EnrichEuropeana+ First release of HTR Plugin

Version 1.0

Documentation Information

Action Number	2020-EU-IA-0075
Project Website	https://pro.europeana.eu/project/enricheuropeana
Contractual Deadline	30 September 2021
Nature	Software prototype
Author	Florian Krull, Frank Drauschke
Contributors	Philip Kahle
Reviewer	Sergiu Gordea
Version	1.0
Document Date	1 October 2021

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 Europeana Transcribe (www.transcribathon.eu) as a service for cultural heritage institutions.

Scope

This document outlines the work carried out in the task 3.1 "HTR web plugin".

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 Europeana Transcribe.
- 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 Europeana Transcribe 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 this Milestone are:

- develop a transcription editor using HTR technology, which can be embedded as a plugin in different Web portals
- develop a prototype showcasing the integration of the transcription editor within the Transcribathon tool

Development of the HTR Plugin

This section describes the work carried out within the Task 3.1 for the development of an embeddable HTR Plugin. It describes the base functionality of the developed plugin and the guidance for integrating the plugin in existing Web portals.

Main functionality of the HTR Plugin

READ COOP developed a new text editor as the one available from TranskribusLite lacked the architectural features needed to be integrated as an easy-to-use plugin. The development goal was to build a modular text editor that can be customized. For that purpose, the Vue[1] library and the open-source text editor framework tiptap[2] were used.

The included image viewer is based on OpenSeadragon[3]. OpenSeadragon is an open-source, web-based viewer for zoomable images, implemented in pure JavaScript. It renders the polygons of text lines detected by the HTR within the original document. This allows the visualization of connections between the transcribed text and corresponding regions in the original document. Consequently, the selected lines can be highlighted and focused both in text and in the original image for easy transcription.

For faster editing of the text 'Redo' and 'Undo' are implemented as buttons and common shortcuts (Ctrl+Y/Ctrl+Z). Additionally the editor enables tagging and editing of tag attributes as demonstrated in the image below. Changing the orientation of the view between text and image is also an option in the new editor.

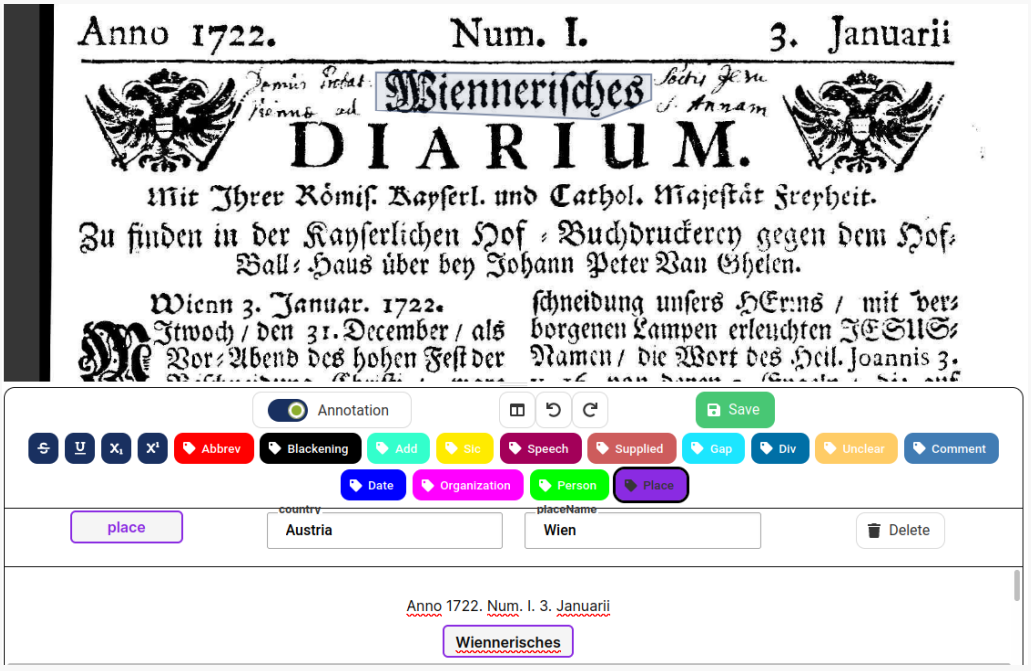

Figure 1. Transcription editor. The zoomable and draggable image to be transcribed is shown on the top. On the bottom, the current transcription is shown. The line highlighted in the image synchronizes to the movement of the cursor in the transcription and vice versa.

[1] https://vuejs.org/
[2] https://tiptap.dev/
[3] https://openseadragon.github.io/

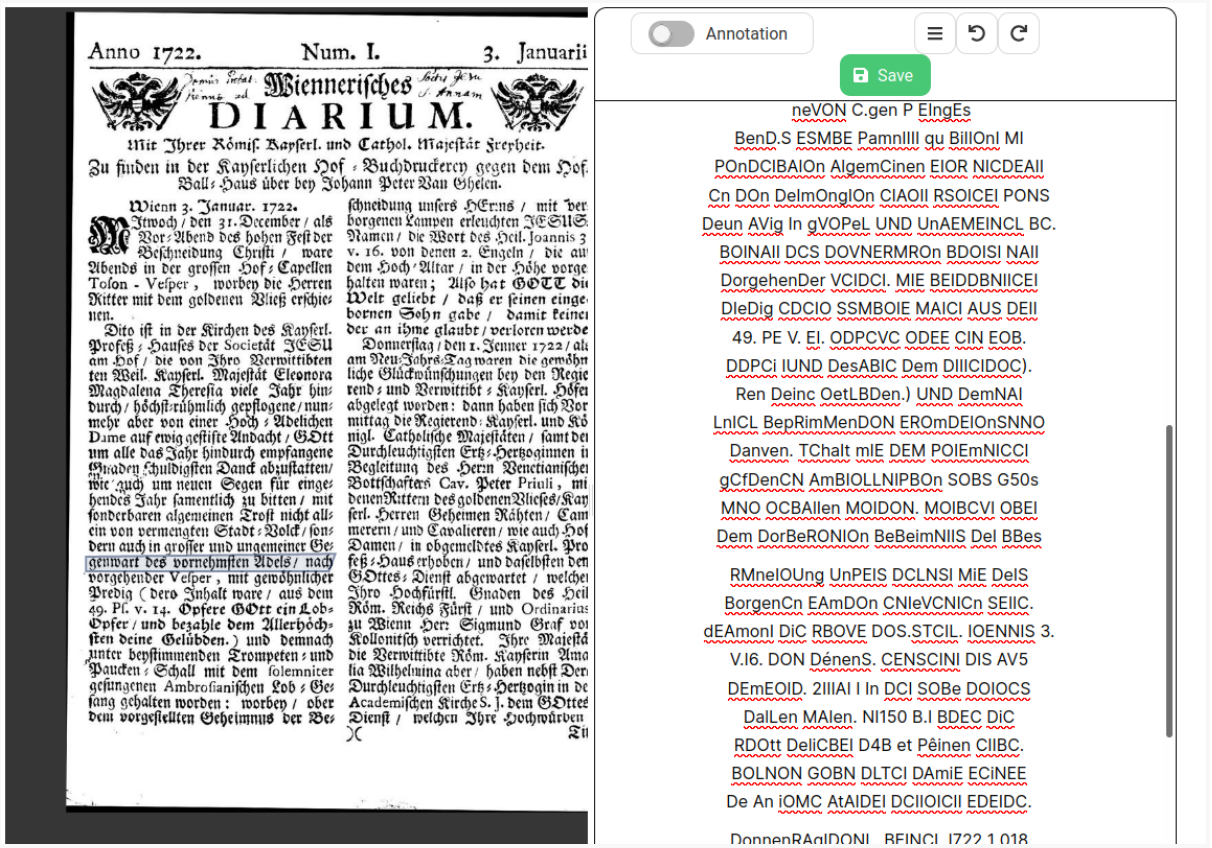

Figure 2. Transcription editor. The user can switch between horizontal and vertical alignment of the viewer components.

Text editor and layout editor are the two components of the handwritten text recognition tool. They communicate with each other using the same data formats, the IIIF URLs and the textual transcriptions (represented in xmlJson). The output of both components enable the Trancribathon platform to provide transcriptions enriched with coordinates to Europeana. They can serve as a basis for further applications such as fulltext search with hit highlighting, analogously to classic OCR output.

The following image displays the current status of the layout editor that is being developed. It allows for corrections to be made to any automatically recognized layout of lines and regions as well as creating the layout markup from scratch. The features include editing, adding and removing baselines as well as regions.

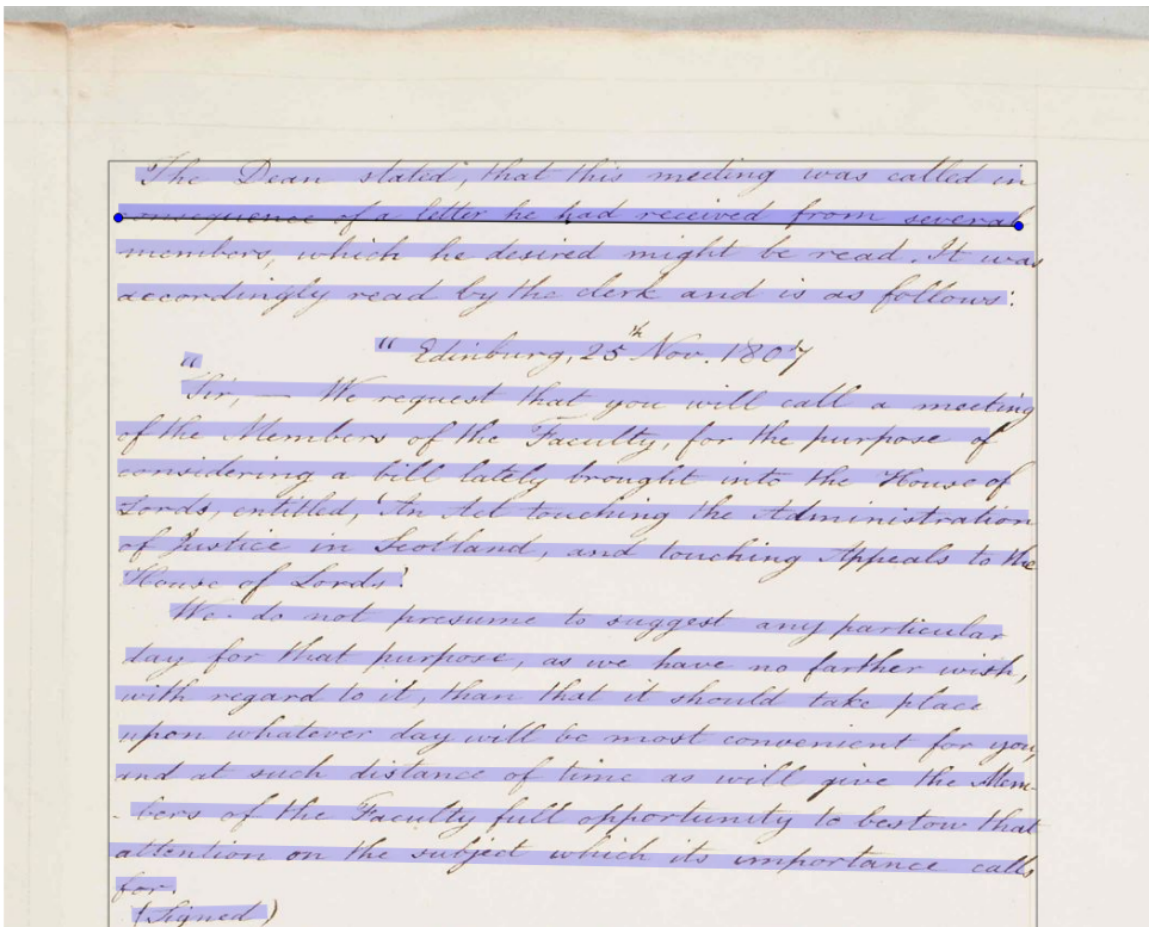

Figure 3. Layout Editor

Guidance for integration in Web portals

The Vue library expects the following properties (custom attributes that can be registered on a component):

- iiifUrl
  - type String
  - example: https://rhus-209.man.poznan.pl/fcgi-bin/iipsrv.fcgi?IIIF=11/8//2020601/https__1914_1918_europeana_eu_contributions_21890/21890.261095.original.tif/info.json
- xmlJson
  - type Object
- xmlJsonString
  - type String
- tags
  - type Object

The **iiifUrl** property requires the image information of an IIIF image according to the following URL pattern {scheme}://{server}{/prefix}/{identifier}/info.json. The Text and Layout attached to the original document can be passed to the editor using either **xmlJson** or **xmlJsonString**. The first expects a Javascript object, while the latter allows passing a string representation of this

object. Additionally, a JSON that describes the allowed tags can be passed, which allows developers to customize the tagging buttons shown above the text editor.

```
"definitions": {
   "annotations": [
    {
      "name": "abbrev",
      "label": "Abbrev",
      "attributes": ["expansion"],
      "extra": [
       { "name": "expansion", "label": "Expansion", "type": "string" }
      ],
      "color": "#ff0000",
      "icon": 8228
    },...
```

The following command is used to integrate the whole text editor in existing Web portals:

```
<div id="app" data-iiif-url="https://files.transkribus.eu/iiif/2/QJDYEHKUDZIDNBCYBYXBLAZT/info.json"
data-xml-json-string='{"declaration": { "attributes": { "version": "1.0", "encoding": "UTF-8"...'>
```

The transcriptions provided by the users are accessible through the data-output attribute of the div-tag and also through the global window object (i.e. window.output attribute). These include the XML serialization of the data, as well as it's transformation into JSON representation.

```
<div id="app" data-output="...." >
```

To illustrate the integration of the text editor into a wordpress based portal[4], the following Figure showcases the integration into the Readcoop website.

[4] https://readcoop.eu/ee-test/

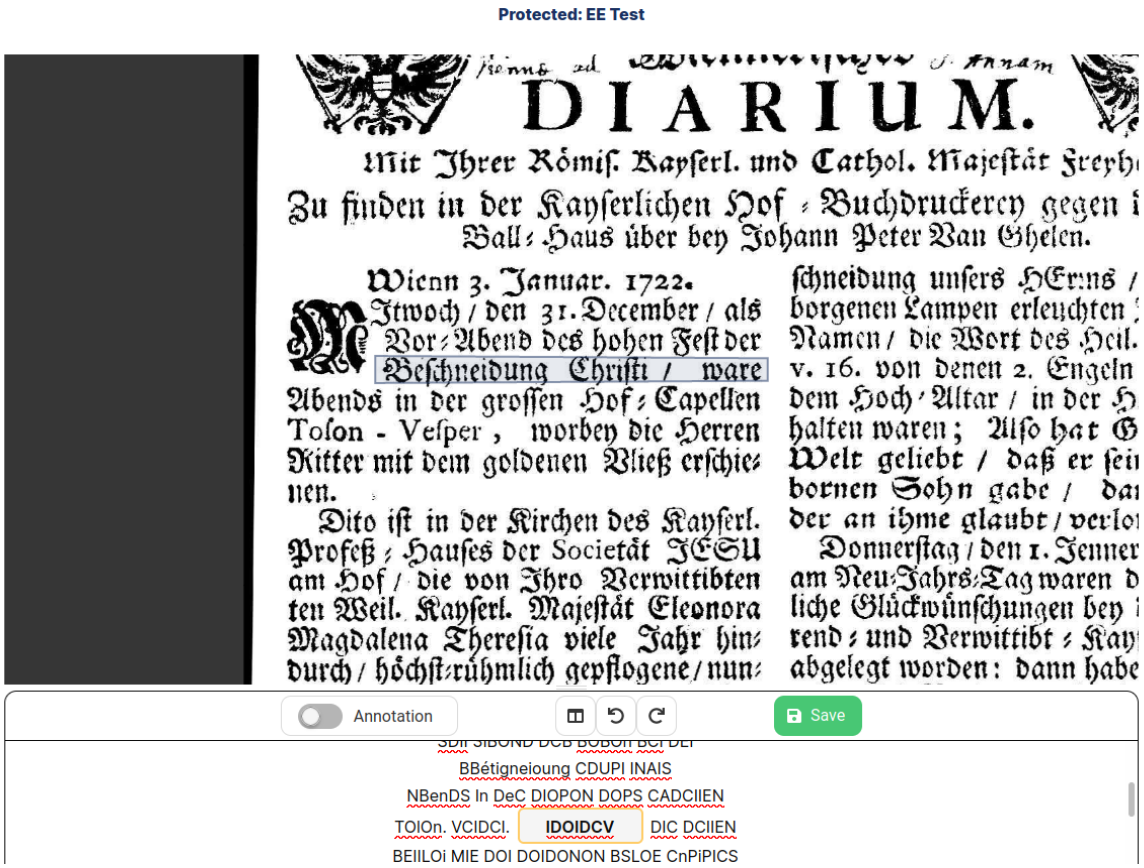

Figure 4. HTR Plugin showcased on Readcoop Website

Also the text editor is an essential part of TranskribusLite, the web version of Transkribus with enhanced usability. It allows users to use most of the features of the expert client on the web, such as uploading documents, starting layout/text recognition, transcribing or searching entire collections.

TranskribusLite[5] uses the text editor since version 1.1.0, which can be seen in the image below:

[5] https://transkribus.eu/lite

Figure 5. HTR Plugin integrated in Transkribus Lite

HTR Plugin integration in the Transcribathon Tool

In the process of integration of the HTR technology and line based layout functionality into the existing Transcribathon plattform Facts & Files started with a test setup. The first task was to evaluate alternatives available for combining the HTR text and the layout information to enhance the functionality of the existing transcription tool. The first evaluated solution uses the Annotorious toolset, which is an open source web annotation tool developed by AIT (https://github.com/recogito/annotorious).

Integrating HTR output through Annotorious library

Similar to the existing Transcribathon tool, Annotorius uses the OpenSeaDragon library for displaying and interacting with images available as IIIF resources. The Annotorious OpenSeaDragon extension allows the user to annotate the canvas of the viewer. It can also be used to load pre-existing annotations. To test Annotorious, it was needed to get layout polygon data and HTR results. In order to get these we used Transkribus manually to run a recognition of a sample document. Transkribus allows users to export the results of the recognition in different formats. We chose TEI and transformed it into a JSON format, readable by Annotorious thanks to an XSL transformation made and run in oXygen. Then, the JSON output was used in a pre-made Annotorious JavaScript function, which was used to load the annotations directly in the OpenSeaDragon viewer. Even as this process was highly manual, it allows one to see how transcriptions could be presented as annotations in the viewer and how the components could work together.

This test setup is loading the JSON object provided as Transkribus HTR output, and renders the layout within the Transcribathon Enrichment tool. The following screenshots depict the IIIF viewer with layout information which is connected to the text editor of the Transcribathon tool. Figure 6. shows the rendering of layout information in image viewer, where the text regions detected by HTR are indicated with polygon shapes.

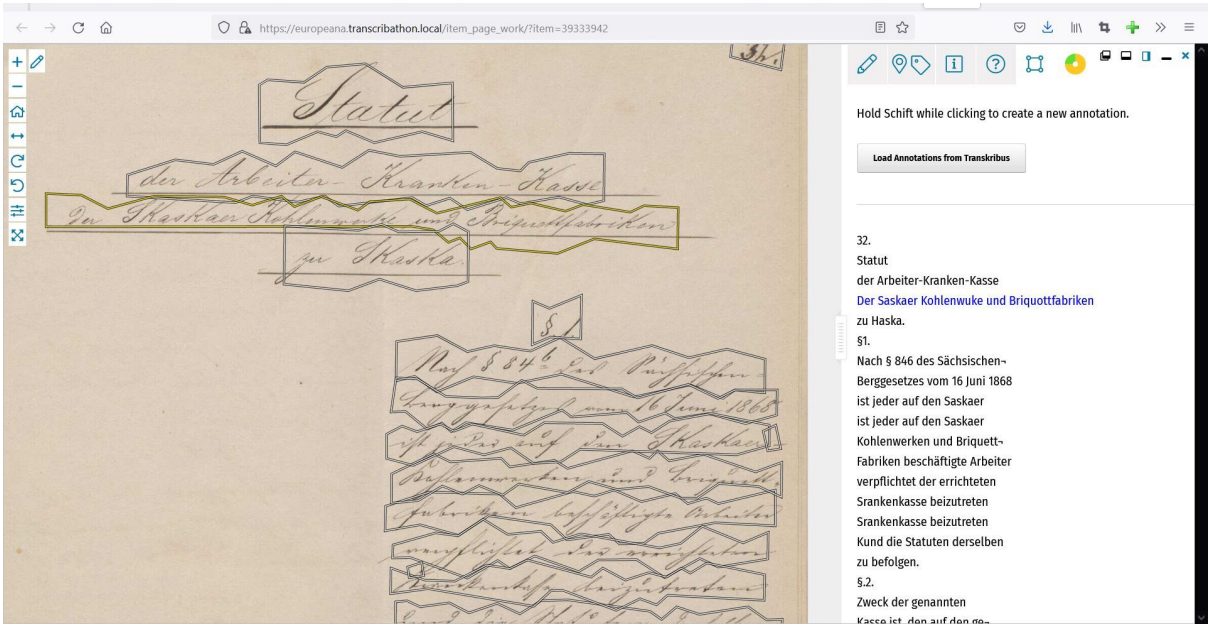

Figure 6. Rendering HTR layout in Transcribathon Tool using Annotorious library.

Figure 7 showcases the overlaid Annotorius editor for text regions. The editor is displayed when a text region is selected and displayed in a floating move. The user has the possibility to edit the text automatically recognized using Transkribus HTR.

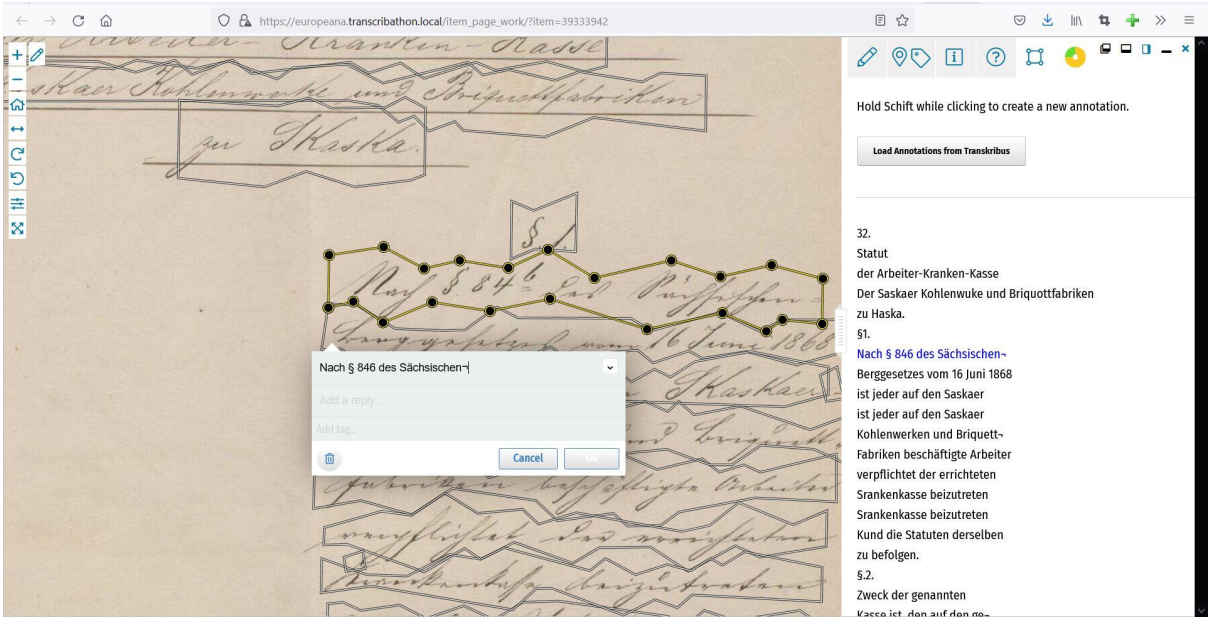

Figure 7. Transcribing text regions using Annotorious.

Using the HTR Plugin within the Transcribathon tool

The integration of the HTR plugin described in previous sections is based on the web packages provided READ COOP. This solution provides an integrated HTR viewer and editor. The following Figures showcase the new plugin integrated within the test Transcribathon website.

The following figures show the results of the READ COOP webpackages' incorporation into the Transcribathon test website[6]. Figure 8 shows the synchronization between the transcribed text and the corresponding region in the original document.

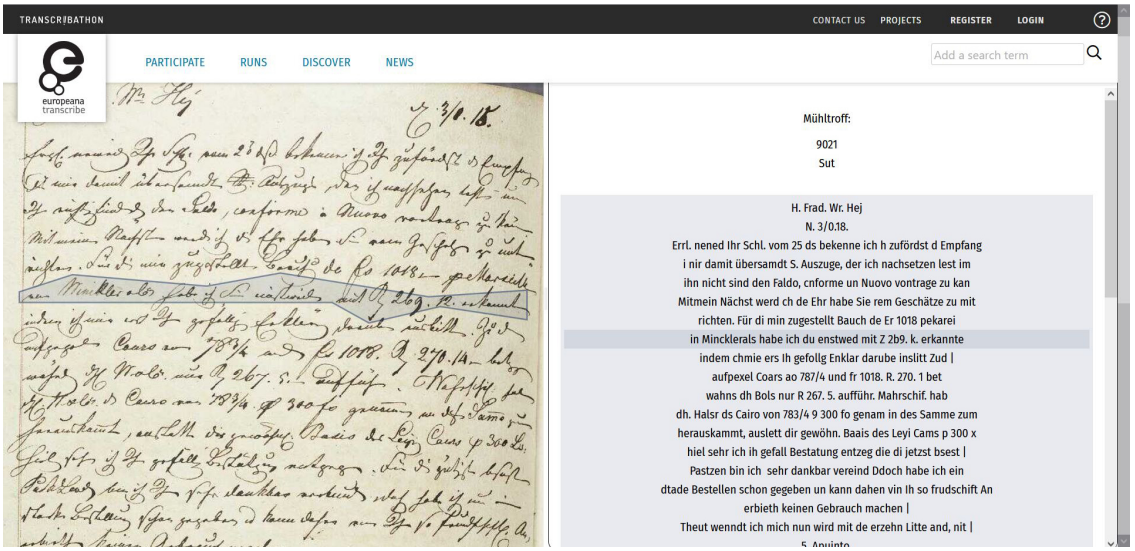

Figure 8. HTR Plugin integrated within the Transcribathon tool

Figure 9 presents the advanced annotation and text editing functionality within the integrated HTR plugin, including text formatting, tagging with persons, dates and locations, keywords, comments, categorizations.

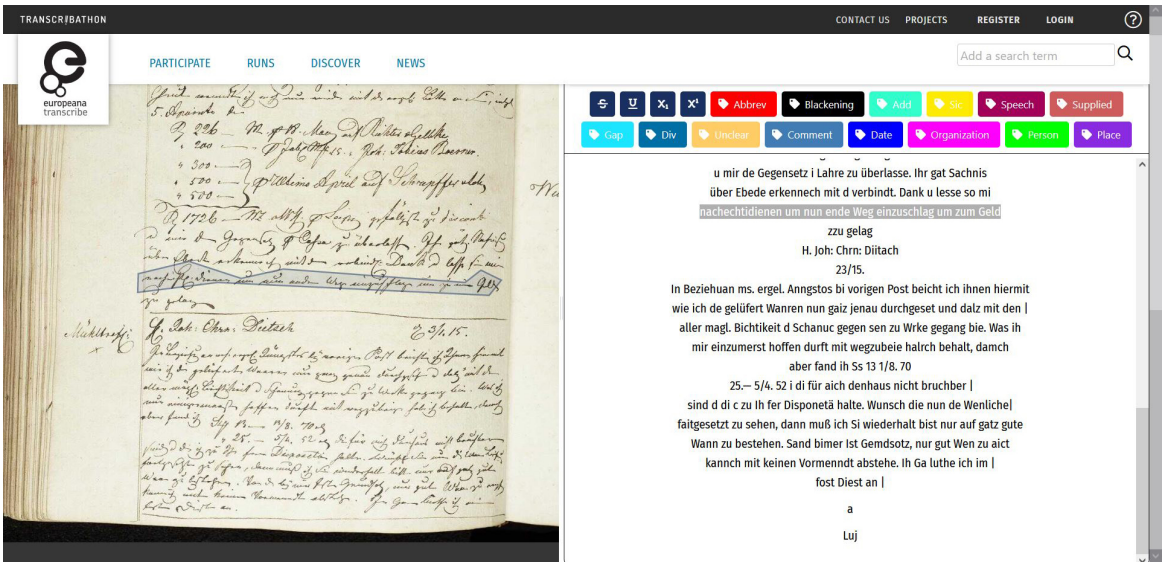

Figure 9. Advanced editing components integrated within the Transcribathon tool

[6] https://europeana.fresenia.man.poznan.pl/dev/documents/story/htr_sample_tst/

Conclusions

This document presents the work carried out for achieving milestone 2 of EnrichEuropeana+ Action, the implementation of the HTR Plugin.

It presents the current state of development for the pluggable transcription editor and the rendering of automatically detected text regions within the historical documents.

A first prototype for integrating the HTR technology within the Transcribathon tool is presented as well.

The HTR detection for selected documents was carried out using the Transkribus Tool. The final integration of the HTR technology will make use of the services developed within the scope of Task 2.1 Handwritten Text Recognition Services, which will be presented in the following milestones MS4: First release of HTR Services and MS5: Enhanced UX in Transcribathon Platform .