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

Contents

1. Why this paper?

1. Why this paper?

Relevancy

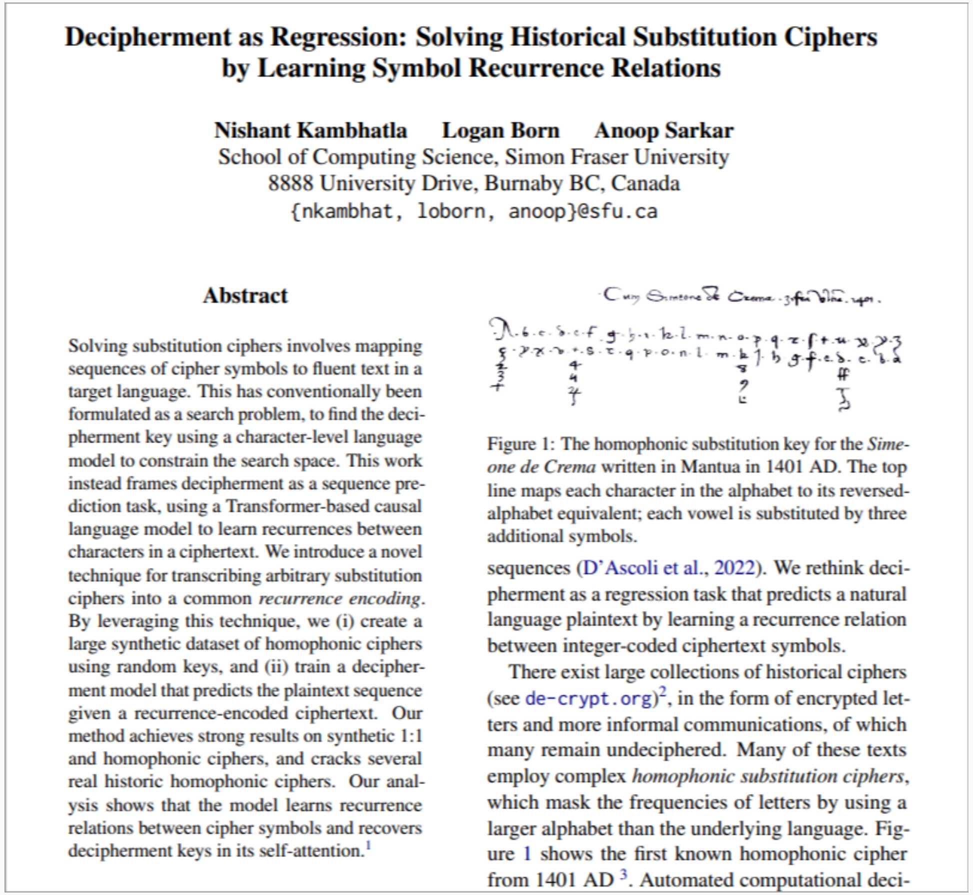
- Homophonic substitution ciphers

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]

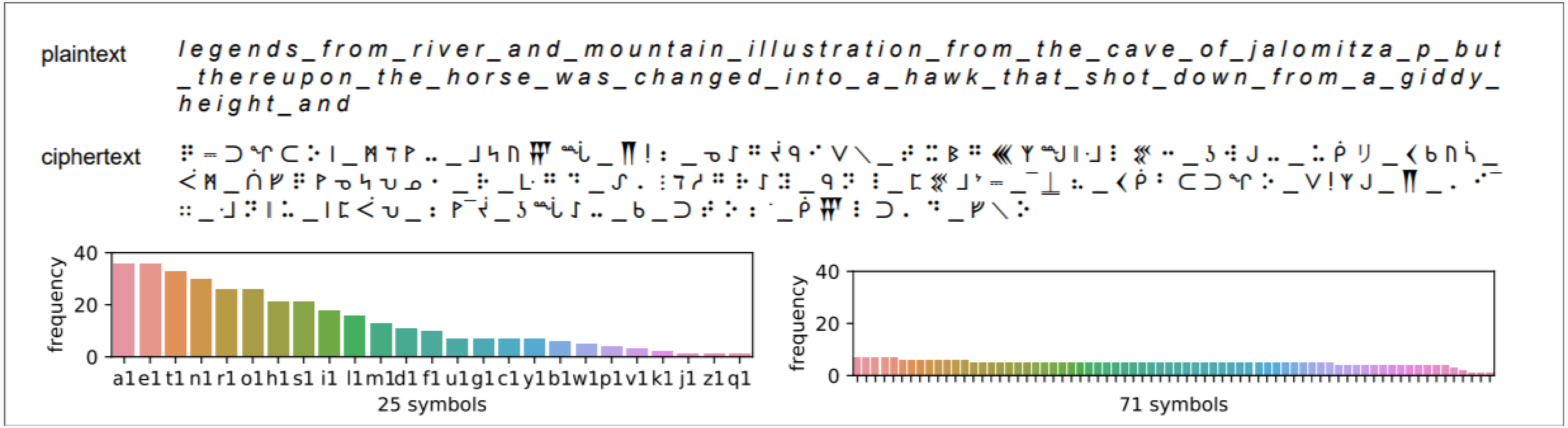
Figure 1: The homophonic substitution key for the *Simeone de Crema* written in Mantua in 1401 AD. The top line maps each character in the alphabet to its reversed-alphabet equivalent; each vowel is substituted by three additional symbols.

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



plaintext legends_from_river_and_mountain_illustration_from_the_cave_of_jalomitza_p_but_thereupon_the_horse_was_changed_into_a_hawk_that_shot_down_from_a_giddy_height_and

Monoalphabetic (1:1)

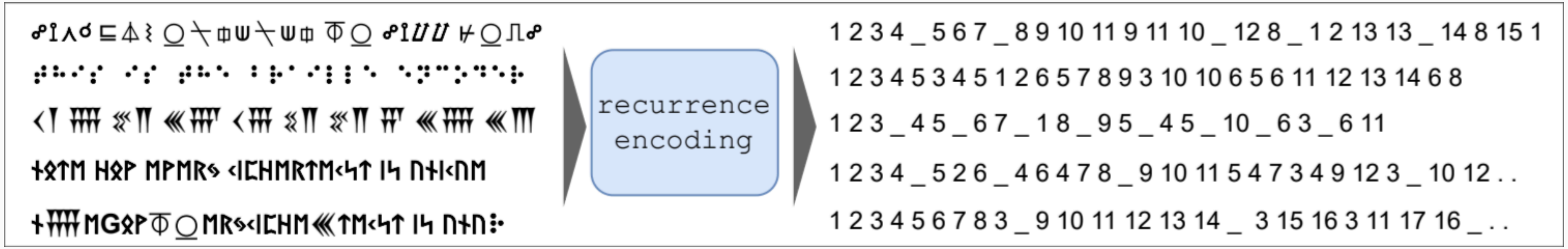
- Trivially solved with frequency analysis

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

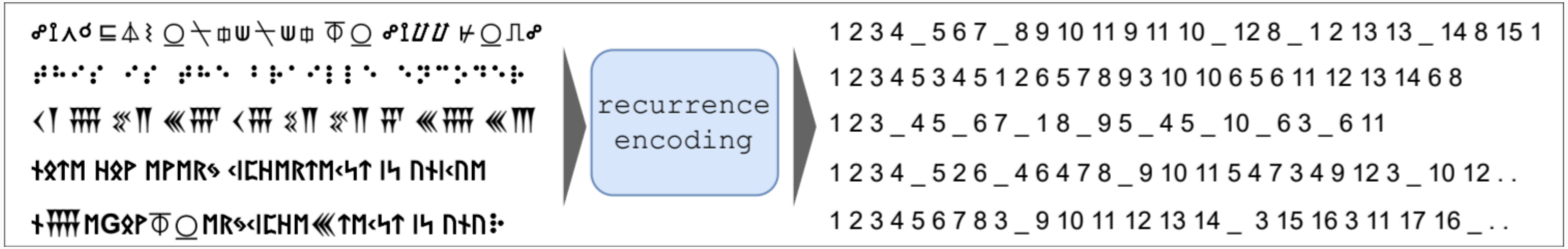


```
1 2 3 4 _ 5 6 7 _ 8 9 10 11 9 11 10 _ 12 8 _ 1 2 13 13 _ 14 8 15 1
1 2 3 4 5 3 4 5 1 2 6 5 7 8 9 3 10 10 6 5 6 11 12 13 14 6 8
1 2 3 _ 4 5 _ 6 7 _ 1 8 _ 9 5 _ 4 5 _ 10 _ 6 3 _ 6 11
1 2 3 4 _ 5 2 6 _ 4 6 4 7 8 _ 9 10 11 5 4 7 3 4 9 12 3 _ 10 12 . .
1 2 3 4 5 6 7 8 3 _ 9 10 11 12 13 14 _ 3 15 16 3 11 17 16 _ . .
```

2.1 Recurrent Integer Sequences

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



```
1 2 3 4 _ 5 6 7 _ 8 9 10 11 9 11 10 _ 12 8 _ 1 2 13 13 _ 14 8 15 1
1 2 3 4 5 3 4 5 1 2 6 5 7 8 9 3 10 10 6 5 6 11 12 13 14 6 8
1 2 3 _ 4 5 _ 6 7 _ 1 8 _ 9 5 _ 4 5 _ 10 _ 6 3 _ 6 11
1 2 3 4 _ 5 2 6 _ 4 6 4 7 8 _ 9 10 11 5 4 7 3 4 9 12 3 _ 10 12 . .
1 2 3 4 5 6 7 8 3 _ 9 10 11 12 13 14 _ 3 15 16 3 11 17 16 _ . .
```

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)

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

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

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

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

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

Remember: CausalLM only looks back!

2.2 Generative Decipherment Model

Loss function

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

- $L^{\text{SRC}} \rightarrow$ 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

2.2 Generative Decipherment Model

Loss function

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

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

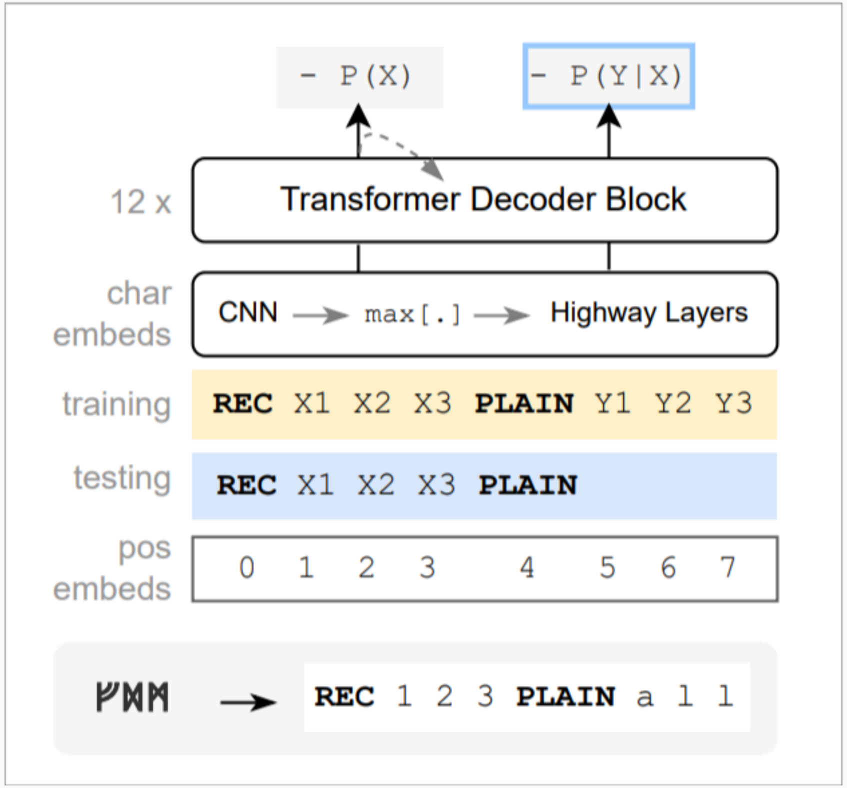
Low probability = high loss, and vice versa

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

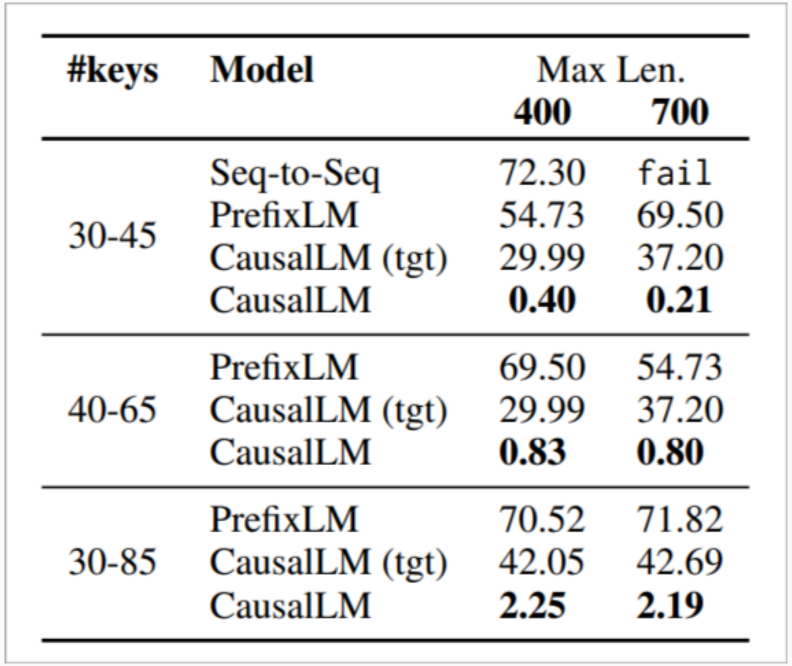
3.1 Synthetic Ciphers

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**

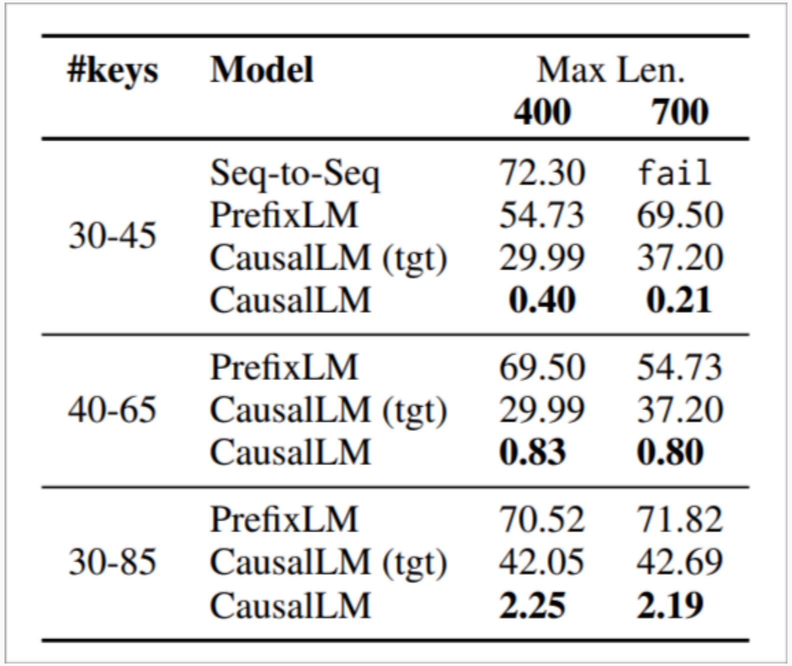
3.1 Synthetic Ciphers

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 🥳



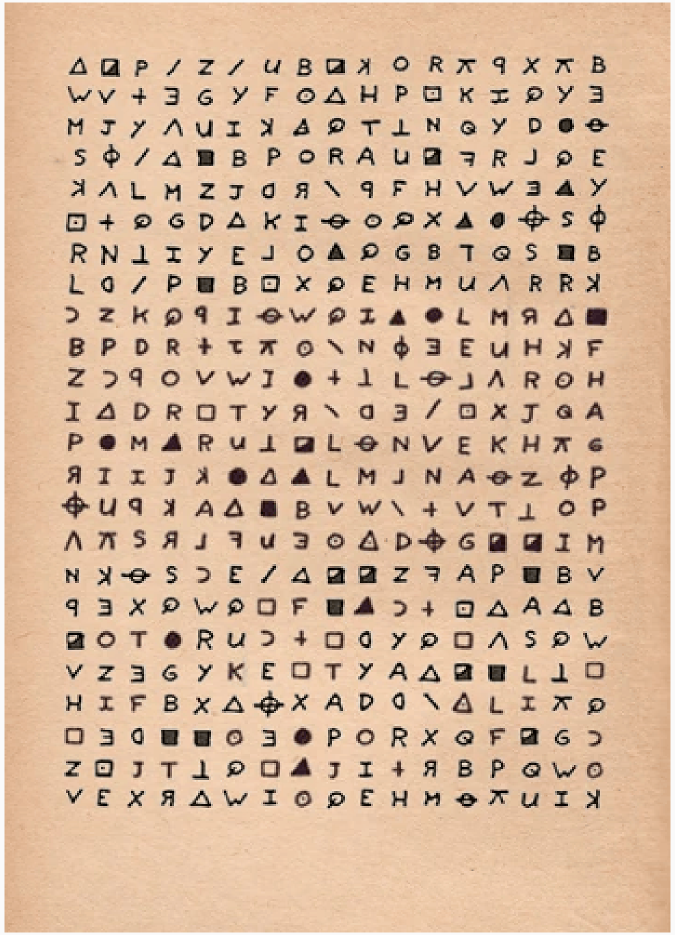
#keys	Model	Max Len.	
		400	**700**
30-45	Seq-to-Seq	72.30	fail
	PrefixLM	54.73	69.50
	CausalLM (tgt)	29.99	37.20
	CausalLM	**0.40**	**0.21**
40-65	PrefixLM	69.50	54.73
	CausalLM (tgt)	29.99	37.20
	CausalLM	**0.83**	**0.80**
30-85	PrefixLM	70.52	71.82
	CausalLM (tgt)	42.05	42.69
	CausalLM	**2.25**	**2.19**

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

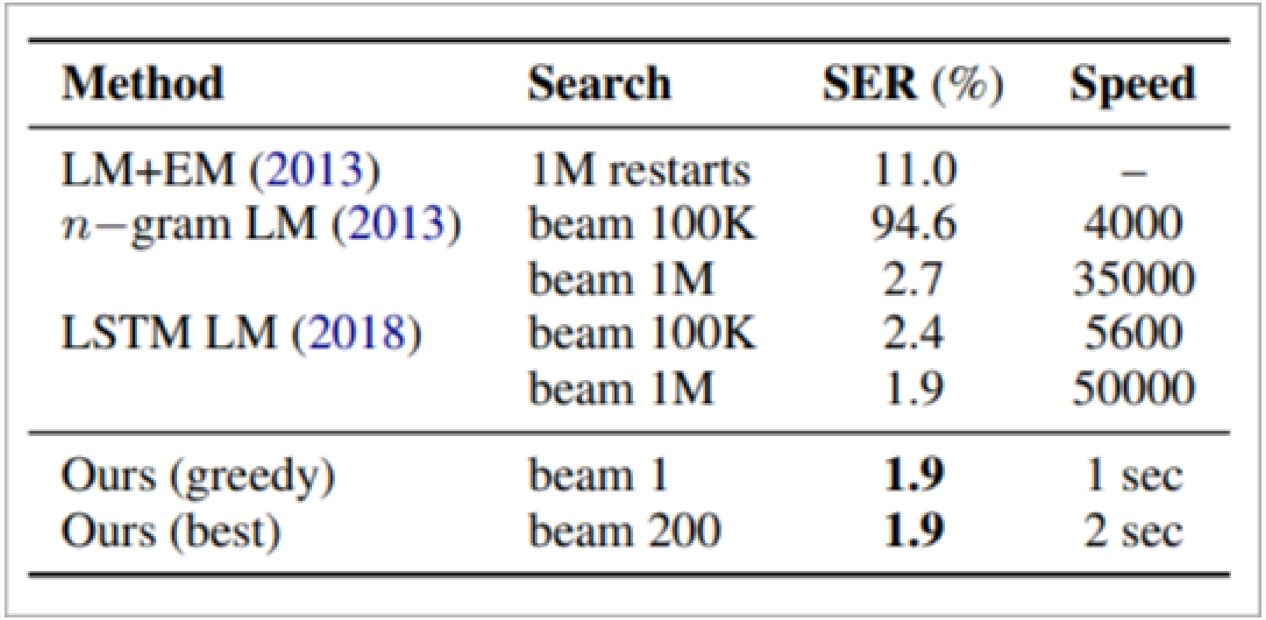
3.2 Z408 Cipher

Hill-climbing

- Keep the best candidates

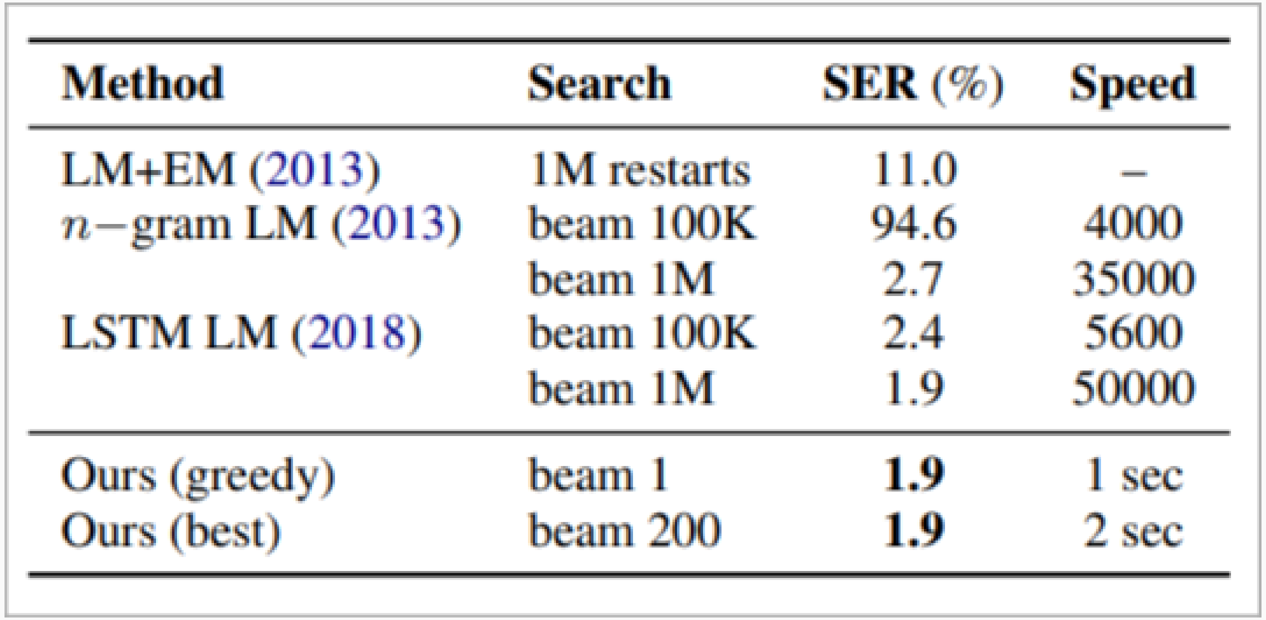
Beam search

- Keep N best candidates



Method	Search	SER (%)	Speed
LM+EM (2013)	1M restarts	11.0	–
n−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



Method	Search	SER (%)	Speed
LM+EM (2013)	1M restarts	11.0	–
n−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

Hill-climbing

- Keep the best candidates

Beam search

- Keep N best candidates

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 reccurences (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 reccurences (3.6 avg)

The homophonic 40-65 key model

- They used beam size 1000
- Achieved 18% SER

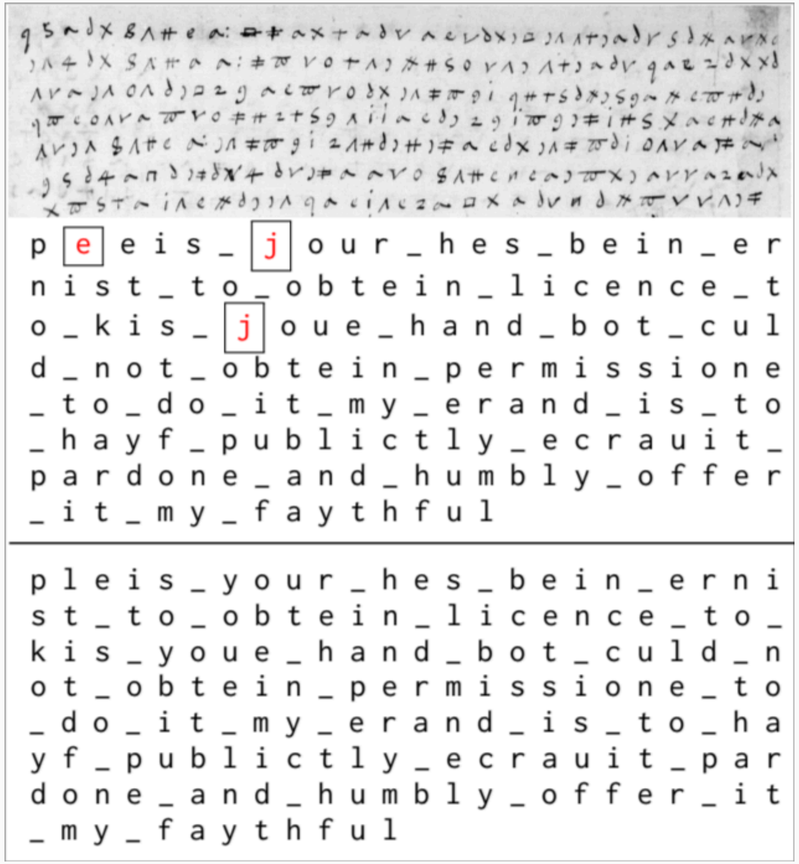
Remember: This is a hard cipher in an out-of-domain language

BnF_fr2988_f01

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

The homophonic 30-45 key model

- Achieved 1.13% SER



```
peeis_jour_hes_bein_er
nist_to_obtein_licence_t
o_kis_joue_hand_bot_cul
d_not_obtein_permissione
_to_do_it_my_erand_is_to
_hayf_publictly_ecrauit_
pardone_and_humbly_offer
_it_my_faythful
```

```
pleis_your_hes_bein_erni
st_to_obtein_licence_to_
kis_youe_hand_bot_culd_n
ot_obtein_permissione_to
_do_it_my_erand_is_to_ha
yf_publictly_ecrauit_par
done_and_humbly_offer_it
_my_faythful
```

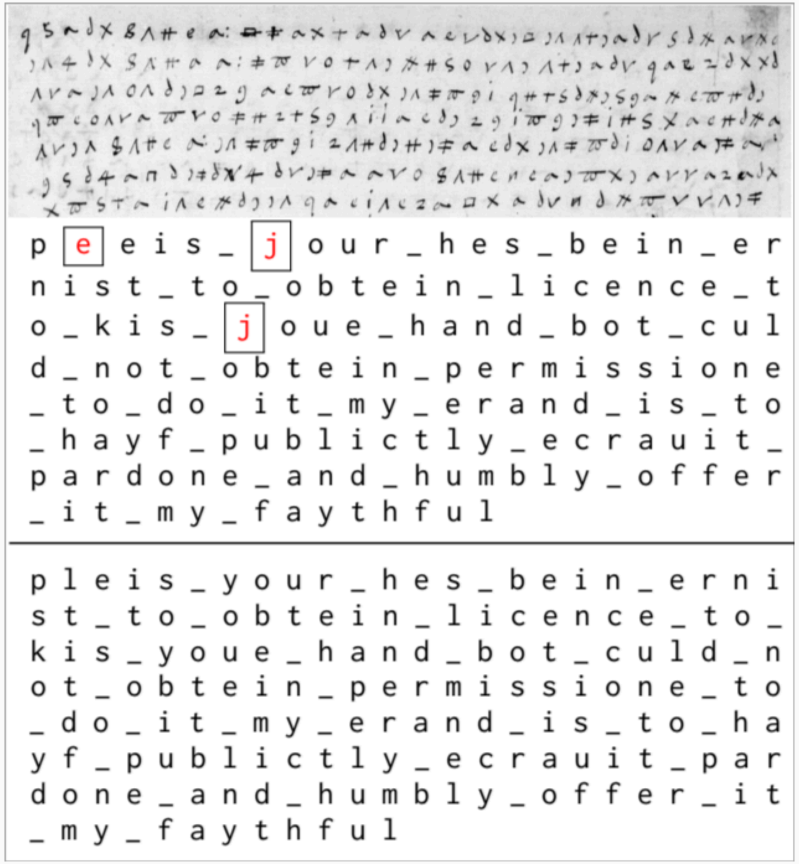
3.3 Historical Ciphers

BnF_fr2988_f01

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

The homophonic 30-45 key model

- Achieved 1.13% SER



```
p e e i s _ j o u r _ h e s _ b e i n _ e r
n i s t _ t o _ o b t e i n _ l i c e n c e _ t
o _ k i s _ j o u e _ h a n d _ b o t _ c u l
d _ n o t _ o b t e i n _ p e r m i s s i o n e
_ t o _ d o _ i t _ m y _ e r a n d _ i s _ t o
_ h a y f _ p u b l i c t l y _ e c r a u i t _
p a r d o n e _ a n d _ h u m b l y _ o f f e r
_ i t _ m y _ f a y t h f u l
```

```
p l e i s _ y o u r _ h e s _ b e i n _ e r n i
s t _ t o _ o b t e i n _ l i c e n c e _ t o _
k i s _ y o u e _ h a n d _ b o t _ c u l d _ n
o t _ o b t e i n _ p e r m i s s i o n e _ t o
_ d o _ i t _ m y _ e r a n d _ i s _ t o _ h a
y f _ p u b l i c t l y _ e c r a u i t _ p a r
d o n e _ a n d _ h u m b l y _ o f f e r _ i t
_ m y _ f a y t h f u l
```

Notice how words are different in old English!

3.4 Monoalphabetic Ciphers

CausalLM + Rec

- Recurrence Integer Sequence

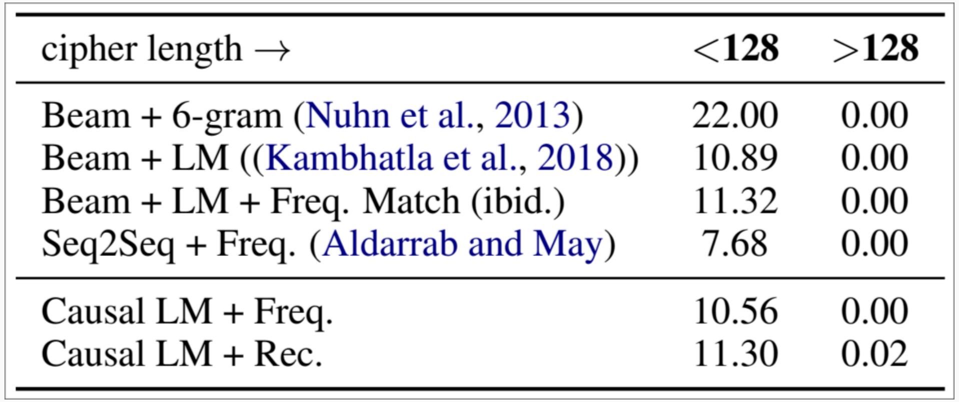
CausalLM + Freq

- 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!

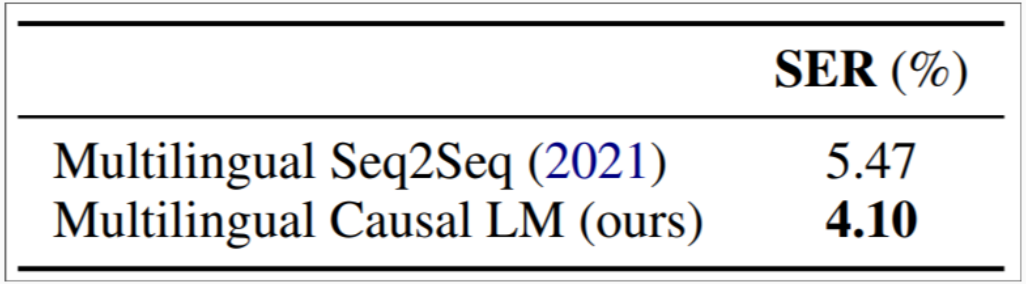


cipher length →	**<128**	**>128**
Beam + 6-gram (Nuhn et al., 2013)	22.00	0.00
Beam + LM ((Kambhatla et al., 2018))	10.89	0.00
Beam + LM + Freq. Match (ibid.)	11.32	0.00
Seq2Seq + Freq. (Aldarrab and May)	7.68	0.00
Causal LM + Freq.	10.56	0.00
Causal LM + Rec.	11.30	0.02

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 evaluates 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 evaluates output
- If they correct 3 words manually:
  - SER 3.89%
  - Pretty good!

Remember:

The main model has never seen Latin before!

4. Contributions

4. 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. 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?

They did not compare with AZDecrypt

5. 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?

They did not compare with AZDecrypt

What happens with transpositioned ciphers like Z340? 😈

6. Relevance

6. Relevance

Does not solve modern encryption ⚠️

- They are too advanced

Historical Value 📚

- Solve other ciphers

Cryptanalysis advancement 🤓

- Deeper knowledge of classic ciphers

More LM use cases 🤖

- What else can we use LMs for?

7. Questions?