

A Web-Based Interactive Transcription Tool for Encrypted Manuscripts

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés

Computer Vision Center
Computer Science Department
Universitat Autònoma de Barcelona
{jchen,msouibgui,afornes}@cvc.uab.es

Beáta Megyesi

Dept. of Linguistics and Philology
Uppsala University, Sweden
beata.megyesi@lingfil.uu.se

Abstract

Manual transcription of handwritten text is a time consuming task. In the case of encrypted manuscripts, the recognition is even more complex due to the huge variety of alphabets and symbol sets. To speed up and ease this process, we present a web-based tool aimed to (semi)-automatically transcribe the encrypted sources. The user uploads one or several images of the desired encrypted document(s) as input, and the system returns the transcription(s). This process is carried out in an interactive fashion with the user to obtain more accurate results. For discovering and testing, the developed web tool is freely available ¹.

1 Introduction

Nowadays, artificial intelligence and pattern recognition are playing an important role in historical manuscript processing and recognition. Some research projects with focus on digital paleography, including the transcription of historical manuscripts are, for example, HIMANIS (Stutzmann et al., 2017), Transkribus (Kahle et al., 2017), and *From Quill to Bytes* (q2b, 2013). For the case of encrypted historical manuscripts analysis, which constitute the main subject of this paper, the project DECRYPT (Megyesi et al., 2020) is joining the expertise in computer vision, computational linguistics, philology, cryptanalysis and history for the aim of making advances in historical cryptology.

The first step toward decrypting a handwritten ciphertext is transcription. Intuitively speaking, the transcription could be done manually

but it turns out to be a time-consuming, error-prone, and expensive task (Piotrowski, 2012). During the last decade, several handwritten text recognition (HTR) methods have been developed and applied successfully to historical handwritten sources, allowing (semi-)automatic transcription (Kahle et al., 2017; Romero et al., 2017). Alternative approaches use word spotting (Santoro et al., 2017), speech recognition (Granell et al., 2018) or even gamification (Chen et al., 2018) for speeding up the manual transcription. However, all these tools have been developed to only deal with known scripts (e.g. Roman alphabet). Indeed, the transcription of encrypted sources is more complicated as they often include symbols that are taken from a wide range of alphabets and symbol sets. For a more generic and flexible transcription within and across ciphers, the use of generic annotation tools such as Alethea (Clausner et al., 2011) or Pixlabeler (Saund et al., 2009) could be preferable. But, the annotation process through these tools is fully manual, leading to a huge cost in term of time especially for encrypted manuscripts with unknown symbol sets. Therefore, semi-automatic image processing tools would be the suitable solution to this kind of applications.

In this paper, we present a tool for transcription of encrypted sources consisting of various symbols sets. The tool processes document images (e.g. scanned images of manuscripts) and outputs the corresponding transcription. The system interacts with the user at certain steps for a more accurate transcription (in a semi-automatic fashion). Users could be paleographers, cryptologists, archive workers, etc. We start by briefly describing previous efforts on (semi-)automatic transcription of ciphers, and then present our interactive tool.

¹<https://cl.lingfil.uu.se/decode/transcription/>

2 Automatic Transcription of Encrypted Sources

The main challenge in HTR is to locate and segment the actual text parts into paragraphs, lines, and individual symbols (glyphs). In addition, the system shall identify the various allographs (variants) of each symbol type (graphem). The system shall also be able to determine the various elements of a graphem, such as dots and commas, and leave out unintentional ink spots, bleed-through, or marks from a damaged paper or parchment. In a fully automatic system, computers handle the entire process in one step, while in a semi-automatic system the user can interact with the system to improve the result during the transcription or as a post-processing step to correct the output of an automatic process.

Experiments on automatic transcription by image processing have been performed on numeric cipher sequences (Fornés et al., 2017) and a wide range of glyphs belonging to alchemic and Zodiac signs, digits, and Roman and Greek letters (Baró et al., 2019). Preliminary results show that image processing can be used as base for transcription followed by a post-processing step with user validation and correction. Even though image processing techniques need to be trained on individual hand-writings to reach high(er) accuracy, unsupervised techniques (i.e. no labelled data is required to train) can also be used for speeding up the transcription. In addition, they might be of great help to identify the symbol set represented in the manuscript and to make clear distinctions between symbols, hence can be used as a support tool for the transcriber.

3 Interactive Transcription Tool

Our interactive transcription tool is generic in the sense that it should be applicable to any symbol sets, and it does not need any labelled data to train the image processing algorithms. The tool consists of three main steps, as illustrated in Figure 1. First, the input cipher images are segmented into lines and symbols. Then these symbols are clustered (grouped) according to their shape similarity. Finally, the transcription is performed, obtaining the final transcribed cipher-

text. Executing these stages in an automatic way leads to the transcription of a given cipher image. But, since the efficacy of each step highly depends on the correctness of the previous step output, it is preferable to use the tool in a semi-automatic way. In other words, if the user intervenes in each stage to validate or correct the intermediate results, then more accurate transcription can be obtained. In what follows, a detailed description of those steps is provided.

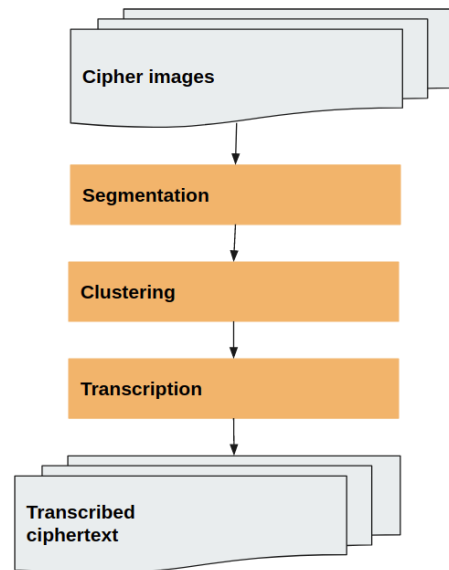


Figure 1: The architecture of the Interactive Transcription Tool.

3.1 Image Upload

First, the user uploads the image(s) into the tool. The system accepts PNG, JPEG or TIFF image file formats. Since the transcription accuracy depends on the images quality, we recommend to use colored images of high resolution (e.g. 300-600 dpi) as stated in (van Dormolen, 2019). This is recommended as well in ISO/TS 19264-1:2017 technical specification for cultural heritage imaging, even though the tool accepts low resolution images as well. It is to note that the image processing algorithms are based on the analysis of the symbols shapes. Thus, the document images should be selected from the same manuscript with the same symbol set and handwriting style to obtain a more reliable transcription. In this stage, the system creates a first JSON-file, it will be used to store all the intermediate results that will be obtained during the

different stages. This file will be sent to the user after each subsequent step of the transcription process.

3.2 Segmentation

The first step of our unsupervised transcription pipeline consists of segmenting the document image(s) into isolated symbols by creating bounding boxes for each symbol to be transcribed. Although the user can manually segment all symbols using our tool, it is a time consuming task. Hence, the optimal choice is to request an automatic segmentation and manually validate the results. The segmentation method consists of applying horizontal projections to detect the text lines, connected components to segment the symbols, and grouping to obtain the final bounding boxes of each symbol. An example of the automatic segmentation obtained can be seen in Figure 2.

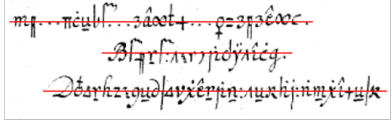
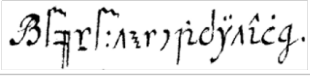
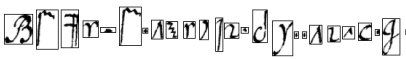
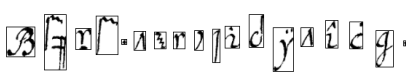
Horizontal Projection	
Segmented Line	
Segmented Symbols	
Symbol Grouping	

Figure 2: The stages to segment a cipher document into isolated symbols by the tool.

Although the segmentation algorithm can run using the default options, our interface provides some advanced options as illustrated in Figure 3, which are very useful for trained and experienced users when applying the automatic segmentation. These advanced options are:

- Symbol size: Big/Small. This value is used to inform on the size of the symbols with respect to the page. For example, the Copiale cipher (Knight et al., 2011) contains small symbols regarding the pages, whereas the Borg cipher (Aldarrab et al., 2017) contains big symbols in the pages.

- Binarize image: Yes/No. The user can chose whether to binarize the image or not. Because our current method works only on binary images, the user will receive an error if it is set "No". This option is added to guarantee scalability, since we are planning to add other segmentation methods to work on colored images as well.
- Minimum line distance: A number (in pixels) indicates the minimum distance between lines. Example: In the Copiale cipher, most lines have 120 pixels of separation.
- Lines threshold: it is a decimal/float number between 0 and 1. This value is used to state that only those lines with an amplitude higher than this threshold will be detected (this acts as a line filter).
- Max. distance symbols: This number (in pixels) indicates the maximum distance between symbols. This parameter is useful when grouping symbols that contain diacritics, super- och subscripts (e.g. dots or accents like á or ÿ). When the segmentation is based on connected components, these small elements are separated. For this reason, the system tends to group nearby symbols, i.e. symbols that are closer to the given threshold distance.
- Min. symbol size: This number (in pixels) indicates the minimum symbol size that could be found in the manuscript. This is used to filter components that are smaller than this size, which usually corresponds to background noise in historical manuscripts.

When the segmentation process ends, the user will receive (in their indicated email) a JSON file containing the results of the segmentation step. To visualize these results, the user should upload the JSON file and the cipher image to the web tool. Figure 4 shows an example of the output from the segmentation part.

Although the user can apply the segmentation algorithm using different setups (i.e. different values in the advanced options interface), it is difficult to obtain a perfect segmentation with an unsupervised segmentation method. The

Figure 3: The interface for the segmentation request, showing the advanced options.

main reason is that the segmentation algorithm is generic, so it has no information on the type of symbol set used in the encrypted source. Moreover, most encrypted manuscripts use a cursive writing, so touching and overlapping symbols are frequent, which make the segmentation even harder. In this stage, the user interaction is highly recommended, so that the clustering stage can be more efficient and less error-prone. Therefore, the tool allows the user to verify and manually correct any segmentation errors. Figure 5 shows and example of correcting a wrong segmentation. It is to note that the users cannot only delete or modify the bounding boxes, but they can also create new ones for any symbol missed by the automatic segmentation.

3.3 Clustering

Once the user obtain the set of isolated symbols (assumed to be correctly segmented), they can proceed to the clustering. Clustering means grouping visually similar symbols in different sets, called clusters. Our tool applies the hierarchical K-Means algorithm for clustering (Arai and Barakbah, 2007). As advanced setting, the

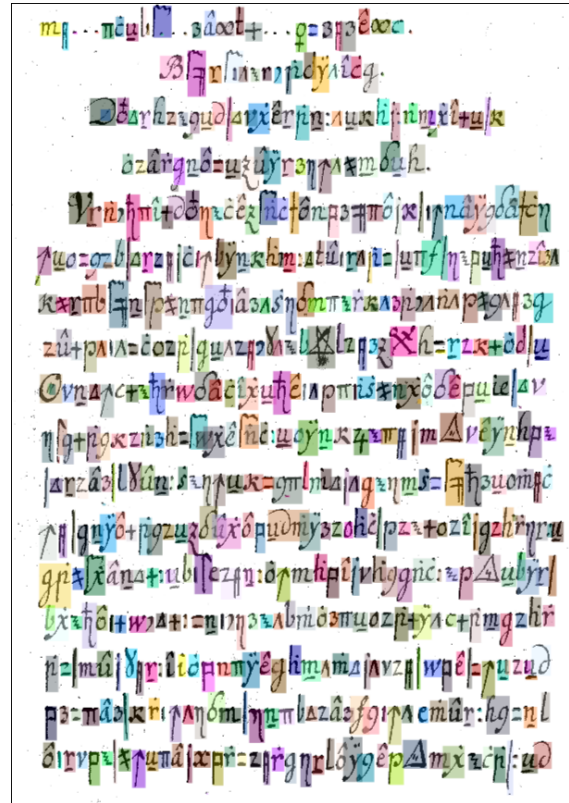


Figure 4: Visualization of the bounding boxes after the segmentation step.

user can define the minimum number of symbols that could be assigned to one cluster, called the Min. cluster images. The K-means algorithm starts by assuming that all the symbols are belonging to a single cluster, then, splitting it recursively until the clusters are no more divisible or when reaching the minimum amount of images per cluster. Figure 6 shows the clustering request interface.

Similar to the segmentation step, the user will receive the results of the clustering via e-mail. The user can visualize the clusters by uploading the received JSON file as shown in Figure 7. The tool bar on the right hand side called "Clusters" shows all the clusters provided by the K-means. The user can press the 'eye' icon to visualize the symbols belonging to each cluster. Figure 8 illustrates the symbols (instances) within a specific cluster.

In the ideal case, each cluster should contain instances from the same symbol. However, there is a high degree of visual similarity between the different symbols in many encrypted sources. As a result, some clusters can contain instances

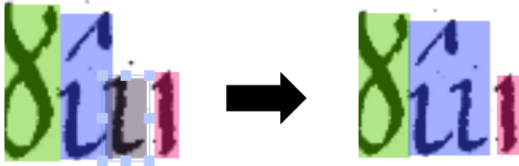


Figure 5: An example of correcting an over-segmented symbol. The grey bounding box must be merged to the previous symbol marked in blue.

Figure 6: Clustering request, showing the advanced options.

from different, although similar symbols. Thus, our tool allows the user to correct errors in the clusters. The user can clean a cluster by removing those symbols that do not belong to that cluster. An illustrative example can be seen in Figure 9.

After cleaning the clusters, the removed symbols remain unlabelled, i.e. not assigned to any cluster. The tool also allows the user to create new clusters, assign symbols to clusters, and

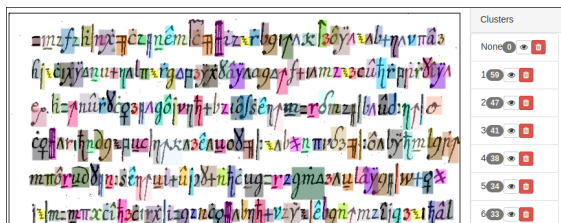


Figure 7: On the right, the system shows the clusters (i.e. group of symbols) obtained by the K-Means algorithm.

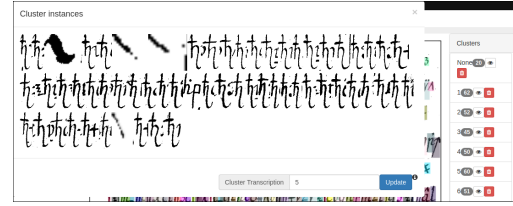


Figure 8: Example of one cluster after the label propagation step.

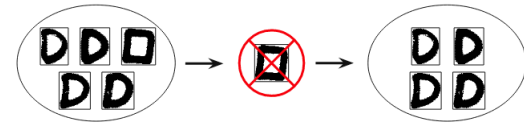


Figure 9: An example of cleaning a cluster: the user removes the symbol that does not belong to this cluster.

change the obtained clusters for the symbols. Cleaning the clusters facilitates the subsequent label propagation step, where symbols will be assigned to the most similar cluster.

3.4 Transcription

After the clustering step, the user can request the actual transcription where a label is assigned to each symbol according to the label of cluster the symbol belongs to. We call this process label propagation. The objective is to propagate the label of the clusters to the unlabeled symbols. The setup of the label propagation request has two options as illustrated in Figure 10:

- **Seeds number:** The number of the most populated clusters that will be used as seeds to propagate labels. This number should be at least equal to the alphabet size (if it is known). After setting the seeds number, the user can visualize the selected clusters in the cluster bar tool. The default value of seed numbers is 10 due to many ciphertext containing digits only (0-9).
- **Change class threshold:** A value between 0 and 1 determines how easy is to propagate a label through the instances. If the value is close to 0, the propagation will be more stable (less changeability), but it can lead to poor results when the user is transcribing few pages. Contrary, if the value is close to 1, it will make the propagation

Figure 10: Label propagation request, showing the advanced options.

unstable (high changeability) which leads sometimes to propagation of wrong labels.

The label propagation determines the final clusters and assigns the labels. The output is the sets of instances in each cluster, as shown in Figure 8.

At this moment, the only user intervention consists in assigning the desired transcription label to each cluster as shown in Figure 11. All the symbols in the cluster will be transcribed with the label assigned to the particular cluster. Note, however, that each symbol has a value between 0 and 1, representing the degree of belonging to this specific cluster. This means that if a symbol has a low value, the system is not confident in labelling the correct transcription. Therefore, the recommendation is to manually transcribe symbols with a low value to increase transcription correctness.

There is a trade-off between transcription correctness (precision) and transcription completeness (recall). As illustrated in Figure 12, a low transcription confidence threshold leads to more complete transcriptions. On the other hand, this leads to a higher possibility of errors. Contrary, a high confidence threshold means that only symbols with a high confidence value will be transcribed, whereas the rest will lack correct transcription. These non-transcribed symbols ap-

pear as "NONE" (or '*'') in the transcription file, and the user shall dedicate more time to manually transcribe those symbols. In order to make a fewer intervention with higher accuracy, we tried to balance this by choosing the threshold confidence to be 0,5. As the final step, the user can download the obtained transcription using the download request with various types of output formats (e.g. text, XML, JSON), see Figure 13.

Figure 13: The downloading interface, where the user can select different kind of output files.

4 Conclusion

We presented a tool serving as an aid for faster and more accurate transcription of encrypted sources with various cipher text alphabets. The transcription system segments the lines and then suggests the segmentation of each individual symbol, which could be corrected by the user. Then, the segmented symbols are clustered into groups on the basis of similarity measures and the symbols in the same cluster receive the same transcription. The user can edit the suggestions given by the system in each step, correct the output, and upload a new, improved versions for further processing.

To the best of our knowledge, there is no similar tool that allows for the (semi)-automatic transcription of manuscripts with various alphabets and scripts. We hope that the ITT tool will be useful for the transcription of the historical and encrypted sources. The tool is under development and we plan to add more image processing techniques in the different transcription steps to

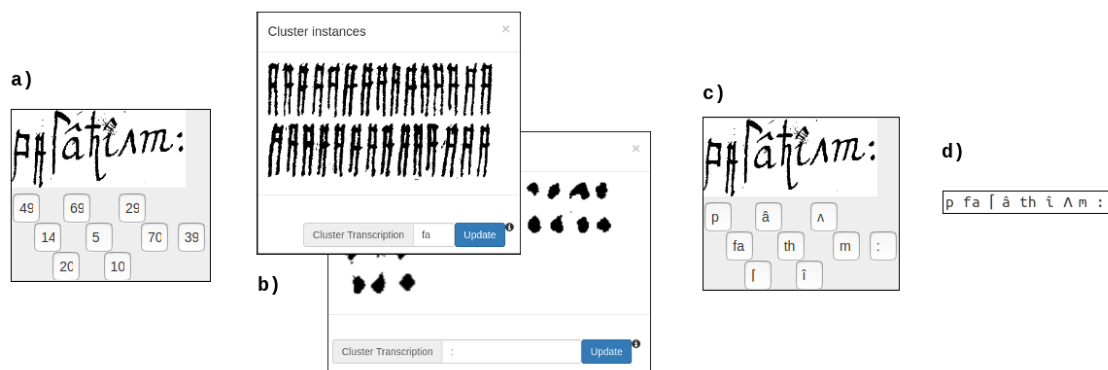


Figure 11: Transcription step. a) Line transcription using default cluster labels (numbers). b) The user changes the cluster labels to the desired transcription. c) Line transcription using the desired transcription. d) A text file with the line transcription.

Reconstructed line	
Ground-truth	q w 1 w v x d 1
Our method Assigned code	Thr 0.4 q w 1 w * x d 1
	Thr 0.6 q w * w * x d 1
	Thr 0.8 q w * w * * * 1

Figure 12: In the transcription phase, by changing the transcription threshold, the symbols with lower confidence than the given threshold will be transcribed as '*'. *

enhance the accuracy and reduce the user intervention.

Acknowledgments

This work has been partially supported by the Swedish Research Council, grant 2018-06074: *DECRYPT - Decryption of historical manuscripts*, the Spanish project RTI2018-095645-B-C21, the Ramon y Cajal Fellowship RYC-2014-16831 and the CERCA Program / Generalitat de Catalunya.

References

Nada Aldarrab, Kevin Knight, and Beáta Megyesi. 2017. The Borg Cipher. <https://cl.lingfil.uu.se/ bea/borg>. Accessed: 2020-01-31.

Kohei Arai and Ali Ridho Barakbah. 2007. Hierarchical K-means: An Algorithm for Centroids Initialization for K-means. *Reports of the Faculty of*

Science and Engineering, Saga University, 36:25–31.

Arnau Baró, Jialuo Chen, Alicia Fornés, and Beáta Megyesi. 2019. Towards a Generic Unsupervised Method for Transcription of Encoded Manuscripts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (DATECH)*, pages 73–78.

Jialuo Chen, Pau Riba, Alicia Fornés, Joan Mas, Josep Lladós, and Joana Maria Pujadas-Mora. 2018. Word-Hunter: A Gamesourcing Experience to Validate the Transcription of Historical Manuscripts. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 528–533. IEEE.

Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. 2011. Aletheia – An Advanced Document Layout and Text Ground-Truthing System for Production Environments. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 48–52. IEEE.

Alicia Fornés, Beáta Megyesi, and Joan Mas. 2017. Transcription of Encoded Manuscripts with Image Processing Techniques. In *Digital Humanities*.

Emilio Granell, Verónica Romero, and Carlos D. Martínez-Hinarejos. 2018. Multimodality, Interactivity, and Crowdsourcing for Document Transcription. *Computational Intelligence*.

Philip Kahle, Sebastian Colutto, Gunter Hackl, and Gunter Muhlberger. 2017. Transkribus – A Service Platform for Transcription, Recognition and Retrieval of Historical Documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 19–24.

Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. The Copiale Cipher. In *Invited talk at*

ACL Workshop on Building and Using Comparable Corpora (BUCC). Association for Computational Linguistics.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of Historical Manuscripts: The DECRYPT Project. *Cryptologia*, 0(0):1–15.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Morgan Claypool Publishers.

q2b. 2013. q2b – From Quill to Bytes. <https://www.it.uu.se/research/project/q2b?lang=sv>. Accessed: 2020-04-21.

Verónica Romero, Vicente Bosch, Celio Hernández-Tornero, Enrique Vidal, and Joan Andreu Sánchez. 2017. A Historical Document Handwriting Transcription End-to-end System. In *8th Iberian Conference on Pattern Recognition and Image Analysis*, pages 149–157. Springer International Publishing.

Adolfo Santoro, Claudio De Stefano, and Angelo Marcelli. 2017. Assisted Transcription of Historical Documents by Keyword Spotting: A Performance Model. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 971–976.

Eric Saund, Jing Lin, and Prateek Sarkar. 2009. Pixlabeler: User Interface for Pixel-Level Labeling of Elements in Document Images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 646–650. IEEE.

Dominique Stutzmann, Jean-François Moufflet, and Sbastien Hamel. 2017. La Recherche en Texte dans les Sources Manuscrites Mdiévalles : Enjeux et Perspectives du Projet HIMANIS pour L’édition Électronique. *Médiévalles*, 73:67–96.

Hans van Dormolen. 2019. Metamorfoze Preservation Imaging Guidelines, version 2.0. In *Archiving Conference*, pages 9–11.