

Transkribus: Reviewing HTR training on (Greek) manuscripts

Transkribus, University of Innsbruck and READ-COOP SCE (ed.), 2019. <https://transkribus.eu/> (Last Accessed: 10.12.2021). Reviewed by Elpida Perdiki (Democritus University of Thrace), eperdiki@helit.duth.gr.



Abstract

Transkribus is a fully developed GUI (graphical user interface) platform offering (among others) the possibility to train HTR models with AI. It supports auto-transcription and searching of historical documents and is oriented towards Archives, Libraries, and researchers. This review describes most of Transkribus' features by outlining the results of a project researching HTR on Greek manuscripts. Transkribus proves to be a useful solution for painlessly implementing state-of-the-art technology on Humanities, regardless of technical expertise or resources' limitations. Any discussion in this review for further development of the platform is only provided given some of the Greek manuscripts' peculiarities.

Introduction

1 OCR and most recently HTR technologies have proven to be of assistance in several scientific fields offering ever-evolving solutions for off-/online text recognition. Among those, Humanities and especially research on manuscript tradition(s) could benefit a lot from the progress of such technologies. Transkribus is one of the very few projects which have fully developed a ready-to-use tool for training HTR models via AI

(Artificial Intelligence) to automate historical documents' transcription.¹ This review of the Transkribus platform is mostly based on personal experience with the software, gained through research regarding my PhD. I have extensively used Transkribus for testing automatic transcription of large document data sets, with a case study of witnesses containing John Chrysostom's *Homilies* of Saint Paul's *Epistles to Titus*. Part of my research results has been published at *Classics@ 18*, "Ancient Manuscripts and Virtual Research Environments" issue ([Perdiki and Konstantinidou 2021](#)). All points of this review take into consideration mostly the usability of the software in my field. No technical details are reviewed, as a result, an expert could cover that aspect.

2 Initially developed by the University of Innsbruck in 2013 and the Horizon 2020 EU research project READ in 2016 ([Kahle et al. 2017](#)), Transkribus is a cross-platform ecosystem continually being developed since 2019 by the READ-COOP SCE, a European Cooperative Society to sustain and evolve the project.² Although not fully open source, its code can be found both in GitHub and in GitLab repositories, where is now solely stored ([Kahle et al. 2017](#)).³ Transkribus offers two main solutions for the automation of manuscript transcription. One could either work with the "Expert Client",⁴ which has a very rich in features GUI (graphical user interface), or the "Transkribus Lite" option on a web browser,⁵ with a cleaner interface, yet a lack of some tools. In the course of my research, I have extensively used both solutions and specifically I have been working with "Expert Client" versions 1.10.0-1.16.1 on Windows (10 & 11, 64bit, Intel® Core™ i5-8265U 1.60GHz, 8GB RAM) and Linux (Mint 19.00, 32 bit, Intel Pentium T3200 dual-core processor 2GHz, 3GB RAM) and with versions 1.0.7-1.3.1 of "Transkribus Lite" on a Chrome browser (versions 93.0.4577.63-97.0.4692.99). Transkribus' "Expert Client" requires a minimum Java version 8 to be properly executed.⁶ Apart from that, no other software or hardware dependency is necessary for the platform to be fully functional.

3 In general, Transkribus has been offered to the public as a freemium model, meaning that all versions (GUI and web-based) come free to use in most of their functionality, except for the ability to automatically transcribe documents with already trained HTR models. For that purpose, all signed up users acquire 500 free credits (equal to a 400 pages transcription) upon subscription and then move on to a payment method depending on their research needs. Other than that, Transkribus offers a scholarship programme as well, for selected students.⁷

Methodology and implementation

4 The need of accessing ancient/historical documents goes back to the first historians. People and especially scientists advance by studying sources of existing knowledge. It goes without saying that in recent years a lot of progress has been made in the field of text recognition – especially HTR, as a means to retrieve more quickly and efficiently historical documents. Transkribus builds upon that progress by integrating into one platform advanced technology tools such as HTR and layout analysis via neural network training, text to image matching, keyword spotting, TEI/XML mark-up tool, and even online collaboration on same document collections, as a form of a Virtual Research Environment ([Perdiki and Konstantinidou 2021](#)). Similar projects are a) eScriptorium,⁸ which seems to offer the same features but as a fully open-source application, b) python systems implemented with TensorFlow library,⁹ c) Kraken, an OCR system for historical documents,¹⁰ and d) Tesseract, the OCR engine developed by Google and mostly used in many projects.¹¹ All the latter however presuppose advanced knowledge of coding and CLI (command-line interface).

5 So, if one would seek for Transkribus' innovation in the field, they should not focus specifically on HTR technology and the applied algorithm (yet not to be denied of course), but rather on the possibility of HTR to be exploited by a vast majority of researchers. Due to its effective and user-friendly environment, researchers, archives/collections, and research centres across the linguistic variety of the world could easily access and make effective use of Transkribus' features leading to general progress in Humanities. Ancient documents, undiscovered for many years, could now be massively and automatically studied, regardless of the language they are written in. As a result, valuable historical, literary, and linguistic information could be retrieved much quicker than in the past. More importantly, despite offering state-of-the-art tools as the pioneering possibility to train your neural network for text recognition at any historical document, emphasis should lay on the fact that this very function is completely possible even with a low-power hardware machine such as a 32- or 63bit laptop and few GBs of RAM (see ibid. “Introduction”, for the technical specifications). That software's performance proves to be liberating for independent researchers or scientific projects with a low budget/workforce ([Perdiki and Konstantinidou 2021](#)).

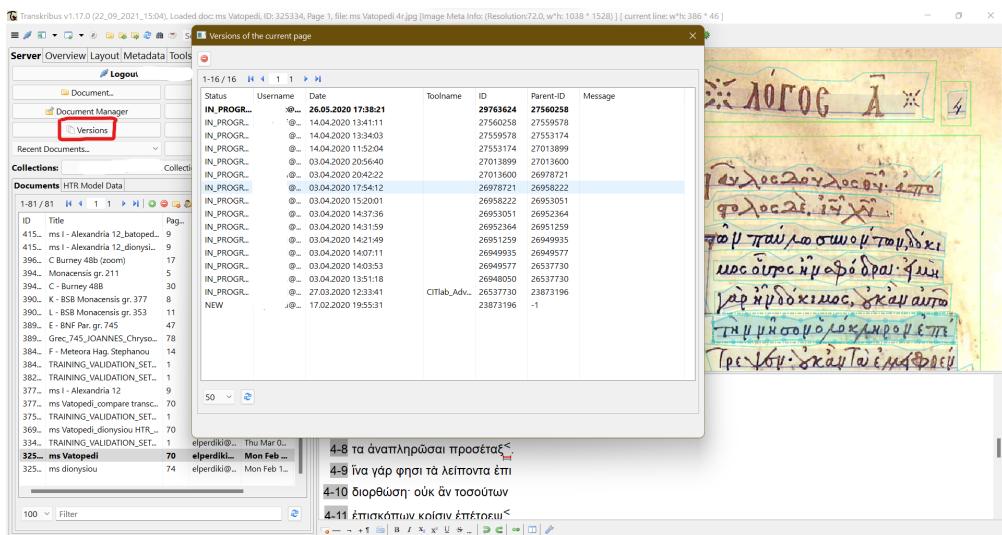


Fig. 1: Rollback to specific versions option.

6 Behind the convenient GUI platform or the web portal of Transkribus lies a compact backend architecture that consists of interconnection between several Java applications, an Apache Tomcat and an Oracle Database (its visual representation can be found at [Kahle et al. 2017](#)). Subscribed users log in to the platform and import selected documents images to private collections, which are only shared with their collaborators to whom the collections manager assigns custom roles of full or restricted access under permission. File images are imported on a server via a) local folder, b) FTP, c) URL of DFG Viewer METS or d) URL of IIIF manifest. Those files are called documents and are organised in collections created by the user. Then a PAGE XML with a unique key is generated for each image – a format framework that stores documents' structural information and image characteristics for recording and evaluating document analysis workflow ([Pletschacher and Antonacopoulos 2010](#); [Kahle et al. 2017](#)). That PAGE XML file, along with relevant metadata of the document stored in the database, is augmented to a new document version every time editing progress is saved. As a result, users can roll back to whichever version, if needed – there is a specific button for that purpose on the main panel of the software, by which users can select from a detailed list of all saved versions (see [fig. 1](#))

7 An Apache Solr indexing solution provides keyword spotting through full-text search. The interface has been written in Java and C++ alongside OpenCV and JNI. All this workflow is documented in detail in the paper “Transkribus - a Service Platform for Transcription, Recognition and Retrieval of Historical Documents” ([Kahle et al. 2017](#)). For layout analysis, certain tools are deployed provided by the National Centre of Scientific Research “Demokritos” (NCSR), the Computer Vision Lab (CVL) of the

Technical University Vienna, and the Computational Intelligence Technology Laboratory (CITlab) at the University of Rostock (more on the matter can be found at Kahle et al. 2017). Text recognition is offered in two different ways: a) ABBYY FineReader 11 SDK is used for OCR (Optical Character Recognition) of printed texts and b) offline HTR engines provide manuscripts transcriptions via HMM (Hidden Markov Models) and language models, or RNN (Recurrent Neural Networks) and no language models, through developments from the Pattern Recognition and Human Language Technology (PRHLT) research centre, University of Valencia (UPVLC) and the CITlab and PLANET intelligent systems GmbH respectively (Kahle et al. 2017). All systems described above are responsible for the neural model training on the data provided by the users. Users annotate via Transkribus GUI relevant points of the manuscript images and provide ground truth data of transcription. Both segmented images and their transcriptions are then considered to be training data and, when selected, can be used to train an HTR model at a certain manuscript collection. A portion of the same data is defined by the user as the validation set, which is used post-training to evaluate the HTR performance of the model (Kahle et al. 2017).

An instance of Transkrib-ing

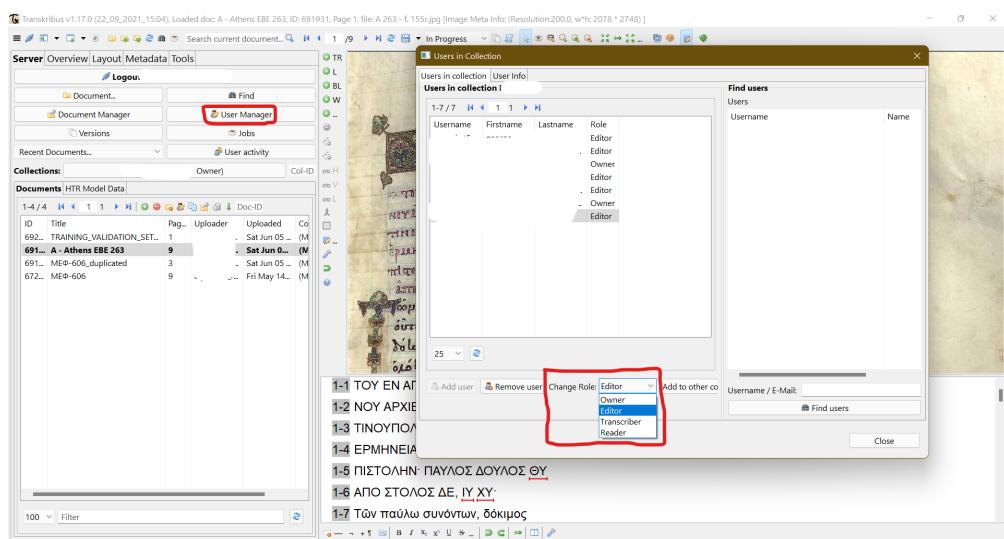


Fig. 2: Setting user roles and permission in collections.

8 An example of usage would showcase more vividly Transkribus' possibilities. So, once logged in (one cannot download the platform unless signing up for an account), a user can import manuscript images organised in collections (as described in the "Methodology and implementation" section ibid.). Each collection can be modified by

more than one user, if users' roles and permissions are assigned (i. e., administrator, editor, viewer etc. – see [fig. 2](#)).

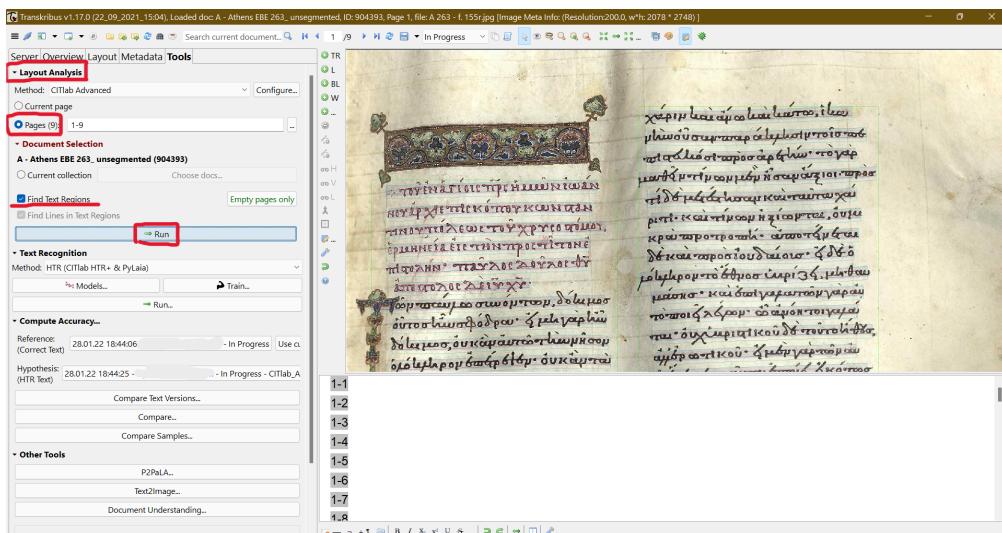


Fig. 3: Transkribus automatic layout analysis.

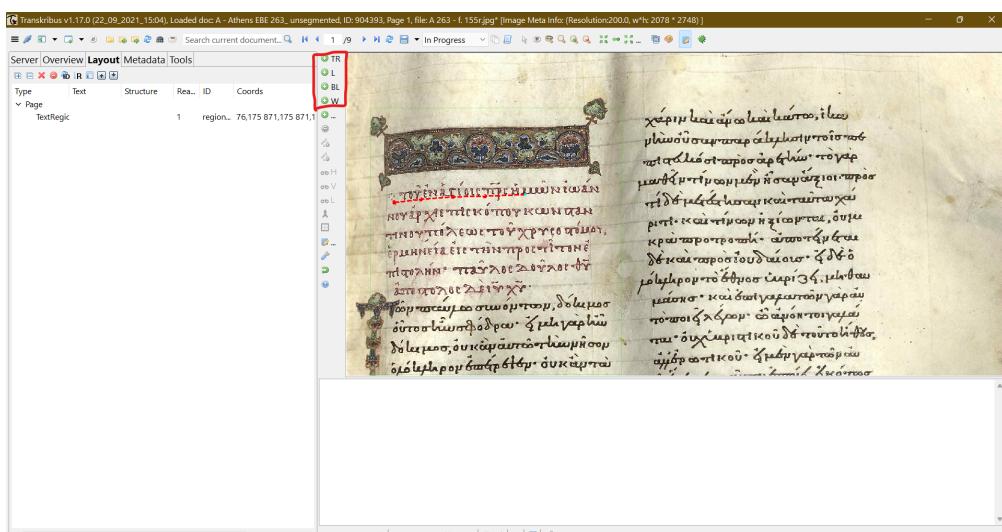


Fig. 4: Transkribus manual layout analysis.

9 That feature ensures that a researcher could work collaboratively with a group of other researchers, or even a crowdsourcing team, without risking any meddling with the training data and most importantly with expediting the research progress. All data in collaborative collections are seamlessly synchronised and connected via the Transkribus server, so any document changes are almost instantly visible (a refresh of the environment is of course required, though, should any recent change happen, a pop-up window informs the user anyhow). From that point on, users can continue to the most important step of the HTR training. Image files should be segmented to define the

baseline of the text, a process known as layout analysis, which can be either automatic (see [fig. 3](#)) or manual (see [fig. 4](#)).

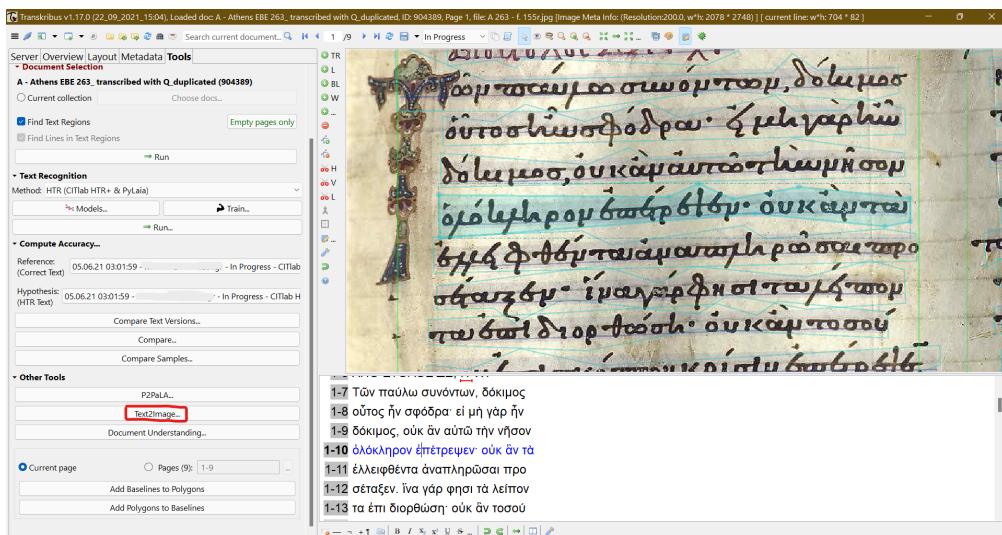


Fig. 5: Transcription and Text2Image tool.

10 After segmentation is complete, the user may provide line per line diplomatic transcriptions of the depicted text, again either manually or automatically via “Text2Image” tool, by importing a .txt file (which must be named exactly like the image file, so Transkribus can successfully synchronise image and relevant transcription – see [fig. 5](#)).

11 Some discussion has arisen on how much data and epochs (number of times a learning algorithm sees the complete dataset) are necessary for successful training of any neural network really but also regarding HTR ([Ströbel, Clematide, and Volk 2020](#)). For instance, Transkribus guidelines recommend a minimum transcription of 5,000–15,000 words and 50 epochs, depending on the format of the text;¹² printed texts require less data than manuscripts, due to the uniformity of the script style. Other researchers have experimented successfully with less data ([Perdiki and Konstantinidou 2021](#)) or have argued about why and when more data and more epochs could result in overfitting, meaning to worsen the HTR results ([Rabus 2019](#)). Decision on the matter should be made after taking into consideration

1. the quality of the images,
2. the uniformity of the script and/or the existence of more than one scribes ([Perdiki and Konstantinidou 2021](#)),

3. the utilisation of an already fully trained model in the same language (as a base model), which could boost up the training process, and
4. the research goals; a searchable text requires less HTR accuracy than a text to be included in a critical edition (some of those aspects are discussed at [Perdiki and Konstantinidou 2021](#)).

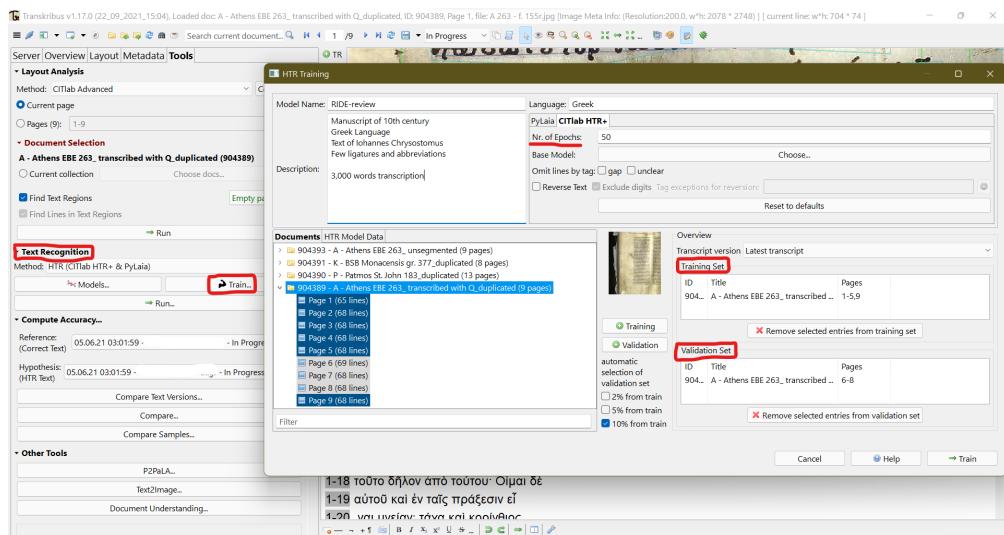


Fig. 6: Training an HTR model with AI.

12 For most projects, an average of 10,000 words should suffice for a successful model. With that step completed the user can proceed to the training of the model. A relevant button can be found at the “Tools” section of the panel and a second window prompts the user to define the details of the training, such as:

1. name and description of the model,
2. the language in which text is written [really any language at all or even more than one in the same document – I have already successfully trained models for Greek manuscripts from the 10th–14th c. A.D. ([Perdiki and Konstantinidou 2021](#)) and there are also few projects, that have worked with non-Latin scripts as Arabic or with multiple languages in the same model¹³,
3. desired № of epochs (system default is 50) and
4. setting the pages that will be utilised as a training set and those to be used in validation (see [fig. 6](#))

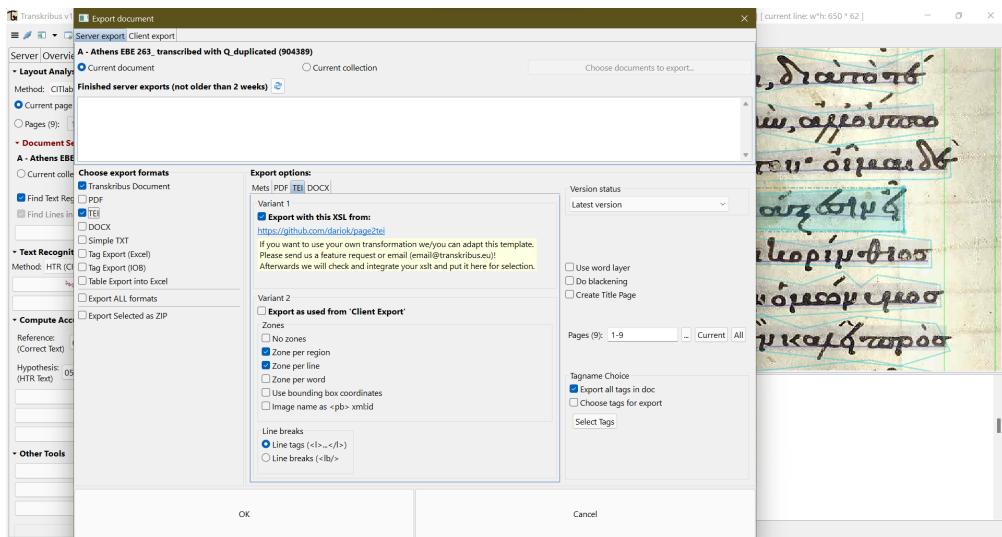


Fig. 7: Possible output formats of Transkribus documents.

13 If needed, one could easily benefit from an existing model in the same language, as a base model. That can be used as a means of data augmentation, as it helps train the model much faster if one has not had enough training data ([Perdiki and Konstantinidou 2021](#)). Depending on the size of the data set, the training will be efficiently completed within a few hours or days. Because training is executed on Transkribus servers, during the training period the user can freely continue to use this or any other software, or even shut down his/her computer, which frees up resources and delivers a performance that does not influence interface responsiveness. Upon completion, Transkribus notifies the user via email of the training progress, and models can be successfully applied at manuscripts transcriptions, further improved if needed and even publicly published for the benefit of the whole community. Transcribed collections can be easily exported to many formats: from simple .txt files to PDF documents or even a fully structured TEI file (see [fig. 7](#)). Exports can be sent via the platform's server to the user's email or saved locally to a selected folder.

14 The accuracy and efficiency of Transkribus as a complete solution of HTR emerge from the high number of DH projects (independent researchers and archive collections) that have successfully applied the software to research related to historical documents.¹⁴ Moreover, Transkribus has been researched by scholars of non-Latin alphabet scripts, such as Greek or Cyrillic alphabets, enlightening thus the methodology relevant to the implementation of HTR technology ([Rabus 2019](#); [Perdiki and Konstantinidou 2021](#); [Burlacu and Rabus 2021](#)). Despite, individual differences originating from the distinctiveness of every historical collection/script, all those projects feature one common conclusion: HTR technology, especially when provided in a user-

friendly environment, can and will undoubtedly speed up and facilitate research in Humanities.

Usability and user's support

15 The scope and the length of this review do not allow for further, more detailed demonstration of all aspects and possibilities Transkribus can provide to manuscripts research. Yet, because of the richness of the platform's tools and the sense of complexity these tools may develop to any user, the Transkribus team has provided extensive and sustained documentation of the platform titled "Resource Centre"¹⁵. What used to be a wiki database, is now an .html format of all relevant information, which inform beginners or expert users for every aspect of the software. From installation instructions (covered by relevant step to step screenshots) to how-to guides of most sophisticated features, FAQs, glossary, and even YouTube video tutorials, Transkribus knowledge is gathered in one efficient portal. The same "Help" function is also available in a button form, from the GUI of both "Expert Client" as well as "Transkribus Lite." An extra section of the documentation offers more information on the REST API provided by Transkribus, should any user like to implement it in another software.¹⁶ In the occasion that none of this documentation offers enough help, users could easily contact the Transkribus team via a relevant email address. From my own experience, a member of the team usually gets back to you within a day and kindly offers a solution or at least guidance towards one. If there are absolutely no solutions to a specific problem or in the case of a malfunction, users can easily report a bug or request a feature directly from the GUI of "Expert Client." A similar feature is not currently available at "Transkribus Lite" version. Apart from that functionality, users cannot provide any actual contribution to the enhancement of the platform. All maintenance is developed under the READ project ([Kahle et al. 2017](#)) and the READ-COOP SCE – hence the freemium model, fundings of which supports Transkribus' further development.¹⁷

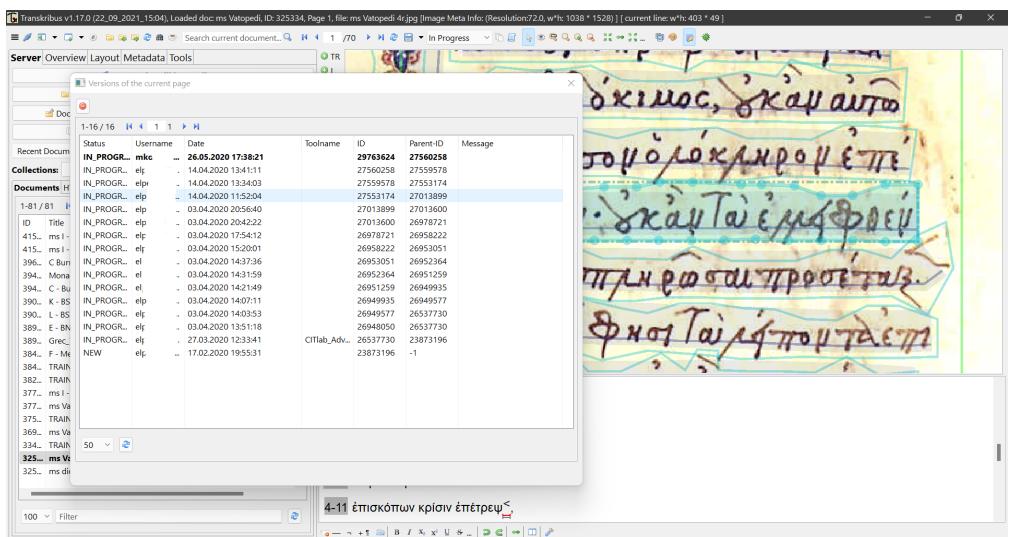


Fig. 8: Indication of multiple users working simultaneously on the same document/collection.

16 In the mentioned above documentation, there is extended information on the installation and execution of the software. Windows users can expect a flawless installation, yet I have encountered some instances in which installation was successful, but the software could not be executed. Transkribus provides guidance for that issue as well, but further action could be to temporarily disable any antivirus software. For Linux users, no issues should impact installation as it is straightforward, provided that one has experience with the Linux terminal and follows religiously Transkribus' instructions. Uninstalling the software is as easy if needed, and updates are automatic and sound. I have no experience with Transkribus on Mac OS (Operating System), so I could not possibly offer insight on the matter. The web version of the platform, "Transkribus Lite", requires only a sign-in step and no other plugin is needed for it to work properly. That portability of Transkribus is the main reason the platform proves to be an excellent VRE solution, especially for small research groups ([Perdiki and Konstantinidou 2021](#)). Since all progress is saved in Transkribus servers, users can work simultaneously on collections and each editing version is marked up with the username of either one (see [fig. 8](#)). The same functionality is available at "Transkribus Lite" as well. The potential of input and output multiple types offers a certain amount of interoperability, given the fact that due to the built-in text annotation feature, one could easily use exported files as data to another software. That is especially true when it concerns the TEI/XML files, which are fully machine-readable and could be used either in text mining techniques as NLP algorithms, or even in the preparation of a digital edition by encoding and structuring the transcription.

17 The other side of Transkribus' portability and interoperability comes (although not seamlessly) in the form of users' data management. Treatment of all users' data is described and determined in the relevant "General terms and conditions" page of Transkribus website,¹⁸ but it can be summarised under the statement: "A collection is private by default. [...] Transkribus team has the right to use uploaded material for testing and improving its services." ([Kahle et al. 2017](#)). An issue may arise regarding that policy because all data is stored and handled in Innsbruck, Austria, servers, meaning that any usage of the platform must comply with the EU GDPR law (General Data Protection Regulation). According to §8.3 of the "General Terms and Conditions" Transkribus page, ¹⁹ users can delete uploaded training and non-personal data, yet READ-COOP SCE may preserve copies of it. Considering the debatable section of GDPR "Data Deletion and Modification" ([Haque et al. 2021](#)), as well as the discussion on whether AI training data might be/contain personal data ([Liu et al. 2021](#)), Transkribus' handling of users' content seems questionable. Equally, unresolved is §8.5 in which is stated that users must ensure copyright licences for all uploaded data. That is a fair point, but a question arises on whether or not one conflicts with copyright law by using protected data in an AI training process, without distributing in public the data per se but only the research results derived from those data – a question partially answered by the §8.4, which declares that READ-COOP SCE "[...] grants the Customer all exclusive, transferable, sublicensable, worldwide indefinite rights [...]" when recognition results concern copyrighted material. Although no violation of data seems to take place, researchers should carefully read "General Terms and Conditions" before deciding if they accept the management of their data.

Interaction, GUI, and visualisation

18 Whilst normally training neural networks requires technical expertise, firm knowledge of programming languages and coding in general, Transkribus provides the user with a compact, fully-featured, yet easy to grasp graphical environment. With the supplement of the detailed documentation, no expert knowledge is necessary for a user to start training an AI model. Thus, Transkribus' users could either be experienced scholars/librarians or even inexperienced, simple users (students or else), which could provide valuable crowdsourcing input.

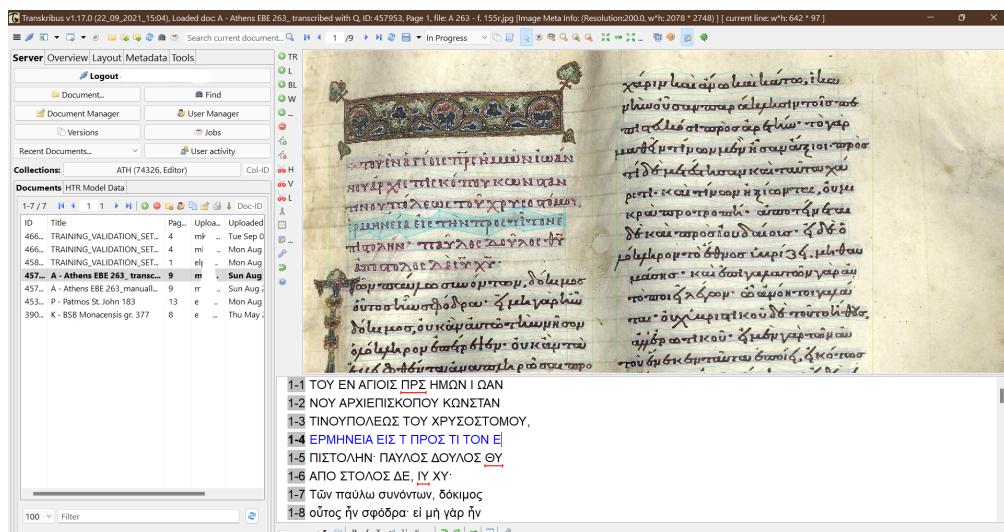


Fig. 9: Transkribus' GUI on a Windows OS.

19 The interface of the “Expert Client” version might seem a little too complicated at the beginning, due to the small size of the numerous buttons (see fig. 9). Although no function allows searching for a certain tool (which would be fairly appreciated, regarding the compact buttons panel), features are arranged in five (5) main tabs: server, overview, layout, metadata, and tools, which imply the relevant group of functionalities. In spite of that functionality structure, even experienced users might develop some difficulties in discovering the appropriate tool for each action. However, documentation provides useful insight for almost every aspect of the platform, so the interface becomes eventually familiar. At certain points of editing collections or training progress, prompting pop-up windows might appear, which explain functionality or remind the user of crucial steps, such as saving or refreshing documents, error messages, etc.

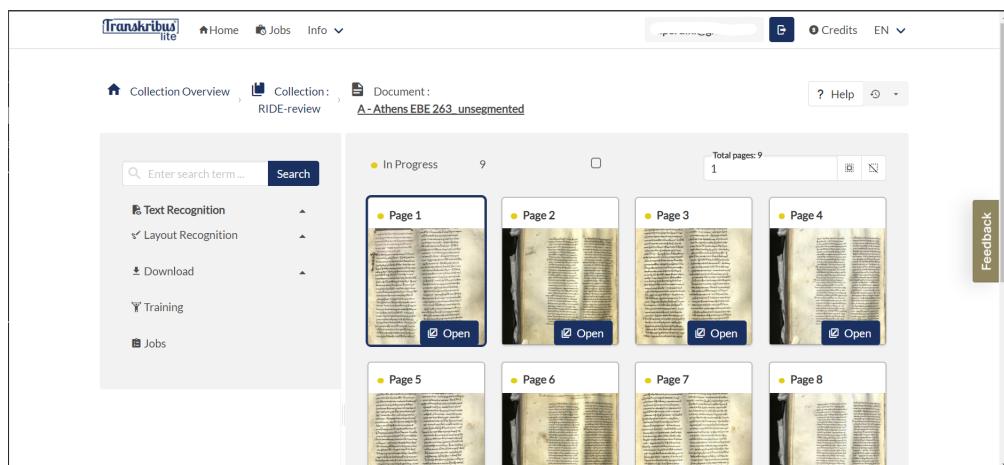


Fig. 10: Transkribus Lite (web version) GUI.

20 As mentioned earlier (see ibid. “Introduction” section), “Transkribus Lite” has a much cleaner and easy to grasp interface, which seems to be more helpful with inexperienced users (see [fig. 10](#)).

21 Up until recently one could only view and transcribe documents in the web version of the platform. Yet in the recent updates more functions became available and for the moment the following are possible: a) collections’ management, b) images’ uploading, c) documents’ view and editing/transcribing, d) text searching, e) HTR training or/and recognition, f) jobs viewing (as in tasks progress) and g) credit manager (as in checking credits’ balance).²⁰ Lastly, despite offering the aforementioned rich in features graphical interface and the web-based version, there is also a possibility to make good use of Transkribus’ functions within third-party applications through a RESTful API ([Kahle et al. 2017](#)). Full documentation of it is provided at the Resources Centre of Transkribus, as mentioned above (see ibid. “Usability and user’s support” section).

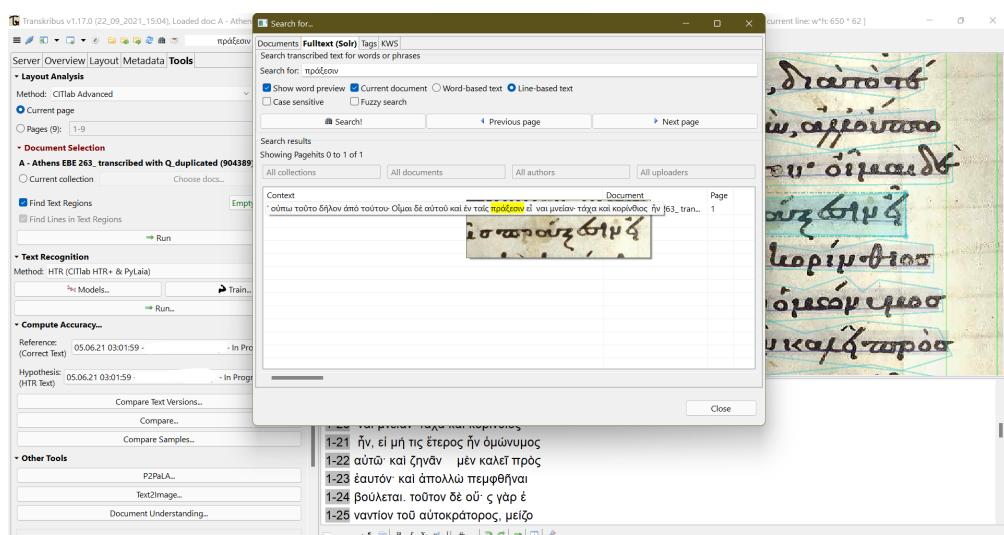


Fig. 11: Keyword spotting feature with returned results.

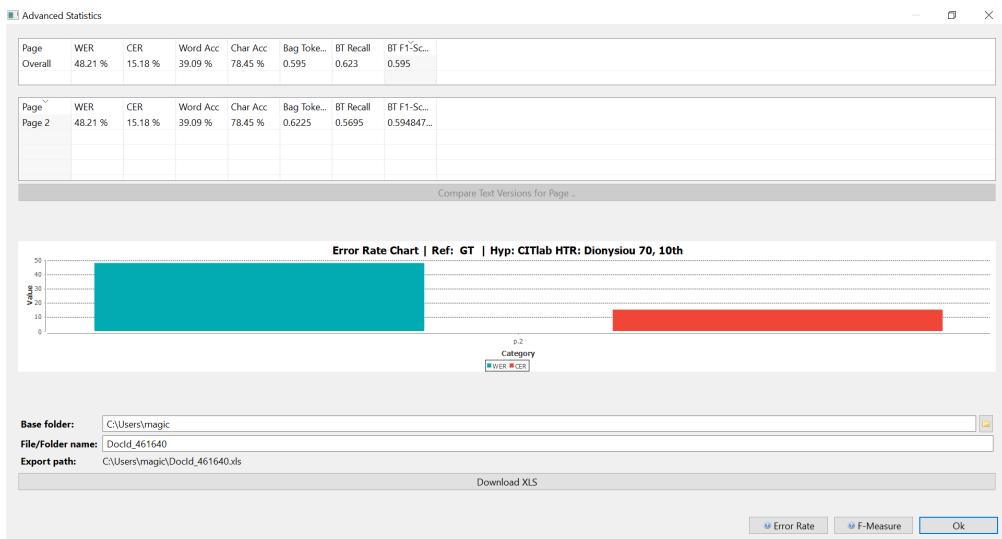


Fig. 12: Visualisation of the HTR validation.

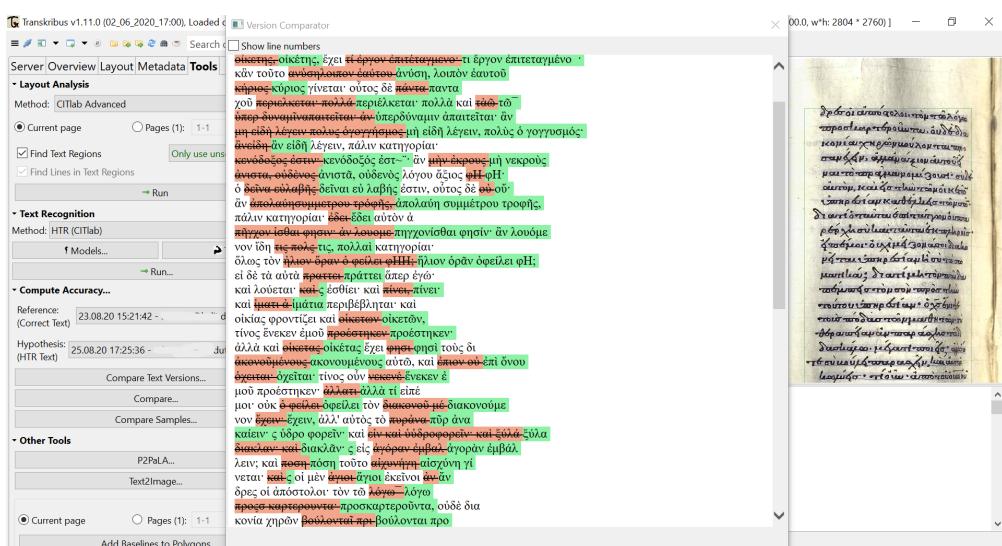


Fig. 13: Text comparison of ground truth and HTR prediction's errors.

22 It becomes obvious then, that the possibilities of the platform are not solely restricted to training HTR models, although this is the most important feature. Transkribus offers a lot of features and tools in the compact form of a simple GUI, some of which are: a) keyword spotting, which returns results both in text and image form as well (see fig. 11), b) TEI export of the completed transcription, c) comparison of text versions, etc. Especially, the text comparison feature proves to be particularly important to the visualisation of HTR accuracy, as it provides detailed statistics and relevant graphics of the CER [Character Error Rate] (see fig. 12). and a ground truth-validation comparison image with colour marking of HTR errors (see fig. 13). All these tools are helpful to evaluate the behaviour and progress of the model and thus decide further steps.

Conclusion-Desiderata

23 In the era of an exponential development of technology and under the real influence of AI in every aspect of the modern world, exploitation of state-of-the-art software/hardware in Humanities comes as a necessity. Accumulated historical knowledge of the past is patiently waiting in archives and collections to be researched. But human experts could not be enough to manually transcribe all that big data of Humanities. With that in mind projects that develop software for HTR, or any other form of ancient text mining are particularly welcome.

24 However real the need for technical assistance in the field of manuscripts' research, scholars do not always prove to be tech-savvy. On the other hand, because software is mostly produced (as expected) by developers-experts in their field, most DH projects are technically sophisticated and cannot thus be mainstream. What is not fully understood or difficult to use, most probably will not be used at all (see for instance the distinction between Windows and Linux users, despite Linux proving to be equally, if not more, efficient). Transkribus differentiation from similar projects is exactly the fact that the platform successfully empowered simple users with the potential of neural networks training. It offered AI in a simple and functional package, which already proves to move forward research in Humanities, as small research groups or independent researchers liberated themselves from expensive or complicated technology.

25 Transkribus capability to be trained in reading any language is of course expected due to the technical architecture of the software (i. e., recognise characters structure and not language semantics), it is admirable nevertheless, especially with non-Latin alphabets. Greek manuscripts, which are the sole core of my experiments with Transkribus, have many peculiarities in style and format. Regardless of the wide chronological range of my data (10th–14th c. A.D.) and experiments with small datasets, Transkribus performance was always formidable. The factor of uniformity that characterises Greek medieval manuscripts was of course helpful to that direction. Even in the case of low-quality manuscript images, automatic layout analysis was hardly erroneous and upon several required transcription data, models would perform in the worst case under 20% of CER. For projects that do not necessitate high accuracy results, or for those with small datasets, that percentage of CER is more than accepted; it is usable.

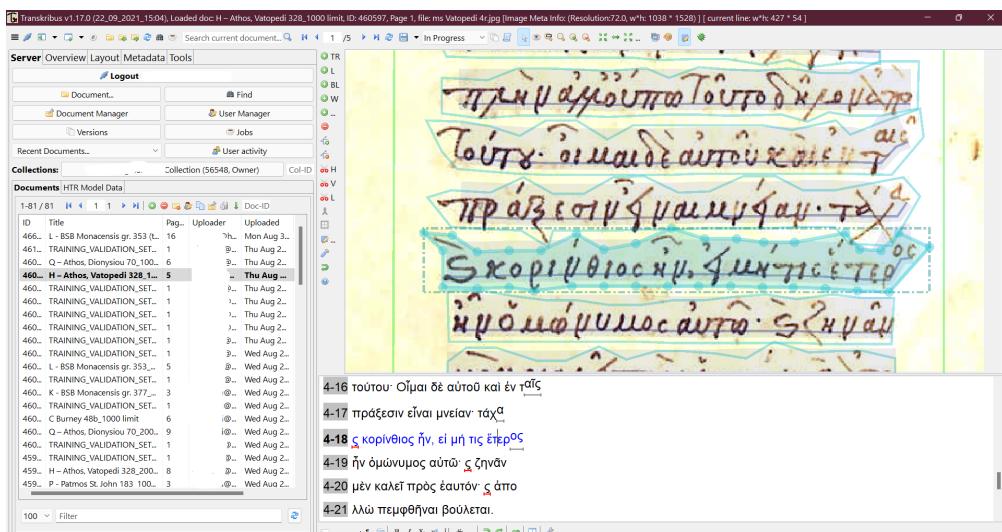


Fig. 14: Ligatures' annotation and recognition.

26 In contrast, the very universality of Transkribus leads to some limitations regarding certain palaeographic peculiarities of Greek manuscripts. One of these aspects is about ligatures and abbreviations. When it comes to defining (and so recognising) ligatures in Greek manuscripts, often it happens to come upon a ligature in which some of the letters are written just above the others, sort of *supra scriptis*. In those cases, despite annotating the characters as *superscript* (an annotation tag offered in Transkribus toolbox), often HTR (indeed not always) fails to understand such combinations of letters – unless the user clearly defines their baseline as separate (see fig. 14). From a point of view, that phenomenon is not a Transkribus' shortcoming per se, but more of a general HTR issue. One should experiment a lot before choosing the best methodology for any project.

27 In the same context, the already existing and extremely useful virtual keyboard of special characters could be further enriched in collaboration with palaeographers. For the moment it is possible to add custom Unicode characters that will represent ligatures or other non-Latin characters of the manuscripts, but Greek ligatures are underrepresented and could be further supplemented. That way one would not have to use incorrect symbols or letters, because in that case there is the risk to confuse the neural network by teaching it to read the same letter in multiple instances: i. e. I was transcribing the ligature representing the word *kai* with a combination of characters, the most visibly similar *ς`* (see fig. 14). Eventually, that choice proved to be successful, but surely it was counterproductive as well. Such minor implementations could be extremely

beneficial for end-users, who prefer a complete-suit software as Transkribus than a CLI package.

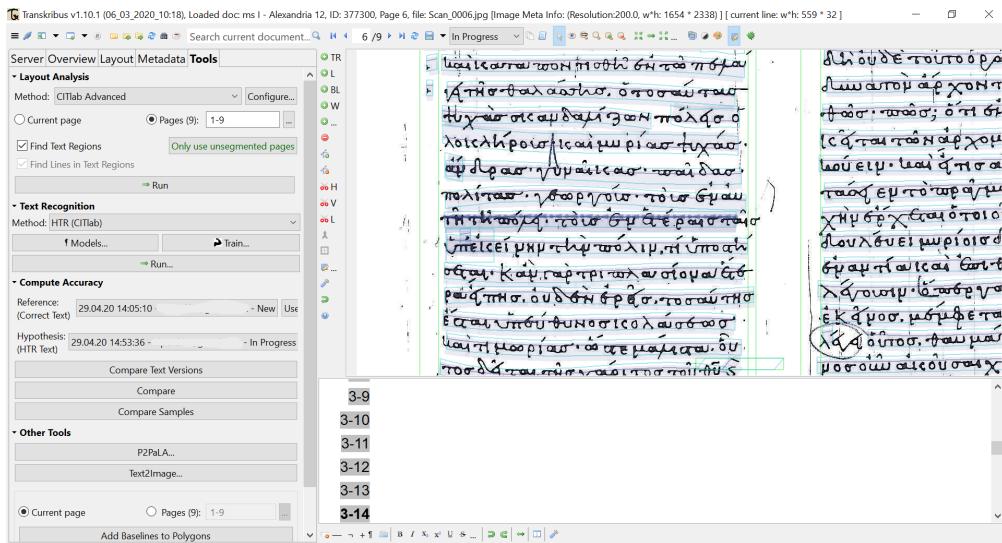


Fig. 15: Baseline vs hanging line misconception.

28 Improvements could also be made in the field of layout analysis, in regard to Greek manuscripts. Segmentation of the image is completed, as stated earlier, by annotating manually or automatically the baseline of a manuscript. Yet there exist instances of Greek medieval manuscripts (mainly from the 14th c. A.D.) that form their text from a hanging line instead of a baseline ([Coulson and Babcock 2020](#)). Most surprisingly, Transkribus' automatic analysis captured that peculiar characteristic, however, because software presupposes that a baseline is needed, line region was misplaced in the void area *supra script* (see [fig. 15](#)). The issue was corrected by manually annotating the layout of the document, but this option would not have been practical for larger collections.

29 Finally, as DH projects do make progress in and exploit the field of AI, we should move on to research the validity of the trained models. It is common practice to evaluate HTR models on the CER percentage of validation sets. Although, due to the complexity of real, raw data, which have not been previously curated by a researcher, evaluation of HTR models probably should not rely only on CER of validation sets, but could also include explanation techniques that may help us decide further steps on the research ([Ribeiro, Singh, and Guestrin 2016](#)). The main questions are: a) how do we know if these metrics are correct or misleading and b) how easy is it to inspect training or validation data when dealing with big manuscript data? Researchers, we, should be able to trust the (trained by us) models.

Notes

1. <https://web.archive.org/web/20211108183341/https://readcoop.eu/transkribus/>
2. <https://web.archive.org/web/20211213175714/https://readcoop.eu/about/>
3. <https://web.archive.org/web/20211213181217/https://github.com/transkribus/> and
<https://web.archive.org/web/20211213184430/https://gitlab.com/readcoop/transkribus> respectively.
4. <http://web.archive.org/web/20211113063459/https://readcoop.eu/transkribus/download/>
5. <http://web.archive.org/web/20220119164148/https://transkribus.eu/lite/>
6. See “Supported Operating Systems” at: <https://web.archive.org/web/20220123181926/https://readcoop.eu/transkribus/howto/how-to-download-install-and-run-transkribus/>.
7. More about pricing options, studentship and credits’ management can be found at: <http://web.archive.org/web/20211118014952/https://readcoop.eu/transkribus/credits/>.
8. More information about the project can be found at: <https://web.archive.org/web/20220102050954/https://escriptorium.fr/>.
9. See for instance “SimpleHTR” project at: <https://web.archive.org/web/20220124180144/https://github.com/githubharald/SimpleHTR>.
10. More information about the project are available at: <https://web.archive.org/web/20220129152558/http://kraken.re/master/index.html>.
11. Tesseract repository can be found at GitHub: <https://web.archive.org/web/20220125061256/https://github.com/tesseract-ocr/tesseract>.
12. As defined in Transkribus’ documentation: <https://web.archive.org/web/20220130213434/https://readcoop.eu/transkribus/howto/use-transkribus-in-10-steps/>.
13. See for instance the multi-language model trained on 6 different languages and centuries data: <https://web.archive.org/web/20220130101732/https://readcoop.eu/model/transkribus-print-multi-language-dutch-german-english-finnish-french-swedish-etc/>, or

the experiments conducted on Arabic manuscripts by Dr Adi Keinan-Schoonbaert, Digital Curator for Asian and African Collections, British Library: <https://web.archive.org/web/20210416133914/https://blogs.bl.uk/digital-scholarship/2020/01/using-transkribus-for-arabic-handwritten-text-recognition.html>.

14. Transkribus has dedicated a page informing for such projects at: <https://web.archive.org/web/20220129182000/https://readcoop.eu/success-stories/>. See also the crowdsourcing UCL's Bentham Project: <https://web.archive.org/web/20220130135024/https://www.ucl.ac.uk/news/2019/aug/transcribing-brunels-illegible-handwriting-using-ai/>.

15. <https://web.archive.org/web/20220130141552/https://readcoop.eu/transkribus/resources/>.

16. <https://web.archive.org/web/20220130144402/https://readcoop.eu/transkribus/docu/rest-api/>.

17. See ibid. "Introduction" and note 3.

18. <https://web.archive.org/web/20211106145137/https://readcoop.eu/terms-and-conditions/>.

19. See ibid note 18.

20. All functions along with relevant screenshots are available at: <https://web.archive.org/web/20220121150815/https://transkribus.eu/lite/>.

References

Burlacu, Constanța, and Achim Rabus. 2021. 'Digitising (Romanian) Cyrillic Using Transkribus: New Perspectives'. *Diacronia* 2021 (14): A196–A196.
<https://doi.org/10.17684/i14A196en>.

Coulson, Frank T., and Babcock, eds. 2020. *The Oxford Handbook of Latin Palaeography*. The Oxford Handbook of Latin Palaeography. Oxford University Press.
<https://doi.org/10.1093/oxfordhb/9780195336948.001.0001>.

Haque, Akm Bahalul, A. K. M. Najmul Islam, Sami Hyrynsalmi, Bilal Naqvi, and Kari Smolander. 2021. ‘GDPR Compliant Blockchains—A Systematic Literature Review’. IEEE Access 9: 50593–606.
<https://doi.org/10.1109/ACCESS.2021.3069877>.

Kahle, Philip, Sebastian Colutto, Gunter Hackl, and Gunter Muhlberger. 2017. ‘Transkribus - A Service Platform for Transcription, Recognition and Retrieval of Historical Documents’. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 19–24. Kyoto: IEEE.
<https://doi.org/10.1109/ICDAR.2017.307>.

Liu, Bo, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. 2021. ‘When Machine Learning Meets Privacy: A Survey and Outlook’. ACM Computing Surveys 54 (2): 31:1-31:36.
<https://doi.org/10.1145/3436755>.

Perdiki, Elpida, and Maria Konstantinidou. 2021. ‘Handling Big Manuscript Data’. Edited by Claire Clivaz and V. Allen Garrick. Classics@ 18 (Ancient Manuscripts and Virtual Research Environments, special issue).
<https://classics-at.chs.harvard.edu/classics18-perdiki-and-konstantinidou/>.

Pletschacher, Stefan, and Apostolos Antonacopoulos. 2010. ‘The PAGE (Page Analysis and Ground-Truth Elements) Format Framework’. In 2010 20th International Conference on Pattern Recognition, 257–60. Istanbul, Turkey: IEEE.
<https://doi.org/10.1109/ICPR.2010.72>.

Rabus, Achim. 2019. ‘Recognizing Handwritten Text in Slavic Manuscripts: A Neural-Network Approach Using Transkribus’. Scripta & E-Scripta 19: 9–32.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier’. ArXiv:1602.04938 [Cs, Stat], August.
<http://arxiv.org/abs/1602.04938>.

Ströbel, Phillip Benjamin, Simon Clematide, and Martin Volk. 2020. ‘How Much Data Do You Need? About the Creation of a Ground Truth for Black Letter and the Effectiveness of Neural OCR’. In Proceedings of the 12th Language Resources and

Evaluation Conference, 3551–59. Marseille, France: European Language Resources Association.

<https://www.aclweb.org/anthology/2020.lrec-1.436>.

Factsheet

Resource reviewed	
Title	Transkribus
Editors	University of Innsbruck and READ-COOP SCE
URI	https://transkribus.eu/
Publication Date	2019
Date of last access	10.12.2021

Reviewer	
Name	Perdiki, Elpida
Affiliation	Democritus University of Thrace
Place	Komotini, Greece
Email	eperdiki (at) helit.duth.gr

General information		
Software type	What type of software is it? (cf. Catalogue 0.1.1)	Virtual Research Environment (VRE)
Identification of the environment	On which platform runs the tool? (cf. Catalogue 1.4)	Operating system
Purpose	For what purpose was the tool developed? (cf. Catalogue 1.5)	developed to accomplish a general task
Funding	Which is the financial model of the tool? (cf. Catalogue 1.6)	Support-based
Maturity	What is the development stage of the tool? (cf. Catalogue 1.5)	Release
Methods and implementation		
Programming Language	Which programming languages and technologies are used? (cf. Catalogue 2.3)	C family, Java, Python, Other: HTML
Reuse	Does the tool reuse portions of other existing software? (cf. Catalogue 2.3)	yes

Input format	Which input formats are supported? (cf. Catalogue 2.4)	.txt, .pdf, Other: .jpg, IIIF
Output format	Which output formats are supported? (cf. Catalogue 2.4)	.xml/tei, .txt, .pdf, Other: .docx
Encoding	Which character encoding formats are supported? (cf. Catalogue 2.4)	latin-1, utf-8, utf-16, Other: MUFI (Medieval Unicode Font Initiative)
Encoding preprocessing	Is a pre-processing conversion included?	yes
Dependencies	Does the documentation list dependencies on other software, libraries or hardware? (cf. Catalogue 3.2)	yes
Dependencies installation	If yes, is the software handling the installation of dependencies during the general installation process (you don't have to install them manually before the installation)?	yes
Documentation and support		
Documentation	Is documentation and/or a manual available? (tool website, wiki, blog, documentation, or tutorial) (cf. Catalogue 3.4)	yes
Documentation format	Which format has the documentation? (cf. Catalogue 3.3)	.html
Documentation parts	Which of the following sections does the documentation contain? (cf. Catalogue 3.3)	'Getting Started' section (installation and configuration), Step-by-step instructions, Examples, Troubleshooting (a selection of possible error messages and related solutions), FAQ, Support
Documentation language	In what languages is the documentation available? (cf. Catalogue 3.3)	English, German, Italian
Support	Is there a method to get active support from the developer(s) or from the community? (cf. Catalogue 3.4)	yes
From of support	Which form of support is offered? (cf. Catalogue 3.4)	Help desk, Forum, Mailing-list
Issue tracker	Is it possible to post bugs or issue using issue tracker mechanisms?	yes

	(cf. Catalogue 3.4)	
Usability and sustainability		
Build and install	Grade how straightforward it is to build or install the tool on a supported platform: (cf. Catalogue 3.6)	straightforward
Tests	Is there a test suite, covering the core functionality in order to check that the tool has been correctly built or installed? (cf. Catalogue 3.7)	yes
Portability and interoperability	On which platforms can the tool/software be deployed? (cf. Catalogue 3.8)	Linux/BSD/Unix, Mac OS X, Windows, Not applicable (if web-based for example)
Devices	On which devices can the tool/software be deployed? (cf. Catalogue 3.8)	Desktop, Laptop, Not applicable (if web-based for example)
Browsers	If the tool is web-based: On which browsers can the tool/software be deployed? (cf. Catalogue 3.8)	Mozilla Firefox, Google Chrome, Safari
Plugins	If the tool is web-based: Does the tool rely on browser plugins? (cf. Catalogue 3.8)	yes
API	Is there an API for the tool? (cf. Catalogue 3.8)	yes
Code	Is the source code open? (cf. Catalogue 3.9)	yes
License	Under what license is the tool released? (cf. Catalogue 3.9)	GNU/GPL
Credits	Does the software make adequate acknowledgement and credit to the project contributors? (cf. Catalogue 3.9)	yes
Registered	Is the tool/software registered in a software repository? (cf. Catalogue 3.9)	yes
Possible contribution	If yes, can you contribute to the software development via the repository/development platform?	yes
Analysability, extensibility, reusability of the code		
Analysability		yes

	Can the code be analyzed easily (is it structured, commented, following standards)? (cf. Catalogue 3.10)	
Extensibility	Can the code be extended easily (because there are contribution mechanisms, attribution for changes and backward compatibility)? (cf. Catalogue 3.10)	yes
Reusability	Can the code be reused easily in other contexts (because there are appropriate interfaces and/or a modular architecture)? (cf. Catalogue 3.10)	yes
Security and privacy	Does the software provide sufficient information about the treatment of the data entered by the users? (cf. Catalogue 3.11)	yes
Supportability and maintenance	Is there information available whether the tool will be supported currently and in the future? (cf. Catalogue 3.12)	yes
Citability	Does the tool supply citation guidelines (e.g. using the Citation File Format)? (cf. Catalogue 3.13)	yes
User interaction, GUI and visualization		
User profile	What kind of users are expected? (cf. Catalogue 4.1)	Humanities researcher, Digital humanist, General public
User interaction	What kind of user interactions are expected? (cf. Catalogue 4.1)	Reading, Text editing, Text analysis, Searching
User Interface	What kind of interface does the tool provide? (cf. Catalogue 4.2 and 0.1.1)	Graphical User Interface (GUI)
Visualization	Does the tool provide a particular visualizations (in terms of analysis) of the input and/or the output data? (cf. Catalogue 4.3)	yes
User empowerment	Is the user allowed to customize the functioning of the tool and the output configuration? (cf. Catalogue 4.4)	yes
Accessibility	Does the tool provide particular features for improving accessibility,	yes

allowing „people with the widest range of characteristics and capabilities” to use it?
(cf. Catalogue 4.5)