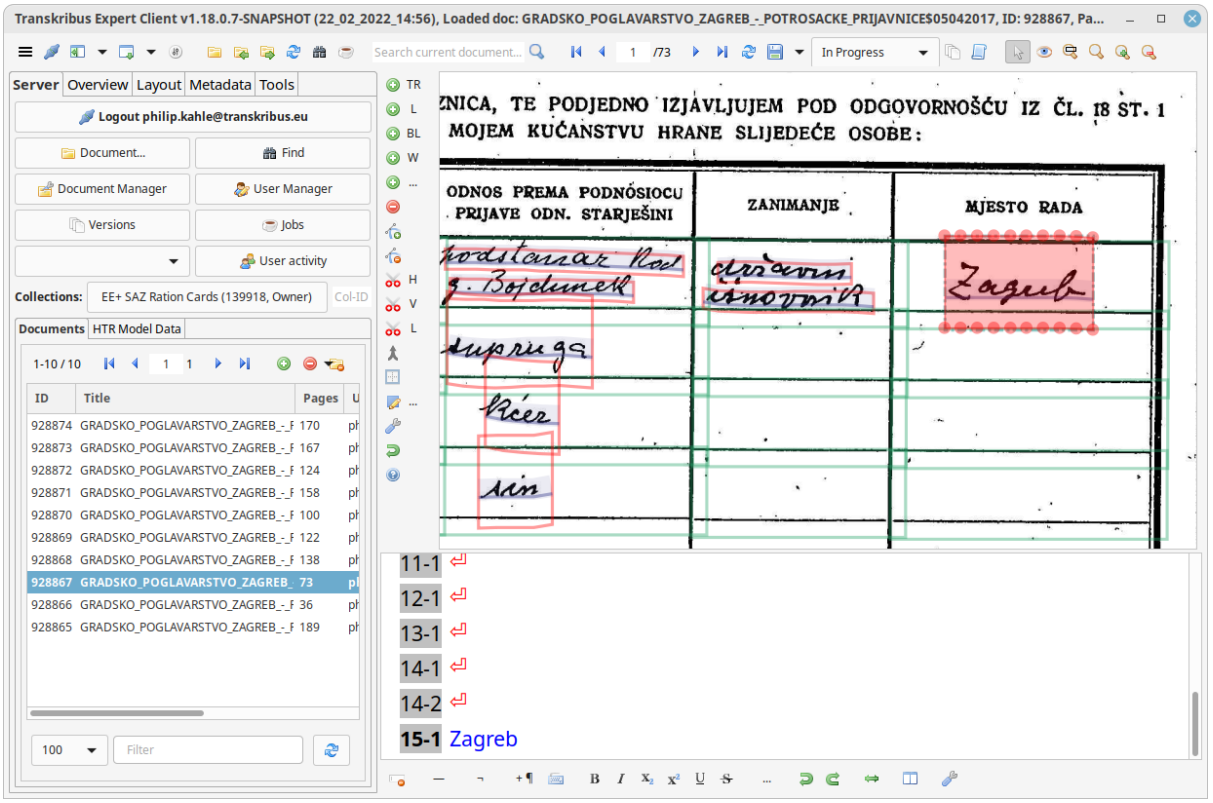

Figure 5: The manual transcription editor of the Transkribus Expert Client

When enough transcriptions are available, the training of a new HTR model can be started. The user will assign the transcribed pages to a training and a validation set, will provide metadata describing the newly created model, and start the training process using the GUI depicted in Figure 6. The stored model will be available through the Models API and can be used for HTR processing of similar documents (see section Transkribus Processing API).

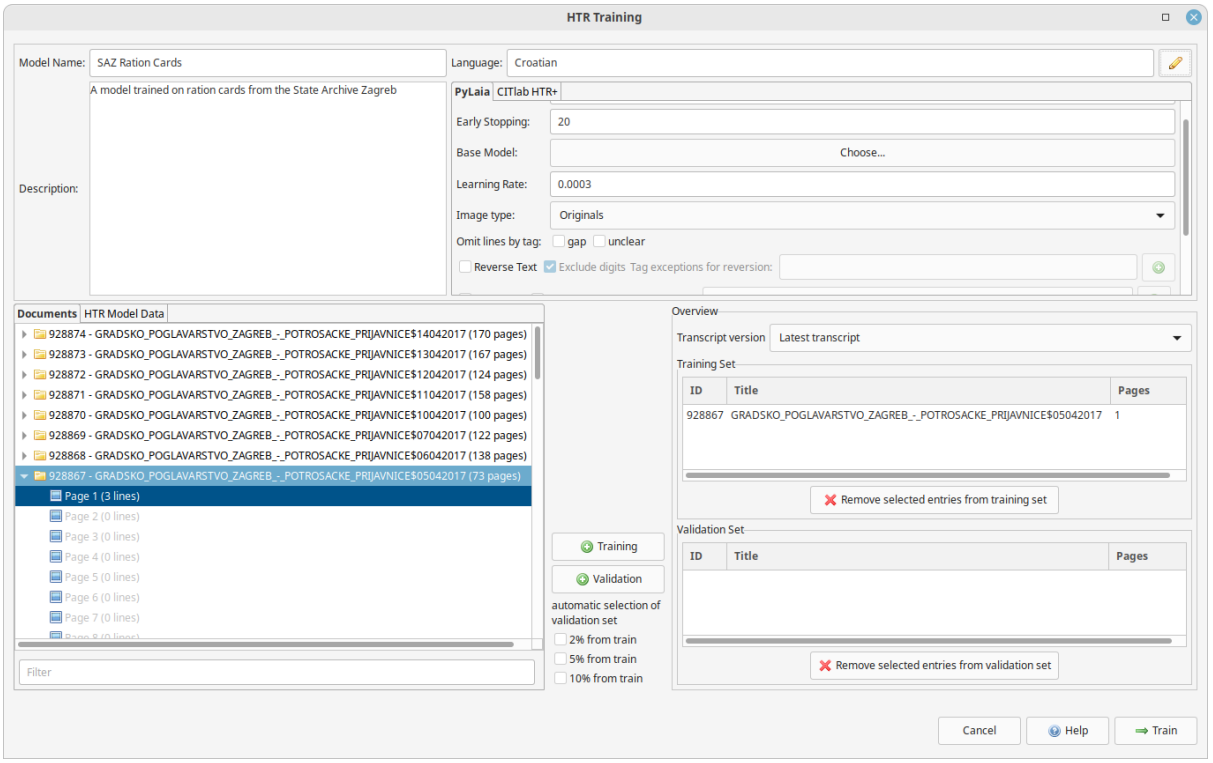

Figure 6: The HTR training interface of the Transkribus Expert Client

Conclusions

This document presents the work carried out within Activity 2 with the goal to facilitate the integration of handwritten text recognition services within the Transcribathon tool. It provides a detailed overview of Transkribus Processing API, showcasing the individual steps required to use a HTR model for recognizing text in handwritten text documents.

The publicly available HTR models can be used within the processing API. However, new models would need to be trained when none of the existing ones is appropriate to effectively process a given set of documents (e.g. document language not supported by existing documents). This is also the case of materials provided by State Archive Zagreb, for which the project partners are currently contributing to build a training set.

Besides general maintenance, future work includes enhancements of the described components' functionality and infrastructure as needed.

Sources

[1] P. Kahle, S. Colutto, G. Hackl and G. Mühlberger, "Transkribus - A Service Platform for Transcription, Recognition and Retrieval of Historical Documents," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 19-24, doi: 10.1109/ICDAR.2017.307.

[2] S. Colutto, P. Kahle, G. Hackl and G. Mühlberger, "Transkribus. A Platform for Automated Text Recognition and Searching of Historical Documents," *2019 15th International Conference on eScience (eScience)*, 2019, pp. 463-466, doi: 10.1109/eScience.2019.00060.