Direct Decipherment and Transcription of Historical Handwritten Ciphered Document Images

Marina Bermúdez Granados

July 1, 2024

Abstract– There are many historical ciphered documents that are still preserved in libraries worldwide, their content still unknown to researchers even after all this time. Since manual decryption is not viable, many researchers have resorted to machine learning practices. The usual techniques use a pipeline approach (first transcription, then decryption), causing a high level of dependance between tasks. The objective of this work is to propose a deep-learning model to transcribe and directly decipher these document images. First, we formed different versions of the same data set with real handwritten images and synthetic replicas. After the image generation, we performed a total of 18 experiments across tasks, data sets, and configurations. Among other findings, we concluded that our model obtains different results depending on the task at hand, despite working with the same data sets and parameters.

Keywords– Decipherment, Historical manuscripts, Image generation, Sequence-to-sequence model, Transcription

Resum– Hi ha molts documents històrics xifrats conservats a biblioteques arreu del món amb continguts encara desconeguts pels investigadors. Com el desxifrat manual no és una opció viable, molts investigadors han acudit a mètodes d'aprenentatge automàtic. Les tècniques habituals utilitzen un enfocament de pipeline (primer transcripció, després desxifrat), provocant una alta dependència entre les tasques. L'objectiu d'aquest treball és proposar un model d'aprenentatge profund per transcriure i desxifrar directament imatges d'aquests documents. Primer es van formar diferents versions del mateix conjunt de dades, imatges de documents reals i rèpliques sintètiques. Després de la generació d'imatges, es va realitzar un total de 18 experiments entre tasques, conjunts de dades i configuracions. Entre altres descobriments, es va concloure que el nostre model obté resultats depenent de les tasques, malgrat treballar amb els mateixos conjunts de dades i paràmetres.

Paraules clau– Desencriptació, Manuscrits històrics, Generació d'imatges, Models sequence-to-sequence, Transcripció

1 Introduction

THROUGHOUT the history of humanity, there has always been the need to hide sensitive information. Many encrypted manuscripts have been collected and preserved in different libraries and archives, their content still unknown to researchers even after all this time. From diplomatic correspondence and intelligence reports to private diaries and secret societies, the details of these manuscripts could aid in historical investigations. Manual methods to decipher classical cryptographic algorithms have been inefficient and time-consuming, which is why the automation or semi-automation of these processes has been the objective of computer scientists interested in historical cryptography. There have been a couple of initiatives to develop techniques to ease the large-scale deciphering process for some ciphers. For instance, the DECRYPT project [1] has developed resources and tools to analyse and decode encrypted manuscripts.

- Major: Computing
- Tutor: Alicia Fornés Bisquerra
- Course 2023/24

The main topic of this bachelor's thesis will be the direct decipherment of historical documents by proposing a joint end-to-end approach. Most of the proposed deciphering methods rely on pipeline structures where the image is transcribed, analysed and deciphered. This approach causes a high degree of dependance between all the stages; the mistakes from the previous phases concatenate into the next one. However, there are not many studies on the possibility of directly deciphering these images. We will explore their direct decipherment by first examining the transcription of historical documents with the intention of comparing their performances.

2 Objectives

The main objective of this bachelor's thesis is to study the direct decryption and transcription of images containing ciphered documents from historical contexts. Before tackling these tasks, a study of the current state of the art will be conducted, and a suitable data set will be assembled. The data set will be formed with real handwritten images of lines from historical manuscripts alongside artificial images meant to mimic the real ones. We will develop methods and functions to generate these synthetic images. Once the main architecture is developed, the model's performance will be tested and evaluated accordingly.

In conclusion, the objectives of this bachelor's thesis can be summarised into five different ambitions:

1. The study of the current methods and techniques in the state of the art.
2. The generation of images containing synthetic ciphertext.
3. The transcription of images containing ciphered text.
4. The direct decipherment of images containing ciphered text.
5. The evaluation and assessment of the implemented models.

3 State of the Art

The decipherment of historical manuscripts usually starts with a transcription of the document image, i.e. from the real-life cipher text contained in a scanned document image to a computer-readable format. In other words, the task is to save the encrypted text in a format that can be used in the following steps. These handwritten ciphers are often written in an unknown alphabet, which also needs to be identified. Other common challenges in the transcription process include handwritten styles, irregularities, alignment imperfections, and deteriorations in the paper. There are many methods that have been successfully applied for automatic transcription, like Recurrent Neural Networks with manual post-correction [2], Siamese Neural Networks with a Gaussian mixture model [3], and other unsupervised models [4-6], among others.

When the transcription has been completed, the ciphertext needs to be deciphered into plaintext. The typical ciphers used in historical documents are either substitution ciphers, transposition ciphers, or a combination of those. The first one substitutes each character for another one according to a key, a substitution table. They can do the replacements one letter per one letter or one letter per multiple ones: monoalphabetic ciphers and homophonic ciphers, respectively. The second kind rearranges the characters of the plaintext.

Since the decryption of ciphers is a complex matter, researchers have found multiple methods to decode them. The chosen strategy may vary depending on each case, too. For instance, there are some cases where some cleartext could be found along the encrypted text. Other times, there are decoded sections within the text that could help crack the cipher. These cases are commonly referred to as known-plaintext attacks. By contrast, ciphertext-only attacks only have access to the ciphertext. The cryptanalysts usually have to detect the cipher type and the language behind the ciphertext, before attempting to find the key. Our case would fall under the ciphertext-only attack, but directly from scanned images. In other words, we aim to directly decipher the images without knowing the plaintext language, the key or the method of encryption.

Some studies view the automatic decipherment of encrypted texts as a natural language processing problem. NLP is a subfield of artificial intelligence that utilises machine learning techniques to manipulate, understand and manage human language. Consequently, these methods are used in a variety of language-related tasks, such as machine translation, text summarization or speech recognition. Although they do not specialise in the decipherment of historical manuscripts, the problem can be framed as a translation task from an encoded language into a decoded one. In particular, studies such as [8-9] consider decipherment as a sequence-to-sequence translation task. Thus, she used a sequence-to-sequence model due to its resistance to noise and its ability to be trained with multilingual data. However, researchers tend to apply a frequency encoding of the ciphertext [7-9] instead of the images. Earlier methods relied on frequency analysis as well, measuring the occurrences of specific letters in the ciphertext in order to discern the cipher type and its properties. Other studies have proposed decipherment through different models and pipelines, including [3] with a three-stage decipherment pipeline using scanned images and a two-step decipherment model with the integration of transcription and decipherment. One-stage decipherment is suggested for future work.

4 Methodology and Planning

The completion of the set objectives depends on the execution of three key phases: The image generation, the experiments on transcription tasks and deciphering tasks. The first step is the collection of a complete data set through image generation for both tasks. There are precisely three different versions for the same data set: One with the synthetic images, one with the handwritten equivalents, and another with a combination of both synthetic and real data. From there, we can begin with the execution of experiments on a transcription task with the synthetic images. Once we ensure that the model can recognise them, we proceed with the direct decipherment tasks and the other data sets. We have determined from the state of the art that the most suitable

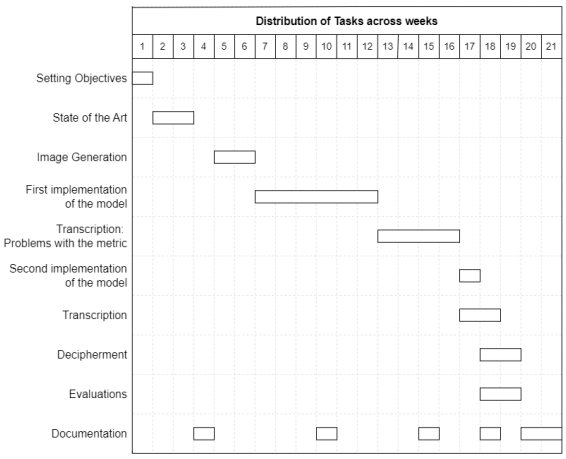


Fig. 1: Gantt Diagram with the tasks completed per week

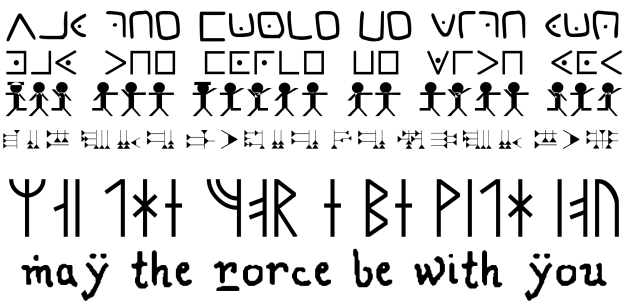


Fig. 2: Synthetic images containing lines from different cipher fonts. From top to bottom: Masonic cipher, Pig pen cipher, Stickman cipher, Keil font cipher, Modern runic and Copiale cipher.

technique to make the predictions would be a sequence-to-sequence one. Lastly, we assess the different configurations and parameters per task and gather the corresponding conclusions from the outcomes.

The planning initially followed the deliveries and the completion of these key steps. All the tasks were mainly distributed in two blocks of work: The first one included the image generation and the implementation of the model, while the second block included the different experiments and evaluations. The time dedicated to each task can be found on Fig. 1.During the early experiments on the transcription task, it was noted that the model had the tendency to predict values that did not exist within the set vocabulary. For instance, if the vocabulary covered glyphs from 0 to 132, the sequence-to-sequence model would sometimes predict values above 132. At the time, there were three different options to handle this issue: Ignore the indices outside the vocabulary file, substitute them for an error character, or reduce the output units of the model. All the experiments were able to eventually reach adequate scores, and the losses indicated that the model could adjust appropriately to the sets. However, that implementation was unreliable, time-consuming and highly dependent on our labels for the glyphs. When the calculations were modified to use the indices directly, the model could no longer learn. Hence, we had to change the model's implementation entirely.

5 The Image Generation

The first step in order to achieve all our objectives is to prepare the data set of images for the models. The real and synthetic images can be separated into two groups: The images of single lines and the images of manuscripts, meaning multiple lines within one image. Regardless of the group, multiple examples will be created with six different cipher alphabets. Two of them are the masonic cipher and the pigpen cipher, two variations of the same substitution cipher that was used to deliver correspondence during the 18th century. The next cipher is a version of the dancing stick man cipher, a fabricated substitution cipher featured in one of the stories of Sherlock Holmes from December 1903. The fourth one is the keil font cipher, another substitution cipher, and the fifth cipher is modern runic, taking advantage of runic alphabets to substitute Latin letters. The last one is the Copiale cipher [11], a homophonic substitution cipher found in a fully encrypted manuscript. Its original text describes the initiation process for a secret society in German. Compared to the rest of ciphers, plaintext characters may map multiple glyphs. In other words, one glyph can decode one plaintext character the same way multiple glyphs may decode it as well. Examples from all six different cipher alphabets can be inspected in Fig. 2, without any applied encryption of any kind. All the synthetic images were generated from either randomly generated strings or files with extracts from books, short stories and quotes

The image generation derives from three Python files that were previously designed to output different images containing texts with specific alphabets. First, the content of the image is either generated or read from a file. Then, its size within the image is calculated with the desired font and size. From there, a blank image is created according to the prior measurements and the text is copied into it. The last step is to add an extra white border before saving the results. When there are multiple lines of text, two extra steps are added to justify the text and filter out missing characters from the cipher fonts.

The cipher alphabets are saved as fonts within files with ttf extensions, also known as True Type Format. These extensions assign a glyph to every character input. In the case of ciphers such as masonic, pigpen or stickman, all the unaccented Latin letters plus some punctuation signs are automatically assigned to their corresponding characters. In addition, the keil font cipher also admits numerical values. Regarding the modern runic cipher, not all the characters have a glyph assigned to them. For instance, there is no translation for the uppercase letters of *X* and *Z*, and for the lowercase letters of *c*, *x* and *z*. The reason behind the missing letters originates from the fact that some of the older Germanic languages were already using variations of the runic alphabet before adopting the Latin alphabet. In other words, some letters from the Latin alphabet do not exist in runic alphabets, the same way letters like *ñ* in Spanish or *ç* in Catalan do not exist in English. Finally, the Copiale cipher [11] uses a mix of unaccented Roman letters with some alternative variants, some Greek letters and some abstract symbols. All these glyphs are assigned randomly to characters within the font file. After further analysing it with the original Copiale alphabet, there were five missing symbols that needed to be drawn and added manually to the

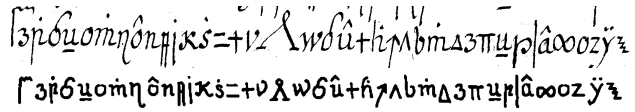


Fig. 3: Synthetic image containing a Copiale line (bottom), resembling a real handwritten image (top).

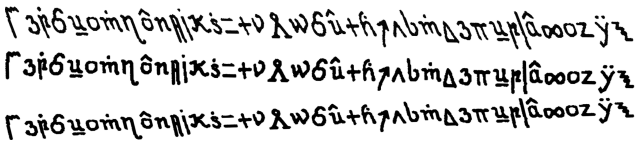


Fig. 4: Three randomised augmentations over the synthetic image from Fig. 3.

existing font file: The star, the smiling faces facing left and right, the cloud, and the gate.

Aside from the previously generated images, another set of synthetic images will be produced with the goal of resembling real handwritten images containing Copiale lines. We already had access to a database consisting of individual lines from the original book, which was initially used for a Handwritten Recognition task in a competition. The transcriptions had to be adapted to the correct input characters from the font file in order to properly show the correct glyphs. This was achieved through Python dictionaries between the transcriptions and the set inputs. The end results can be examined in Fig. 3.

Among all of the presented ciphers, the Copiale cipher will be the centre of our experiments due to the availability of handwritten transcriptions. However, the deciphered data had to be found among the 105 pages of the original book. In other words, we only had our own transcriptions of some lines and the entire transcribed and deciphered book. The original ground truths from the book were divided by lines and each transcription could easily be matched with its corresponding plaintext. Nonetheless, our version of the transcriptions used different terminologies. For instance, the book refers to the star symbol as *star*, while our transcriptions refer to it as *Pentagram*. Thus, we first had to adapt the transcriptions from the book to our own transcriptions, so as to properly match them to their deciphered equivalents. Although not all the translations were perfect due to some exceptions, the majority of lines perfectly matched and the deciphered lines could be assigned accordingly. The remaining cases could also be assigned by only matching the beginning or end of each sample.

Since using the exact same figures would be considerably straightforward for any model, we had to apply an augmentation to the synthetic images. A Python class has been adapted to randomly morph any image. The augmentations include erosions, dilatations, additions of noise, gamma corrections, shearings, rotations and scalings. The entire process is controlled by a list of parameters to ensure that the content will still be readable. Some of the results can be found in Fig. 4.

6 The Sequence-to-Sequence Model

A sequence-to-sequence model is an approach with an encoder-decoder structure typically used to solve sequence

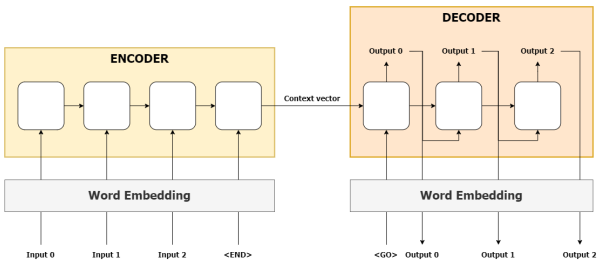


Fig. 5: Simplified Encoder-Decoder Architecture from a sequence-to-sequence model.

modelling problems, where both the input and output are sequences. The encoder-decoder architecture consists of two components: The encoder processes each token within the input into a context vector that is received by the decoder, which makes predictions token by token. A summary of the entire architecture can be found in Fig. 5. The inputs and outputs need a word embedding before being processed.

Both structures are composed of multiple long short-term memory networks, a type of recurrent neural network designed to retain information for longer periods of time. Each network within the encoder reads a token from the original sequence and sends their insights into the next one. The final state receives all the internal data from past cells and a special token that conveys the ending of the sequence, usually represented as $<EOS>$ or $<END>$. Its output is known as the context vector, which encodes all the information from the source. The initial cell from the decoder receives this vector and makes the first prediction, along with another special token that indicates the start of the output sequence. They often appear in documentation as $<BOS>$ or $<GO>$. The subsequent predictions will use the last output and the internal information from the previous network.

The predictions within the model are usually evaluated through a cross-entropy loss function. Commonly used in classification problems, they are able to measure the differences between an estimated probability and the desired outcome. During the learning process, the decoder obtains a probability distribution for the predictions of the next token. Then, the loss function compares the token with the highest probability to the correct output. The corresponding loss is then evaluated with the intention of properly updating the probabilities. The probability for the correct output is maximised, while the others are minimised.

Although they are very effective models for tasks like translation, text generation and language modelling, they may face difficulties handling long sequences. The encoder may not be capable of properly encapsulating all the relevant information, while the decoder may need different key insights at different stages. However, an attention mechanism can be introduced to enable the decoder to select the most important insights. Before every prediction, the decoder needs to pass its internal state to an attention function. Its task is to measure the relevancy of each encoder state through scalar values. Then, a softmax function transforms all these results into a probability distribution. At the end, these probabilities and internal states are combined into a weighted sum for the decoder to use as a new context vector. The next tokens are predicted as usual, but using the most important parts from the encoders. An example of the

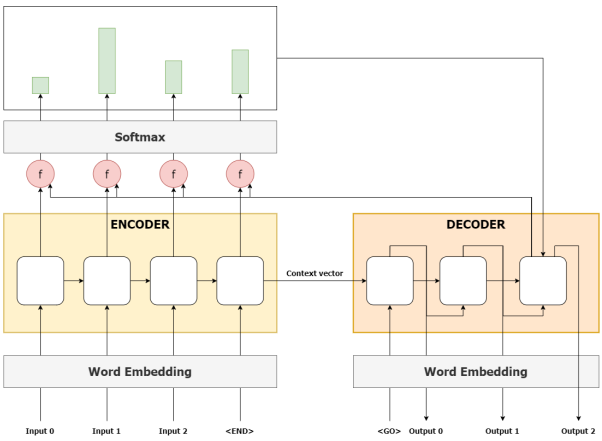


Fig. 6: The Encoder-Decoder Architecture applying an Attention mechanism to the decoder's last internal state.

encoder-decoder structure with an attention mechanism can be found in Fig. 6.

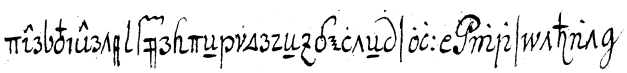
There are many types of attention mechanisms, depending on how they measure the relevance of the encoder states. Each method has its advantages, disadvantages and common use cases. For instance, content-based attention measures attention scores by computing the cosine similarity between the internal states of the encoder and decoders. As the name indicates, the content of a section is what indicates its relevance in this approach. They are widely adopted in a myriad of applications, including natural language processing, computer vision and speech recognition. However, content-based attention does not explicitly consider positional information in its calculations. Since it could be beneficial when working with visual data, we will use a similar mechanism known as location-based attention. In this case, relevance is also placed in the locations within the image.

Depending on the task at hand, the inputs and outputs will vary in type and length. In other words, the data does not necessarily have to be formed only by sequences, regardless of the name of the model. For instance, an image captioning task takes one input in the form of an image and many outputs in the form of a sequence of words. Our transcription and decipherment tasks also have a one-to-many approach, with one image as the input and a sentence as the output, either the corresponding transcription or the decoded message.

6.1 Specifications for Transcription

The idea of the transcription task is to accurately transcribe all the glyphs from the images. The inputs are the images containing the Copiale lines, while the outputs are their respective transcriptions. Each visual glyph directly corresponds to a single label in the records. There is an example in Fig. 7 of an image with its transcription. Furthermore, the predictions will need a vocabulary of terms for the probability distributions. In this instance, the vocabulary is formed by a list of all the labels used in the ground truths. Hence, the predictions will be completed symbol by symbol.

The code implementation of our model employs Json files to configure its inner variables and structures, including the process of loading the data. Consequently, all the



SmallPi i^^ 3 b Earth SmallIota u^^ 3 CapitalLambda LatinSmallLigatureFi l CapitalGamma LatinLongLigatureFi 3 h SmallPi u__ p v Fire 3 z u__ ScriptSmallZ SmallDelta SleepingSymbol c^. CapitalLambda u__ InsularD VerticalLine o^. c^. : e BigP m^. p^. VerticalLine w CapitalLambda Saturn n^. CapitalLambda ScriptSmallG

der vierte für deN ordeN gelteN soll was that

Fig. 7: Examples of the possible outputs. The first row is a real handwritten image. The following paragraph is the equivalent transcription, while the last row is its corresponding plaintext.

text files need to be converted into separate Json files to form the training, testing and validation sets. The vocabulary file will store and index unique values within the ground truths. Thus, each line of text can be embedded into a list of indices for the model to predict.

6.2 Specifications for Decipherment

During the deciphering tasks, the goal is to decipher the images into their original contents. The inputs are still the same, but now the outputs are plaintext. Since the Copiale cipher is a homophonic cipher, each glyph may or may not correspond to a specific letter. The sequence-to-sequence models already accommodate length disparities between inputs and outputs, especially when we view a deciphering task as a translation from an encoded language into a decoded one. It is common in machine translation to have equivalent sentences with different lengths. An example of this situation can be inspected in Fig. 7. As can be seen in the last line, the deciphered text is much shorter than the glyphs in the image. Unlike other papers, our approach skips the transcriptions and predicts the plaintext directly. The predictions will still need a vocabulary of terms, which will be formed by all the letters within the German alphabet. Aside from the outputs and lexicon, there are no other technical differences between tasks.

6.3 Evaluation

The Character Error Rate (CER) metric measures the percentage of character-level mistakes in a prediction compared to its ground truth. Consequently, it is considered a minimising metric; the best results are the ones closer to zero percent. It is calculated by dividing the sum of substitutions (S), deletions (D), and insertions (I) by the total number of characters (N) in the ground truth sample. A substitution occurs when a character seems to have been replaced, while deletions and insertions are recognised through characters that appear to be either missing or added. The lower the scores, the better the performance. The scores of our implementation range from 0 to 1.

$$CER = \frac{S + I + D}{N}$$

The performance of the evaluations will also include the losses of each set for each epoch, a metric that assesses how well the model fits the validation set. Altogether, the evaluations will include the CER scores and losses from the train, test and validation partitions.

TABLE 1: Parameter Configurations

	Layers	Batch size	Label Smoothing	Learning Rate
Configuration 0	2	16	0.4	0.0003
Configuration 1	4	16	0.4	0.00003
Configuration 2	4	12	0.2	0.00003

7 Experimental Results

The experiments can be divided by the task at hand and the type of data used. We will explore the performance of the model in a transcription and decipherment task with synthetic data, real data, and a combination of them. Both the synthetic and real datasets consist of 1502 images plus three augmentations per sample. In total, each data set has 6008 images from Copiale. The 80% of those samples will form the training set, while the rest will be divided equally for the testing and validation sets. Alternatively, the combined dataset will use 6008 synthetic images for training and 1502 real images for testing and validation.

In addition to various tasks and data sets, we will also explore three different configurations. The first set of parameters is going to be based on paper [12], which presents a sequence-to-sequence model for handwritten word recognition. The initial configuration will have 2 layers for both the encoder and decoder, 512 hidden layers, and 50% dropout with a 15% teacher rate. Their findings also suggest that label smoothing is helpful, so we will employ it as well. Taking into account the size of their data set compared to ours, we will reduce the batches from 32 to 16 samples. We will use a higher learning rate of $3 \cdot 10^{-4}$, too. For the second configuration, we will increase the number of layers to 4 while decreasing the learning rate to $3 \cdot 10^{-5}$. The last set of parameters will explore the model's performance when the batch size and label smoothing are further diminished. The summary of all these configurations can be found in Table 1.

The combination of data sets, tasks and configurations amounts to 18 different experiments to be executed. We will start with synthetic transcription in order to guarantee that the model can distinguish the synthetic images. Furthermore, each experiment will be repeated at least once so as to ensure the accuracy of the results. The rest of the experiments will be conducted. All the final results for each permutation can be found in Table 2. The first letter of the experiment's names indicates the applied task: 'T' for transcription and 'D' for decipherment. The following number determines which configuration of parameters was used. After the task and configuration, the words 'MIX', 'REAL', and 'SINT' denote which data set version was employed: The combined, the real, or the synthetic one, respectively.

7.1 Synthetic results

From all the synthetic transcriptions, the third configuration was the only one that managed to obtain suitable results. It was able to obtain a final test CER of 19% alongside a test loss of 2.38, while the other two configurations were only able to secure CER scores of 83% and 93% with somewhat higher losses. Although the results have room for improvement, they are enough to confirm that our sequence-to-sequence model can recognise the synthetic images. The CER scores overtime can be seen in Fig. 8.

TABLE 2: Experimental Results

	Duration	Last epoch	Train CER	Train Loss	Test CER	Test Loss
T0 MIX	2h 13m	61	93%	3.19	118%	4.15
T0 REAL	4h 56m	53	154%	3.75	129%	4.06
T0 SINT	16h 32m	188	81%	2.66	83%	2.74
T1 MIX	2h 17m	61	88%	3.66	147%	3.97
T1 REAL	9h 54m	104	156%	3.93	124%	4.10
T1 SINT	10h 48m	115	92%	3.72	93%	3.84
T2 MIX	2h 33m	58	87%	3.21	146 %	3.83
T2 REAL	9h 52m	90	166%	3.51	129%	3.56
T2 SINT	**10h 25m**	**103**	**21%**	**2.22**	**19%**	**2.38**
D0 MIX	2h 40m	59	104%	3.46	122%	3.53
D0 REAL	**4h 8m**	**180**	**33%**	**2.83**	**30%**	**2.98**
D0 SINT	**12h 50m**	**156**	**0.3%**	**2.35**	**1%**	**2.39**
D1 MIX	2h 19m	63	64%	3.21	121%	3.61
D1 REAL	3h 15m	138	74%	3.31	81%	3.40
D1 SINT	**13h 25m**	**153**	**14%**	**2.69**	**13%**	**2.96**
D2 MIX	2h 33m	59	61%	2.67	114%	3.65
D2 REAL	1h 58m	68	74%	2.97	80%	3.09
D2 SINT	**18h 49m**	**182**	**15%**	**1.18**	**14%**	**2.17**

Fig. 8: CER Scores from the synthetic transcription task.

The following experiments were centred around deciphering the synthetic data set. All the configurations had improvements compared to their transcription counterparts. However, the first one got the best overall metrics, with CER scores of 1% on the testing set and a test loss of 2.39. The other sets of parameters also had a drastic improvement, massively reducing their CER test scores to 13% and 14%, respectively. It seems that our model, for the same parameters, has better performance in decipherment tasks than in transcription. The scores across epochs can be inspected in Fig. 9.

The decipherment results are quite surprising, especially considering that Copiale is a homophonic substitution cipher. In homophonic ciphers, each ciphertext character stands for a particular plaintext character, but several ciphertext characters may encode the same plaintext char-

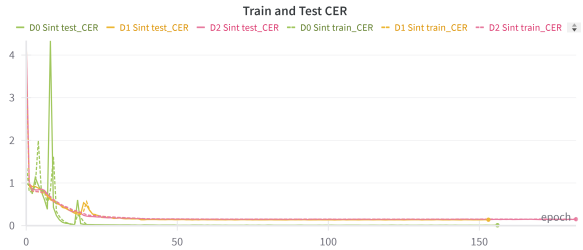


Fig. 9: CER scores from the syntethic deciphering task.

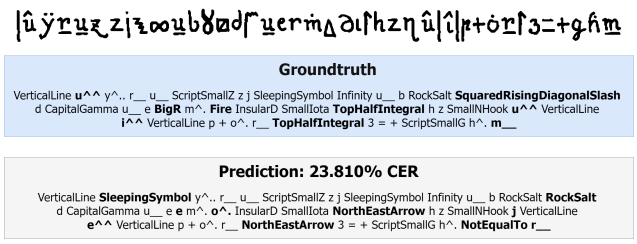


Fig. 10: Qualitative results from the synthetic transcription task using the last configuration (2).

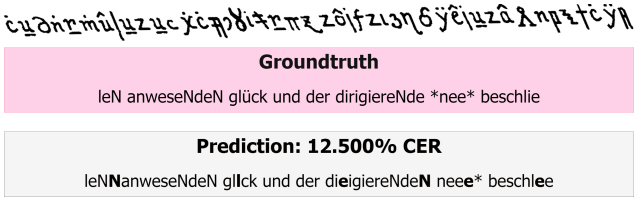


Fig. 11: Qualitative results from the synthetic decipherment task using the last configuration (2).

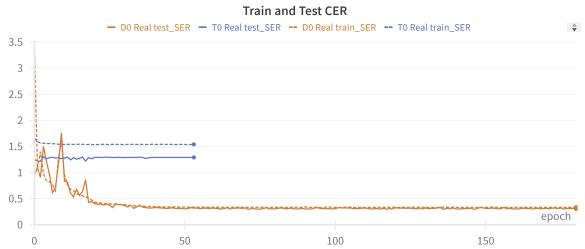


Fig. 12: Example of differences between real transcription and real decipherment tasks.

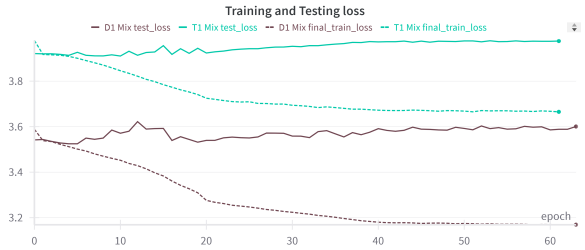


Fig. 13: Overfitting example from the experiments using the combined data set.

acter. Hence, we can infer that our sequence-to-sequence model uses the context of the sentence to predict the next values. In other words, the predictions are not character by character.

Some qualitative results can be found in Figs. 10 and 11. Both of them include a prediction from the best synthetic tasks during the last epochs. The bold sections indicate the differences between the ground truth and the prediction. There are only a few disparities, but the beginnings of all the experiments usually start with randomly selected tokens repeated multiple times. The plots show these peaks during the first epochs, until the results stabilise themselves. Most experiments follow this pattern, one way or another.

7.2 Real results

Unlike the latter results for transcription, no set of parameters was able to reach CER scores below 120%. The best results for the decipherment task originate from the first configuration, with CER scores around 30%. The other two are only capable of securing CER scores of 80% on the testing set. Once again, the experiments from the decipherment task greatly outperform the ones corresponding to the transcription task. This difference in results can be further inspected in Fig. 12, where both tasks with the initial configuration are displayed. All experiments were set to conclude their run when there had been no improvement for 50 consecutive epochs. We can see that our model has difficulties with the real handwritten transcriptions.

7.3 Mixed results

The hope for the combined data set was to train the model with the synthetic data so it could generalise the handwritten data. However, all permutations appear to have signs of overfitting due to the disparities in metrics between the training and testing sets. These differences usually range around 55%; the smallest ones derive from the first configuration. It is clear that our model cannot extrapolate the synthetic training to real testing, at least with the current arrangements. An example of the overfitting tendencies in these experiments can be found in Fig. 13, with the losses from both tasks in the second configuration. We can still observe the trend of transcription results having higher metrics than the decipherment ones.

7.4 Final Discussions

After studying all the experiments, we can conclude that our model achieves better results in deciphering tasks. The first configuration generally obtains the best metrics, despite stemming from a handwritten word recognition paper. In relation to the transcription tasks, the leading results occur between the first and third sets of parameters. Nonetheless, it seems that the tasks and datasets at play are the biggest predictors of our model's outcomes. In other words, the same model with the same parameters does not yield the same results for different tasks. Although it was expected, we can corroborate that synthetic images are always easier to predict than real handwritten ones.

8 Conclusions and Future Work

In this work, we have trained a sequence-to-sequence model in order to directly decrypt and transcribe images of historical handwritten ciphered documents. The first step was to study the various approaches in the state of the art. Then, we formed three different versions of the same data set using real handwritten images from the Copiale cipher and synthetically generated replicas. Our approach to transcribe and decipher these images relies on a sequence-to-sequence model, based on our perspective of the problem as a translation task. We planned for a total of 18 experiments with different combinations of parameters, tasks and datasets. Our findings indicate that the deciphering tasks often outperform the transcription tasks, even with the exact same configurations. The same model behaves differently depending

on the task. Since we are working with a substitution homophonic cipher, we can also deduce that our model is able to make the predictions using the context of the sentence rather than the following token. The synthetic data set always reaches the best results, while the real and combined data sets have some more difficulties. Only one decipherment experiment with the real data set is able to obtain good metrics; the experiments with the combined data set conclude in training processes with overfitting. Even with these issues, this exploratory study already secures promising results. The future avenues of joint end-to-end approaches to decipherment and transcription tasks are definitely worth investigating.

We have successfully transcribed and deciphered historical handwritten ciphered documents with synthetic images and partially with real handwritten images. However, there are many more actions that can be taken to build upon our findings. For instance, other experiments can be set up with other data sets and configurations. It is possible that our results could be improved with an increased dataset or more suitable parameters. The experiments on the combined data set may benefit the most from these changes. In addition, other NLP models could potentially bring better metrics, such as transformers. With further work, the next experiments may be able to be focused on images with manuscripts instead of lines.

9 Acknowledgements

The completion of this bachelor's thesis could not have been possible without the help and support of Alicia Fornés Bisquerra and Pau Torras Coloma. On one hand, Alicia did an excellent job as a tutor, guiding this bachelor's thesis in the right direction while helping me solve any problems along the way. On the other hand, Pau consistently provided invaluable technical support and insights every time I had doubts. The success of this project would not have been possible without them. Lastly, a final appreciation to all my friends and family for being there for me.

References

[1] Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker & Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project, Cryptologia, 44:6, 545-559, DOI: 10.1080/01611194.2020.1716410

[2] Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker & Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project, Cryptologia, 44:6, 545-559, DOI: 10.1080/01611194.2020.1716410

[3] Alicia Fornés, Beáta Megyesi & Joan Mas. 2017. Transcription of Encoded Manuscripts with Image Processing Techniques. In Digital Humanities Conference, pages 441-443.

[4] Xusen Yin, Nada Aldarrab, Beáta Megyesi & Kevin Knight. 2019. Decipherment of Historical Manuscript Images. In ICDAR, pages 78–85.

[5] Arnau Baró, Jialuo Chen, Alicia Fornés & Beáta Megyesi. 2019. Towards a Generic Unsupervised Method for Transcription of Encoded Manuscripts. In DATECH, pages 73–78.

[6] Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés & Beáta Megyesi. 2020. A Web-Based Interactive Transcription Tool for Encrypted Manuscripts. In HistoCrypt 2020, pages 52–59.

[7] Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés & Beáta Megyesi. 2021. Unsupervised Alphabet Matching in Historical Encrypted Manuscript Images. In HistoCrypt 2021, pages 34-37.

[8] Nada Aldarrab. 2017. Decipherment of Historical Manuscripts. DOI: 10.25549/usctheses-c40-351927

[9] Nada Aldarrab. 2022. Automatic Decipherment of Historical Manuscripts. DOI: 10.25549/usctheses-oUC112195841

[10] Nada Aldarrab & Jonathan May. 2020. Can Sequence-to-Sequence Models Crack Substitution Ciphers?, DOI: 10.48550/arXiv.2012.15229

[11] Kevin Knight, Beáta Megyasi & Christine Schaefer. 2011. The Copiale Cipher, In 4th Workshop on Building and Using Comparable Corpora, pages 2-9

[12] Lei Kang, J.Ignacio Toledo, Pau Riba, Mauricio Villegas, Alicia Fornés & Marçal Rusiñol. 2018. Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition, In German Conference on Pattern Recognition, pages 459-472