Decipherment of Substitution Ciphers with Neural Language Models

Nishant Kambhatla, Anahita Mansouri Bigvand, Anoop Sarkar
School of Computing Science
Simon Fraser University
Burnaby, BC , Canada
{nkambhat,amansour,anoop}@sfu.ca

Abstract

Decipherment of homophonic substitution ciphers using language models (LMs) is a well-studied task in NLP. Previous work in this topic scores short local spans of possible plaintext decipherments using n-gram LMs. The most widely used technique is the use of beam search with n-gram LMs proposed by Nuhn et al. (2013). We propose a beam search algorithm that scores the entire candidate plaintext at each step of the decipherment using a neural LM. We augment beam search with a novel rest cost estimation that exploits the prediction power of a neural LM. We compare against the state of the art n-gram based methods on many different decipherment tasks. On challenging ciphers such as the Beale cipher we provide significantly better error rates with much smaller beam sizes.

1 Introduction

Breaking substitution ciphers recovers the *plaintext* from a *ciphertext* that uses a 1:1 or homophonic cipher key. Previous work using pretrained language models (LMs) for decipherment use n-gram LMs (Ravi and Knight, 2011; Nuhn et al., 2013). Some methods use the Expectation-Maximization (EM) algorithm (Knight et al., 2006) while most state-of-the-art approaches for decipherment of 1:1 and homophonic substitution ciphers use beam search and rely on the clever use of n-gram LMs (Nuhn et al., 2014; Hauer et al., 2014). Neural LMs globally score the entire candidate plaintext sequence (Mikolov et al., 2010). However, using a neural LM for decipherment is not trivial because scoring the entire candidate partially deciphered plaintext is computationally challenging. We solve both of these problems in this paper and provide an improved beam search based decipherment algorithm for homophonic ciphers that exploits pre-trained neural LMs for the first time.

2 Decipherment Model

We use the notation from Nuhn et al. (2013). Ciphertext $f_1^N = f_1..f_i..f_N$ and plaintext $e_1^N = e_1..e_i..e_N$ consist of vocabularies $f_i \in V_f$ and $e_i \in V_e$ respectively. The beginning tokens in the ciphertext (f_0) and plaintext (e_0) are set to "\$" denoting the beginning of a sentence. The substitutions are represented by a function $\phi : V_f \rightarrow V_e$ such that 1:1 substitutions are *bijective* while homophonic substitutions are *general*. A cipher function ϕ which does not have every $\phi(f)$ fixed is called a partial cipher function (Corlett and Penn, 2010). The number of fs that are fixed in ϕ is given by its cardinality. ϕ' is called an extension of ϕ, if f is fixed in ϕ' such that $\delta(\phi'(f), \phi(f))$ yields true $\forall f \in V_f$ which are already fixed in ϕ where δ is Kronecker delta. **Decipherment** is then the task of finding the ϕ for which the probability of the deciphered text is maximized.

$$\hat{\phi} = \arg\max_{\phi} p(\phi(f_1)...\phi(f_N)) \tag{1}$$

where $p(.)$ is the language model (**LM**). Finding this argmax is solved using a beam search algorithm (Nuhn et al., 2013) which incrementally finds the most likely substitutions using the language model scores as the ranking.

2.1 Neural Language Model

The advantage of a neural LM is that it can be used to score the entire candidate plaintext for a hypothesized partial decipherment. In this work, we use a state of the art byte (character) level neural LM using a multiplicative LSTM (Radford et al., 2017).

Consider a sequence $S = w_1, w_2, w_3, ..., w_N$. The LM score of S is $\texttt{SCORE}(S)$:

$$\begin{aligned} P(S) &= P(w_1, w_2, w_3, ..., w_N) \\ P(S) &= \prod_{i=1}^{N} P(w_i \mid w_1, w_2, ..., w_{i-1})) \\ \texttt{SCORE}(S) &= -\sum_{i=1}^{N} log(P\left(w_i \mid w_{<i}\right)) \end{aligned} \tag{2}$$

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 869–874
Brussels, Belgium, October 31 - November 4, 2018.

2.2 Beam Search

Algorithm 1 is the beam search algorithm (Nuhn et al., 2013, 2014) for solving substitution ciphers. It monitors all partial hypotheses in lists H_s and H_t based on their quality. As the search progresses, the partial hypotheses are extended, scored with `SCORE` and appended to H_t. `EXT_LIMITS` determines which extensions should be allowed and `EXT_ORDER` picks the next cipher symbol for extension. The search continues after pruning: $H_s \leftarrow$ `HISTOGRAM_PRUNE`(H_t). We augment this algorithm by updating the `SCORE` function with a neural LM.

Algorithm 1 Beam Search for Decipherment

```
function (BEAM_SEARCH (EXT_ORDER, EXT_ LIMITS))
    initialize sets $H_s$, $H_t$
    CARDINALITY = 0
    $H_s$.ADD(($\emptyset$,0))
    while CARDINALITY < $|V_f|$ do
        $f$ = EXT_ORDER[CARDINALITY]
        for all $\phi \in H_s$ do
            for all $e \in V_e$ do
                $\phi$' := $\phi \cup \{(e, f)\}$
                if EXT_LIMITS($\phi$') then
                    $H_t$.ADD($\phi$',SCORE($\phi$'))
        HISTOGRAM_PRUNE($H_t$)
        CARDINALITY = CARDINALITY + 1
        $H_s = H_t$
        $H_t$.CLEAR()
    return WINNER($H_s$)
```

3 Score Estimation (`SCORE`)

Score estimation evaluates the quality of the partial hypotheses ϕ. Using the example from Nuhn et al. (2014), consider the vocabularies $V_e = \{a, b, c, d\}$ and $V_f = \{A, B, C, D\}$, extension order (B, A, C, D), and ciphertext `$ ABDDCABCDADCABDC $`. Let $\phi = \{(a, A), (b, B))\}$ be the partial hypothesis. Then `SCORE`(ϕ) scores this hypothesized partial decipherment (only A and B are converted to plaintext) using a pre-trained language model in the hypothesized plaintext language.

3.1 Baseline

The initial rest cost estimator introduced by Nuhn et al. nuhnbeam computes the score of hypotheses only based on partially deciphered text that builds a shard of n adjacent solved symbols. As a heuristic, n-grams which still consist of unsolved cipher-symbols are assigned a trivial estimate of probability 1. An improved version of rest cost estimation (Nuhn et al., 2014) consults lower order n-grams to score each position.

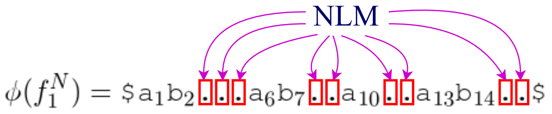
3.2 Global Rest Cost Estimation

The baseline scoring method greatly relies on local context, *i.e.* the estimation is strictly based on partial character sequences. Since this depends solely on the n-gram LM, the true conditional probability under Markov assumption is not modeled and, therefore, context dependency beyond the window of $(n-1)$ is ignored. Thus, attempting to utilize a higher amount of context can lower the probability of some tokens resulting in poor scores.

We address this issue with a new improved version of the rest cost estimator by supplementing the partial decipherment $\phi(f_1^N)$ with predicted plaintext text symbols using our neural language model (NLM). Applying $\phi = \{(a, A), (b, B))\}$ to the ciphertext above, we get the following partial hypothesis:

$$\phi(f_1^N) = \texttt{\$a}_1\texttt{b}_2\texttt{...a}_6\texttt{b}_7\texttt{..a}_{10}\texttt{..a}_{13}\texttt{b}_{14}\texttt{..\$}$$

We introduce a scoring function that is able to score the entire plaintext including the missing plaintext symbols. First, we sample[1] the plaintext symbols from the NLM at all locations depending on the deciphered tokens from the partial hypothesis ϕ such that these tokens maintain their respective positions in the sequence, and at the same time are sampled from the neural LM to fit (probabilistically) in this context. Next, we determine the probability of the entire sequence including the scores of sampled plaintext as our rest cost estimate.


NLM
$\phi(f_1^N) = \texttt{\$a}_1\texttt{b}_2\boxed{.}\boxed{.}\boxed{.}\texttt{a}_6\texttt{b}_7\boxed{.}\boxed{.}\texttt{a}_{10}\boxed{.}\boxed{.}\texttt{a}_{13}\texttt{b}_{14}\boxed{.}\boxed{.}\texttt{\$}$

In our running example, this would yield a score estimation of the partial decipherment, $\phi(f_1^N)$:

$$\phi(f_1^N) = \$\ \mathrm{a}_1\mathrm{b}_2\mathrm{d}_3\mathrm{c}_4\mathrm{c}_5\mathrm{a}_6\mathrm{b}_7\mathrm{c}_8\mathrm{d}_9\mathrm{a}_{10}\mathrm{d}_{11}\mathrm{d}_{12}\mathrm{a}_{13}\mathrm{b}_{14}\mathrm{d}_{15}\mathrm{c}_{16}\ \$$$

Thus, the neural LM is used to predict the score of the full sequence. This method of global scoring evaluates each candidate partial decipherment by scoring the entire message, augmented by the sam-

[1] The char-level sampling is done incrementally from left to right to generate a sequence that contains the deciphered tokens from ϕ at the exact locations they occur in the above $\phi(f_1^N)$. If the LM prediction contradicts the hypothesized decipherment we stop sampling and start from the next character.

Cipher	Length	Unique Symbols	Obs/ symbol
Zodiac-408	408	54	7.55
Beale Pt. 2	763	180	4.23

Table 1: Homophonic ciphers used in our experiments.

pled plaintext symbols from the NLM. Since more terms participate in the rest cost estimation with global context, we use the plaintext LM to provide us with a better rest cost in the beam search.

3.3 Frequency Matching Heuristic

Alignment by frequency similarity (Yarowsky and Wicentowski, 2000) assumes that two forms belong to the same lemma when their relative frequency fits the expected distribution. We use this heuristic to augment the score estimation ($\texttt{SCORE}$):

$$\texttt{FMH}(\phi') = \left| log\left(\frac{\nu(f)}{\nu(e)}\right)\right| \qquad f \in V_f, \quad e \in V_e \tag{3}$$

$\nu(f)$ is the percentage relative frequency of the ciphertext symbol f, while $\nu(e)$ is the percentage relative frequency of the plaintext token e in the plaintext language model. The closer this value to 0, the more likely it is that f is mapped to e.

Thus given a ϕ with the $\texttt{SCORE}(\phi)$, the extension ϕ' (Algo. 1) is scored as:

$$\texttt{SCORE}(\phi') = \texttt{SCORE}(\phi) + \texttt{NEW}(\phi') - \texttt{FMH}(\phi') \tag{4}$$

where $\texttt{NEW}$ is the score for symbols that have been newly fixed in ϕ' while extending ϕ to ϕ'. Our experimental evaluations show that the global rest cost estimator and the frequency matching heuristic contribute positively towards the accuracy of different ciphertexts.

4 Experimental Evaluation

We carry out 2 sets of experiments: one on letter based 1:1, and another on homophonic substitution ciphers. We report Symbol Error Rate (SER) which is the fraction of characters in the deciphered text that are incorrect.

The character NLM uses a single layer multiplicative LSTM (mLSTM) (Radford et al., 2017) with 4096 units. The model was trained for a single epoch on mini-batches of 128 subsequences of length 256 for a total of 1 million weight updates. States were initialized to zero at the beginning of each data shard and persisted across updates to simulate full-backprop and allow for the forward propagation of information outside of a given subsequence. In all the experiments we use a character NLM trained on English Gigaword corpus augmented with a short corpus of plaintext letters of about 2000 words authored by the Zodiac killer[2].

4.1 1:1 Substitution Ciphers

In this experiment we use a synthetic 1:1 letter substitution cipher dataset following Ravi and Knight (2008), Nuhn et al. (2013) and Hauer et al. (2014). The text is from English Wikipedia articles about history[3], preprocessed by stripping the text of all images, tables, then lower-casing all characters, and removing all non-alphabetic and non-space characters. We create 50 cryptograms for each length 16, 32, 64, 128 and 256 using a random Caesar-cipher 1:1 substitution.

Length	Beam	SER(%) 1	SER(%) 2
64	100	4.14	4.14
	1,000	1.09	1.04
	10,000	0.08	0.12
	100,000	0.07	0.07
128	100	7.31	7.29
	1,000	1.68	1.55
	10,000	0.15	0.09
	100,000	0.01	0.02

Table 2: Symbol Error Rates (SER) based on Neural Language Model and beam size (Beam) for solving 1:1 substitution ciphers of lengths 64 and 128, respectively. SER 1 shows beam search with global scoring, and SER 2 shows beam search with global scoring with frequency matching heuristic.

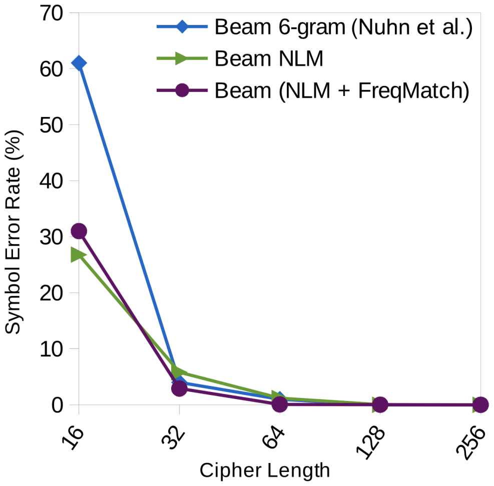


Figure 1: Symbol error rates for decipherment of 1:1 substitution ciphers of different lengths. The beam size is 100k. Beam 6-gram model uses the beam search from Nunh et al. (2013).

[2] https://en.wikisource.org/wiki/Zodiac_Killer_letters

[3] http://en.wikipedia.org/wiki/History

I	H	A	V	E	D	E	P	O	S	I	T	E	D	I	N	T	H	E	C	O	U	N	T	Y	O
115	73	24	807	37	52	49	17	31	62	647	22	7	15	140	47	29	107	79	84	56	239	10	26	811	5
F	B	E	D	F	O	R	D	A	B	O	U	T	F	O	U	R	M	I	L	E	S	F	R	O	M
196	308	85	52	160	136	59	211	36	9	46	316	554	122	106	95	53	58	2	42	7	35	122	53	31	82
B	U	F	O	R	D	S	I	N	A	N	E	X	C	A	V	A	T	I	O	N	O	R	V	A	U
77	250	196	56	96	118	71	140	287	28	353	37	1005	65	147	807	24	3	8	12	47	43	59	807	45	316
L	T	S	I	X	F	E	E	T	B	E	L	O	W	T	H	E	S	U	R	F	A	C	E	O	F
101	41	78	154	1005	122	138	191	16	77	49	102	57	72	34	73	85	35	371	59	196	81	92	191	106	273
T	H	E	G	R	O	U	N	D	T	H	E	F	O	L	L	O	W	I	N	G	A	R	T	I	C
60	394	620	270	220	106	388	287	63	3	6	191	122	43	234	400	106	290	314	47	48	81	96	26	115	92
L	E	S	B	E	L	O	N	G	I	N	G	J	O	I	N	T	L	Y	T	O	T	H	E	P	A
158	191	110	77	85	197	46	10	113	140	353	48	120	106	2	607	61	420	811	29	125	14	20	37	105	28

Figure 2: First few lines from part two of the Beale cipher. The letters have been capitalized.

Fig 1 plots the results of our method for cipher lengths of 16, 32, 64, 128 and 256 alongside Beam 6-gram (the best performing model) model (Nuhn et al., 2013)

4.2 An Easy Cipher: Zodiac-408

Zodiac-408, a homophonic cipher, is commonly used to evaluate decipherment algorithms.

Beam	SER (%) 1	SER (%) 2
10k	3.92	3.18
100k	2.40	1.91
1M	1.47	**1.22**

Table 3: Symbol Error Rates (SER) based on Neural Language Model and beam size (Beam) for deciphering Zodiac-408, respectively. SER 1 shows beam search with global scoring, and SER 2 shows beam search with global scoring with the frequency matching heuristic.

Our neural LM model with global rest cost estimation and frequency matching heuristic with a beam size of 1M has SER of 1.2% compared to the beam search algorithm (Nuhn et al., 2013) with beam size of 10M with a 6-gram LM which gives an SER of 2%. The improved beam search (Nuhn et al., 2014) with an 8-gram LM, however, gets 52 out of 54 mappings correct on the Zodiac-408 cipher.

4.3 A Hard Cipher: Beale Pt 2

Part 2 of the Beale Cipher is a more challenging homophonic cipher because of a much larger search space of solutions. Nunh et al. (2014) were the first to automatically decipher this Beale Cipher.

With an error of 5% with beam size of 1M vs 5.4% with 8-gram LM and a pruning size of 10M, our system outperforms the state of the art (Nuhn et al., 2014) on this task.

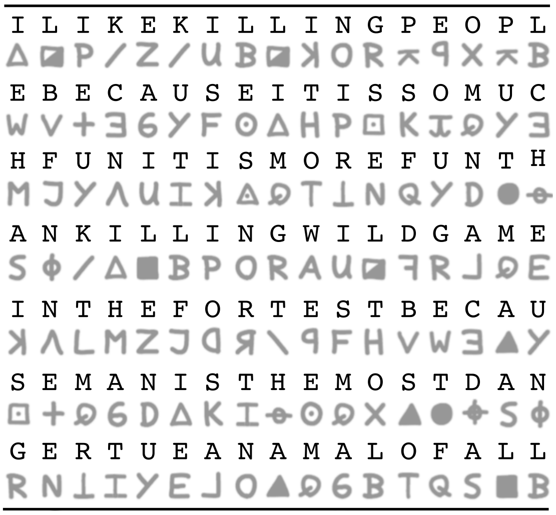

I L I K E K I L L I N G P E O P L
E B E C A U S E I T I S S O M U C
H F U N I T I S M O R E F U N T H
A N K I L L I N G W I L D G A M E
I N T H E F O R T E S T B E C A U
S E M A N I S T H E M O S T D A N
G E R T U E A N A M A L O F A L L

Figure 3: First 119 characters from deciphered Zodiac-408 text. The letters have been capitalized.

Beam	SER (%) 1	SER (%) 2
10k	41.67	48.33
100k	7.20	10.09
1M	**4.98**	5.50

Table 4: Symbol Error Rates (SER) based on Neural Language Model and beam size (Beam) for deciphering Part 2 of the Beale Cipher. SER 1 shows beam search with global scoring, and SER 2 shows beam search with global scoring with the frequency matching heuristic.

5 Related Work

Automatic decipherment for substitution ciphers started with dictionary attacks (Hart, 1994; Jakobsen, 1995; Olson, 2007). Ravi and Knight (2008) frame the decipherment problem as an integer linear programming (ILP) problem. Knight et al. (2006) use an HMM-based EM algorithm for solving a variety of decipherment problems. Ravi and Knight (2011) extend the HMM-based EM approach with a Bayesian approach, and report the first automatic decipherment of the Zodiac-408 cipher.

Berg-Kirkpatrick and Klein (2013) show that a large number of random restarts can help the EM approach.Corlett and Penn (2010) presented an efficient A* search algorithm to solve letter substitution ciphers. Nuhn et al. (2013) produce better results in faster time compared to ILP and EM-based decipherment methods by employing a higher order language model and an iterative beam search algorithm. Nuhn et al. (2014) present various improvements to the beam search algorithm in Nuhn et al. (2013) including improved rest cost estimation and an optimized strategy for ordering decipherment of the cipher symbols. Hauer et al. (2014) propose a novel approach for solving mono-alphabetic substitution ciphers which combines character-level and word-level language model. They formulate decipherment as a tree search problem, and use Monte Carlo Tree Search (MCTS) as an alternative to beam search. Their approach is the best for short ciphers.

Greydanus (2017) frames the decryption process as a sequence-to-sequence translation task and uses a deep LSTM-based model to learn the decryption algorithms for three polyalphabetic ciphers including the Enigma cipher. However, this approach needs supervision compared to our approach which uses a pre-trained neural LM. Gomez et al. (2018) (CipherGAN) use a generative adversarial network to learn the mapping between the learned letter embedding distributions in the ciphertext and plaintext. They apply this approach to shift ciphers (including Vigenère ciphers). Their approach cannot be extended to homophonic ciphers and full message neural LMs as in our work.

6 Conclusion

This paper presents, to our knowledge, the first application of large pre-trained neural LMs to the decipherment problem. We modify the beam search algorithm for decipherment from Nuhn et al. (2013; 2014) and extend it to use global scoring of the plaintext message using neural LMs. To enable full plaintext scoring we use the neural LM to sample plaintext characters which reduces the beam size required. For challenging ciphers such as Beale Pt 2 we obtain lower error rates with smaller beam sizes when compared to the state of the art in decipherment for such ciphers.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful remarks. The research was also partially supported by the Natural Sciences and Engineering Research Council of Canada grants NSERC RGPIN-2018-06437 and RGPAS-2018-522574 and a Department of National Defence (DND) and NSERC grant DGDND-2018-00025 to the third author.

References

Taylor Berg-Kirkpatrick and Dan Klein. 2013. Decipherment with a million random restarts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 874–878.

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. Association for Computational Linguistics.

Aidan N. Gomez, Scng Huang, Ivan Zhang, Bryan M. Li, Muhammad Osama, and ukasz Kaiser. 2018. Unsupervised cipher cracking using discrete gans. *arXiv preprint arXiv:1801.04883*.

Sam Greydanus. 2017. Learning the enigma with recurrent neural networks. *arXiv preprint arXiv:1708.07576*.

George W Hart. 1994. To decode short cryptograms. *Communications of the 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.

Thomas Jakobsen. 1995. A fast method for cryptanalysis of substitution ciphers. *Cryptologia*, 19(3):265–274.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 499–506. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

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)*, volume 1, pages 1568–1576.

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.

Edwin Olson. 2007. Robust dictionary attack of short simple substitution ciphers. *Cryptologia*, 31(4):332–342.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 812–819. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Bayesian inference for zodiac and other homophonic ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 239–247. Association for Computational Linguistics.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 207–216. Association for Computational Linguistics.