Computing Melville

An automated HTR campaign on Transkribus

Federica Bonifazi – federica.bonifazi@studio.unibo.it
Orsola Maria Borrini – orsolamaria.borrini@studio.unibo.it
21/11/2022

Index

Introduction – definition of the problem

Computing Melville[1] is a small-scale annotation campaign on a collection of Herman Melville's manuscripts regarding his last novella, "Billy Budd", left unfinished in 1891, and "Rip Van Winkle's Lilac", an unpublished experimental combination of prose and poetry.

The campaign has been carried out using Transkribus, a platform for the digitisation, AI-powered text recognition, transcription and searching of historical documents.

Our aim is to develop a Machine Learning system able to perform a transcription task on Melville's handwritten documents: the model is trained on a selection of chapters from "Billy Budd" manually annotated by the team and tested on some of "Rip Van Winkle's Lilac" manuscript's leaves.

Annotation Pipeline

Task definition

The first step of the workflow was to define the task: to perform supervised training on a ML model it was crucial to first provide it with an annotated corpus to be used as a training set and a raw corpus for the evaluation phase.

Starting from *Transkribus'* guidelines[2] on how to manage an annotation campaign, we decided to proceed with a corpus composed of the first 16 chapters of "Versions of Billy Budd" taken from the *Melville Electronic Library*[3], consisting of a total of 175 pages and ca. 17000 words[4] to use as our training set. The entire manuscript is available on the website as a diplomatic edition[5], displaying photos of each page of the manuscript and its correspondent transcription; unfortunately, nor the images nor the transcription could be downloaded or exported so we were forced to take screenshots of each page (drastically reducing the quality of the images), but we could use the transcription as a base for the one produced by the annotators during the campaign.

The initial corpus was then expanded with the validation set chosen for our HTR model: the poem "Rip Van Winkle's Lilac", located at the end of the homonymous manuscript and digitised and displayed with the permission of Houghton Library in *Melville Electronic Library*[6]. This time no diplomatic transcription was provided, while digital images of the manuscripts' leaves were accessible for download on the website's GitHub repository[7].

Analysing the so chosen corpus before starting the transcription campaign some challenges were noted and specific solutions were employed and declared to allow annotators to start from a common ground: in particular, the original manuscript showcased many leaves and leaf fragments added and removed, clearly suggesting the many revisions put in place by the author. This was an issue that risked complicating the ML task and which we decided to solve by considering and transcribing only those leaves that were closer to the final authorial version (that is, most of the time, the leaves with the mount).

---

[1] Website available at https://orsolamborrini.github.io/ComputingMelville/ - last visited 21.11.2022

[2] *How To Transcribe Documents with Transkribus – Introduction* (url: https://readcoop.eu/transkribus/howto/how-to-transcribe-documents-with-transkribus-introduction/ - last visited 17.11.2022)

[3] All rights are reserved to the original owner and publisher https://mel.netlify.app/manuscripts - last visited 17.11.2022

[4] Cfr. 2 "[…] that you start the training process with between 5,000 and 15,000 words (around 25-75 pages) of transcribed material. If you are working with printed rather than handwritten text, a smaller amount of training data is usually required."

[5] Here to the first page of the first chapter of "Versions of Billy Budd" diplomatic edition url: https://app.textlab.org/transcriptions/16900 - last visited 17.11.2022

[6] All rights are reserved to the original owner and publisher https://mel.netlify.app/rip-van-winkle-lilac - last visited 17.11.2022

[7] https://github.com/performant-software/mel-website - last visited 17.11.2022

For the same reason, every addition to the text has been left in its original position on the page, as we are more interested in showcasing an "analytic" transcription of Melville's manuscripts and all numbers present on the pages, both at the top and bottom, have been ignored, as the original source of the image does not state clearly their provenance and we wanted to stay focused on Melville's handwriting only. Similarly, to avoid any confusion in the recognition of the characters, section breaks' glyphs and any other mark (circles, pencil smears and even underlinings) on the pages have been ignored and not transcribed. For everything else we relied on *Transkribus'* Transcription Conventions[8], except for other two cases that we considered worth of particular attention, and for which a more detailed case-by-case analysis can be found in the documentation of the project at the original website[9]:

- Strikethrough, mainly tagged as such when appearing in line with the rest of the text line and ignored when co-occurring with superscript-tagged text
- Superscript, tagged as such when the x-height of the characters was included in the main line area (highlighted in purple on Transkribus), or included in a dedicated extra line, considering it as normal text (preferred solution in edge cases too)

Pilot

Following the so defined annotation guidelines from the previous paragraphs, the annotators started to process 10 leaves each from the "Billy Budd"'s manuscript on *Transkribus*, doing both the layout parsing and the transcription. Then the datasets were swapped and respectively checked to find possible controversial situations. Cases of disagreements were discussed, and annotation and transcription parameters were changed accordingly. This first step corresponded also to our first pilot campaign, that mainly resulted in improving the guidelinesfor the handling of the superscript and strikethrough text passages and illegible text not transcribed by *MEL's* experts, that we decided to ignore as well.

The second pilot was carried out on 20 other leaves of the manuscript (11-20 of the first chapter, 11-14 of the second and 1-6 of the fourth chapter) and allowed us to refine the handling of superscripts, especially in cases where multiple spaced out superscript text passages were present. This second pilot was followed by a third and last one that was carried out randomly during the first moments of the final annotation campaign and allowed further refining to the guidelines.

Campaign

The proper annotation campaign started right after the second pilot and was conducted following the declared guidelines on the remainder of the selected corpus.

The annotated data were then used to train two different recognition-models (*Melville Handwriting 3.1* and *Melville Handwriting Base Model*) on the downloaded version of *Transkribus,* both having as Training Set the whole 16 chapters from "Billy Budd" and as Validation Set the pages from "Rip Van Winkle's Lilac". However, a base model for English Handwriting was added to the *Melville Handwriting Base Model* training to refine the recognition process.

Both models were trained relying on the PyLaia HTR engine supported by *Transkribus* and the paramethers were defined according to the guidelines for HTR Models' Training[10]: for both models we sticked to a default early stopping of 50 epochs and a learning rate of 0,0001%, while for the model training with the addition of the English Handwriting base model the total number of epochs was lowered from 250 to 150 to avoid overfitting.

---

[8] https://readcoop.eu/transkribus/howto/transkribus-transcription-conventions/ - last visited 17.11.2022

[9] Cfr.1 in section *The transcription pipeline – Annotation Guidelines* at *Computing Melville*

[10] *How To Train and Apply Handwritten Text Recognition Models in Transkribus* https://readcoop.eu/transkribus/howto/how-to-train-a-handwritten-text-recognition-model-in-transkribus/ - last visited 17.11.2022

For the *Melville Handwriting Base* Model the chosen parameters resulted in a satisfying outcome straight away, whereas the training for the *Melville Handwriting Model* was slightly more challenging. In fact, three other versions were tested before arriving to the final version *Model Handwriting 3.1*:: the main refinements concerned the number of epochs (at first set too low at just 125/150) and the learning rate (first set at the default 0,0003%, then considered a little too harsh and lowered for better results in the last trials).

The accuracy of the final two models can be compared by analysing their Learning Curves (*Figure 1* and *Figure 2*) indicating the variation of the Character Error Rate (i.e., the percentage of characters that have been transcribed incorrectly by the Text Recognition model) for number of epochs.

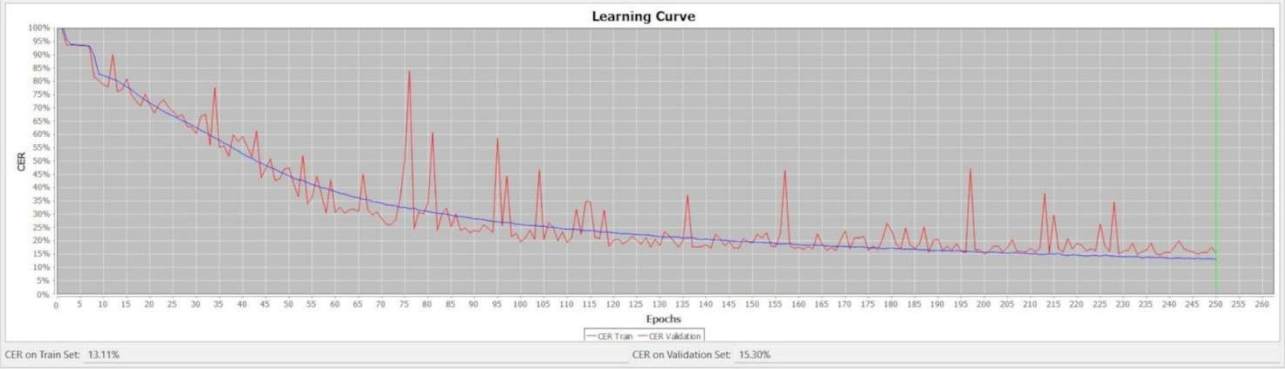

Figure 1: Melville Handwriting 3.1 - Learning curve of the trained HTR model

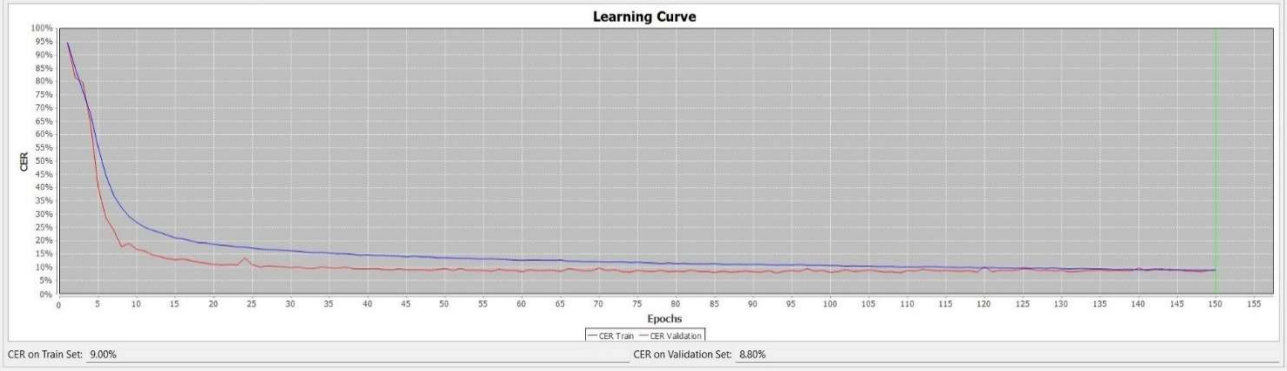

Figure 2: Melville Handwriting Base Model - Learning curve of the trained HTR model

In the graphs above the blue line represents the progress made by the model on the Training Set whereas the red one represents the progress of evaluations on the Validation Set, on which the program tests itself after the training. The trend and final value of the CER for the Validation Set is of course the most significant as it shows how the model is capable of generalising, performing on pages that it has not been trained on. The results of *Melville Handwriting Base Model* are slightly better performative as a CER of 10% or below can be seen as very efficient for automated transcription; however even *Melville Handwriting 3.1* resulting in a CER on Validation Set lower than 20% proved itself to be more than sufficient to start working with and could definitely be improved in further trials.

Annotation and use

In designing our annotation campaign, we have tried to apply the FAIR principles for data publication, making the results of our research findable, accessible, interoperable and reusable.

Outcomes and criticalities

Given the obtained results, the amount of the starting corpus selected and the small team behind the annotation, the outcome of the campaign was considered undeniably satisfactory, although a lot can still be improved, starting from some criticalities that emerged along the process.

Comparing the transcription made on the Validation Set by the annotators and by the transcription models there are two main criticalities that we consider worth of notice, and they regard what was perceived as a challenge from the beginning of the campaign: strikethrough and superscript. It appears clear that none of the models has been able to recognize these labels (*Figure 3*), despite them being thoroughly tagged throughout the entire selected corpus. Probably more instances of the two phenomena were required for a better recognition by the model, but a lot of difficulties were encountered during the tagging in *Transkribus* of the two cases[11]: this may be an issue worth reporting to the developers or to further analyse to inform future users on the best way to approach it.

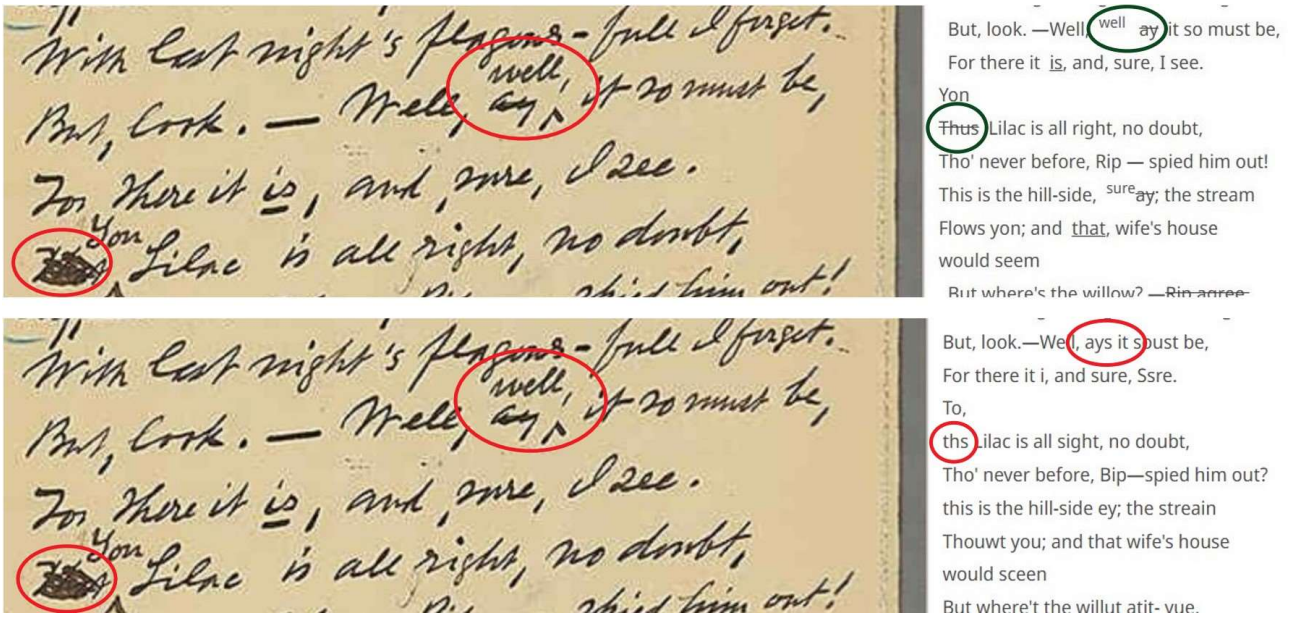

Figure 3: Comparison between annotated version of *RIP Van Winkle's Lilac* pg. 1 by the annotators' (top) and the *Melville Handwriting 3.1* model (bottom)

Further works

Certainly, this is only the beginning of what could be a much more extent campaign on Melville's original manuscripts. The limited dimensions of our team and, consequently, of the corpus we annotated, prevented us from tackling more in-depth research on the topic. Some improvements could certainly be made by expanding the training corpus and by using HQ images.

However, we are fairly convinced that this project could stand as an inspiring push towards authorial annotation campaigns by means of AI and ML systems.

[11] For instance, if pieces of texts were both superscript and strikethrough *Transkribus,* despite apparently allowing tagging both, was indeed able to show and probably recognize just the superscript, removing strikethroughs.

Sitography

Melville Electronic Library: https://mel.netlify.app/

Transkribus Lite: https://transkribus.eu/lite/it

Transkribus How To Guides: https://readcoop.eu/transkribus/resources/how-to-guides/

- *Transkribus Transcription Conventions*: https://readcoop.eu/transkribus/howto/transkribus-transcription-conventions/
- *How To Transcribe Documents with Transkribus – Introduction*: https://readcoop.eu/transkribus/howto/how-to-transcribe-documents-with-transkribus-introduction/
- *How To Train and Apply Handwritten Text Recognition Models in Transkribus:* https://readcoop.eu/transkribus/howto/how-to-train-a-handwritten-text-recognition-model-in-transkribus/