CipherBusters: automatically busting classical ciphers 👻

John Chung, Alex Ding, Megan Frisella

Introduction

Classical encryption schemes such as Caesar and Vigenère ciphers are traditionally cracked using statistical methods like frequency analysis. Instead, we model decryption as a language translation task and use deep learning models as an alternative solution, which has the potential to generalize to arbitrary cipher schemes.

Data

We use the English Wikipedia corpus as plaintext and ran various enciphering schemes (Caesar, Vigenère, and substitution) to create ciphertext for training. We reduce the character space to 26 English letters, 10 Arabic numbers, and whitespace by lowercasing, normalizing spacing between words, and removing all other characters. We then tokenize and one-hot encode fixed-sized windowed character sequences as matrices.

We sample from 20,000 Wikipedia articles for a total of 58,222,708 characters. For Caesar cipher, we encode and store the data using all 36 possible keys. For Vigenère and substitution ciphers, due to the huge key space, we dynamically sample encryption keys and encode the plaintext during training whenever a new batch is requested.

Plaintext	the 1935 washington huskies football team was an american football team
Caesar (Key=3)	wkh 4c68 zdvklqjwrq kxvnlhv irrwedoo whdp zdv dq dphulfdq irrwedoo whdp
Vigenère (Key=Bust)	u1w kann fbcz1o0b7o 1cbl2wb g86ccu34 uys5 xua to u4xs2uto z67uvs4m dwtn
Substitution	j1p hial tcf15v0jqv 1bfs5pf uqqj2c66 jpco tcf cv cop85dcv uqqj2c66 jpco

Table 1. Examples of all three ciphering schemes.

Methodology

We tackle increasingly difficult decryption tasks while exploring a variety of model architectures.

Decryption Task

1. Caesar cipher with one particular key.
2. Generalized Caesar ciphers.
3. Generalized Vigenère ciphers.
4. Generalized substitution ciphers

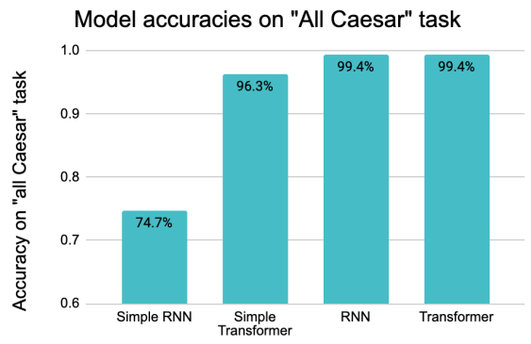
Architecture

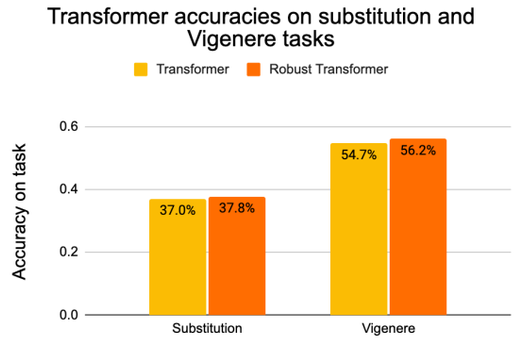
In general, our architectures process the ciphertext as character-by-character sequential data and predict probability distributions amongst the vocabulary. We create a simple LSTM and a simple transformer, each consisting of just the LSTM/transformer layer and a few fully connected layers. We add an encoding layer that synthesizes information from the input to improve on the simple architectures. Lastly, we create a robust transformer by stacking multiple transformer layers, as well as increasing parameter sizes.

Training

For all experiments, we train our model for 20 epochs, using an Adam optimizer with learning rate 0.003, and we use window size 50. We use crossentropy loss, and accuracy is evaluated as the percentage of correctly decoded characters.

Results





Encrypted	Decrypted
mtzwzrt343 nzyqt2x zn4z053p3 3xl24 m54 wtvp 4sp9 34tww zn4z053p3 wp43 yz4 rp4 lsplo zq z523pw6p3	biologists confirm octopuses smart but like they still octopuses lets not get ahead of ourselves
fgy60d ywd7ea9 d0fdwyfe x4ff0d d08wd6e w1f0d 149w77k 20ff492 wyy0bf0z fa xdai9	tucker carlson retracts bitter remarks after finally getting accepted to brown
qjy ymj wzqnsl hqfxxjx ywjrgqj fy f htrrzsnxynh wj0tqzynts	let the ruling classes tremble at a communistic revolution

Table 2. Examples of trained transformer decrypting arbitrary Caesar ciphers.

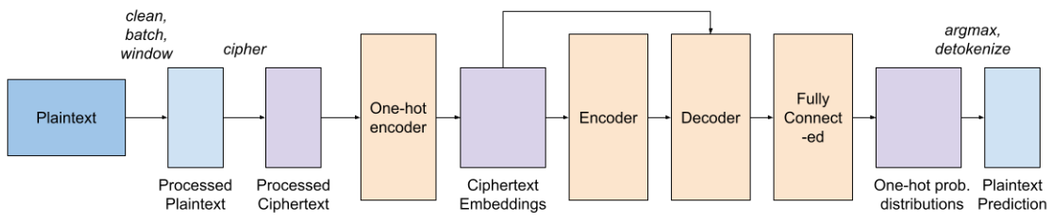


Figure 1. Our model's pipeline. The encoder and decoder can be either LSTMs or transformers.

Discussion

Our best models are able to achieve near-perfect accuracy on generalized Caesar ciphers. However, even the robust transformer struggles to learn generalized Vigenère and substitution ciphers, though the model has clearly learned some insight into the decryption, since it vastly outperforms random guessing. While existing statistical methods can crack these ciphers with near perfect accuracy, deep learning methods underperform in more complicated ciphers. It is important to note that existing methods assume knowledge of the type of cipher used, whereas our methods have the potential to generalize to multiple types of ciphers and form an end-to-end decryption pipeline that can handle any arbitrary ciphertext.

One direction of future work is to add an adversarial component to our training pipeline to punish implausible decryptions. As it is, our model often devolves into guessing the most common letters during training. The adversarial loss can help mitigate such loss-gaming behavior.

References

1. Focardi, R. and Luccio, F.L., 2018. Neural Cryptanalysis of Classical Ciphers. In ICTCS (pp. 104-115).
2. Kopal, N., 2020, May. Of Ciphers and Neurons–Detecting the Type of Ciphers Using Artificial Neural Networks. In Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020 (No. 171, pp. 77-86). Linköping University Electronic Press.
3. Z., D., 2021, February. An Exposition of Neural Cryptanalysis of Classical Ciphers.

BROWN

https://devpost.com/software/cipherbusters
https://github.com/cipherbusters/cipherbusters