

Can Sequence-to-Sequence Models Crack Substitution Ciphers?

Nada Aldarrab and Jonathan May

University of Southern California

Information Sciences Institute

{aldarrab, jonmay}@isi.edu

Abstract

Decipherment of historical ciphers is a challenging problem. The language of the target plaintext might be unknown, and ciphertext can have a lot of noise. State-of-the-art decipherment methods use beam search and a neural language model to score candidate plaintext hypotheses for a given cipher, assuming the plaintext language is known. We propose an end-to-end multilingual model for solving simple substitution ciphers. We test our model on synthetic and real historical ciphers and show that our proposed method can decipher text without explicit language identification while still being robust to noise.

1 Introduction

Libraries and archives have many enciphered documents from the early modern period. Example documents include encrypted letters, diplomatic correspondences, and books from secret societies (Figure 1). Previous work has made historical cipher collections available for researchers (Pettersson and Megyesi, 2019; Megyesi et al., 2020). Decipherment of classical ciphers is an essential step to reveal the contents of those historical documents.

In this work, we focus on solving 1:1 substitution ciphers. Current state-of-the-art methods use beam search and a neural language model to score candidate plaintext hypotheses for a given cipher (Kambhatla et al., 2018). However, this approach assumes that the target plaintext language is known. Other work that both identifies language and deciphers relies on a brute-force guess-and-check strategy (Knight et al., 2006; Hauer and Kondrak, 2016). We ask: Can we build an end-to-end model that deciphers directly without relying on a separate language ID step?

The contributions of our work are:

- We propose an end-to-end multilingual decipherment model that can solve 1:1 substi-

tution ciphers without explicit plaintext language identification, which we demonstrate on ciphers of 14 different languages.

- We conduct extensive testing of the proposed method in different realistic decipherment conditions; different cipher lengths, no-space ciphers, and ciphers with noise, and demonstrate that our model is robust to these conditions.
- We apply our model on synthetic ciphers as well as on the Borg cipher, a real historical cipher.¹ We show that our multilingual model can crack the Borg cipher using the first 256 characters of the cipher.

2 The Decipherment Problem

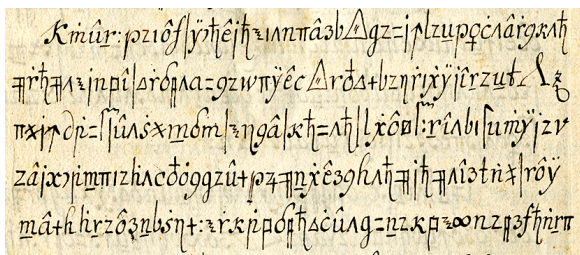
Decipherment conditions vary from one cipher to another. For example, some cleartext might be found along with the encrypted text, which gives a hint to the plaintext language of the cipher. In other cases, called **known-plaintext attacks**, some decoded material is found, which can be exploited to crack the rest of the encoded script. However, in a **ciphertext-only attack**, the focus of this paper, the cryptanalyst only has access to the ciphertext. This means that the encipherment method, the plaintext language, and the key are all unknown.

In this paper, we focus on solving 1:1 substitution ciphers. We follow Nuhn et al. (2013) and Kambhatla et al. (2018) and use machine translation notation to formulate our problem. We denote the ciphertext as $f_1^N = f_1 \dots f_j \dots f_N$ and the plaintext as $e_1^M = e_1 \dots e_i \dots e_M$.²

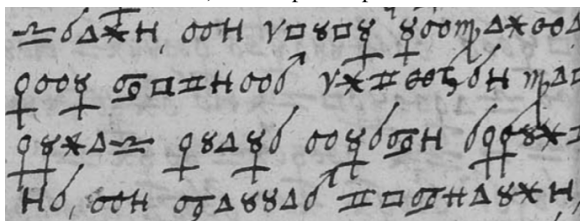
In a **1:1 substitution cipher**, plaintext is encrypted into a ciphertext by replacing each plaintext character with a unique substitute according

¹<https://cl.lingfil.uu.se/~bea/borg/>

²Unless there is noise or space restoration, $N = M$; see Sections 5.4 and 5.2.



a) The Copiale cipher.³



b) The Borg cipher.

Figure 1: Historical cipher examples.

to a substitution table called the **key**. For example: the plaintext word “doors” would be enciphered to “KFFML” using the substitution table:

Cipher	Plain
K	d
F	o
M	r
L	s

The decipherment goal is to recover the plaintext given the ciphertext.

3 Decipherment Model

Inspired by character-level neural machine translation (NMT), we view decipherment as a sequence-to-sequence translation task. The motivation behind using a sequence-to-sequence model is:

- The model can be trained on multilingual data (Gao et al., 2020), making it potentially possible to obtain end-to-end multilingual decipherment without relying on a separate language ID step.
- Due to transcription challenges of historical ciphers (Section 5.4), ciphertext could be noisy. We would like the model to have the ability to recover from that noise by inserting, deleting, or substituting characters while generating plaintext. Sequence-to-sequence models seem to be good candidates for this task.

³<https://cl.lingfil.uu.se/~bea/copiale/>

3.1 Decipherment as a Sequence-to-Sequence Translation Problem

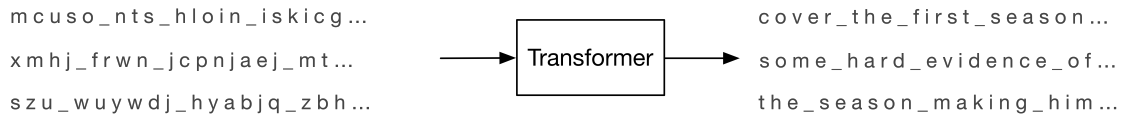
To cast decipherment as a supervised translation task, we need training data, i.e. pairs of $\langle f_1^N, e_1^M \rangle$ to train on. We can create this data using randomly generated substitution keys (Figure 2a). We can then train a character-based sequence-to-sequence decipherment model and evaluate it on held-out text which is also encrypted with (different) randomly generated substitution keys. However, if we attempt this experiment using the Transformer model described in Section 3.3, we get abysmal results (see Section 5.1 for scoring details).

Increasing the amount of training data won’t help; there are $26! \approx 4 \times 10^{26}$ possible keys for English ciphers, and even if every key is represented, most of the training data will still be encoded with keys that are not used to encode the test data. In fact, since each training *example* uses a different key, we cannot assume that a character type has any particular meaning. The fundamental assumption behind embeddings is therefore broken. In the next section, we describe one way to overcome these challenges.

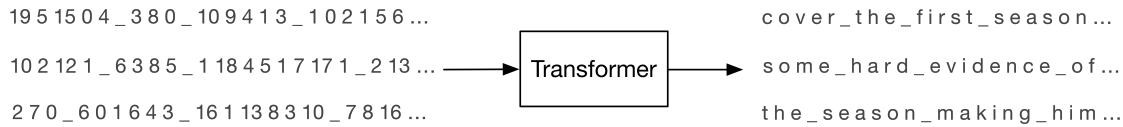
3.2 Frequency Analysis

To address the aforementioned challenges, we employ a commonly used technique in cryptanalysis called **frequency analysis**. Frequency analysis is attributed to the great polymath, Al-Kindi (801-873 C.E.) (Dooley, 2013). This technique has been used in previous decipherment work (Hauer and Kondrak, 2016; Kambhatla et al., 2018). It is based on the fact that in a given text, letters and letter combinations (n-grams) appear in varying frequencies, and that the character frequency distribution is *roughly* preserved in any sample drawn from a given language. So, in different pairs of $\langle f_1^N, e_1^M \rangle$, we expect the frequency distribution of characters to be similar.

To encode that information, we re-map each ciphertext character to a value based on its frequency rank (Figure 2b). This way, we convert any ciphertext to a “frequency-encoded” cipher. Intuitively, by frequency encoding, we are reducing the number of possible substitution keys (assuming frequency rank is roughly preserved across all ciphers from a given language). This is only an approximation, but it helps restore the assumption that there is a coherent connection between a symbol and its type embedding. For example, if the letters “e” and “i”



(a) Input: Example ciphers encoded in random keys. Output: Plaintext in target language.



(b) Input: Example ciphers encoded according to frequency ranks in descending order. Output: Plaintext in target language.

Figure 2: Decipherment as a sequence-to-sequence translation problem. (a) shows the original ciphers being fed to the model. (b) shows the same ciphers after frequency encoding.

are the most frequent characters in English, then in any 1:1 substitution cipher, they will be encoded as “0” or “1” instead of a randomly chosen character.

3.3 The Transformer

We follow the character-based NMT approach in Gao et al. (2020) and use the Transformer model (Vaswani et al., 2017) for our decipherment problem. The Transformer is an attention-based encoder-decoder model that has been widely used in the NLP community to achieve state-of-the-art performance on many sequence modeling tasks. We use the standard Transformer architecture, which consists of six encoder layers and six decoder layers as described in Gao et al. (2020).

4 Data

For training, we create 1:1 substitution ciphers for 14 languages using random keys. For English, we use English Gigaword (Parker et al., 2011). We scrape historical text from Project Gutenberg for 13 other languages, namely: Catalan, Danish, Dutch, Finnish, French, German, Hungarian, Italian, Latin, Norwegian, Portuguese, Spanish, and Swedish.⁴ Table 1 summarizes our datasets. Following previous literature (Nuhn et al., 2013; Aldarrab, 2017; Kambhatla et al., 2018), we lowercase all characters and remove all non-alphabetic and non-space symbols. We make sure ciphers do not end in the middle of a word. We strip accents for languages other than English.

5 Experimental Evaluation

To make our experiments comparable to previous work (Nuhn et al., 2013; Kambhatla et al., 2018),

⁴Our dataset is available at <https://github.com/NadaAldarrab/s2s-decipherment>

Language	Words	Characters
Catalan	915,595	4,953,516
Danish	2,077,929	11,205,300
Dutch	30,350,145	177,835,527
Finnish	22,784,172	168,886,663
French	39,400,587	226,310,827
German	3,273,602	20,927,065
Hungarian	497,402	3,145,451
Italian	4,587,027	27,786,754
Latin	1,375,804	8,740,808
Norwegian	706,435	3,673,895
Portuguese	10,841,171	62,735,255
Spanish	20,165,731	114,663,957
Swedish	3,008,680	16,993,146

Table 1: Summary of data sets obtained from Project Gutenberg.

we create test ciphers from the English Wikipedia article about History.⁵ We use this text to create ciphers of length 16, 32, 64, 128, and 256 characters. We generate 50 ciphers for each length. We follow the same pre-processing steps to create training data.

We carry out four sets of experiments to study the effect of cipher length, space encipherment/removal, unknown plaintext language, and transcription noise. Finally, we test our models on a real historical cipher, whose plaintext language was not known until recently.

As an evaluation metric, we follow previous literature (Kambhatla et al., 2018) and use Symbol Error Rate (SER). SER is the fraction of incorrect symbols in the deciphered text. For space restoration experiments (Section 5.2), we use Translation Edit Rate (TER) (Snover et al., 2006), but on the

⁵<https://en.wikipedia.org/wiki/History>

character level. We define character-level TER as:

$$\text{TER} = \frac{\# \text{ of edits}}{\# \text{ of reference characters}} \quad (1)$$

where possible edits include the insertion, deletion, and substitution of single characters. When the ciphertext and plaintext have equal lengths, SER is equal to TER.

We use FAIRSEQ to train our models (Ott et al., 2019). We mostly use the same hyperparameters as Gao et al. (2020) for character NMT, except that we set the maximum batch size to 10K tokens and use half precision floating point computation for faster training. The model has about 44M parameters. Training on a Tesla V100 GPU takes about 110 minutes per epoch. We train for 20 epochs. Decoding takes about 400 character tokens/s. We use a beam size of 100. Unless otherwise stated, we use 2M example ciphers to train, 3K ciphers for tuning, and 50 ciphers for testing in all experiments. We report the average SER on the 50 test ciphers of each experiment.

5.1 Cipher Length

We first experiment with ciphers of length 256 using the approach described in Section 3.1 (i.e. we train a Transformer model on pairs of $\langle f_1^N, e_1^M \rangle$ without frequency encoding). As expected, the model is not able to crack the 50 test ciphers, resulting in an SER of 71.75%. For the rest of the experiments in this paper, we use the frequency encoding method described in Section 3.2.

Short ciphers are more challenging than longer ones. Following previous literature, we report results on different cipher lengths using our method. Table 2 shows decipherment results on ciphers of length 16, 32, 64, 128, and 256. For the 256 length ciphers, we use the aforementioned 2M train and 3K development splits. For ciphers shorter than 256 characters, we increase the number of examples such that the total number of characters remains nearly constant, at about 512M characters. We experiment with training five different models (one for each length) and training a single model on ciphers of mixed lengths. In the latter case, we also use approx. 512M characters, divided equally among different lengths. The results in Table 2 show that our model achieves comparable results to the state-of-the-art model of Kambhatla et al. (2018) on longer ciphers, including perfect decipherment for ciphers of length 256. The table also

shows that our method is more accurate than Kambhatla et al. (2018) for shorter, more difficult ciphers of lengths 16 and 32. In addition, our method provides the ability to train on multilingual data, which we use to attack ciphers with an unknown plaintext language as described in Section 5.3.

5.2 No-Space Ciphers

The inclusion of white space between words makes decipherment easier because word boundaries can give a strong clue to the cryptanalyst. In many historical ciphers, however, spaces are hidden. For example, in the Copiale cipher (Figure 1a), spaces are enciphered with special symbols just like other alphabetic characters (Knight et al., 2011). In other ciphers, spaces might be omitted from the plain text before enciphering, as was done in the Zodiac-408 cipher (Nuhn et al., 2013). We test our method in four scenarios:

1. Ciphers with spaces (comparable to Kambhatla et al. (2018)).
2. Ciphers with enciphered spaces. In this case, we treat space like other cipher characters during frequency encoding as described in Section 3.2.
3. No-space ciphers. We omit spaces in both (source and target) sides.
4. No-space ciphers with space recovery. We omit spaces from source but keep them on the target side. The goal here is to train the model to restore spaces along with the decipherment.

Table 3 shows results for each of the four scenarios on ciphers of length 256. During decoding, we force the model to generate tokens to match source length. Results show that the method is robust to both enciphered and omitted spaces. In scenario 4, where the model is expected to generate spaces and thus the output length differs from the input length, we limit the output to exactly 256 characters, but we allow the model freedom to insert spaces where it sees fit. The model generates spaces in accurate positions overall, leading to a TER of 1.88%.

5.3 Unknown Plaintext Language

While combing through libraries and archives, researchers have found many ciphers that are not accompanied with any cleartext or keys, leaving the plaintext language of the cipher unknown (Megyesi

	Cipher Length				
	16	32	64	128	256
Beam NLM (Kambhatla et al., 2018)	26.80	5.80	0.07	0.01	0.00
Beam (NLM + FreqMatch) (Kambhatla et al., 2018)	31.00	2.90	0.07	0.02	0.00
Transformer + Freq + separate models (this work)	20.62	1.44	0.41	0.02	0.00
Transformer + Freq + single model (this work)	19.38	2.44	1.22	0.02	0.00

Table 2: SER (%) for solving 1:1 substitution ciphers of various lengths using our decipherment method.

Cipher Type	TER(%)
Ciphers with spaces	0.00
Ciphers with enciphered spaces	0.00
No-space ciphers	0.77
No-space ciphers + generate spaces	1.88

Table 3: TER (%) for solving 1:1 substitution ciphers of length 256 with different spacing conditions.

et al., 2020). To solve that problem, we train a single multilingual model on the 14 different languages described in Section 4. We train on a total of 2.1M random ciphers of length 256 (divided equally among all languages). We report results as the number of training languages increases while keeping the total number of 2.1M training examples fixed (Table 4). Increasing the number of languages negatively affects performance, as we expected. However, our experiments show that the 14-language model is still able to decipher 700 total test ciphers with an average SER of 0.68%. Since we are testing on 256-character ciphers, this translates to no more than two errors per cipher on average.

5.4 Transcription Noise

Real historical ciphers can have a lot of noise. This noise can come from the natural degradation of historical documents, human mistakes during a manual transcription process, or misspelled words by the author, as in the Zodiac-408 cipher. Noise can also come from automatically transcribing historical ciphers using Optical Character Recognition (OCR) techniques (Yin et al., 2019). It is thus crucial to have a robust decipherment model that can still crack ciphers despite the noise.

Hauer et al. (2014) test their proposed method on noisy ciphers created by randomly corrupting $\log_2(N)$ of the ciphertext characters. However, automatic transcription of historical documents is

very challenging and can introduce more types of noise, including the addition and deletion of some characters during character segmentation (Yin et al., 2019). We test our model on three types of random noise: insertion, deletion, and substitution. We experiment with different noise percentages for ciphers of length 256 (Table 5). We report the results of training (and testing) on ciphers with only substitution noise and ciphers that have all three types of noise (divided equally). We experimentally find that training the models with 10% noise gives the best overall accuracy, and we use those models to get the results in Table 5. Our method is able to decipher with up to 84% accuracy on ciphers with 20% of random insertion, deletion, and substitution noise. Figure 3 shows an example output for a cipher with 15% noise. The model recovers most of the errors, resulting in a TER of 5.86%. One of the most challenging noise scenarios, for example, is the deletion of the last two characters from the word “its.” The model output the word “i,” which is a valid English word. Of course, the more noise there is, the harder it is for the model to recover due to error accumulation.

5.5 The Borg Cipher

The Borg cipher is a 400-page book digitized by the Biblioteca Apostolica Vaticana (Figure 1b).⁶ The first page of the book is written in Arabic script, while the rest of the book is enciphered using astrological symbols. The Borg cipher was first automatically cracked by Aldarrab (2017) using the noisy-channel framework described in Knight et al. (2006). The plaintext language of the book is Latin. The deciphered book reveals pharmacological knowledge and other information about that time.

We train a Latin model on 1M ciphers and use

⁶http://digi.vatlib.it/view/MSS_Borg_lat.898.

# lang	ca	da	nl	en	fi	fr	de	hu	it	la	no	pt	es	sv	avg
3	-	-	-	0.04	-	0.23	-	-	-	-	-	-	0.39	-	0.29
7	-	-	-	0.08	-	0.34	0.30	-	1.23	1.38	-	0.48	0.40	-	0.60
14	0.34	1.29	0.79	0.25	0.20	0.20	0.41	0.64	1.52	1.43	0.41	0.69	0.72	0.70	0.68

Table 4: SER (%) for solving 1:1 substitution ciphers using a multilingual model trained on a different number of languages. Each language is evaluated on 50 test ciphers generated with random keys.

Source	3 2 11 11 2 6 4 15 0 _ 16 0 1 6 _ d 20 12 9 i5 2 4 3 1 _ 2 3 _ d 15 0 3 6 _ 2 s22 _ 18 i16 0 9 9 _ 2 1 _ 6 13 0 _ 1 4 i7 19 3 4 5 4 10 2 3 i13 10 0 _ 7 5 _ 8 d 5 5 0 11 0 3 6 _ 10 2 14 1 0 i21 1 _ 2 3 8 _ 0 5 5 0 10 6 1 i0 _ 13 4 1 6 7 s5 4 2 3 s6 _ 2 9 1 7 i18 _ 8 0 16 2 6 0 _ 6 13 0 _ 3 2 6 14 d 0 _ s3 5 _ d 4 1 6 7 d 17 _ s5 3 8 _ 4 d d _ 14 1 0 5 s0 9 3 0 1 1 _ 16 17 _ 8 i5 4 1 10 14 1 1 4 s23 19 _ s2 13 0 _ 1 s11 14 s24 17 _ 7 5 _ 6 13 i21 0 _ 8 4 1 10 4 12 9 4 3 0 _ 2 1 _ 2 i7 3 _ 0 3 8 _ 4 3 _ s5 6 1 0 s14 s12 _ 2 3 8 _ 1 d _ 2 _ 18 d 17 _ 7 i20 5 i9 _ 12 11 7 15 4 8 4 s2 19 _ 12 0 11 i12 0 d d 10 d 4 15 0
Target	narrative_best_explains_an_event_as_well_as_the_signific ance_of_different_causes_and_effect s_historians_also_de bate_the_nature_of_h istor y_and_i ts _usefulness_by_disc ussing_the_study_of_the_discipline_as_an_end_in_itself_a nd_as_a_way_of_providing_perspectiv e
Output	narrative_best_explains_an_event_as_well_as_the_signific ance_of_different_causes_and_effect ive _historians_also_ debate_the_nature_of_v is i tor s _and_i _usefulness_by_d iscussing_the_study_of_the_discipline_as_an_end_in_itself _and_as_a_way_of_providing_perspectiv _

Figure 3: Example system output for a cipher with 15% random noise (shown in red). Substitutions, insertions, and deletions are denoted by letters s, i, and d, respectively. The system recovered 34/40 errors (TER is 5.86%). Highlighted segments show the errors that the system failed to recover from.

% Noise	Noise Type	
	sub	sub, ins, del
5	1.10	2.87
10	2.40	5.87
15	5.28	10.58
20	11.48	16.17
25	17.63	27.43

Table 5: TER (%) for solving 1:1 substitution ciphers with random insertion, deletion, and substitution noise. These models have been trained with 10% noise.

the first 256 characters of the Borg cipher to test our model. Our model is able to decipher the text with an SER of 3.91% (Figure 4). We also try our 14-language multilingual model on this cipher, and obtain an SER of 5.47%. While we cannot directly compare to Aldarrab (2017), who do not report SER, this is a readable decipherment and can be easily corrected by Latin scholars who would be interested in such a text.

6 Anagram Decryption

To further test the capacity of our model, we experiment with a special type of noise. In this section, we address the challenging problem of solving substitution ciphers in which letters within each word have been randomly shuffled. Anagramming is a technique that can be used to further disguise substitution ciphers by permuting characters. Various theories about the mysterious Voynich Manuscript, for example, suggest that some anagramming scheme was used to encode the manuscript (Reddy and Knight, 2011). Hauer and Kondrak (2016) propose a two-step approach to solve this problem. First, they use their 1:1 substitution cipher solver (Hauer et al., 2014) to decipher the text. The solver is based on tree search for the key, guided by character-level and word-level n-gram language models. They adapt the solver by relaxing the letter order constraint in the key mutation component of the solver. They then re-arrange the resulting deciphered characters using a word trigram language model.

Π	ϛ	ϛ	ϛ	∞	ϛ	H	*	H	η	*	ϛ	*			
c	a	l	a	m	e	n	t	i	t	h	i	m	i		
♀	Δ	ϛ	∞	ϛ	*	Π	ϛ	ϛ	γ	Δ	*				
<u>b</u>	<u>u</u>	<u>l</u>	<u>e</u>	<u>g</u>	<u>i</u>	<u>c</u>	<u>a</u>	<u>r</u>	<u>d</u>	<u>u</u>					
ϛ	∞	ϛ	∞	γ	*	Π	H	*	ϛ	□	△	ϛ	Δ	ϛ	
<u>v</u>	<u>e</u>	<u>n</u>	<u>e</u>	<u>d</u>	<u>i</u>	<u>c</u>	<u>t</u>	<u>i</u>	<u>r</u>	<u>o</u>	<u>s</u>	<u>a</u>	<u>r</u>	<u>u</u>	<u>m</u>
ϛ	∞	ϛ	H	η	∞	Π	ϛ	*	△	♀	∞				
<u>m</u>	<u>e</u>	<u>n</u>	<u>t</u>	<u>h</u>	<u>e</u>	<u>c</u>	<u>r</u>	<u>i</u>	<u>s</u>	<u>e</u>					
ϛ	ϛ	ϛ	ϛ	*	△	*	4	∞	ϛ	*	Π	Δ	ϛ	*	
<u>a</u>	<u>n</u>	<u>a</u>	<u>n</u>	<u>i</u>	<u>s</u>	<u>i</u>	<u>p</u>	<u>e</u>	<u>n</u>	<u>i</u>	<u>c</u>	<u>u</u>	<u>l</u>	<u>i</u>	
□	□	*	ϛ	*	Δ	ϛ	H	η	*	Π	∞				
<u>o</u>	<u>f</u>	<u>i</u>	<u>m</u>	<u>i</u>	<u>u</u>	<u>r</u>	<u>t</u>	<u>h</u>	<u>i</u>	<u>c</u>					
ϛ	ϛ	∞	H	*	ϛ	ϛ	ϛ	∞	ϛ	*	Π	∞			
<u>a</u>	<u>n</u>	<u>e</u>	<u>t</u>	<u>i</u>	<u>a</u>	<u>n</u>	<u>g</u>	<u>e</u>	<u>l</u>	<u>i</u>	<u>c</u>	<u>e</u>			
4	∞	ϛ	*	Π	Δ	ϛ	*	ϛ	ϛ	H	η	∞	∞		
<u>p</u>	<u>e</u>	<u>n</u>	<u>i</u>	<u>c</u>	<u>u</u>	<u>l</u>	<u>i</u>	<u>a</u>	<u>l</u>	<u>t</u>	<u>h</u>	<u>e</u>	<u>e</u>		
△	η	Δ	*	ϛ	ϛ	∞	*	ϛ	*	γ	*	△			
<u>s</u>	<u>q</u>	<u>u</u>	<u>i</u>	<u>l</u>	<u>l</u>	<u>e</u>	<u>i</u>	<u>r</u>	<u>i</u>	<u>d</u>	<u>i</u>	<u>s</u>			

Figure 4: The first 132 characters of the Borg cipher and its decipherment. Errors are underlined. Correct words are: pulegi, benedicti, crispe, ozimi, and feniculi.

We try a one-step, end-to-end anagram decryption model. In our sequence-to-sequence formulation, randomly shuffled characters can confuse the training. We thus represent an input cipher as a bag of frequency-mapped characters, nominally presented in frequency rank order (Figure 5). We use the English Gigaword dataset to train a 256 character model on the sorted frequencies and test on the aforementioned test set of 50 ciphers (after applying random anagramming). Following Hauer and Kondrak (2016), we report word accuracy on this task. Our model achieves a word accuracy of 95.82% on the 50 Wikipedia ciphers.

Hauer and Kondrak (2016) report results on a test set of 10 long ciphers extracted from 10 Wikipedia articles about art, Earth, Europe, film, history, language, music, science, technology, and Wikipedia. Ciphers have an average length of 522 characters. They use English Europarl to train their language models (Koehn, 2005). To get comparable results, we trained a model on ciphers of length 525 created from the English side of the Spanish-English Europarl dataset. Our model achieved a word accuracy of 96.05% on Hauer and Kondrak’s test set. Training on English Gigaword gave a word accuracy of 97.16%, comparable to the 97.72% word accuracy reported by Hauer and Kondrak (2016). This shows that our simple model can crack randomly anagrammed ciphers, which hopefully inspires future work on other cipher types.

7 Related Work

Deciphering substitution ciphers is a well-studied problem in the natural language processing community, e.g., (Hart, 1994; Olson, 2007; Ravi and Knight, 2008; Corlett and Penn, 2010; Nuhn et al., 2013, 2014; Hauer et al., 2014; Aldarrab, 2017). Many of the recent proposed methods search for the substitution table (i.e. cipher key) that leads to a likely target plaintext according to a character n-gram language model. The current state-of-the-art method uses beam search and a neural language model to score candidate plaintext hypotheses from the search space for each cipher, along with a frequency matching heuristic incorporated into the scoring function (Kambhatla et al., 2018). This method, which is comparable in results to our method on longer ciphers and slightly weaker on shorter ciphers, assumes prior knowledge of the target plaintext language. Our method, by contrast, can solve substitution ciphers from different languages without explicit language identification.

Recent research has looked at applying other neural models to different decipherment problems. Greydanus (2017) find an LSTM model can learn the decryption function of polyalphabetic substitution ciphers when trained on a concatenation of <key + ciphertext> as input and plaintext as output. Our work looks at a different problem. We target a ciphertext-only-attack for short 1:1 substitution ciphers. Gomez et al. (2018) propose CipherGAN, which uses a Generative Adversarial Network to find a mapping between the character embedding distributions of plaintext and ciphertext. This method assumes the availability of plenty of ciphertext. Our method, by contrast, does not require a large amount of ciphertext. In fact, all of our experiments were evaluated on ciphers of 256 characters or shorter.

Early work on language identification from ciphertext uses the noisy-channel decipherment model (Knight et al., 2006). Specifically, the expectation-maximization algorithm is used to learn mapping probabilities, guided by a pre-trained n-gram language model. This decipherment process is repeated for all candidate languages. The resulting decipherments are ranked based on the probability of the ciphertext using the learned model, requiring a brute-force guess-and-check approach that does not scale well as more languages are considered. Hauer and Kondrak (2016) use techniques similar to ours, incorporating character

```

(1) t h e _ i n v e n t i o n _ o f _ w r i t i n g _ s y s t e m s
(2) j c z _ m r b z r j m k r _ k f _ w u m j m r e _ a o a j z g a
(3) c j z _ k z m r b r j m r _ f k _ e w u j m m r _ z g o a j a a
(4) 6 0 3 _ 5 3 1 2 7 2 0 1 2 _ 8 5 _ 11 9 10 0 1 1 2 _ 3 13 12 4 0 4 4
(5) 0 3 6 _ 0 1 1 2 2 2 3 5 7 _ 5 8 _ 0 1 1 2 9 10 11 _ 0 3 4 4 4 12 13
(6) t h e _ i n v e n t i o n _ o f _ b r i t a i n _ s y s t e m s

```

Figure 5: Example anagram encryption and decryption process: (1) original plaintext (2) after applying a 1:1 substitution key (3) after anagramming (this is the ciphertext) (4) after frequency encoding (5) after sorting frequencies. This is fed to Transformer (6) system output (errors are highlighted).

frequency, decomposition pattern frequency, and trial decipherment in order to determine the language of a ciphertext.

8 Conclusion and Future Work

In this work, we present an end-to-end decipherment model that is capable of solving simple substitution ciphers without the need for explicit language identification. We use frequency analysis to make it possible to train a multilingual Transformer model for decipherment. Our method is able to decipher 700 ciphers from 14 different languages with less than 1% SER. We apply our method on the Borg cipher and achieve 5.47% SER using the multilingual model and 3.91% SER using a monolingual Latin model. In addition, our experiments show that these models are robust to different types of noise, and can even recover from many of them. To the best of our knowledge, this is the first application of sequence-to-sequence neural models for decipherment.

We hope that this work drives more research in the application of contextual neural models to the decipherment problem. It would be interesting to develop other techniques for solving more complex ciphers, e.g. homophonic and polyalphabetic ciphers.

Acknowledgements

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental

purposes notwithstanding any copyright annotation thereon.

Ethics Statement

This work, like all decipherment work, is concerned with the decoding of encrypted communications, and thus the methods it describes are designed to reveal information that has been deliberately obfuscated and thus violate the privacy of the authors. However, the class of problems it addresses, 1:1 substitution ciphers, are known to be relatively weak forms of encryption, once popular, but long considered obsolete. Thus, the major practical use of this work as a decryption tool is in the ability to quickly decode ancient ciphertexts, such as the Borg cipher, the contents of which are interesting for historical purposes but are not in danger of revealing secrets of any living person. Modern encryption schemes such as RSA, Blowfish, or AES cannot be defeated by the methods presented here.

We have demonstrated our work’s effectiveness on ciphers of 14 alphabetic languages. The approaches presented here may be less effective on other orthographic systems such as abjads (which have fewer explicit symbols and more inherent ambiguity), abugidas (which have more explicit symbols and thus are conceivably less tractable), or logographic systems (which have many more explicit symbols). We caution that more exploration needs to be done before relying on the methods presented here when decoding ancient historical ciphertexts that are not encodings of alphabetic plaintext.

It is possible, though unlikely, that incorrect conclusions can be drawn if the approaches presented in this work yield false results. For instance, in Figure 1b, the word decoded as *peniculi* (towels) should in fact be decoded as *feniculi* (fennel); similar examples can be seen in Figure 3. The translation “seed of towels” being far less likely than

“seed of fennel“ in context, we would expect easy detection of this kind of error. We recommend that these methods not be trusted exclusively, but rather that they be used as one tool in a cryptologist’s kit, alongside language expertise and common sense, such that incoherent decodings may be given a careful look and correction.

References

- Nada Aldarrab. 2017. [Decipherment of historical manuscripts](#). Master’s thesis, University of Southern California.
- Eric Corlett and Gerald Penn. 2010. [An exact A* method for deciphering letter-substitution ciphers](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1040–1047, Uppsala, Sweden. Association for Computational Linguistics.
- John F. Dooley. 2013. *A Brief History of Cryptology and Cryptographic Algorithms*. Springer International Publishing.
- Yingqiang Gao, Nikola I. Nikolov, Yuhuang Hu, and Richard H.R. Hahnloser. 2020. [Character-level translation with self-attention](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604, Online. Association for Computational Linguistics.
- Aidan N. Gomez, Sicong Huang, Ivan Zhang, Bryan M. Li, Muhammad Osama, and Lukasz Kaiser. 2018. [Unsupervised cipher cracking using discrete GANs](#). *CoRR*, abs/1801.04883.
- Sam Greydanus. 2017. [Learning the enigma with recurrent neural networks](#). *CoRR*, abs/1708.07576.
- George W. Hart. 1994. [To decode short cryptograms](#). *Commun. ACM*, 37(9):102–108.
- Bradley Hauer, Ryan Hayward, and Grzegorz Kondrak. 2014. [Solving substitution ciphers with combined language models](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2314–2325, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Bradley Hauer and Grzegorz Kondrak. 2016. [Decoding anagrammed texts written in an unknown language and script](#). *TACL*, 4:75–86.
- Nishant Kambhatla, Anahita Mansouri Bigvand, and Anoop Sarkar. 2018. [Decipherment of substitution ciphers with neural language models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 869–874, Brussels, Belgium. Association for Computational Linguistics.
- Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. [The copiale cipher](#). In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 2–9, Portland, Oregon. Association for Computational Linguistics.
- Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. [Unsupervised analysis for decipherment problems](#). In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 499–506, Sydney, Australia. Association for Computational Linguistics.
- Philipp Koehn. 2005. [Europarl: A parallel corpus for statistical machine translation](#). In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.
- Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. [Decryption of historical manuscripts: the decrypt project](#). *Cryptologia*, 44(6):545–559.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. [Beam search for solving substitution ciphers](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1576, Sofia, Bulgaria. Association for Computational Linguistics.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2014. [Improved decipherment of homophonic ciphers](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1764–1768, Doha, Qatar. Association for Computational Linguistics.
- Edwin Olson. 2007. [Robust dictionary attack of short simple substitution ciphers](#). *Cryptologia*, 31(4):332–342.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [FAIRSEQ: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. [Gigaword fifth edition LDC2011T07](#).
- Eva Pettersson and Beata Megyesi. 2019. [Matching keys and encrypted manuscripts](#). In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 253–261, Turku, Finland. Linköping University Electronic Press.
- Sujith Ravi and Kevin Knight. 2008. [Attacking decipherment problems optimally with low-order N-gram models](#). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language*

Processing, pages 812–819, Honolulu, Hawaii. Association for Computational Linguistics.

Sravana Reddy and Kevin Knight. 2011. [What we know about the voynich manuscript](#). In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 78–86, Portland, OR, USA. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Xusen Yin, Nada Aldarrab, Beata Megyesi, and Kevin Knight. 2019. [Decipherment of historical manuscript images](#). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 78–85.