Transcription of Historical Encrypted Manuscripts

*Evaluation of an automatic interactive transcription tool.*

Kajsa Johansson

Uppsala University
Department of Linguistics and Philology
Language Technology Programme
Bachelor’s Thesis in Language Technology, 15 ECTS credits

June 12, 2019

Supervisors:
Beáta Megyesi, Uppsala University
Abstract

Countless of historical sources are saved in national libraries and archives all over the world and contain important information about our history. Some of these sources are encrypted to prevent people from reading it. This thesis examines a semi-automated Interactive Transcription Tool based on unsupervised learning without any labelled training data that has been developed for transcription of encrypted sources and compares it to manual transcription. The interactive transcription tool is based on handwritten text recognition techniques and the system identifies cluster of symbols based on similarity measures. The tool is evaluated on ciphers with number sequences that have previously been transcribed manually to compare how well the transcription tool performs. The weaknesses of the tool are described and suggestions on how the tool can be improved are proposed. Transcription based on HTR techniques and clustering shows promising results and the unsupervised method based on clustering should be further investigated on ciphers with various symbol sets.
# Contents

Preface ............................................. 5

1 Introduction .................................... 6
   1.1 Purpose and Research Questions .......... 7
   1.2 Thesis Outline .............................. 8

2 Background ...................................... 9
   2.1 Image Processing .......................... 9
   2.2 Transcription of Encrypted Historical Sources .... 10
      2.2.1 Ciphers ............................... 10
      2.2.2 Transcription of Ciphers ............. 11

3 Resources and Tools ............................ 13
   3.1 Data ....................................... 13
      3.1.1 Training .............................. 13
      3.1.2 Testing .............................. 15
      3.1.3 Gold Standard ....................... 16
   3.2 Interactive Transcription Tool ............ 17
      3.2.1 Upload Image ......................... 17
      3.2.2 Segmentation ......................... 18
      3.2.3 Clustering .......................... 20
      3.2.4 Classification ....................... 21
      3.2.5 Transcription ....................... 22

4 Transcribing Ciphers ........................... 24
   4.1 Training ................................... 24
      4.1.1 Discoveries During the Training Phase ... 25
   4.2 Testing .................................... 25
   4.3 Evaluation .................................. 26

5 Results .......................................... 27
   5.1 Result Experiments ......................... 27
   5.2 Using the ITT .............................. 32
      5.2.1 Segmentation ......................... 32
      5.2.2 Clustering .......................... 33
      5.2.3 Classification ....................... 34

6 Discussion and Suggestions for Improvements .... 36
   6.1 Discussion ................................ 36
      6.1.1 Default Setting ...................... 36
      6.1.2 M10 Setting .......................... 37
      6.1.3 M20 Setting .......................... 37
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2 Suggestions for Improvements</td>
<td>38</td>
</tr>
<tr>
<td>7 Conclusion and Future Directions</td>
<td>40</td>
</tr>
<tr>
<td>8 References</td>
<td>42</td>
</tr>
</tbody>
</table>
Preface

I would first like to thank Beáta Megyesi for being my supervisor and for letting me be a part of this fascinating project. Thank you for giving me guidance when writing this thesis and for sharing all your knowledge about decrypting historical manuscripts.

I would like to thank the group behind this entire project, the DECODE group. I would especially want to give a big thanks to Jialuo Chen who has been patient and answered all my questions about the ITT.

Lastly, I would like to thank everyone that has encouraged me to finish this degree, you have all been a big support during this time. Thank you to those who have proofread and showed interest in what I have been doing. I am so thankful for your help and encouragement.
1 Introduction

Countless of historical sources are saved in national libraries and archives all over the world and contain important information about our history. These manuscripts give us a better understanding of what happened through the centuries and expose knowledge that has disappeared with earlier generations (Dooley, 2018).

Some of these manuscripts contain information so important that they were encrypted to hide the content from others except the intended receiver. Only the person(s) that were meant to be reading the source could understand it by decrypting it using a key. Religious writings, political messages between regents, and illegal opinions or movements are examples of texts that were not intended for everyone to read. The encrypted manuscripts are called ciphers or codes, and most of these texts are still secrets to us centuries later. They are made up by different glyphs (a symbol) in symbol systems, some might be Roman or Greek letters, zodiac signs or made up squiggles to name a few. The underlying meaning of the glyphs are unknown unless there is a mapping key explaining the encoding. The mapping key reveals what each glyph represents in the original language, something that the writer could have created and sent to the receiver beforehand. However, in most cases a mapping key is missing, so the message must be decrypted by reconstructing the key (Gobble, 2018).

It is a complicated process to reach the goal of understanding these encrypted manuscripts. In order to do so, the manuscripts need to be digitized, i.e. transformed into a computer readable format, and transcribed before being decrypted. It is not enough to have the manuscripts in a computer readable format, it must be possible to extract information from it as well. The concept of making it possible to search and extract information from that format is known as digitalization. It is easy to scan a page from a historical document into a computer, but it is much more complicated to make the historical text searchable (Gobble, 2018). It is a challenge to convert these historical texts into digital form for large-scale analysis, but a necessary step to preserve our cultural heritage. What is described here is part of digital philology, the art of transforming oral and written historical sources into digital form.

One way of digitizing manuscripts is manual transcription, where a person types in each symbol in a sequence as represented in the manuscript image. The symbols are transcribed following a transcription scheme where each symbol type is represented in some character representation format, often in Unicode or ASCII. Manual transcription is time-consuming and expensive but has for long been the most reliable solution (Piotrowski, 2012). However, in recent years handwritten text recognition (HTR) methods in image processing have been developed and applied successfully to historical handwritten sources, allowing automatic
transcription. The main challenge is to make the computer understand where the actual text, lines, and glyphs are located on the page and what characters the glyphs represent. It must also determine what counts as a character, and what is only unintentional ink spots or bleed-through. Semi- or fully automatic recognition and transcription of glyphs or words can be an alternative to manual transcription. In a fully automatic system, the computers handle the whole process by itself whilst in a semi-automatic system the user can interact with it to improve the result, either during the transcription or as a post-processing step to correct the output of the automatic transcription (Piotrowski, 2012).

Ciphers are usually meticulously written, and the characters are often clearly segmented to avoid any kind of ambiguity to make decryption by the receiver easier. The extra challenge with encrypted historical manuscripts is that ciphers consist of a sequence of symbols that are not understandable until it is decrypted. The cipher symbols can come from various alphabets which computers do not necessarily recognise by standard HTR techniques. The system cannot rely on word spotting or regular language models as these are not frequently occurring in the encrypted sources. Therefore, besides page and line segmentation, the challenge is to segment and identify the glyphs in the cipher and suggest appropriate transcription of the symbol sequence. The transcribed document can then be analysed in order to decrypt its content by guessing the underlying language and applying cryptanalysis (Dooley, 2018).

1.1 Purpose and Research Questions

The aim of the thesis is to apply HTR techniques to encrypted historical sources, more specifically, to examine how well a semi-automated transcription tool (ITT) developed for the transcription of a subset of encrypted sources performs compared to manual transcription. The transcription tool is an interactive tool where the system identifies cluster of symbols based on similarity measures and the user can edit the system’s suggestions of clusters integrated into a graphical user interface as a basis for transcription. The ITT is under development and the aim of the work is to evaluate the tool and make suggestions for improvements and changes to increase its usability.

The following questions are researched:

• Can HTR techniques based on clustering be used for semi-automatic transcription of images containing ciphers with a numerical symbol set?

• How well does the ITT perform on numerical ciphers compared to manual transcription?

• How can the ITT be improved to fasten the transcription process?

The work is part of the DECRYPT project, aiming at the collection, description and automatic decryption of historical encrypted sources (http://cl.lingfil.uu.se/decode).
1.2 Thesis Outline

The outline of the thesis is as follows:

Chapter 2 introduces the background to transcription of ciphers and gives a short overview of Handwritten Text Recognition and Optical Character Recognition.

Chapter 3 presents the Interactive Transcription Tool that the thesis is based on. The ciphers that have been used for the training and evaluation are displayed.

Chapter 4 explains how the evaluation of the interactive transcription tool was carried out.

In Chapter 5 the results from the experiments are presented.

Chapter 6 contains a discussion about the results and presents suggestions for improvements for the Interactive Transcription Tool.

Chapter 7 ends the paper with a conclusion of the experiments and gives directions for future research on automated transcriptions of encrypted sources.
2 Background

In order to process historical handwritten manuscripts, they must be digitalized for further computer-aided processing. All manuscripts originally written on parchment or paper are photographed or scanned to create a digital version to work with. At this stage the computer only sees it as an image, it does not understand that the image contains written text. The uploaded image must then be processed either manually or (semi-)automatically by transcribing the document to turn into a computer-readable text format. Manual transcription is time-consuming and error-prone, and above all expensive. Automatic processing aiding the transcription is preferable which typically involves the detection of pages and lines, and the location and identification of the actual symbols written in the source, so-called glyphs (Berg-Kirkpatrick et al., 2013). In the following section, we give an overview of how image processing can be applied to historical manuscripts for transcription.

2.1 Image Processing

One solution to extract text from images is optical character recognition (OCR). OCR is an automatic method applied to the transcription of printed text. It scans each character and compares it with a large collection of documents with different types of computer fonts. It relies on finding the correct match with the help of dictionaries in uncertain cases. If the OCR cannot separate two glyphs it uses the dictionaries to look up which character is the most likely depending on the neighbouring characters within the same word boundary. This solution relies on natural language processing resources in terms of language models and works well for printed, computer-typed documents, but less applicable to handwritten texts (Piotrowski, 2012). OCR techniques can be assumed to even be more error-prone when applied to enciphered manuscripts with various symbol sets. A large amount of datasets is needed in order to achieve reasonable accuracy (Yin et al., 2019).

Handwritten text recognition (HTR) is divided into two different fields, on-line handwriting and off-line handwriting (Plamondon, 2000). On-line handwriting is made on a touchscreen with a special pen. The advantage is that the output is in a computer format from the beginning. Off-line handwriting, on the other hand, represents handwritten text on a piece of papyrus, parchment, vellum or paper, that we find in historical manuscripts. HTR tools are often developed to be applied to off-line handwriting. HTR can be used for dating manuscripts, scribe recognition, word identification or symbol detection. HTR techniques analyse various patterns in the handwriting from the detection of the text areas to lines, words and symbols. Automatic recognition of handwritten texts is a complex problem since people’s handwriting is unique and varies even within the same
type of writing style, and by the same person (Plamondon, 2000).

Some HTR methods are based on word spotting whilst others are based on symbol segmentation. The challenge for the system is to recognize what is just a blob of ink and what actually is meaningful text (Firmani et al., 2018). One of the main challenges for HTR techniques based on symbol segmentation is connected symbols since the system does not know where to separate them (Piotrowski, 2012). In other cases, parts belonging to the same symbol should be recognized by the system as one symbol, e.g. the dot over i shall be recognized as belonging to the bar below. Another problem is when symbols are dropping down to the line below or reaching the line above, which makes it hard for the HTR system to decide which line the symbol belongs to (Firmani et al., 2018).

HTR methods nowadays are based on advanced deep learning methods. There are supervised and unsupervised methods for glyph recognition. Supervised methods are when there are training sets that have been labelled with the correct answer. The system uses the training set to learn from to generate a model for predictions and the test set to see how well the generated model performs. For unsupervised learning, there are no prespecified labels and the system uses clustering instead to find similar objects (Dickinson et al., 2013). The supervised HTR methods based on deep learning have proved to be satisfying, but it becomes a problem when the cipher contains unique glyphs that it does not recognise. Therefore, a semi-automated method based on unsupervised methods might work better (Baró et al., 2019).

There is a lot of research on automated transcription and different techniques. There is fully automated transcription without any user interaction at all, as well as semi-automated transcription where the user can be involved in the transcription process. HTR techniques developed so far do not perform well when applied to encrypted manuscripts. Handwritten text recognition methods based on deep learning have shown promising results, but there are not always enough labelled data to train supervised HTR models. That is why unsupervised approaches are preferable (Baró et al. 2019). Current HTR methods are based on deep learning architectures such as Recurrent Neural Networks.

2.2 Transcription of Encrypted Historical Sources

What makes automated transcription for historical documents even more challenging is that they are in many cases made up of an unknown set of symbols. The transcription tools must, therefore, be adapted to handle both known and unknown symbol sets with various handwriting styles on various types of material.

2.2.1 Ciphers

As said in the Introduction, a cipher is an encrypted text where the text has been encrypted in a systematic way using some encryption method. The most well-known and straightforward method is simple substitution where each letter
in the plaintext language (the original text) is replaced by some other symbol. To make it more difficult to decrypt the message, the most commonly occurring characters in the alphabet can be encrypted by several symbols, which is called homophonic substitution. Not only characters but also frequent words, people, or geographical names can have its own symbol. Such lists are called nomenclatures.

In order to encrypt a document, any symbol set can be used for the encoding of the characters in the alphabet, and a large variety of symbols do occur across ciphers. Many ciphers are encrypted with numbers only, where each character in the alphabet is encoded with one or several unique numbers as decided by the encryptor (Megyesi et al., 2019). For example, an “a” might be encoded as 101 or 103, while a “b” can be encoded as 301 or 1003, or any other unique number. There are also ciphers that are encrypted with alphabetical characters, graphical symbols such as zodiac or alchemical signs, or a mixture of these (Knight et al., 2017).

Recently, a group of researchers have gathered ciphers to create an online database, DECODE (Megyesi et al., 2019). The goal of the DECODE database is to collect ciphertexts, codes, and keys from various libraries and make it public to everyone interested in historical cryptology. Each cipher is described by a set of metadata including information about the cipher’s current location, origin, and content (Megyesi et al., 2019).

2.2.2 Transcription of Ciphers

The first step in decrypting a cipher is to turn the cipher image into a computer readable text format, i.e. to transcribe the document. Manual transcription has its challenges. It is time-consuming and error-prone, it is easy for the transcriber to lose focus and make mistakes. Before the transcriber starts executing the transcription a transcription scheme needs to be created. The transcription scheme displays how each symbol in the cipher should be transcribed, usually in Unicode or ASCII. Each glyph that is detected in the cipher must be included in the scheme and given a representation. Glyph recognition could be problematic for a human as well as for a computer when there are less frequent or made up symbols. The transcriber can have problems deciding where the character boundaries are. For ciphers, it is hard to predict the symbols given its context, for a normal natural language it might be possible for the transcriber to guess which symbol it is if he, or she, is familiar with the language.

The automatic transcription of historical ciphers has not been recognized until recent years. In a paper written by Fornés et al. (2017) HTR techniques were applied to encoded manuscripts and manual transcription was compared to a fully automated transcription based on Deep Neural Networks followed by manual validation. The researchers ran their tests on 14 different unseen cipher pages while documenting the times for each trial. The result of all tests was that manual transcription is usually 15% slower compared to the automatic transcription with post-editing if the accuracy of the image processing is above 90%. When the accuracy was lower the validation time increased because all incorrect symbols
needed to be localized. The conclusion of the paper shows that image processing can be used for base transcription followed by a postprocessing step involving the user.

Another paper written by Baró et al. (2019) proposes an unsupervised method for transcribing encrypted manuscripts based on clustering and label propagation. The authors implemented the methods used into a web-based transcription tool, namely the Interactive Transcription Tool (ITT), which will be described in the next session. The method is based on three separate modules; pre-processing with line and character segmentation, clustering of symbols, and transcription based on the clusters. The researchers have used ciphers with three different types of symbol sets to test their idea and compared the result of supervised HTR methods. The conclusion of the experiments was that the results are promising, they received up to 62.7% correct symbol classification.
3 Resources and Tools

The aim of this thesis is to apply HTR techniques to encrypted historical sources by using a semi-automated transcription tool. The goal is to compare how well the ITT performs compared to manual transcription and to suggest improvements for a more user-friendly interaction. The ITT is a semi-automated tool that is based on HTR techniques using segmentation, clustering, and classification to terminate in a proposed transcription of a cipher.

3.1 Data

There are a lot of different ciphers, with different symbol sets and different handwritings. The data used for this project has been taken from the DECODE database (Megyesi et al, 2019). 14 different sets from the same cipher type have been made available for this project, some of them used for training and the others for testing. The majority of the sets contain only one page, while some contain two. The sets have been divided into two parts, so the tool is given previously unseen sets of ciphers for the evaluation. The data that has been used for the experiments are numerical ciphers located at the Secret Archives of the Vatican. Some of the pages are dated back to the 15th century. The ciphers originate from different countries in Europe such as Spain, France and Portugal. The ciphers consist of numbers from 0 - 9 and occasionally plaintext of a known natural language, mostly Italian, Spanish and/or Latin. The plaintext could be embedded amongst the number sequences or at the top or bottom of the page. Some pages also include emblems from the Vatican.

The scanned images can be a bit noisy. On some pages the text has bleed-through, there might be ink stains, and some contain the margin from the following page when scanned. It is visually clear that the handwriting varies across ciphers, indicating several scribes. Some pages are very clear, the glyphs are clearly segmented by spaces which should make character recognition of symbols easier. Other pages contain connected symbols, the symbols are very close to each other and some are touching.

3.1.1 Training

The training data was mainly used for learning how the tool works and for practising the transcription supported by ITT. Table 3.1 lists the ciphers that were used for training, all taken from the DECODE database. The ciphers in the training dataset are encrypted with various symbol sets: numerical, and a combination of symbols from Latin, and Greek letters as well as graphical symbols, such as zodiac and alchemical signs. Among the training data, four ciphers use numbers only for encryption, while two contains a mixture of alphabets and
graphical signs to learn a bit more about how the tool performs on ciphers with various symbol sets.

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Writer</th>
<th>Pages</th>
<th>No. of line per page</th>
<th>Symbols</th>
<th>Manual Transcription Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>France_64_7_068</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>Numbers</td>
<td>10 min</td>
</tr>
<tr>
<td>Spain_423_2_294</td>
<td>5</td>
<td>2</td>
<td>21+21=42</td>
<td>Numbers</td>
<td>50+50=100 min</td>
</tr>
<tr>
<td>Spain_423_6_384</td>
<td>5</td>
<td>2</td>
<td>21+2=23</td>
<td>Numbers</td>
<td>60+5=65 min</td>
</tr>
<tr>
<td>Spain_423_9_415</td>
<td>5</td>
<td>2</td>
<td>21+13=34</td>
<td>Numbers</td>
<td>30+20=50 min</td>
</tr>
<tr>
<td>Borg</td>
<td>6</td>
<td>3</td>
<td>15+8+12=35</td>
<td>Mixture</td>
<td></td>
</tr>
<tr>
<td>Copiale</td>
<td>7</td>
<td>2</td>
<td>21+19=40</td>
<td>Mixture</td>
<td></td>
</tr>
</tbody>
</table>

As Table 3.1 shows, the cipher France_64_7_068v is written by one writer out of the sets used for training. It contains one page only and that page has 16 lines. Spain_423_2_294v, Spain_423_6_384v, and Spain_423_9_415v are written by the same person and they consist of 2 pages each. The sets include 42, 23, and 34 lines, respectively. All four ciphers are based on numbers (0-9), as illustrated in Figure 3.1.

The remaining two ciphers, Borg and Copiale, contain various symbols consisting of numbers, Roman and Greek letters, and graphical sign. They are longer books so random pages have been selected for the experiments.

The Copiale cipher is an encrypted manuscript from 1736. The cipher contains over 90 types of glyphs, a mixture of Roman letters, Greek letters and abstract symbols (The Copiale Cipher*, https://cl.lingfil.uu.se/bea/copiale/). An
example is shown in Figure 3.2.

The Borg cipher consists of 34 different glyph types and is assumed to be from the 17th century (https://cl.lingfil.uu.se/bea/borg/). An example from the cipher is shown in Figure 3.3.

![Figure 3.2: Example from the Copiale cipher.](image1)

![Figure 3.3: Example from the Borg cipher.](image2)

3.1.2 Testing

For the evaluation of the interactive transcription tool, only the numerical ciphers were included in the final test. The reason for excluding ciphers with many various
types of glyphs was that the ITT performed the best on these numerical ciphers at the time of the evaluation. The tool had just become functional when the testing started, wherefore all additional functions did not work.

Eight different sets from the numerical ciphers were selected for the final test, see Table 3.2 for a detailed list. The sets were chosen to include various handwritings and writers. In the test set, the ciphers were written by four individuals, two of them are the same as in the training set (scribe 4 and 5). The ciphers contain one page each, but the number of lines varies ranging from 3 up to 25 lines. The time that is recorded is how long it took for the transcriber to transcribe the page, from start to finish.

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Writer</th>
<th>Pages</th>
<th>No. of line per page</th>
<th>Symbols</th>
<th>Manual Transcription Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>France_4_1_221</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>Numbers</td>
<td>5 min</td>
</tr>
<tr>
<td>France_18_2_206</td>
<td>3</td>
<td>1</td>
<td>24</td>
<td>Numbers</td>
<td>45 min</td>
</tr>
<tr>
<td>France_64_6_064</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>Numbers</td>
<td>10 min</td>
</tr>
<tr>
<td>France_64_5_060</td>
<td>4</td>
<td>1</td>
<td>25</td>
<td>Numbers</td>
<td>20 min</td>
</tr>
<tr>
<td>Spain_423_5_374</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>Numbers</td>
<td>15 min</td>
</tr>
<tr>
<td>Spain_423_3_297</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td>Numbers</td>
<td>15 min</td>
</tr>
<tr>
<td>Spain_423_8_391</td>
<td>5</td>
<td>1</td>
<td>14</td>
<td>Numbers</td>
<td>15 min</td>
</tr>
<tr>
<td>Spain_423_10_491</td>
<td>5</td>
<td>1</td>
<td>21</td>
<td>Numbers</td>
<td>25 min</td>
</tr>
</tbody>
</table>

3.1.3 Gold Standard

Each set of the numerical cipher contained the scanned images and a file with the gold standard transcription made manually before this project started. The transcriptions were carried out by students according to transcription guidelines developed for ciphers (Megyesi et al., 2019). Each glyph in the cipher was given an ASCII character that a transcriber could write easily and fast. The transcriber then typed each symbol in the cipher, symbol by symbol, line by line while keeping the line breaks. It is important that spaces, punctuation, and cleartext is kept in the transcription as in the image. When the manual transcription was done the result was validated to make sure that the transcription represents the original image, so it can be used as Gold Standard (Megyesi et al., 2019).

The manually transcribed documents were used to compare the result from the ITT to measure its accuracy as the gold standard. The ITT was never in contact with the gold standard transcriptions, it was only used for the evaluation of our study separately after the ITT has already produced its result.
3.2 Interactive Transcription Tool

The interactive transcription tool is a semi-automated transcription tool created mainly for the transcription of ciphers and was under development while this project was executed (Baró et al., 2019). The tool consists of different steps of which some are made by the machine and some by human interaction. The main modules in the tool are illustrated in Figure 3.4 and include i) Segmentation which carries out page segmentation and symbol segmentation, ii) Clustering which includes identification and categorization of the segments, iii) Classification which performs label propagation and lastly iv) Transcription which is the actual transcription result of using the tool. Below we describe the transcription process step by step.

![Figure 3.4: The transcription process in the ITT.](image)

3.2.1 Upload Image

The first step in the ITT is to upload the images that are desired to be transcribed, see Figure 3.5. The images must be in some kind of image format (e.g. jpeg, png, ...). The images should preferably be from the same cipher, with the same symbol set and handwriting style to receive a more reliable transcription. This uploading function is used for the processed JSON-files that the user will receive after each step in the process as well. More about that in the upcoming chapters.

![Figure 3.5: Image upload.](image)
3.2.2 Segmentation

Once the image(s) of the cipher is uploaded, the user marks the part of the image to be transcribed. If possible, the user should try to make the margins of the page segmentation close to the actual margins of the glyphs to avoid empty segments. In this step, the system also analyses where the line breaks are located. Figure 3.6 shows the page segmentation in the ITT and how the user gets to mark the area to be transcribed in the edit tab. The edit tab contains functionalities of viewing the page segmentation (Edit documents BB), the character segmentation (Edit instances) and the line segmentation (Line Label). The user must then request the segmentation in the Download/Req. Ground Truth tab by entering an email address and then pressing Request Segmentation. The page is binarized by the system and the lines and symbols are segmented.

![Image](image.png)

**Figure 3.6:** Image preprocessing: segmenting the text area to be transcribed.

The segmentation tool segments each row by computing the horizontal projection based on the amount of ink in each row. The lines are located where there are most ink centred. The system connects the symbols to the closest row. If there are symbols that touch two different rows the symbol is cut in the middle. When the line segmentation is performed the system continues segmenting the symbols. It computes the centre of mass of each ink group (Baró et al., 2019). Character segmentation is performed to segment each symbol in the image. For the segmentation request there are six different advanced settings that the advanced, experienced user could modify to change the result:

- Symbol size: Big/Small. Big pages usually mean big symbols, small pages usually mean small symbols. What is big and small is rather arbitrary, the best way is to test and see how correct the line segmentation is. This is hard for the user to determine in the beginning, it gets easier with some practice using the system.
• Binarize image: Yes/No. If the image is in colour mode, it needs to be binarized.

• Minimum line distance: Minimum distance between lines in pixels.

• Lines threshold: Threshold to filter lines. A decimal number between 0 and 1. This is a normalized value. Only those lines with an amplitude higher than this threshold will be detected.

• Max. distance symbols: Maximum distance between symbols in pixels. This parameter is for symbols that have dots or other diacritics. The detection is made first by components and then the system groups those symbols that are close to each other.

• Min. symbol size: The minimum size of the symbol we can find on the page. It is used to filter those components that are smaller than this size.

When the line and symbol segmentation are carried out the user receives an email with a JSON-file that contains the segmented text. The JSON-file must be uploaded in the system the same way the images were uploaded. When the character segmented file is uploaded in the system, the user can see the segmented characters. When looking at the edit tab in the ITT now, the user can see the segmented characters, each type marked by a special colour, see Figure 3.7. To improve the result the user can go through all segments in the cluster toolbar and remove the incorrect ones. At this step in the process, it is very time-consuming and tedious and therefore not recommended.

Figure 3.7: Character segmentation.
3.2.3 Clustering

After character segmentation, the tool creates clusters of glyphs with all segmented symbols. The program calculates which symbols are most like each other and groups similar symbols into one cluster, representing a certain symbol type. This is done by using a hierarchical K-mean algorithm to create the initial clusters. K-mean is a clustering algorithm that stores k centroids to define clusters. A segment is considered to be in a particular cluster if it is closer to the centroid of that cluster than any other centroid (Piech, 2013). The K-mean algorithm builds a hierarchy of clusters in a top-down approach, starting with a big cluster containing a wide range of glyphs and then split it recursively until each cluster has a few similar looking glyphs (Baró, et al., 2019). The user might also decide how many symbols a cluster shall contain minimum. Currently, the default value of the minimum number of symbols per cluster is set to 3. The user needs to request the K-mean and wait for a new email. This request only has one advanced setting:

- Min. cluster images: Minimum number of segments in each cluster. The system uses this parameter to compare with all obtained clusters. If the number of segments inside the cluster is bigger than this value, the system continues splitting that cluster until the number of segments is smaller than this value.

When the email has arrived, the user should upload the attached file as in previous steps and will then be able to see the different clusters. Figure 3.8 exemplifies the output of the clustering where the system suggests several clusters of the symbols in the manuscript to the user. In Figure 3.8 eight different clusters are visible, ranked from the cluster containing most symbols to the one containing the least. It is possible to view each cluster by clicking on the “eye” or to remove a cluster by clicking on the red garbage can. Symbols that have not been assigned to a specific cluster are gathered in a separate cluster.

![Figure 3.8: Cluster toolbar.](image)
The user can look at each potential cluster and see which symbols the system have clustered together. Figure 3.9 is an example of when the system has correctly clustered a glyph representing “3”. In this step of the transcription process, there is another opportunity for the user to remove incorrect symbols. The symbols have been grouped together in clusters and some of these clusters only contain incorrect ones, which makes it easier to remove them then in the Segmentation process.

Figure 3.9: Example fo cluster.

3.2.4 Classification

After clustering, label propagation is carried out to assign labels to unlabelled glyphs in an interactive fashion with the user. It starts off with a subset of labelled glyphs, called the seed. The user can change the number of seeds when requesting the label propagation under advanced settings. The default value is set to 10 which means that it will be 10 different clusters at the end. It uses the most populated clusters from the K-mean algorithm in order to obtain the final label for each glyph. During the propagation, the label can change depending on its nearby neighbours. The algorithm consists of two main parameters, kernel for the propagation and alpha which is used for defining the changeability. The Kernel knn creates a graph by connecting the elements with their closest neighbours based on k. Alpha is a value between 0 and 1: if the value is closer to 0 the symbols with an assigned label will remain unchanged no matter what its neighbour is labelled as. If the value is closer to 1 it is easily changed based on its neighbours. The algorithm is repeated until convergence (Baró et al. (2019).

The user needs to request the Label propagation and then upload the received script. The label propagation request has two advanced settings:

- Change class threshold: Change of class frequency value. A value between 0 and 1. This parameter is used to change segments with labels. Close to 1 means more changes and close to 0 means fewer changes.

- Seeds number: Number of seeds. This is the amount of Cluster that the user gets to label for the transcription.
Figure 3.10 shows how the different clusters are visualized in the toolbar, the same amount of clusters as chosen seeds. The figure shows that the system has been unable to identify 350 symbols, but has placed 127 glyphs in cluster 1, 62 glyphs in cluster 2 etcetera. Each cluster contains the glyphs that the ITT has calculated to be alike. This example output was created using the default settings, which means that the user did not change any of the advanced settings. It was given 10 different clusters to divide all symbols into, additional to the one containing the unidentified symbols.

![Figure 3.10: Classification.](image)

### 3.2.5 Transcription

The actual transcription is the last step, where the user can decide how he, or she, wants to transcribe each glyph, i.e. what the transcription of each symbol type shall look like. The user needs to tell the system how he or she wants each cluster to be transcribed as the final result. There is by now no general transcription scheme for all ciphers available describing how each symbol shall be transcribed but each user decides how they want each glyph to be represented in the transcription. This has to be made manually by the user in the tool. There are two ways of doing it. Either the system can give each cluster a number to represent the actual glyph in the transcription, or the user can label every cluster as they wish to transcribe. As Figure 3.11 illustrates, the user may prefer that all symbols in this cluster should be transcribed as “2”, because the cluster represents the glyph number 2. If the user lets the system decide, it could be any number, which would make the result more confusing if the transcription shall replicate a more natural and (preferably) interpretable content of the image.
When the user has decided how they want the transcription to be represented for each glyph, they can use the downloading function to receive the transcribed image. The user can download a zip file containing the original picture, the transcription, and the Ground Truth as a text file (.txt). The Ground Truth is a JSON-file that makes it possible to modify the final transcription afterwards, as a post-processing step. When opening the text file, the user will see the transcription of the image as they labelled each cluster, as illustrated in Figure 3.12. That is the result, and the image has turned from an image, probably originated from a historical manuscript, into a format that allows further processing for decryption. Now it is up to the user to translate it into an understandable language.
4 Transcribing Ciphers

The purpose of this paper is to explore whether handwritten text recognition techniques based on clustering can be used for the semi-automatic, interactive transcription of ciphers and how well an interactive transcription tool performs compared to manual transcription. Different ciphers were used for practice and to learn how the tool worked with different symbol sets. When the training was done the tool was evaluated on unseen ciphers from different time periods and with different handwritings. See Figure 4.1 for method process.

![Method of the thesis.](image)

4.1 Training

The first step for being able to conduct the experiments to be evaluated was to learn how the ITT worked. Due to the reason that the tool was not fully functioning when this thesis project started the training was a very lengthy process, testing each function as it was developed. The learning period included experimenting with different settings and functions to see how it changed the result. The main purpose of the training was to decide which settings were the best to use for the real experiments on the numerical cipher to provide as promising results as possible. The tool was not fully developed when the training started, it became functional a few days before the testing took place. The training phase consisted of trying each module when functioning and then repeating it due to occurred errors or detected problems. Not all advanced settings were useful at the time, therefore only the ones that showed impact on the result were used for the testing.

For learning the tool, pages from all three cipher types mentioned in Chapter 3.1; Copiale, Borg and the numerical ciphers, were used. The different sets used for training is displayed in Table 1. All ciphers were tested with different settings to see what effect specific settings had on the result. The ciphers were tested with both big and small sizes of symbols, binarized or not, different minimum number of images in each cluster and different amounts of seeds.
4.1.1 Discoveries During the Training Phase

There were two problems that all experiments during the training phase had and therefore the experiments during the test phase is based on improving them. The problems were:

The training phase showed that the tool segmented bleed-through, page margins and folds from the paper as symbols. The tool had problems identifying what was symbols and what was not. Therefore an interesting act would be to remove these incorrect segments and see how that impacted the result.

Another noticeable complication was that the default value of 10 seeds was never enough. The optimal resolution would be that each glyph would have its own cluster, but this was not the case. The tool produced multiple clusters for the same glyph, therefore there were not enough clusters for each glyph to get their own and a lot of symbols ended up in the unidentified cluster.

4.2 Testing

The experiments were conducted on a test data set, images that the system had not come across before. The test data consisted of 8 different ciphers from the numerical cipher set. All sets were tested using 3 different types of setting, based on the result from the training phase;

- Default settings. Will be referred to as Default. This is when the user does not change anything, the process runs with the standard settings of the tool.

- Manually manipulated classification with default settings. Will be referred to as M10. The user has removed all incorrect segments to improve the result. As for the Default, the user does not change any of the advanced settings. During the training period, it was obvious that some segments made by the tool were incorrect, that is why this test and the next one is made with user interaction concerning removing the incorrect segments after clustering.

- Manually manipulated classification with a higher seed number (amount of clusters), will be referred to as M20. The user has removed all the incorrect segments and has changed the advanced setting Amount of Seeds to 20 instead of 10 which is the default value. This setting will result in the most user interaction because the user must go through all clusters, the same amount as for the other manipulated experiment, but it will also have a higher amount of clusters for the user to label which can take some time too.

All three tests are based on the same segmentation and clustering. The two JSON-files are created during the first trial and are then reused for the other two tests as well. All three test for the same set uses the same JSON-files that has been produced during the segmentation and Clustering. In the Classification module, the changes for each test is performed and then they are not alike anymore and will thereby produce its own JSON-file that the final transcription is based on, the process is described in Figure 4.2.
Figure 4.2: The transcription process with the acquired JSON-files.

4.3 Evaluation

The output consists of a text file with the proposed transcription by the ITT. The transcription is then compared to the gold standard transcription of each cypher that was done manually beforehand. The evaluation is carried out by comparing each glyph in the proposed transcription with the glyphs in the gold standard. Sometimes the transcription misses glyphs, then the pattern is compared to what comes after it. If the following glyphs match the gold standard it is possible to assume that the tool has missed segments, not transcribed them incorrectly. The system is expected to generate some errors, so the length of the proposed transcription is compared to the length of the gold standard. Out of the number of glyphs, the ITT found it is counted how many of these that are correct, incorrect and wrongly added.

Precision and Recall are calculated to more clearly show how well the tool performed. Precision is a measurement of how many transcribed symbols generated by the tool are correct, while Recall measures how many symbols did the tool return correctly compared to how many it should have returned. F1-score is the weighted average of precision and recall. (Dickinson et al., 2013). For this evaluation precision, and recall and F1-score has been calculated as follows:

\[
Precision = \frac{Correct\ Transcribed\ Segments}{Amount\ of\ Segments\ by\ Tool}
\]

\[
Recall = \frac{Correct\ Transcribed\ Segments}{Amount\ of\ Segments\ Gold\ Standard}
\]

\[
Recall = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}
\]

To be able to compare ITT with manual transcription, transcription time was also recorded, both for manual transcription and for ITT. The time has been recorded in minutes for the whole process from uploading an image to receiving the final transcription. The time for the automatic analysis was compared to how long each manual transcription of each cipher page took.
5 Results

In this chapter, the results from the experiments described in the previous chapter are presented and discussed. All useful information for each test has been documented, the transcription outcome from the tool has been compared to the Gold Standard, and Precision and Recall, and F1-score have been calculated for each test.

5.1 Result Experiments

The test was executed as explained in Chapter 4. Each set of ciphers displayed in Table 3.2 were processed through the tool three times; once with the default settings (D), once with default settings but with user removal of empty segments (M10) and once with user removal of empty segments but with the setting of 20 seeds (M20), instead of 10 seeds as default. The results are presented in Table 5.1 where the name of the cipher from the DECODE database, followed by which setting that was used, how long the transcription took, how many lines and glyphs the final transcription contained, and how many of these symbols that were correct, incorrect, missing and wrongly added are presented. The time is how long it took for the whole process from uploading the first scanned document until receiving the final transcription. The manual gold standard transcription (GS) is included as the first setting for each set in the table, so it is easier to see the differences between the manual (GS) and semi-automatic transcription results with various settings (D, M10 and M20 respectively).

Precision and recall for each test are presented in Table 5.2. The accuracy of the manual transcription is included to be compared with the transcription results of the ITT. The manual transcription accuracy is how well the transcriber performed before the validation.

The results show that it was always faster to use the ITT than manually transcribed ciphers, except in one case when the ITT and the manual transcription performed similarly. That was in the shortest set that only contained three lines.

There were two sets that distinguished themselves from the rest, France_4_1_221 and France_64_5_060. France_4_1_221 was the only test where the default setting was not the fastest test out of the three settings. Both M10 and M20 were faster, but only with one minute. This set was the one that contained the least amount of line, only three. It was only for France_4_1_221 the tool managed to transcribe the exact amount of lines, and it did that for all three settings. For both France_4_1_221 and France_64_5_060 the test with default settings found less segmented symbols than there were in the Gold Standard. France_64_5_060 was
<table>
<thead>
<tr>
<th>Cipher</th>
<th>Setting</th>
<th>Time</th>
<th>Lines</th>
<th>Symbols</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Missing</th>
<th>Wrongly Added</th>
</tr>
</thead>
<tbody>
<tr>
<td>France_4_1_221</td>
<td>GS</td>
<td>5 min</td>
<td>3</td>
<td>118</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>6 min</td>
<td>3</td>
<td>117</td>
<td>74</td>
<td>39</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>5 min</td>
<td>3</td>
<td>117</td>
<td>80</td>
<td>33</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>5 min</td>
<td>3</td>
<td>117</td>
<td>87</td>
<td>23</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Spain_423_3_297</td>
<td>GS</td>
<td>15 min</td>
<td>8</td>
<td>288</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>7 min</td>
<td>15</td>
<td>324</td>
<td>220</td>
<td>43</td>
<td>19</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>8 min</td>
<td>14</td>
<td>294</td>
<td>219</td>
<td>40</td>
<td>24</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>10 min</td>
<td>14</td>
<td>294</td>
<td>257</td>
<td>17</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>France_18_2_206</td>
<td>GS</td>
<td>45 min</td>
<td>24</td>
<td>1290</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>14 min</td>
<td>50</td>
<td>1458</td>
<td>428</td>
<td>893</td>
<td>59</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>19 min</td>
<td>35</td>
<td>1278</td>
<td>427</td>
<td>757</td>
<td>53</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>21 min</td>
<td>36</td>
<td>1278</td>
<td>959</td>
<td>272</td>
<td>57</td>
<td>47</td>
</tr>
<tr>
<td>Spain_423_5_374</td>
<td>GS</td>
<td>15 min</td>
<td>10</td>
<td>420</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>6 min</td>
<td>19</td>
<td>458</td>
<td>381</td>
<td>33</td>
<td>8</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>8 min</td>
<td>15</td>
<td>428</td>
<td>381</td>
<td>31</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>9 min</td>
<td>15</td>
<td>428</td>
<td>389</td>
<td>22</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>France_64_6_064</td>
<td>GS</td>
<td>10 min</td>
<td>16</td>
<td>763</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>8 min</td>
<td>29</td>
<td>776</td>
<td>374</td>
<td>356</td>
<td>35</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>9 min</td>
<td>18</td>
<td>733</td>
<td>461</td>
<td>259</td>
<td>40</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>10 min</td>
<td>17</td>
<td>733</td>
<td>559</td>
<td>161</td>
<td>43</td>
<td>13</td>
</tr>
<tr>
<td>Spain_423_8_391</td>
<td>GS</td>
<td>15 min</td>
<td>14</td>
<td>532</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>8 min</td>
<td>31</td>
<td>583</td>
<td>255</td>
<td>261</td>
<td>16</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>8 min</td>
<td>24</td>
<td>550</td>
<td>333</td>
<td>180</td>
<td>22</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>10 min</td>
<td>25</td>
<td>550</td>
<td>468</td>
<td>43</td>
<td>21</td>
<td>39</td>
</tr>
<tr>
<td>France_64_5_060</td>
<td>GS</td>
<td>20 min</td>
<td>25</td>
<td>1311</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>10 min</td>
<td>35</td>
<td>1263</td>
<td>382</td>
<td>847</td>
<td>113</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>17 min</td>
<td>37</td>
<td>1129</td>
<td>593</td>
<td>451</td>
<td>222</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>18 min</td>
<td>36</td>
<td>1129</td>
<td>904</td>
<td>125</td>
<td>229</td>
<td>100</td>
</tr>
<tr>
<td>Spain_423_10_491</td>
<td>GS</td>
<td>25 min</td>
<td>21</td>
<td>876</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>13 min</td>
<td>60</td>
<td>1061</td>
<td>441</td>
<td>383</td>
<td>18</td>
<td>237</td>
</tr>
<tr>
<td></td>
<td>M10</td>
<td>24 min</td>
<td>36</td>
<td>894</td>
<td>544</td>
<td>316</td>
<td>17</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>M20</td>
<td>25 min</td>
<td>39</td>
<td>894</td>
<td>670</td>
<td>186</td>
<td>19</td>
<td>38</td>
</tr>
</tbody>
</table>

GS: Gold standard.
D: Default, when no user interaction has been made and only the default settings.
M10: When the user has removed all empty glyphs but with default settings.
M20: When the user has removed all empty glyphs and changed the number of seeds to 20.
Correct segments: The ones that are on the correct place and the correct number.
Incorrect segments: Right place, wrong number, all question marks in the transcription will be here.
Missing segments: Where there should be a segment, but the ITT has not managed to find it.
Wrongly added segments: Where there should not be a segment but there are.
<table>
<thead>
<tr>
<th>Cipher</th>
<th>Setting</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>France_4_1_221</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.2%</td>
<td>62.7%</td>
<td>62.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68.4%</td>
<td>67.8%</td>
<td>68.1%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.4%</td>
<td>73.7%</td>
<td>74.0%</td>
</tr>
<tr>
<td>Spain_423_3_297</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67.9%</td>
<td>76.4%</td>
<td>71.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.5%</td>
<td>76.0%</td>
<td>75.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87.4%</td>
<td>89.2%</td>
<td>88.3%</td>
</tr>
<tr>
<td>France_18_2_206</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29.4%</td>
<td>33.2%</td>
<td>31.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>33.4%</td>
<td>33.1%</td>
<td>33.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.0%</td>
<td>74.3%</td>
<td>74.6%</td>
</tr>
<tr>
<td>Spain_423_5_374</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>83.2%</td>
<td>90.7%</td>
<td>86.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>89.0%</td>
<td>90.7%</td>
<td>89.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>91.0%</td>
<td>92.6%</td>
<td>91.8%</td>
</tr>
<tr>
<td>France_64_6_064</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48.1%</td>
<td>49.0%</td>
<td>48.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>62.9%</td>
<td>60.4%</td>
<td>61.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>76.3%</td>
<td>73.3%</td>
<td>74.8%</td>
</tr>
<tr>
<td>Spain_423_8_391</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>43.7%</td>
<td>48.0%</td>
<td>45.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60.5%</td>
<td>62.6%</td>
<td>61.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.1%</td>
<td>87.9%</td>
<td>86.5%</td>
</tr>
<tr>
<td>France_64_5_060</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30.2%</td>
<td>29.1%</td>
<td>29.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52.5%</td>
<td>45.2%</td>
<td>48.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80.1%</td>
<td>69.0%</td>
<td>74.1%</td>
</tr>
<tr>
<td>Spain_423_10_491</td>
<td>GS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>41.6%</td>
<td>50.3%</td>
<td>45.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60.9%</td>
<td>62.1%</td>
<td>61.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.0%</td>
<td>76.5%</td>
<td>75.7%</td>
</tr>
</tbody>
</table>
the only one where the M10 transcribed more lines than the Default, 37 lines compared to 35 lines. For the amount of wrongly added segments France_64_5_060 distinguished itself, it was the only test where M20 had the most wrongly added ones.

If we disregard France_4_1_221 and France_64_5_060, all other tests share the same behaviour. The experiments with the Default setting contained the most amount of segmented symbols in each trial and the most exceeded amount of lines. The Default trials also contained the smallest amount of correct segments and the highest amount of incorrect segments. The M10 trials always performed a little bit better than Default, but not as well as the M20. The M20 usually had a high amount of correct segments and a very low amount of incorrect segment, but unfortunately for the vast majority missed a lot of segments compared to the gold standard and also had the problem with wrongly added segments where it should not be segments.

Examples have been provided to give the user a clearer view of the transcription file and the problems that exist. Figure 5.1 is a screenshot from the transcription file of Spain_423_3_297 M10. The glyphs have been colour-coded to better mark the differences. The green segments are correct, the yellow ones are incorrect, the red symbols are the wrongly added ones and the blue marked dots are the missing symbols.

![Figure 5.1: Transcription Spain_423_3_297 with setting M10 to the left and Gold Standard to the right.](image)

In figure 5.1, six lines are marked in red and these are the supernumerary lines that were documented in Table 5.1 when compared to the amount of lines of the manual transcription. The transcription should only contain eight lines. A conclusion that can be drawn is that the tool sometimes has a problem with finding the lines after segmentation, or it might be the symbols that touch two different lines (a clearly problematic case as expected and explained in Chapter 2.1). Comparison of the generated transcription to the Gold Standard manual transcription (see fig. 5.1), shows that the transcription does not contain any “1”. The tool did not managed to classify “1” into its own cluster so we can assume that they are spread out between other clusters.
Figure 5.2 represents a transcription that had an unidentified cluster, compared to figure 5.1 where all symbols have been transcribed with a number. Figure 5.2 shows a smaller part from the transcription of Spain_423_8_391 with Default settings. The question marks represent symbols that have ended up in the unidentified cluster. The tool shows the same problem with the wrongly added segments making out whole lines as mentioned before. The tests with the default value got a precision score of 43.7%, a recall score of 48.0% and its weighted F1-score was 45.7%. All these scores demonstrate that the tool did not perform well on this cipher with the default settings. On the test with the M20 setting, however, the tool performed much better. The tests precision score was 85.1%, recall was 87.9% and its F1-score was 86.5%, which are acceptable results. The tool managed to transcribe 468 symbols out of 532 symbols correct with the M20 setting. Only 43 symbols were incorrectly transcribed compared to the Default test which had 261 incorrect transcribed symbols.

Looking at the results in Table 5.2, one set of cipher stood out, Spain_423_5_374. It was the only test where all three tests had a higher recall score than the accuracy of the Gold Standard. For M10 and M20 the precision score was higher as well. Besides Spain_423_5_374 the Accuracy of the Gold Standard is higher than the best result for each cipher, which in all cases were the M20.

Excluding Spain_423_5_374, the result for all tests with the Default setting shows that it is not that reliable to run the tool with only standard settings. Spain_423_3_297 got the highest precision score for Default with 67.9%, which at least is promising. The lowest precision score for Default, however, was France_18_2_206 with 29.4%. That means that less than a third of all segments were classified correctly. It is clear that the tool performed better when it got more clusters to work with, and when the incorrect segments were removed. M20 achieved the highest score for both Precision and Recall for all sets.
5.2 Using the ITT

Each module in the ITT has its advantages and disadvantages. The final result is very much relying on how well each step in the process is performed as errors generated in one step are propagated to the next. To be able to understand what impact each step has had on the final transcription we illustrate the whole transcription process with an example taken from the experiments. The document with the highest Precision and Recall scores, i.e. with the highest F-score, namely Spain_423_5_374, has been chosen to visualise the positive and negative aspects of the ITT’s different modules. Spain_423_5_374 is a short cipher with neat handwriting where each symbol is segmented well and written clearly, lined up in a clear horizontal structure. The cipher is shown in Figure 5.3. The same Segmentation- and K-mean JSON-file has been used for all three different settings.

![Figure 5.3: Spain_423_5_374.](image)

5.2.1 Segmentation

Figure 5.4 shows how the text area on the page has been marked by the user by the red square. Then the symbols have been segmented into glyphs where each glyph is marked by a specific colour (fig. 5.5). The tool has segmented the page as if it contains 458 symbols when it should contain 420 symbols instead (see Table 5.1). It is clear that the tool had some trouble distinguishing the character boundaries of some symbols. On the penultimate line, there are two symbols 6 and 9, that are wrongly segmented as one “69” twice. The tool was not able to separate their different centroids probably due to that sense that “6” and “9” have the same shape, only displayed in different angles.

Figure 5.6 displays a screenshot of the first lines of the line segmentation. The tool segmented the page as if it contained 19 lines when it actually contains 10
lines (see Table 5.1). This is probably why there are so many supernumerary lines in the final transcription. We can assume that the first line is correct, but the second and third lines contain almost no symbols, they only consist of empty segments. As for line seven and eleven, they only contain the lower part of the symbol “9” from the row above.

5.2.2 Clustering

The K-mean algorithm has created 77 clusters out of the original 458 symbols. The biggest cluster contains 31 symbols and is displayed in Figure 5.7. For this cluster, the tool managed to cluster most of them correctly, but there are four glyphs that are circled in red that should not be in this cluster. In the second biggest cluster, the tool managed to cluster all symbols correctly, all 24 represent the symbol “7”. Figure 5.8 shows a cluster containing four glyphs that were not segmented correctly. All glyphs are composed of two symbols, one of them being a “6” for all glyphs and the other being a “3”, “7” or something that is probably only a part of another symbol (the last symbols in Figure 5.8). There are also some
clusters that only contain empty segments. These clusters have been removed for the M10 and M20 tests to improve the result.

![Figure 5.7: Cluster of the symbol "2".](image)

5.2.3 Classification

To show how the classification module behaved examples from the test with the setting of 20 seeds and the empty segments removed have been used. This was the setting that got the best score for this set for both precision and recall. The result of the classification is 20 different clusters and the one with unidentified glyphs, but in this case, that cluster is empty. The biggest cluster contained 43 symbols, of which nine were incorrect. The second biggest cluster contains 33 symbols, whereas only two of them were incorrect. There were three different clusters that
all consisted of the symbol “3”, except a few errors. These can be merged together and transcribed as one. Figure 5.9 shows the cluster of the glyphs representing “6”, but the problem is that these glyphs have not been segmented correctly. They contain the symbol “6” but also a part of another glyph. This might be the reason why some glyphs are missing in the final transcription.
6 Discussion and Suggestions for Improvements

The overall impression of the results from the Interactive Transcription Tool is that it is very promising. It performs well on the shorter ciphers with clean handwriting, and in those cases, it is significantly faster than manual transcription. Even though there are possible improvements that can be done, ITT shows that it can produce a functional, good-enough result to help the user to get a transcription.

6.1 Discussion

The biggest advantage of the tool is that it is much faster to use, compared to manual transcription when one has learned how the system works. Depending on the length of the cipher, if it is only a few lines it takes about the same time to use the tool as transcribing it manually. However, if the cipher is long, using the tool might be quicker depending on how much involvement it needs from the user.

Most of the problematic areas are grounded in the segmentation process, and since that is the first module in the tool, all the following modules are affected. The segmentation needs to be improved to avoid error propagation to the subsequent modules. The line segmentation is the most problematic area because the tool generates lines that do not exist. The Clustering and Classification modules perform as they should and there is nothing to remark concerning those. The Transcription module is reasonable and creates the result based on what it has been given by the previous modules. The methods that the tool is based on should not change but the segmentation module needs to improve for the tool to perform better.

It is important that all symbols are included because missing symbols can have a big negative impact during decryption. In some of the tests whole lines were missing and that could make it more complicated to decode the transcription. It is necessary that all symbols in the original cipher are somehow marked. Unrecognized symbols will then end up in the unidentified cluster and be transcribed with a question mark or something similar that show that something should be in that position.

6.1.1 Default Setting

The tests that were carried out with the default settings had a very diverse result when it comes to precision and recall. The precision scores were between 29.4% and 83.2%, and the recall scores varied between 29.1% and 90.7%.

The most valuable score to look at in this case is the Recall score, which is...
the amount of correctly transcribed segments by the tool divided by the number of segments in the gold standard. Spain_423_5_374 which got the highest score for both precision and recall was one of the shorter sets of ciphers. The tool only missed 8 symbols in the segmentation and it was 44 symbols that were wrongly added, and there were only 33 symbols that had been incorrectly transcribed. For this set, the tool has performed very well.

For the sets with the lowest scores, France_18_2_206 which had the lowest precision score and France_64_5_060 which had the lowest recall score, the tool did not perform well at all. Their F1-scores are 31.2% and 29.6% respectively and these scores are low and would not be reliable to base the transcription on. For France_18_2_206 the tool transcribed 893 segments incorrect, 78 segments were added wrongly, and it missed 59 symbols. In France_64_5_060 847 symbols were transcribed incorrect, 34 were wrongly added and the tool missed to segment 113 symbols. Compared to Spain_423_5_374 with the highest score, we can see that these two sets both missed a lot more segments and more than half of the symbols were incorrectly transcribed. However, all three of the ciphers had a lot of wrongly added segments.

6.1.2 M10 Setting

The test that was made with the M10 setting, where the user removed the incorrect segments but used the default settings concluded in an average result, better than the Default test but worse than the M20 scores. The time it took to perform the experiments did not differ a lot from the M20 test either, and they got a better score. This shows that the result improved when removing the incorrect segments which strengthen the discussion about the importance to improve the segmentation.

6.1.3 M20 Setting

Both Precision, and Recall, and F1-score showed good values for each M20 test. The tool performed the best on the M20 tests, where the incorrect segments had been removed and 20 seeds were used instead of the default value of 10. The recall scores were all between 69.0% and 92.6% for the M20 tests. The recall score gives the most precise calculation of how the tool performed because it shows how many of all expected segments the tool managed to transcribe correctly. It takes into account that the tool could have missed some symbols, which it did, compared to the Precision score which only shows how many symbols out of the ones it found that was correct. All precision scores for the tests were between 74.4% and 91.0%, which make it seem like the M20 test with the lowest score got a higher score than it deserved since it missed a lot of segments. The F1-scores for the tests were acceptable and with some improvements, the tool can perform even better in the future.
6.2 Suggestions for Improvements

The tool shows promising results but there are some improvements that must be done to get a more reliable transcription and make it more user-friendly. If the tool is too complicated to use it does not matter how well it performs, the user will find other alternatives. The most important thing that the ITT needs, is to improve the user experience. The tool must be easier to use and more straightforward. A suggestion on how the process and its appearance could be improved follows. All the modules already exist but they need to change to make it more user-friendly:

1. The user uploads the image(s) that shall be transcribed, then there is a “ready-button” on the same page that redirects the user to the page segmentation.

2. The user marks where the page segmentation should be, then there is a button for requesting the Segmentation with the possible advanced settings. The user gets a message in the tool like the existing one “an email has been sent” and is redirected to a page where it is possible to upload the JSON-file that has been received.

3. When the file has been uploaded the page redirects to an edit page where it is possible to see the line- and character segmentation. There should be a request button for the K-means on the same page with the advanced settings.

4. The user is then redirected to a page where the corresponding JSON-file is uploaded and following that, redirected to a page that shows all the clusters. At this stage it must be possible for the user to view the different clusters and to remove inaccurate ones. The page should contain a Request Label Propagation button with the additional settings.

5. The user is redirected to a new page where the Label Propagation file can be uploaded and then directed to the final page, transcription.

6. On the transcription-page, the user will be able to decide how each cluster should be transcribed and then there should be a button for downloading the transcription.

The goal of the tool is to have a minimum of user involvement but still get good results. Some suggestions on how to reduce user involvement:

- Apply a more user-friendly viewing of the different glyphs. One way could be to display all segments at once in the edit tab and make it possible to “cross out (delete)” the incorrect segments. As it is now, the user must examine every single cluster and then press delete on only that specific cluster, and then open the next one and repeat the process until all clusters have been examined. This will probably affect the duration of the process in a positive way.
• Improve the viewing of each cluster when it is time for the user to decide how each cluster should be transcribed. Make it possible to view all clusters at once in one window and then decide for each cluster how it should be transcribed.

One of the biggest problems with the tool is located to the first module, the line segmentation, and that impacts the entire process in a negative way. If the segmentation was more reliable there would not be many things to pay attention to in the Clustering or Classification. It does not matter how fast the tool is to use, if the result does not reach the user’s expectation. Here follows some suggestions or necessities linked to the performance of the ITT:

• Add some kind of post-processing on the page segmentation. Either involve the user more and make it possible to remove/change the line segmentation. Or train the tool to detect when there are lines that should not be there. This will reduce the amount of wrongly added symbols substantially and by that highly improve the result.

• Make it possible for the user to edit the segmentation, the symbol boundaries. It could be possible for the user to zoom in on the segmented page and change the segmentation on the symbols that have been inaccurately segmented as one when there should be two different ones. There should be a function for moving the boundaries that mark a glyph and add a new box to the one that is left out, which is then made into its own segment.

• Make the default value of the amount of seeds higher. There will probably not be many ciphers that have less unique symbols than this numerical cipher and the experiments shows better result with a higher seed number. It might be possible to add a box where the user can write the estimated amount of unique symbols because the ultimate goal would be that each unique symbols have its own cluster. That would be a change in how the user interprets the amount of seeds and it might not improve the speed but would give a better result.

• Make it possible for the user to move symbols between clusters. If the symbol has been clustered incorrectly it should be possible for the user to move it to the correct cluster.
7 Conclusion and Future Directions

The aim of the thesis was to apply HTR techniques to encrypted historical sources, more specifically, to examine how well a semi-automated transcription tool (ITT) developed for the transcription of encrypted sources using numbers for encryption performs compared to manual transcription.

The interactive transcription tool consist of page, line and character segmentation, followed by clustering of similar glyphs into groups. The tool is based on unsupervised learning where no labelled data is needed. The tool allows user experiments with various settings with regard to the number of clusters. The tool has been tested and evaluated based on Precision, Recall, F-score. The experiments were performed on numerical ciphers divided into eight sets of one or two images. The sets were written by different people, had different properties and appearances. All sets were tested three times, with different settings: one with the default settings, one where the user removed all incorrect segments using default settings, and one where the user removed all empty segments and changed the amount of seeds to 20 instead of 10.

The result showed that the ITT is very promising. The tool performs very well on the shorter sets with neat handwriting but has more trouble with the longer ones with many symbols on each line. The recall is the best value to look at when evaluating the system because it shows how many symbols that the tool transcribed correctly compared to the number of symbols in the Gold Standard. The highest received Recall score was 92.6%, which is a very satisfying result. The highest Precision score was 91.0%, a satisfying result as well. The F1-score for the M20 tests ranged between 74.0% and 91.0%, which are all acceptable results and shows promises for even better results when the tool has been improved. The tool needs some improvements before it can be considered reliable. The segmentation module which all other modules is based on needs to be improved. The tool adds a lot of lines that should not be there, and it sometimes misses symbols that are necessary for the decryption to be correct.

The thesis has shown that HTR techniques based on clustering can be used for semi-automatic transcription on ciphers with a numerical symbol set. The method seems very promising and with some improvements, it can be expected to work well. The tool has proved to work faster than manual transcription, even though it does not achieve as good result as manual transcription at the moment. There is a big chance that it will do so in the future. There are some necessary adjustments that the tool must go through, the segmentation must be improved, and a post-processing module should be implemented. HTR techniques based on clustering has proved to be a promising method for semi-automatic transcription and should be examined and developed further. There should be more research
done based on this method with regard to the line and character segmentation, and the Interactive Transcription Tool should be upgraded. In this thesis, the ITT was only tested on numerical ciphers so in the future a bigger study should be performed on ciphers with different symbol sets. There should also be a more intense study on how the different advanced settings impact the result depending on what cipher type to be transcribed.
8 References


Knight, K., Megyesi, B., and Schaefer, Ch. The Copiale Cipher* URL: https://cl.lingfil.uu.se/ bea/copiale/. Date: //2019


