

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?	3
2. Methodology	5
2.1 Recurrent Integer Sequences	6
2.2 Generative Decipherment Model	8
3. Results	13
3.1 Synthetic Cipher	14
3.2 Z408 Cipher	15
3.3 Historical Ciphers	17
3.4 Monoalphabetic Ciphers	19
3.5 Unseen Language Ciphers	20
4. Contributions	22
4.1 Contributions	23
5. Limitations	24

5.1 Limitations 25

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

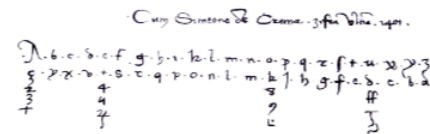


Figure 1: The homophonic substitution key for the *Simeone de Crema* written in Mantua in 1401 AD. The top line maps each character in the alphabet to its reversed-alphabet equivalent; each vowel is substituted by three additional symbols.

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.

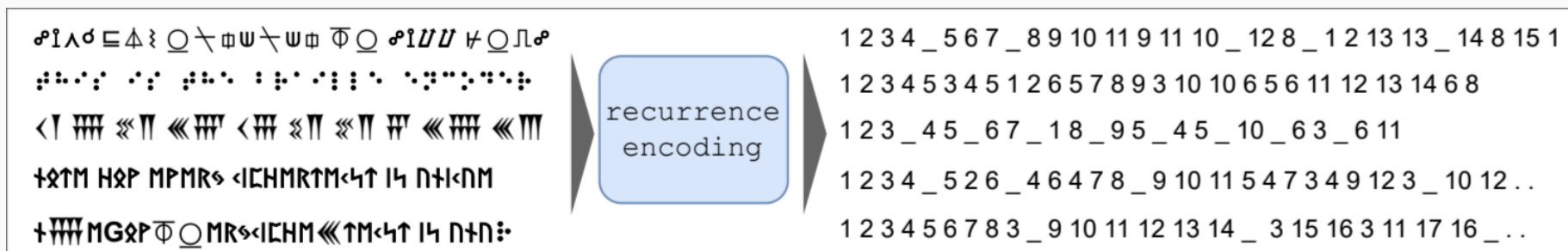
There exist large collections of historical ciphers (see de-crypt.org)², in the form of encrypted letters and more informal communications, of which many remain undeciphered. Many of these texts employ complex *homophonic substitution ciphers*, which mask the frequencies of letters by using a larger alphabet than the underlying language. Figure 1 shows the first known homophonic cipher from 1401 AD³. Automated computational deci-

2. Methodology

2.1 Recurrent Integer Sequences

Capturing first/repeated symbol occurrences

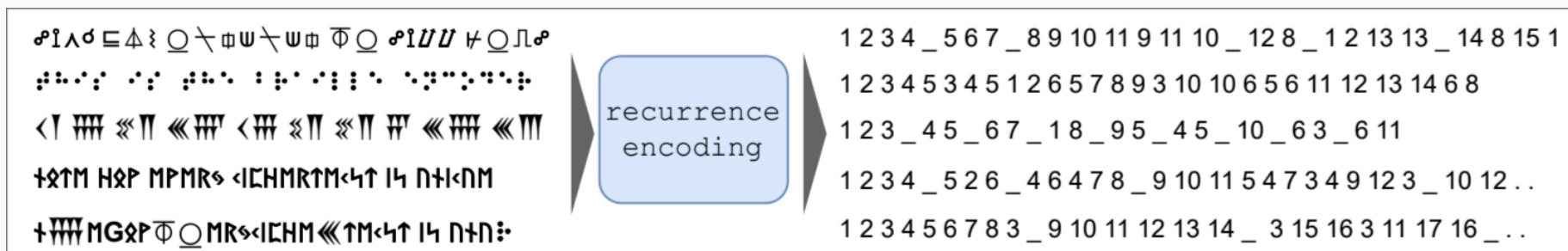
- Spaces denoted as **underscore**
- Unseen symbols denoted as **incremental integer**
- Recurring symbols denoted as represented **previous integer**
- Works for ciphers with different symbol sets



2.1 Recurrent Integer Sequences

Capturing first/repeated symbol occurrences

- 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)

2.2 Generative Decipherment Model

CausalLM

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

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

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

2.2 Generative Decipherment Model

CausalLM

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

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

- $X^{l-1} \rightarrow$ Cipher at layer previous to l
- $Y^{l-1} \rightarrow$ Text at layer previous to l
- SelfAttn \rightarrow Captures positions related to previous symbols/letters
- Mask \rightarrow The attention mask used by SelfAttn
- 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!

2.2 Generative Decipherment Model

Loss function

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

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

2.2 Generative Decipherment Model

Loss function

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

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

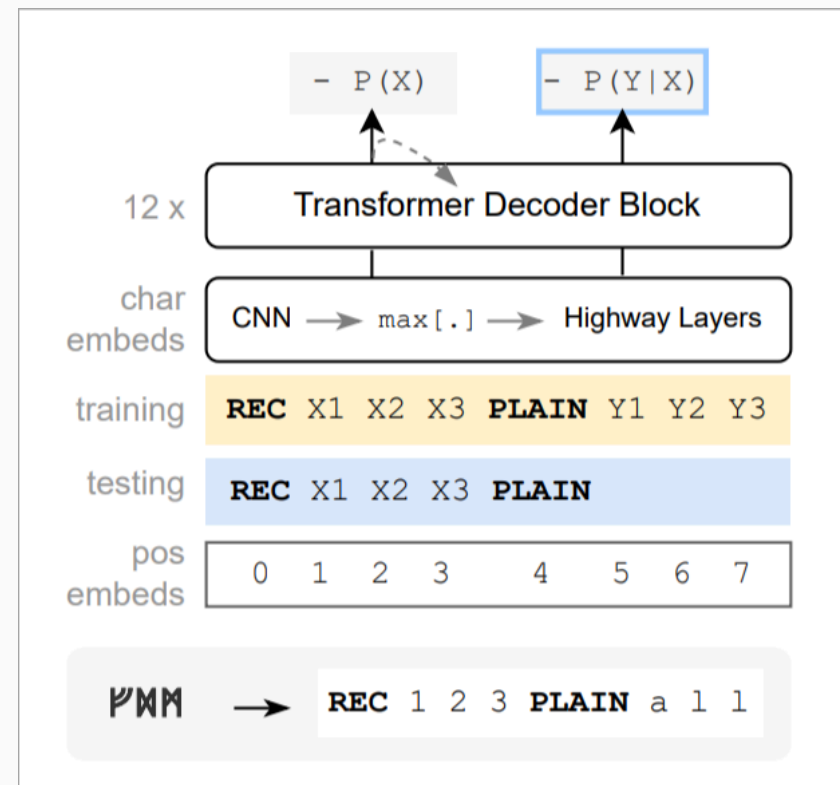
Low probability = high loss, and vice versa

Probability can be seen as confidence

2.2 Generative Decipherment Model

Why CausalLM?

- Predict the cipher symbols in a sequence
- Predict plaintext in the sequence
- Model learns the mappings



2.2 Generative Decipherment Model

Considered models

- Seq2seq
- Target-Only CausalLM
- PrefixLM

Why are they weaker?

- Only predicts plaintext
- Does not learn cipher symbol recurrence patterns

3. Results

3.1 Synthetic Cipher

How can we measure?

- Symbol Error Rate (SER)
- 0% = Perfect decipherment
- 100% Total gibberish

What can we observe?

- Between 400 - 700 chars
- Three key ranges
- CausalLM outperforms the others 🎉

#keys	Model	Max Len.	
		400	700
30-45	Seq-to-Seq	72.30	fail
	PrefixLM	54.73	69.50
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.40	0.21
40-65	PrefixLM	69.50	54.73
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.83	0.80
30-85	PrefixLM	70.52	71.82
	CausalLM (tgt)	42.05	42.69
	CausalLM	2.25	2.19

3.1 Synthetic Cipher

How can we measure?

- Symbol Error Rate (SER)
- 0% = Perfect decipherment
- 100% Total gibberish

What can we observe?

- Between 400 - 700 chars
- Three key ranges
- CausalLM outperforms the others 🎉

#keys	Model	Max Len.	
		400	700
30-45	Seq-to-Seq	72.30	fail
	PrefixLM	54.73	69.50
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.40	0.21
40-65	PrefixLM	69.50	54.73
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.83	0.80
30-85	PrefixLM	70.52	71.82
	CausalLM (tgt)	42.05	42.69
	CausalLM	2.25	2.19

Fun observation: seq2seq does not even converge at longer ciphers 🤖

3.2 Z408 Cipher

But what about real ciphers?

- Z408 = 408 characters
- 54 symbols
- From the 1960's
- No spaces !



<https://zodiackiller.fandom.com/wiki/408-cipher>

3.2 Z408 Cipher

Hill-climbing

- Keep the best candidates

Beam search

- Keep N best candidates

Method	Search	SER (%)	Speed
LM+EM (2013)	1M restarts	11.0	–
<i>n</i> –gram LM (2013)	beam 100K	94.6	4000
	beam 1M	2.7	35000
LSTM LM (2018)	beam 100K	2.4	5600
	beam 1M	1.9	50000
Ours (greedy)	beam 1	1.9	1 sec
Ours (best)	beam 200	1.9	2 sec

3.2 Z408 Cipher

Hill-climbing

- Keep the best candidates

Beam search

- Keep N best candidates

Method	Search	SER (%)	Speed
LM+EM (2013)	1M restarts	11.0	–
<i>n</i> –gram LM (2013)	beam 100K	94.6	4000
	beam 1M	2.7	35000
LSTM LM (2018)	beam 100K	2.4	5600
	beam 1M	1.9	50000
Ours (greedy)	beam 1	1.9	1 sec
Ours (best)	beam 200	1.9	2 sec

CausalLM 🎮

- Faster (it does not search)
- Better (even with smaller beam)

3.3 Historical Ciphers

What about historical ciphers?

TNA_SP106/5

- 1624, UK
- Homophonic substitution
- 171 characters
- 47 symbols to 27 letters
- Not many recurrences (3.6 avg)

The homophonic 40-65 key model

- They used beam size 1000
- Achieved 18% SER

3.3 Historical Ciphers

What about historical ciphers?

TNA_SP106/5

- 1624, UK
- Homophonic substitution
- 171 characters
- 47 symbols to 27 letters
- Not many recurrences (3.6 avg)

The homophonic 40-65 key model

- They used beam size 1000
- Achieved 18% SER

Remember: This is a hard cipher in an out-of-domain language

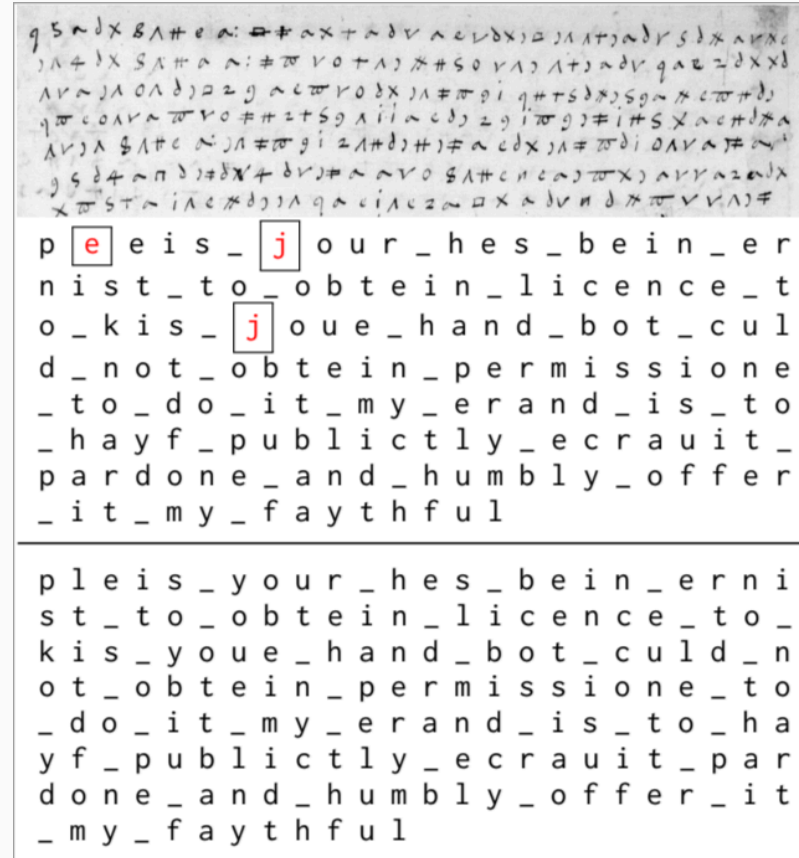
3.3 Historical Ciphers

BnF_fr2988_f01

- 1524-1549, Italy
- Homophonic substitution
- 2 pages long
- 35 symbols
- More recurrences but older language

The homophonic 30-45 key model

- Achieved 1.13% SER



Notice how words are different in old English!

3.4 Monoalphabetic Ciphers

CausalLM + Rec

- Recurrence Integer Sequence

CausalLM + Freq

- Described in another paper
- Summary:
 - encoded with frequency rank
 - unlike REC which is left to right order

CausalLM Observations

- CausalLM
 - Weaker on short ciphers
 - Still comparable to other models!

(Near) perfect SER on >128 ciphers!

cipher length →	<128	>128
Beam + 6-gram (Nuhn et al., 2013)	22.00	0.00
Beam + LM ((Kambhatla et al., 2018))	10.89	0.00
Beam + LM + Freq. Match (ibid.)	11.32	0.00
Seq2Seq + Freq. (Aldarrab and May)	7.68	0.00
Causal LM + Freq.	10.56	0.00
Causal LM + Rec.	11.30	0.02

3.5 Unseen Language Ciphers

What if we don't know the language of the cipher?

- Multilingual model
- Trained on 13 languages (Latin included)
- No language ID's during training!
- Frequency based encoding
 - Likely due to Zipfian consistency

	SER (%)
Multilingual Seq2Seq (2021)	5.47
Multilingual Causal LM (ours)	4.10

Results on the monoalphabetic Borg cipher in 17th century Latin

3.5 Unseen Language Ciphers

What about the main model?

Zero-shot on 400 chars of Borg

- SER 45.14%
- Not too good

But in real life...

- Domain expert evaluate output
- If they correct 3 words manually:
 - SER 3.89%
 - Pretty good!

3.5 Unseen Language Ciphers

What about the main model?

Zero-shot on 400 chars of Borg

- SER 45.14%
- Not too good

But in real life...

- Domain expert evaluate output
- If they correct 3 words manually:
 - SER 3.89%
 - Pretty good!

Remember:

The main model has never seen Latin before!

4. Contributions

4.1 Contributions

1. The Seq2seq dataset

2. Novel Recurrence Integer Sequence

- Captures repetition and position
- Works for both mono- and homophonic

3. Analysis of REC in Transformer LM

- Faster and more accurate

4. Practical application of solver

- Fully automated
- Solved real historical ciphers

5. Limitations

5.1 Limitations

Cipher Sizes

- Not superior for shorter ciphers <128 with less frequencies
- Inefficient for longer ciphers >1500

The paper tries a lot!

- The model tries to do it all
- ... But it does it well!

Where is the data from?

- Authors use 3 different datasets + ciphers
- Hard to keep track of which and when?

5.1 Limitations

Cipher Sizes

- Not superior for shorter ciphers <128 with less frequencies
- Inefficient for longer ciphers >1500

The paper tries a lot!

- The model tries to do it all
- ... But it does it well!

Where is the data from?

- Authors use 3 different datasets + ciphers
- Hard to keep track of which and when?

What happens with transpositioned ciphers like Z340? 🤩