Deciphering of historic letters

Deciphering a directory-based encryption

Maxence MONCEL & Loïc ROUQUETTE
EPITA - Laboratoire de Recherche de l'EPITA (LRE)
16 juillet 2026

Table of contents

01

Context

Context

- Ciphered letters from the $XVIII - XIX^{th}$ century
- Correspondence letters with French colonies in America
- Letters are part of a bigger collection
- The collection also contains some of the keys
- Some letters are already deciphered[Pierrot et al., 2023]

Context - Letters

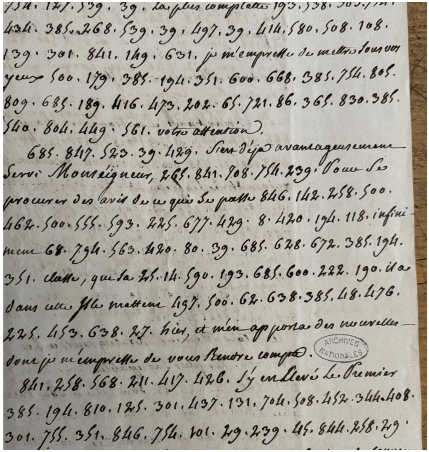

Figure 1 : A section of the letters we are trying to decipher. Please notice that plaintext is inserted between two cyphertext sections

Context - Ciphers

- Exclusively substitutions ciphers[Lucks, 1988]
- A mix between nomenclator and homophonic[Dhavare et al., 2013] substitution ciphers cipher
- Also called *directory-based encryption*

Context - Ciphers

- Exclusively substitutions ciphers[Lucks, 1988]
- A mix between nomenclator and homophonic[Dhavare et al., 2013] substitution ciphers cipher
- Also called *directory-based encryption*
- Nomenclator -> Simple substitution but plaintext symbols can be syllables and words

Context - Ciphers

- Exclusively substitutions ciphers[Lucks, 1988]
- A mix between nomenclator and homophonic[Dhavare et al., 2013] substitution ciphers cipher
- Also called *directory-based encryption*
- Nomenclator -> Simple substitution but plaintext symbols can be syllables and words
- Homophonic substitution -> Simple substitution but multiple cypher symbols for a single plaintext symbol

Context - Key

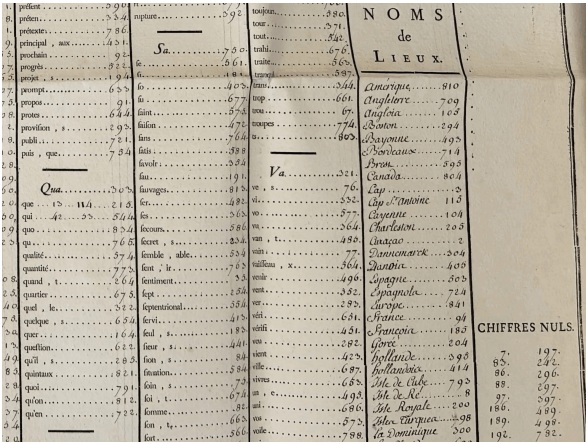

Figure 2 : A section of the key used to cypher the text. Please note the section reserved for the name of places and for null cypher symbols

Context - DECRYPT project[Megyesi et al., 2020]

- Collection of old ciphered texts and (sometimes) the associated keys
- Collection of tools to help deciphering (Cryptool2)
- Helping historians decipher historic texts

Figure 3 : The Cryptool2 logo that was taken from the official Cryptool2 website

02

Perplexity

Perplexity

”Does this text make sense ?”

Perplexity

"Does this text make sense ?"

- A scoring function evaluating a key by its ability to decipher french text that makes sense
- An optimization function will optimize this scoring function

Perplexity

"Does this text make sense ?"

- A scoring function evaluating a key by its ability to decipher french text that makes sense
- An optimization function will optimize this scoring function
- Based on one of Markov chains applications (n-grams [Megyesi et al., 2023])
- First a training phase, then we can compute perplexity

Perplexity

"Does this text make sense ?"

- A scoring function evaluating a key by its ability to decipher french text that makes sense
- An optimization function will optimize this scoring function
- Based on one of Markov chains applications (n-grams [Megyesi et al., 2023])
- First a training phase, then we can compute perplexity
- Show how *"perplex"* is the program towards the given text

Perplexity - Usage

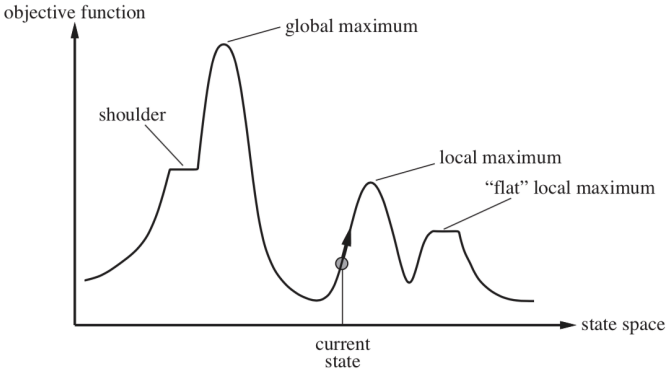

Figure 4 : An example of a scoring function that we are trying to minimize/maximize the value of. There are a lot of pitfalls that hinder our ability to find the global minimum/maximum. Source : https ://www.cs.cmu.edu/ 15281-s23/coursenotes/localsearch/images/utilitygraph.png

Perplexity - Tokens

How to abstract from the plaintext symbols ?

Perplexity - Tokens

How to abstract from the plaintext symbols ?
Answer : Encapsulate them in a class and give them a name !

Perplexity - Tokens

How to abstract from the plaintext symbols ?
Answer : Encapsulate them in a class and give them a name !

- Differentiate them from plain strings
- Ensure structure and coherence within the deciphering process (1 token <-> 1 cypher symbol)

Perplexity - Tokens

How to abstract from the plaintext symbols ?
Answer : Encapsulate them in a class and give them a name !

- Differentiate them from plain strings
- Ensure structure and coherence within the deciphering process (1 token <-> 1 cypher symbol)

Hypothesis

Ciphering by hand is a time-consuming process, thus we assumed that the writers always used the longest plaintext symbol available.

Perplexity - Markov chains

"What happens next depends only on the state of affairs now"

Perplexity - Markov chains

"What happens next depends only on the state of affairs now"

$$\forall n \geq 0, (i_0, \dots, i_{n-1}, i, j) \in E^{n+2} \text{ such as } \mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) > 0,$$
$$\mathbb{P}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) = \mathbb{P}(X_{n+1} = j | X_n = i).$$

With E, the set of all possible states

Perplexity - Markov chains

"What happens next depends only on the state of affairs now"

$$\forall n \geq 0, (i_0, \dots, i_{n-1}, i, j) \in E^{n+2} \text{ such as } \mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) > 0,$$
$$\mathbb{P}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) = \mathbb{P}(X_{n+1} = j | X_n = i).$$

With E, the set of all possible states

How about n-grams ?

A set of n consecutive tokens in the text.
A Markov chain of order $n - 1$ as we want to find the probability of the last token in the n-gram considering all the others.
Ex : Probability of 4-gram *["bon", "jour", "a", "tous"]* is the probability of the tokens *["bon", "jour", "a"]* to be followed by the token *"tous"*.

Perplexity - Computing score

- Training on text dealing with the same subjects, same time period and in the same language as the cyphertext [Megyesi et al., 2023]
- Simply computing probabilities for all possible n-grams in the training texts and saving them

Perplexity - Computing score

- Training on text dealing with the same subjects, same time period and in the same language as the cyphertext [Megyesi et al., 2023]
- Simply computing probabilities for all possible n-grams in the training texts and saving them

$$\log\left(\frac{times\ ngram\ appeared}{amount\ of\ ngrams}\right)$$

Perplexity - Computing score

- Training on text dealing with the same subjects, same time period and in the same language as the cyphertext [Megyesi et al., 2023]
- Simply computing probabilities for all possible n-grams in the training texts and saving them

$$\log\left(\frac{times\ ngram\ appeared}{amount\ of\ ngrams}\right)$$

- Computing the mean average probability of all n-grams in the deciphered text (naive approach)
- Penalty if we haven't learned a specific n-gram

Perplexity - Naive approach results

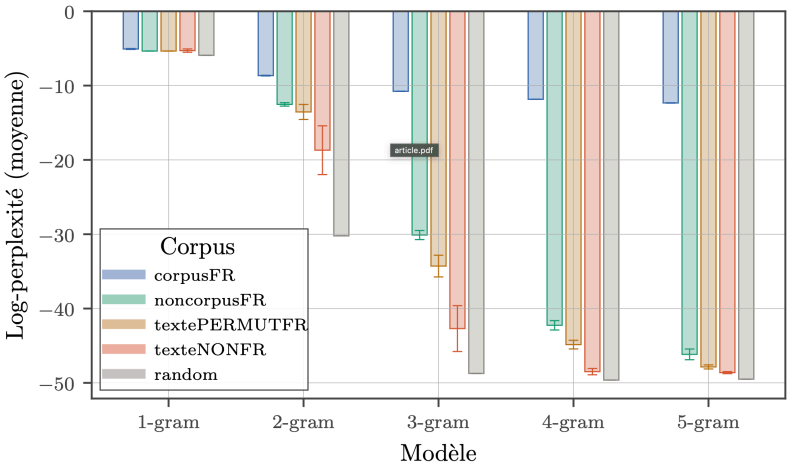

Figure 5 : A table showing the results of our naive perplexity approach on test text of roughly a paragraph of length (about 60 words each)

03

Hill-Climbing

Hill-Climbing

We now can detect French text on the topic of the letter we are trying to decipher
Now we *just* have to find the key

Hill-Climbing

We now can detect French text on the topic of the letter we are trying to decipher
Now we *just* have to find the key

- Multiple approaches
- Initialize the key and get searching
- Need a big enough text sample to ensure correct deciphering

Hill-Climbing

We now can detect French text on the topic of the letter we are trying to decipher
Now we *just* have to find the key

- Multiple approaches
- Initialize the key and get searching
- Need a big enough text sample to ensure correct deciphering

Be careful!

We are going to manipulate a key with 1000 elements.
It's like trying to optimize a 1000 parameters function...

Hill-Climbing - Key initialization

- We know most of the plaintext symbols
- There are plaintext symbols for places and persons names

Hill-Climbing - Key initialization

- We know most of the plaintext symbols
- There are plaintext symbols for places and persons names
- For now we will ignore them and consider them as null symbols
- We just shuffle all the plaintext symbols and add enough null symbols to have 1000 elements in the key

Hill-Climbing - Naive hill climbing[Russell et Norvig, 2020]

- Most basic and naive hill climbing
- Swap 2 elements of the key and keep it if the perplexity score improved
- Used as a baseline to compare to, not as an actual solution

Hill-Climbing - Naive hill climbing[Russell et Norvig, 2020]

- Most basic and naive hill climbing
- Swap 2 elements of the key and keep it if the perplexity score improved
- Used as a baseline to compare to, not as an actual solution

Swap intelligently

We still make sure to avoid objectively useless swaps such as swapping the same elements or two elements not used in the cyphertext.

Hill-Climbing - Taboo hill climbing

- Naive hill-climbing but with a *taboo list*
- If a swap is unsuccessful, add it to the *taboo list* so we may avoid it

Hill-Climbing - Taboo hill climbing

- Naive hill-climbing but with a *taboo list*
- If a swap is unsuccessful, add it to the *taboo list* so we may avoid it
- Helps converging faster to local maximum than the naive approach
- Hyperparameters that can influence the outcome of the algorithm (number of iterations, max size of taboo list)

Hill-Climbing - Taboo hill climbing

- Naive hill-climbing but with a *taboo list*
- If a swap is unsuccessful, add it to the *taboo list* so we may avoid it
- Helps converging faster to local maximum than the naive approach
- Hyperparameters that can influence the outcome of the algorithm (number of iterations, max size of taboo list)

Taboo *"list"*?

The taboo list works more like a FIFO structure rather than an actual list because it makes no sense to forbid a swap that used to be bad several iterations ago.
If it is full, the oldest taboo is removed.

Hill-Climbing - Results

gram size	score
3	−42.4
4	−49.1
5	−49.52

Figure 6 : The results of the naive hill-climbing depending on the size of the n-grams

Hill-Climbing - Results

gram size	score
3	−42.4
4	−49.1
5	−49.52

Figure 6 : The results of the naive hill-climbing depending on the size of the n-grams

gram size	score
3	−41.21
4	−49.15
5	−49.48

Figure 7 : The results of the naive hill-climbing with taboo list depending on the size of the n-grams

04

Human-In-The-Loop

Human-In-The-Loop

This is the part where we take care of the plaintext symbols referring to the names of places.

- Not yet tested! Only implemented
- The algorithm is asking for help
- Allows us to escape local maximums
- Allows us to manage the case of plaintext symbols referring to place names

Human-In-The-Loop

This is the part where we take care of the plaintext symbols referring to the names of places.

- Not yet tested! Only implemented
- The algorithm is asking for help
- Allows us to escape local maximums
- Allows us to manage the case of plaintext symbols referring to place names

Cross disciplinary research

We will need the help of the LRE's human and social researchers in order to find the places these letters are referring to.

Human-In-The-Loop - Functionalities

- Adding new plaintext tokens

Human-In-The-Loop - Functionalities

- Adding new plaintext tokens
- Checking the current key

Human-In-The-Loop - Functionalities

- Adding new plaintext tokens
- Checking the current key
- Checking the deciphered text with the current key

Human-In-The-Loop - Functionalities

- Adding new plaintext tokens
- Checking the current key
- Checking the deciphered text with the current key
- Making manual swaps

Human-In-The-Loop - First look

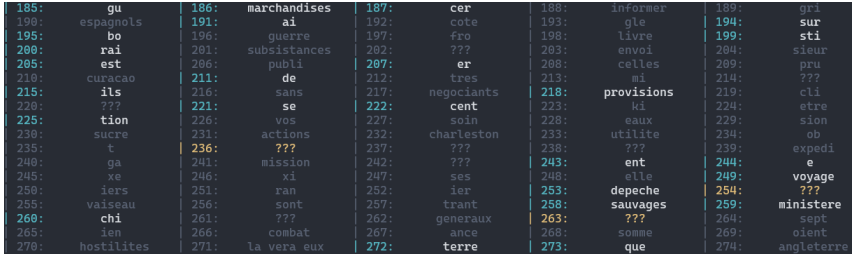

Figure 8 : A first look at the Human-In-The-Loop feature displaying the key

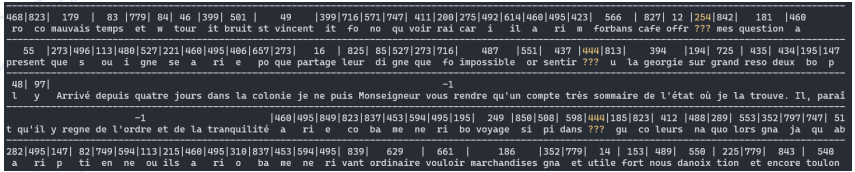

Figure 9 : A first look at the Human-In-The-Loop feature displaying the current text being deciphered

05

Conclusion

Conclusion

- Perplexity method to judge quality of deciphered text

Conclusion

- Perplexity method to judge quality of deciphered text
- First simple hill-climbing algorithms (will have to do better than that)

Conclusion

- Perplexity method to judge quality of deciphered text
- First simple hill-climbing algorithms (will have to do better than that)
- First steps for Human-In-The-Loop features

Conclusion

- Perplexity method to judge quality of deciphered text
- First simple hill-climbing algorithms (will have to do better than that)
- First steps for Human-In-The-Loop features

And also...

My predecessor on this subject (baptiste.cormoran) had already done some work that I documented.

06

Opening

Opening

- Improve hill climbing (Simulated annealing)
- Play around with Hyperparameters

Opening

- Improve hill climbing (Simulated annealing)
- Play around with Hyperparameters
- How ”bad” of a hill-climbing algorithm can we get away with Human-In-The-Loop ?
- Locking parts of the key during Human-In-The-Loop to ease deciphering

Opening

- Improve hill climbing (Simulated annealing)
- Play around with Hyperparameters
- How ”bad” of a hill-climbing algorithm can we get away with Human-In-The-Loop ?
- Locking parts of the key during Human-In-The-Loop to ease deciphering
- Improving perplexity computation (”smoothing” and using various n-grams sizes)

Opening

- Improve hill climbing (Simulated annealing)
- Play around with Hyperparameters
- How ”bad” of a hill-climbing algorithm can we get away with Human-In-The-Loop ?
- Locking parts of the key during Human-In-The-Loop to ease deciphering
- Improving perplexity computation (”smoothing” and using various n-grams sizes)
- OCR for full deciphering pipeline[Yin et al., 2019] ?

Thank you for your attention

Bibliography i

- Dhavare, A., Low, R. M., & Stamp, M. (2013)
  Efficient cryptanalysis of homophonic substitution ciphers.
  Cryptologia.

- Lucks, M. (1988)
  A constraint satisfaction algorithm for the automated decryption of simple substitution ciphers.
  Conference on the Theory and Application of Cryptography.

- Megyesi, B., Esslinger, B., Fornés, A., Kopal, N., Láng, B., Lasry, G., de Leeuw, K., Pettersson, E., Wacker, A., & Waldispühl, M. (2020)
  Decryption of historical manuscripts : the DECRYPT project.
  Cryptologia.

- Megyesi, B., Sikora, J., Fornmark, F., Waldispühl, M., Kopal, N., & Mikhalev, V. (2023)
  Historical Language Models in Cryptanalysis : Case Studies on English and German.
  International Conference on Historical Cryptology.

- Pierrot, C., Desenclos, C., Gaudry, P., & Zimmermann, P. (2023)
  Deciphering Charles Quint (A diplomatic letter from 1547). In C. Dahlke & M. Göggerle (Éd.), *Linköping Electronic Conference Proceedings*.

Bibliography ii

- Russell, S., & Norvig, P. (2020)
  Artificial Intelligence : A Modern Approach (4th Edition)
  .Pearson.

- Yin, X., Aldarrab, N., Megyesi, B., & Knight, K. (2019)
  Decipherment of historical manuscript images.
  2019 International Conference on Document Analysis and Recognition (ICDAR).