Decipherment as Regression

Solving Historical Substitution Ciphers by Learning Symbol Recurrence Relations

Nishant Kambhatla, Logan Born, Anoop Sarkar

May 2023

Findings of the Association for Computational Linguistics: EACL 2023

Presented by Morten Munk Andersen

Contents

1. Why this paper? 2
2. Methodology 4
   2.1 Recurrent Integer Sequences 5
   2.2 Generative Decipherment Model 7

1. Why this paper?

1. Why this paper?

Relevancy

- Homophonic substitution ciphers

Ranking

- Core2023 Ranking: A

Recency

- May 2023

Decipherment as Regression: Solving Historical Substitution Ciphers by Learning Symbol Recurrence Relations

Nishant Kambhatla **Logan Born** **Anoop Sarkar**
School of Computing Science, Simon Fraser University
8888 University Drive, Burnaby BC, Canada
{nkambhat, loborn, anoop}@sfu.ca

Abstract

Solving substitution ciphers involves mapping sequences of cipher symbols to fluent text in a target language. This has conventionally been formulated as a search problem, to find the decipherment key using a character-level language model to constrain the search space. This work instead frames decipherment as a sequence prediction task, using a Transformer-based causal language model to learn recurrences between characters in a ciphertext. We introduce a novel technique for transcribing arbitrary substitution ciphers into a common *recurrence encoding*. By leveraging this technique, we (i) create a large synthetic dataset of homophonic ciphers using random keys, and (ii) train a decipherment model that predicts the plaintext sequence given a recurrence-encoded ciphertext. Our method achieves strong results on synthetic 1:1 and homophonic ciphers, and cracks several real historic homophonic ciphers. Our analysis shows that the model learns recurrence relations between cipher symbols and recovers decipherment keys in its self-attention.[1]

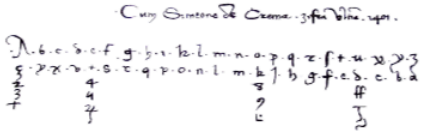


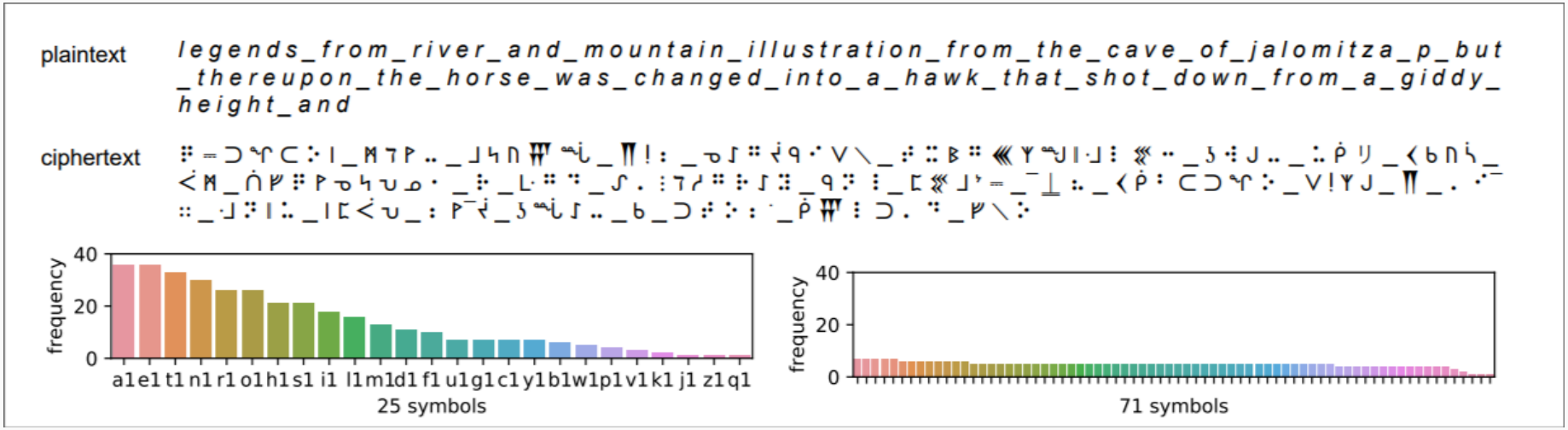
Figure 1: The homophonic substitution key for the *Simeone de Crema* written in Mantua in 1401 AD. The top line maps each character in the alphabet to its reversed-alphabet equivalent; each vowel is substituted by three additional symbols.

sequences (D'Ascoli et al., 2022). We rethink decipherment as a regression task that predicts a natural language plaintext by learning a recurrence relation between integer-coded ciphertext symbols.

There exist large collections of historical ciphers (see de-crypt.org)[2], in the form of encrypted letters and more informal communications, of which many remain undeciphered. Many of these texts employ complex *homophonic substitution ciphers*, which mask the frequencies of letters by using a larger alphabet than the underlying language. Figure 1 shows the first known homophonic cipher from 1401 AD [3]. Automated computational deci-

2. Methodology

2.1 Recurrent Integer Sequences



Monoalphabetic (1:1)

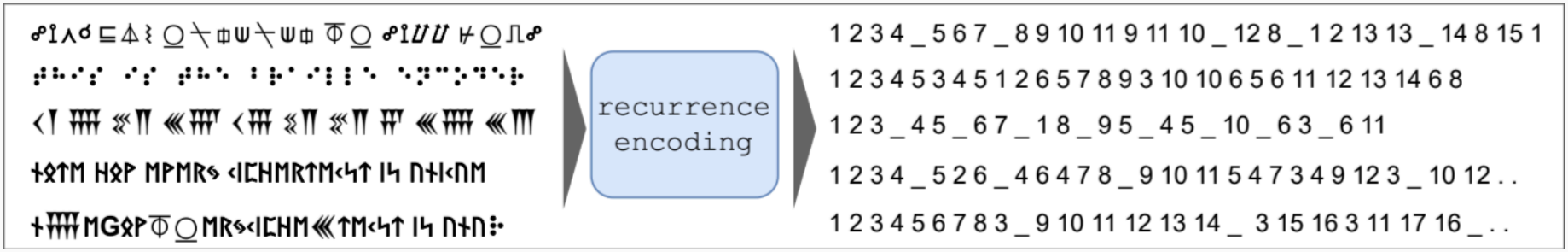
- Trivially solved with frequency analysis

Homophonic (1:>0)

- Harder to solve - frequencies can be hidden 😔
- More symbols = More mappings

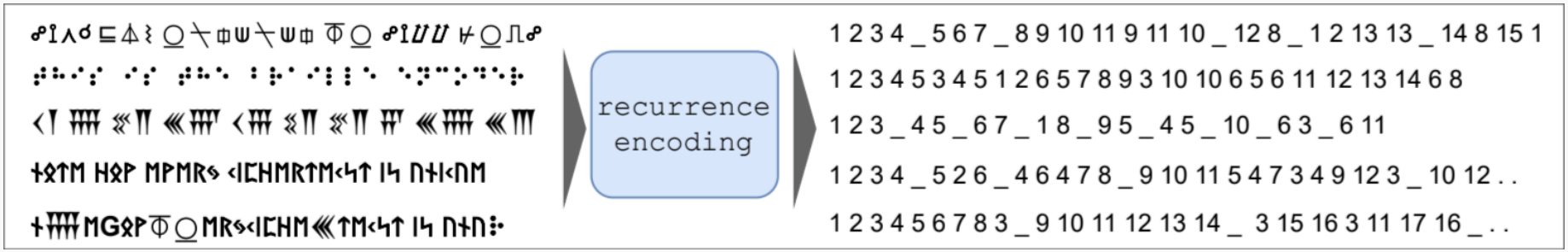
Capturing first/repeated symbol occurences

- Spaces denoted as **underscore**
- Unseen symbols denoted as **incremental integer**
- Recurring symbols denoted as represented **previous integer**
- Works for ciphers with different symbol sets



Capturing first/repeated symbol occurences

- Spaces denoted as **underscore**
- Unseen symbols denoted as **incremental integer**
- Recurring symbols denoted as represented **previous integer**
- Works for ciphers with different symbol sets



The authors consider this a novel approach

2.2 Generative Decipherment Model

Remember: Ciphertext is now a Recurrent Integer Sequence

This makes every cipher comparable

Dataset made by authors

- 2 million unique homophonic substitution ciphers
- Including their corresponding plaintexts
- Uses Modern English

2.2 Generative Decipherment Model

CausalLM

- Reads from left to right - can only look back
- Past words affect predicted words - (sort of like autocorrect)

CausalLM

- Reads from left to right - can only look back
- Past words affect predicted words - (sort of like autocorrect)

$$[X^l, Y^l] = \text{FFN} \circ \text{SelfAtn}\big([X^{l-1}, Y^{l-1}], \text{Mask}\big)$$

- $X^{l-1} \rightarrow$ Cipher at layer previous to l
- $Y^{l-1} \rightarrow$ Text at layer previous to l
- SelfAtn $\rightarrow$ Captures positions related to previous symbols/letters
- Mask $\rightarrow$ The attention mask used by SelfAtn
- FFN $\rightarrow$ Result is fed to Feed-Forward Neural Network X

CausalLM

- Reads from left to right - can only look back
- Past words affect predicted words - (sort of like autocorrect)

$$[X^l, Y^l] = \text{FFN} \circ \text{SelfAtn}([X^{l-1}, Y^{l-1}], \text{Mask})$$

- $X^{l-1} \rightarrow$ Cipher at layer previous to l
- $Y^{l-1} \rightarrow$ Text at layer previous to l
- SelfAtn $\rightarrow$ Captures positions related to previous symbols/letters
- Mask $\rightarrow$ The attention mask used by SelfAtn
- FFN $\rightarrow$ Result is fed to Feed-Forward Neural Network X

Above produces the representation at $[X^l, Y^l]$

Remember: CausalLM only looks back!

Loss function

$$L^{\text{CLM}}(X, Y) = L^{\text{SRC}} + L^{\text{TGT}} = -\text{logP}(X) - \text{logP}(Y|X)$$

- $L^{\text{SRC}} \rightarrow$ Source loss - error predicting cipher seq
- $L^{\text{TGT}} \rightarrow$ Target loss - error predicting plaintext seq
- $-\text{logP}(X) \rightarrow$ Probability of reproducing correct cipher symbols
- $-\text{logP}(X|Y) \rightarrow$ Probability of predicting plaintext given cipher

Loss function

$$L^{\text{CLM}}(X, Y) = L^{\text{SRC}} + L^{\text{TGT}} = -\text{logP}(X) - \text{logP}(Y|X)$$

- $L^{\text{SRC}} \rightarrow$ Source loss - error predicting cipher seq
- $L^{\text{TGT}} \rightarrow$ Target loss - error predicting plaintext seq
- $-\text{logP}(X) \rightarrow$ Probability of reproducing correct cipher symbols
- $-\text{logP}(X|Y) \rightarrow$ Probability of predicting plaintext given cipher

Low probability = high loss, and vice versa

Loss function

$$L^{\text{CLM}}(X, Y) = L^{\text{SRC}} + L^{\text{TGT}} = -\text{logP}(X) - \text{logP}(Y|X)$$

- $L^{\text{SRC}} \rightarrow$ Source loss - error predicting cipher seq
- $L^{\text{TGT}} \rightarrow$ Target loss - error predicting plaintext seq
- $-\text{logP}(X) \rightarrow$ Probability of reproducing correct cipher symbols
- $-\text{logP}(X|Y) \rightarrow$ Probability of predicting plaintext given cipher

Low probability = high loss, and vice versa

Probability can be seen as confidence