Towards a Generic Unsupervised Method for Transcription of Encoded Manuscripts

Arnau Baró, Jialuo Chen, Alicia Fornés
Computer Vision Center, Computer Science Department,
Universitat Autònoma de Barcelona
Bellaterra, Spain
{abaro,jchen,afornes}@cvc.uab.es

Beáta Megyesi
Linguistics and Philology, Uppsala University
Uppsala, Sweden
beata.megyesi@lingfil.uu.se

ABSTRACT

Historical ciphers, a special type of manuscripts, contain encrypted information, important for the interpretation of our history. The first step towards decipherment is to transcribe the images, either manually or by automatic image processing techniques. Despite the improvements in handwritten text recognition (HTR) thanks to deep learning methodologies, the need of labelled data to train is an important limitation. Given that ciphers often use symbol sets across various alphabets and unique symbols without any transcription scheme available, these supervised HTR techniques are not suitable to transcribe ciphers. In this paper we propose an unsupervised method for transcribing encrypted manuscripts based on clustering and label propagation, which has been successfully applied to community detection in networks. We analyze the performance on ciphers with various symbol sets, and discuss the advantages and drawbacks compared to supervised HTR methods.

CCS CONCEPTS

• **Applied computing** → **Document management and text processing**; **Document capture**; **Optical character recognition**;

KEYWORDS

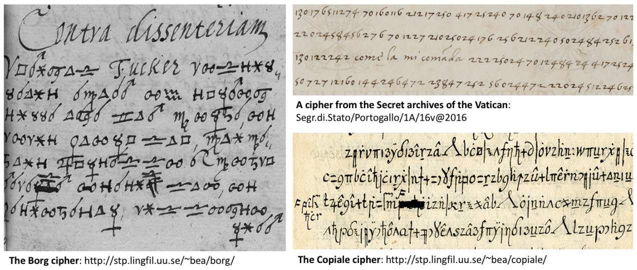
Handwritten text recognition, Encoded manuscripts, Unsupervised methods.

ACM Reference Format:
Arnau Baró, Jialuo Chen, Alicia Fornés and Beáta Megyesi. 2019. Towards a Generic Unsupervised Method for Transcription of Encoded Manuscripts. In *3rd International Conference on Digital Access to Textual Cultural Heritage (DATeCH2019), May 8–10, 2019, Brussels, Belgium.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3322905.3322920

1 INTRODUCTION

Historical manuscripts constitute a key component of our collective memory, without which an understanding of our common background would be severely limited. A special type of handwritten historical records are ciphers, encrypted messages to keep their content hidden from others than the intended receiver(s). Examples of such materials are diplomatic correspondence, scientific writings, private letters, diaries, and manuscripts related to secret societies.

There are many ways to encode information. One common encryption used during the early modern period is substitution where alphabetical characters in the original message are replaced by other symbols in a systematic way. Single characters, letter combinations, syllables, morphemes, words, phrases, or even sentences can be substituted. The symbol set in a cipher might consist of digits, existing alphabets, special symbols such as alchemical or zodiac signs, or other unique, made-up symbols, and not infrequently a mixture of these. Figure 1 illustrates different ciphers. The encoded sequences are usually meticulously written and often segmented symbol by symbol to avoid any kind of ambiguity when decoding the content, but connected symbols also appear. To hide information about word and sentence boundaries, *scriptio continua*, i.e., writing without any spaces or punctuation marks is also common. In addition, the ciphertext might be embedded in cleartext, i.e., non-encrypted text, as illustrated in the third line of the second example in Figure 1.



A cipher from the Secret archives of the Vatican: Segr.di.Stato/Portogallo/1A/16v@2016

The Borg cipher: http://stp.lingfil.uu.se/~bea/borg/

The Copiale cipher: http://stp.lingfil.uu.se/~bea/copiale/

Figure 1: Ciphers from the 16th, 17th and 18th centuries.

To decode the secret writing, it is necessary to digitize and transcribe the ciphers. Transcription can be manually performed by trained expert transcribers. First, each unique symbol, i.e. glyph in the encrypted text, must be identified, and a transcription scheme for glyphs has to be developed. Then, based on this transcription scheme, the user types in every symbol. Manual transcription is time-consuming and expensive, and prone to errors, especially for ciphers with unique symbol sets. Therefore, (semi-)automatic transcription methods are preferable.

Nowadays, Handwritten Text Recognition (HTR) methods based on deep learning have good performance. However, ciphers may contain a unique symbol set with unknown symbols, so transcription is rarely available to train these supervised HTR models. Thus, we believe that unsupervised methods are preferable, because they can be applied to any cipher and symbol set, without any need of

DATeCH2019, May 8–10, 2019, Brussels, Belgium

ACM ISBN 978-1-4503-7194-0/19/05…$15.00
https://doi.org/10.1145/3322905.3322920

labelled data. In fact, creating training data may not be worth for unique ciphers. Besides, only experts can provide reliable labels.

In this paper, we propose an unsupervised transcription method based on clustering and grouping of glyphs by label propagation. Our model can automatically segment and group the most likely (frequent) individual symbols in the cipher, thereby easing the identification of the glyphs and the transcription process. We analyze its performance for ciphers with different symbol sets, and discuss its strengths and weaknesses compared to supervised methods.

2 STATE OF THE ART

Although some unsupervised methods [5] have shown to improve the OCR accuracy of degraded printed documents, the transcription of handwritten documents usually require learning-based (supervised) systems. Current HTR techniques are based on deep learning architectures [11].

Although the performance of these approaches has significantly improved in the last decade, in the case of historical manuscripts, the inherent variability of handwriting styles and the amount of different languages and scripts, still make HTR an open research problem. Consequently, the user is often included in the transcription process. For example, in [3, 9], manuscripts from the Vatican Secret Archives are transcribed through character segmentation and recognition. Two different techniques are used to segment the words into characters. First, the ink pixels are counted for each column and any local minima is considered as possible character boundaries. Second, the upper and the lower contours for each connected component is analyzed. The candidates for character boundaries are then validated through crowdsourcing. Afterwards, Convolutional Neural Networks are used for transcription. Since these manuscripts are written in Latin, n-grams obtained from a medieval Latin corpus are used as language models for disambiguation. Another example of user intervention can be found in [10], where Recurrent Neural Networks are used to transcribe numerical ciphers from the Vatican Secret Archives. This system also includes a post-validation step by manual correction of the automatic transcriptions. In [15], the symbols in the ciphers are segmented using a generative model, and a Siamese Neural Network with character n-gram model is applied for transcription.

We believe that supervised methods are not really suitable for transcribing encrypted documents with unknown symbol sets. First, there is no labelled data available for unknown ciphers. Secondly, the transcription system cannot benefit from any language model because the cipher key is, a priori, unknown. Third, supervised methods based on deep learning architectures are data hungry, which is an important limitation for developing generic transcription methods. Besides these reasons, it is also true that the arcane nature of the symbols in the cipher could require the study of techniques closer to symbol recognition rather than text recognition.

3 UNSUPERVISED TRANSCRIPTION

As stated in the introduction, we focus on unsupervised models without the need of manually transcribed data. Thus, the model can be applied to recognize any symbol set in historical ciphers. The proposed method is composed of the following steps. First, each page is binarized and the lines and symbols are segmented. Then, the symbols are clustered according to their shape similarity and the most populated clusters are used as seeds for propagating their labels through the rest of symbols. At the end of this process, the final labels for each symbol are used to output the transcription for each line. These steps are explained in detail next.

3.1 Preprocessing and Segmentation

First, the image is binarized, the margins of the page are removed, and the connected components are computed. For segmenting lines in the page, we compute the horizontal projection (see Fig. 2), i.e., the amount of ink in each row. Here, the peaks of ink determine the lines. Finally, each connected component is linked to the nearest line. In case a cipher contains touching symbols from consecutive lines, this large connected component is cut by the middle so that each part is linked to its corresponding line.

Once the lines are segmented (see Fig. 2 [Segmented Line]), we proceed to segment the symbols within each line. As stated by the Sayre's paradox, handwritten text cannot be properly recognized without being segmented and vice versa. However, in ciphers with unknown symbol sets, not even the scholar knows which is the exact alphabet of symbols. For this reason, the segmentation step does not separate touching symbols unless they are located in different lines. In addition, we must take into account that there may be symbols composed of several connected components (e.g. the character *i* is a compound symbol, composed of a straight line with a dot above). For this reason, our symbol segmentation is based on connected components and grouping rules as follows. First, the connected components previously obtained (see Fig. 2 [Segmented Symbols]) are analyzed to determine which connected components must be grouped. Two connected components will be grouped if their centers of mass are very close, or one above the other (see Fig. 2 [Symbols Grouping]). Note that some symbols that were originally separated could be now wrongly joined.

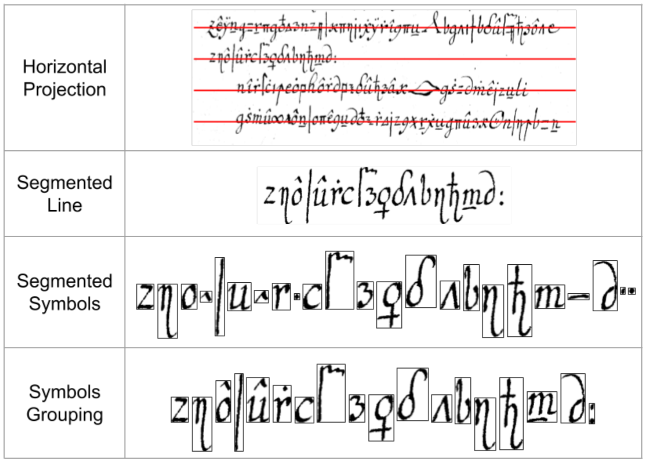


Figure 2: Segmentation example.

3.2 Clustering

Once the symbols are segmented, they are clustered to obtain the most likely symbol alphabet in the cipher. For this purpose, we

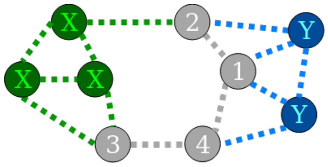


Figure 3: Example of the *knn* kernel for label propagation. There are two labels (X and Y) and four unlabeled elements.

use a hierarchical k-means algorithm to obtain the initial clusters. Then, the most populated clusters are automatically selected. These n clusters are used as initial seeds for the label propagation step, which will determine the final label for each symbol.

3.2.1 Hierarchical Clustering. The hierarchical k-means algorithm is used to obtain the initial seeds for the Label Propagation algorithm. First, we describe each symbol using the SIFT descriptor, which is commonly used for text [2]. Then, we build the hierarchy of clusters in a top-down (divisive) approach, starting with one big cluster and splitting it recursively. This process ends when the clusters have few symbols. We have used the Squared Euclidean distance as the similarity metric.

We can set up two parameters for better results and/or speeding up the process. The first parameter limits the number of generated clusters, which is useful for ciphers with many symbols. The second parameter limits the number of symbols in each cluster. The more symbols in a cluster, the more initial seeds for label propagation. Please notice that many seeds will introduce noise in the label propagation step, especially when a ciphertext is short.

3.2.2 Label Propagation. The Label Propagation algorithm [7] consists of assigning labels to unlabeled elements. This semi-supervised algorithm starts with a subset of labeled elements, namely the seeds, that are propagated through all the unlabelled elements. This algorithm has been typically applied to graph analysis and community structure detection in networks [14].

Since we propose an unsupervised method, the seeds are not manually labelled. Instead, we use the most populated clusters (provided by the hierarchical k-means) as seeds in order to obtain the final label for each segmented symbol. Therefore, the process starts with these n populated clusters, so we have n different labels to diffuse through all the space. During the propagation, the label of a symbol can change depending on the labels of its neighbors.

The algorithm has two main parameters. The first one is the *kernel*, used for the propagation. The kernel *knn* (k-nearest neighbors) creates a graph by connecting the elements with their closest neighbours based on k. The kernel rbf (radial basis function) creates a fully-connected graph and propagates the label based on the distance between the labeled samples and the non-labeled ones. Here, the SIFT descriptor is used for comparing the neighbourhood between symbols. Empirically, the *knn* kernel provides better results and a lower execution time, so we have chosen the *knn* configuration and $k = 11$. Figure 3 shows a representation of the *knn* configuration for $k = 3$ (although there are exceptions, such as the symbol 1, with 4 neighbours).

The second parameter is *alpha* (value between 0 and 1), which defines the changeability or chameleon-like degree (in other words, how easy is to change the label of a symbol). If *alpha* is low (close to 0), symbols with an assigned label remain unchangeable, no matter if their neighbours have a different label. Contrary, if *alpha* is close to 1, symbols with already assigned labels can easily change, based on their neighbours' labels (a higher changeability degree).

This algorithm is repeated until convergence. The output of the algorithm can be a hard-assignment or a soft-assignment of label probabilities. For a better control of the assignation of labels, we have chosen the soft-assignment. We have used the label propagation implementation from the Scikit-learn library [13].

3.3 Transcription

Once the labels have been propagated, we must obtain the transcription for each line in the manuscript. So, for each line, we output the final label of each symbol within that line, from left to right.

As we have chosen the soft-assignment as the output of the label propagation, all symbols (except the seeds) will have a vector of probabilities between 0 and 1. Therefore, in this last step we have set a confidence threshold for determining which is the final label for each symbol. So, for each symbol, if its most probable label has a probability higher than a given confidence threshold, then, we will assign that label to that symbol. Otherwise, that symbol will be labelled as unknown (the label * is used for symbols without a consensus, and therefore they will not be transcribed). This decision is based on the fact that users prefer symbols without labels rather than symbols with wrong labels. However, please note that, although a high confidence threshold ensures lower incorrect transcriptions, the amount of unknown symbols increases.

4 EXPERIMENTS

We test the proposed method on different ciphers, and discuss the advantages and disadvantages compared to a supervised method.

4.1 Datasets

To analyze the performance and generalization degree of our unsupervised method, we have chosen three cipher types from the DECODE database[1]. They are from various time periods and countries, with different symbol sets and handwriting styles, all with freely available transcriptions or transliterations.

The Borg cipher[2] is a 408 pages manuscript, probably from the 17th century, located at the Biblioteca Apostolica Vaticana, called MSS-Borg.lat.898[3]. The cipher consists of 34 different characters, comprising all from abstract, esoteric symbols to Roman letters, and some diacritics. Word boundaries are marked with space. Almost the entire manuscript is encoded with the exception of the first and last two pages, and some headings in Latin that are found in the first part of the manuscript. The cipher has been transcribed, transliterated and deciphered [1] and brought to light a text in Latin (and partly Italian). An extract from this cipher is shown in Fig. 1.

The Copiale cipher[4], dated back to 1730-1760, is a 105 pages manuscript containing 99 different symbols, comprising all from

[1] https://stp.lingfil.uu.se/decode/database
[2] https://stp.lingfil.uu.se/~bea/borg/
[3] https://digi.vatlib.it/view/MSS_Borg.lat.898
[4] https://stp.lingfil.uu.se/~bea/copiale/

Roman and Greek letters, to diacritics and abstract symbols, represented as *scriptio continua*. The manuscript has been transcribed, decrypted and translated [12] and revealed a German text about a secret society. An extract is shown in the third example in Fig. 1.

The third type of cipher, henceforth the Numerical cipher, is encrypted with digits (0-9), where accents or ornaments may appear above or bellow digits. The ciphers, from the Secret Archives of the Vatican, contain letter correspondences between the Vatican and France and Spain during 1573 and 1736 [4] (see Fig. 1).

Noteworthy that, although cleartext sequences (i.e., not encoded words written in the original language) may appear in these ciphers, they are not recognized in our experiments (the recognition of handwritten text is out of the scope of this work).

4.2 Evaluation

We use the Symbol Error Rate (SER) metric, which is based on the well-known Character Error Rate (CER) used in text recognition. The SER is defined as the minimum number of edit operations to convert the system's output into the ground-truth one. Formally:

$$SER = \frac{S + D + I}{N}, \quad (1)$$

where S is the number of substitutions, D of deletions, I of insertions and N the ground-truth's length. The lower the SER, the better.

4.3 Results

Here, we analyze the performance of our method on the three cipher types and for different parameters. Since our method is unsupervised, we do not need any pages/lines for training. Thus, we provide the Symbol Error Rate (SER) for all pages in each cipher.

Figure 4 shows the results for the Copiale cipher. The 3 coloured bars (yellow, orange and red) show the SER (0-1) for different confidence threshold (0.4, 0.6, and 0.8 respectively), alpha (the changeability degree of nodes during label propagation) and different number of clusters (50 or 150 respectively). Here, a higher confidence threshold means that the system will only transcribe a symbol if the confidence is high, otherwise, the symbol will not be transcribed. For illustrating the effects of this parameter, the grey bars within the coloured bars show the percentage of missing symbols in each scenario. The green doted line shows the percentage of symbols in the cipher alphabet that are covered when selecting 50 or 150 clusters as seeds in the label propagation.

We observe that the higher the confidence threshold, the lower the SER (i.e., higher performance). However, the percentage of missing symbols increases because the confidence threshold is more restrictive. The number of clusters that are used as seeds in the label propagation is also important because 150 clusters provide better results. The reason is that the Copiale cipher is composed of 99 different symbols, so, propagating the labels of 50 clusters barely covers the symbol alphabet in this cipher. Indeed, since there are some symbols that are more frequent than others, the top 150 clusters contain slight variations of the same glyph. As a result, 150 clusters do not cover all the symbols in this alphabet. Besides, we can observe that a higher alpha (which means that the chameleon-like degree is higher) helps when there are more clusters.

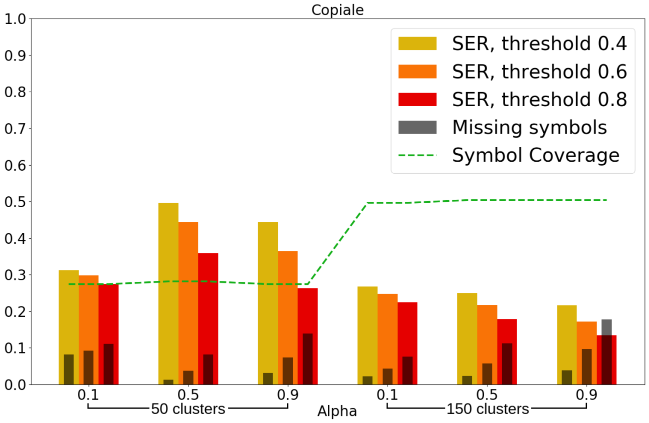


Figure 4: Results for the Copiale cipher.

Figure 5 shows the results for the Borg cipher. Compared to the Copiale cipher, the SER is worse due to the higher number of touching symbols in the manuscript. We could improve the performance by setting a more restrictive confidence threshold value, but, as a consequence, the amount of missing symbols would increase. In general, a higher alpha fosters that the symbols' labels in a cluster change, thereby obtaining better results. The reason is that clusters are rather noisy due to segmentation errors. Furthermore, we observe that using 50 or 150 clusters as seeds barely affect the performance (SER is quite similar), although the symbol coverage increases. Our assumption is that the amount of clusters makes little difference because, although the Borg cipher contains 34 different symbols, the frequency of those symbols is unbalanced. Indeed, a simple analysis of symbol frequency shows that the 15 most frequent symbols appear in more than 80% of cases.

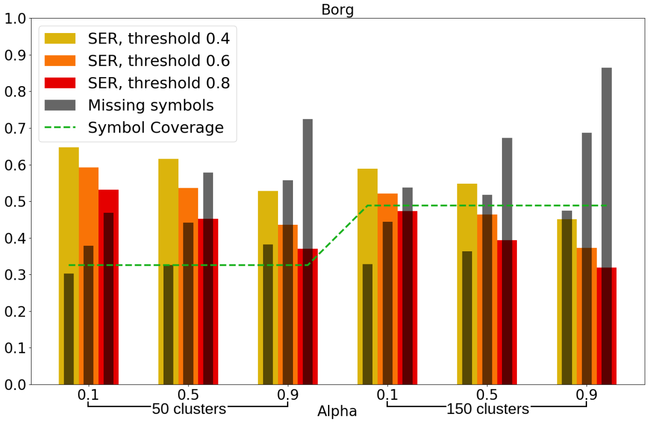


Figure 5: Results for the Borg cipher.

Finally, Figure 6 shows the results for the Numerical ciphers. Since there are 5 different writers, we have clustered and transcribed each handwriting style separately. For this reason, we show the SER for each writer. We observe that, in most cases, more clusters result in better SER. As in the Borg cipher, the higher the alpha and confidence threshold, the better the performance. However, the number of missing symbols significantly increases.

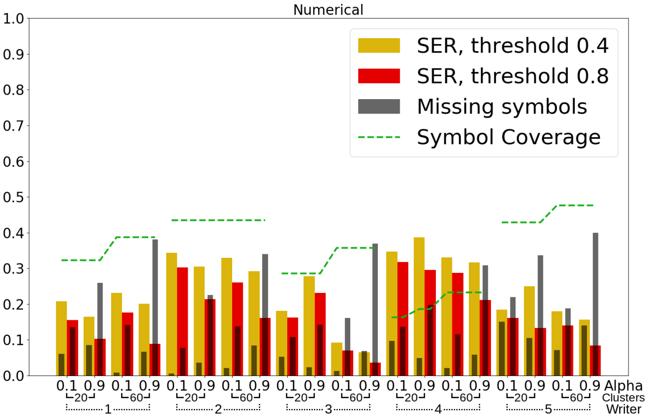
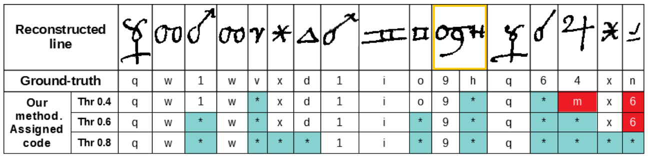
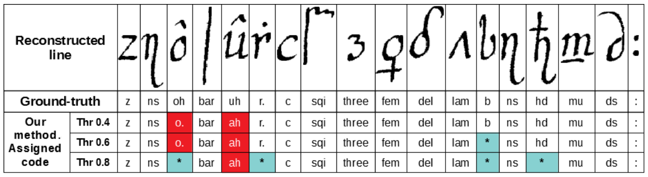


Figure 6: Results for the Numerical cipher.



Reconstructed line																		
Ground-truth		q	w	1	w	v	x	d	1	i	o	9	h	q	6	4	x	n
Our method. Assigned code	Thr 0.4	q	w	1	w	*	x	d	1	i	o	9	*	q	*	m	x	6
	Thr 0.6	q	w	*	w	*	x	d	1	i	*	9	*	q	*	*	x	6
	Thr 0.8	q	w	*	w	*	*	*	1	i	*	9	*	q	*	*	*	*

Figure 7: Borg. Transcription example.



Reconstructed line																			
Ground-truth		z	ns	oh	bar	uh	r.	c	sqi	three	fem	del	lam	b	ns	hd	mu	ds	:
Our method. Assigned code	Thr 0.4	z	ns	o.	bar	ah	r.	c	sqi	three	fem	del	lam	b	ns	hd	mu	ds	:
	Thr 0.6	z	ns	o.	bar	ah	r.	c	sqi	three	fem	del	lam	*	ns	hd	mu	ds	:
	Thr 0.8	z	ns	*	bar	ah	*	c	sqi	three	fem	del	lam	*	ns	*	mu	ds	:

Figure 8: Copiale. Transcription example.

Some examples of the output transcriptions are shown in Figures 7 and 8. The (unlabelled) missing symbols are shown in blue color, and the incorrect symbols in red color. Obviously, a higher confidence threshold results in more missing symbols. This aspect is good when the symbol was incorrectly recognized (see the third symbol in the Copiale example), but it also prevents that correctly labelled symbols are finally transcribed (see the third symbol in the Borg example). We can also observe that errors in the segmentation highly affect the clustering and thereby the final transcription (see the symbol highlighted in yellow color in the Borg example).

4.4 Comparison to a Supervised Method

We compare our method to a supervised HTR method based on Multi-Dimensional Long Short-Term Memory Recurrent Neural Networks (MDLSTM), used for numerical ciphers [10]. We discuss the advantages and disadvantages of both approaches. Table 1 contains information about the cipher images that have been used for training (including validation) and test, for the supervised approach. Data with detailed description is available [5]. For a fair comparison, we run our unsupervised method and compute the SER on the same test images that were used in [10].

Dataset	Symbols	# Classes	# Writers	# Pages Train Set	# Pages Test Set
Borg	Mixture	34	1	14	16
Copiale	Mixture	99	1	51	52
Numerical	Numbers with ornaments	31	5	15	14

Table 1: Training and test sets.

Tables 2 and 3 show the comparison between the methods applied to the three types of ciphers. In both approaches, the higher the confidence threshold, the better the results, but the percentage of missing symbols increases. This means that the user must manually transcribe all these non-transcribed symbols.

Besides the SER values, we also pay attention to the required user effort for generating labeled data needed for training. In the case of the supervised approach (MDLSTM), an expert user must transcribe different amount of pages, concretely, the 26%, 34%, 42% and 50% of the full cipher (e.g. for Copiale, it means transcribing 25, 34, 42 or 51 pages, respectively). Not surprisingly, the more training (i.e. required user effort), the better results indicated by lower SER values and lower percentage of missing symbols.

In order to analyze the user intervention in depth, we consider two different scenarios for our unsupervised method: in the first case, there is no user intervention (in the Table: user intervention = *None*), as explained in the previous subsection. In the second case (Ours, user intervention = *Select Clusters*), the user must select one cluster for each glyph given the cipher's symbol set (e.g. for Borg, the user should select 34 clusters given the 34 glyphs). The user should also remove outliers in these clusters. An outlier is a symbol that has been incorrectly classified into a certain cluster. This ensures two important aspects: first, there will be one seed for each glyph in the symbol set (so, the symbol coverage is 100%), and second, all the elements used as seeds will have a correct label during the label propagation step. We have asked a non-expert user to do these tasks, because, as stated in [6], an expert is not required for tasks related to shape similarity. The selection of clusters takes about 1h for each cipher, so the user effort is much lower than transcribing several pages (required for the MDLSTM).

From the results, we observe that, in the unsupervised method, if we compare the two scenarios, a low user intervention (i.e. select clusters) means a better SER, but more importantly, the percentage of missing symbols significantly decreases. This scenario could be comparable to the MDLSTM with 26% labelled data, where the MDLSTM results are significantly worse, both in terms of SER and missing symbols. Indeed, our unsupervised method with low intervention obtains results very similar to the MDLSTM when using 50% of labelled data. Taking into account that the human effort is significantly lower in our approach, we could assume that this scenario is most suitable for users.

5 CONCLUSION AND FUTURE WORK

We proposed an unsupervised method to transcribe ciphers with various kinds of symbol sets. It can be automatically applied to the

[5] https://cl.lingfil.uu.se/decode/publ.html

Method	User Intervention	Threshold	SER	Missing Symbols
Ours	None Select Clusters	0.4	0.275 0.189	0.053 0.013
Ours	None Select Clusters	0.6	0.225 0.167	0.137 0.033
Ours	None Select Clusters	0.8	0.197 0.146	0.231 0.056
MDLSTM	Label 50% of data	-	0.12	-

Table 2: Comparative results on the Numerical cipher: Our unsupervised vs the supervised MDLSTM method [10]. In our method, the user is requested to select one cluster for each symbol (Select Clusters). In the MDLSTM, the user transcribes the 50% of the cipher (15 pages).

Method	User Intervention	Confidence Threshold	Borg		Copiale	
			SER	Missing Symbols	SER	Missing Symbols
Ours	None	0.4	0.542	0.377	0.444	0.031
	Select Clusters		0.522	0.173	0.201	0.010
MDLSTM	Label 26% of data	0.4	0.715	0.154	0.131	0.008
	Label 34% of data		0.662	0.069	0.120	0.009
	Label 42% of data		0.693	0.061	0.084	0.006
	Label 50% of data		0.556	0.035	0.075	0.003
Ours	None	0.6	0.445	0.547	0.365	0.073
	Select Clusters		0.464	0.282	0.173	0.024
MDLSTM	Label 26% of data	0.6	0.554	0.399	0.113	0.068
	Label 34% of data		0.523	0.280	0.109	0.076
	Label 42% of data		0.551	0.269	0.078	0.048
	Label 50% of data		0.450	0.212	0.074	0.038
Ours	None	0.8	0.377	0.713	0.264	0.138
	Select Clusters		0.418	0.385	0.144	0.045
MDLSTM	Label 26% of data	0.8	0.546	0.513	0.110	0.154
	Label 34% of data		0.437	0.435	0.111	0.170
	Label 42% of data		0.465	0.421	0.087	0.116
	Label 50% of data		0.365	0.365	0.081	0.098

Table 3: Comparative results on the Borg and Copiale ciphers: unsupervised vs supervised MDLSTM method [10]. The results are given with different threshold and user intervention. In our method, the user is requested to select one cluster for each symbol (Select Clusters). In the MDLSTM, the user labels a percentage of pages of the full cipher. The size of Copiale is 103 pages and Borg is 30 pages.

whole cipher, avoiding to ask the user to transcribe any pages to train. In addition, the method can help the user to classify the symbol set of the manuscript, based on glyph recognition. We observe that the obtained results (up to 62.7% correct symbol classification) are promising taking into account that the user effort is zero. However, if there is a low user intervention (i.e. the user selects one cluster for each symbol in the alphabet), the results are similar (or even better) than the ones obtained by a supervised method based on MDLSTMs with a high amount of labelled training data.

Since we aimed to explore the suitability of unsupervised methods, we have used standard techniques for clustering and label propagation. Given the encouraging results, we will focus on more powerful techniques for segmentation, able to deal with touching elements, such as [8]. Also, we will explore zero-shot learning and domain adaptation techniques for improving the classification of ciphers with new type of symbol sets. The combination of supervised and unsupervised methods, as well as interactive transcription methods are also promising research directions.

Finally, we are developing a web-based platform so that scholars can upload their ciphers, run the transcription method and correct the output transcriptions in a user-friendly interface. Our long-term hope is to provide a tool where unsupervised HTR methods can be used for transcribing historical ciphers of any symbol set, minimizing the human intervention.

ACKNOWLEDGMENTS

This work has been partially supported by the Swedish Research Council, grants E0067801 and 2018-06074, the Spanish projects TIN2015-70924-C2-2-R and RTI2018-095645-B-C21, the Ramon y Cajal Fellowship RYC-2014-16831, the CERCA Program/ Generalitat de Catalunya, and the FI fellowship AGAUR 2019 FI_B1 00196 (with the support of the Secretaria d'Universitats i Recerca of the Generalitat de Catalunya and the Fons Social Europeu).

REFERENCES

[1] Nada Aldarrab. 2017. *Decipherment of historical manuscripts.* Master's thesis. Master's thesis, University of Southern California.

[2] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. 2014. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence* 36, 12 (2014), 2552–2566.

[3] Serena Ammirati, Donatella Firmani, Marco Maiorino, Paolo Merialdo, Elena Nieddu, and Andrea Rossi. 2017. In codice ratio: Scalable transcription of historical handwritten documents. In *Italian Symposium on Advanced Database Systems.*

[4] Archivio Segreto Vaticano ASV. 2016. ASV16:Segr.di.Stato/Francia/ and Spagna Archivio Segreto Vaticano. All rights reserved.

[5] Taylor Berg-Kirkpatrick, Greg Durrett, and Dan Klein. 2013. Unsupervised transcription of historical documents. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Vol. 1. 207–217.

[6] 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).* IEEE, 528–533.

[7] Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. 2005. Efficient Non-Parametric Function Induction in Semi-Supervised Learning. In *AISTATS.*

[8] David Fernández-Mota, Josep Lladós, and Alicia Fornés. 2014. A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition* 17, 3 (2014), 293–312.

[9] Donatella Firmani, Marco Maiorino, Paolo Merialdo, and Elena Nieddu. 2018. Towards Knowledge Discovery from the Vatican Secret Archives. In Codice Ratio - Episode 1: Machine Transcription of the Manuscripts. In *ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining.* 263–272.

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

[11] Volkmar Frinken and Horst Bunke. 2014. Continuous handwritten script recognition. In *Handbook of Document Image Processing and Recognition.* Springer, 391–425.

[12] Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2012. The Secrets of the Copiale Cipher. In *Journal for Research into Freemasonry and Fraternalism.*

[13] Scikit learn library for Python. [n. d.]. Semi-supervised. http://scikit-learn.org/stable/modules/label_propagation.html (last entry: 15/01/2019).

[14] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E* 76, 3 (2007), 036106.

[15] Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. 2018. Decipherment of Historical Manuscript Images. *CoRR* (2018).