IBM Research Report

A Speech Recognition Solution to an Ancient Cryptography Problem

Peder Olsen[1], John Hershey[2], Steven Rennie[1], Vaibhava Goel[1]

[1]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

[2]Mitsubishi Electrics Research Laboratory

A Speech Recognition Solution to an Ancient Cryptography Problem

Peder Olsen[1], *John Hershey*[2], *Steven Rennie*[1] *and Vaibhava Goel*[1],
[1]T.J. Watson Research Center, IBM, {pederao,sjrennie,vgoel}@us.ibm.com,
[2]Mitsubishi Electrics Research Lab, hersheymerl.com

January 25, 2011

Abstract

The common substitution cipher that shifts every letter in the alphabet n letters to the left or right is known to every cereal box reader. Perhaps less known is its extension to *poly-alphabetic substitution ciphers* invented in 1553 where consecutive letters are shifted by different amounts. This can be according to a *periodic key* or a *one-time pad*, introduced in 1917. The randomly generated non-repeating key, i.e. one-time pad or running key cipher, is not possible to decipher. However, if the non-repeating key is text, a common encryption mistake, then the message can occasionally be deciphered. In this paper we show how to decipher using the Viterbi algorithm. The case of repeating keys, both English text or random, will be treated as well. We measure the error rate of the decryption on a test set where the messages and keys are known. We further resolve a cipher text for which the original message and key was not available to us.

Index Terms: Vigenère cipher, poly-alphabetic substitution cipher, Viterbi, Speech Separation

1 Introduction

In this paper we assume a standard English alphabet consisting of letters A=0, B=1, ..., Z=25 represented by corresponding integers in $\mathbb{Z}_{26}$. Caesar's code, also known as a shift cipher, is a mono alphabetic substitution cipher. Each letter of the alphabet is replaced by another letter. Julius Caesar was known to have used this cipher as early as the first century A.D. For example

Alphabet:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:	DEFGHIJKLMNOPQRSTUVWXYZABC

The Vigenère cipher was first described in 1553 by Giovan Battista Bellaso. It was considered virtually unbreakable up until the 19th century. It uses a series of different shift ciphers based on the letter of a key. Given a letter in the key, k, and a letter in the message, m, the encrypted cipher letter, c, is computed by modular arithmetic in $\mathbb{Z}_{26}$: $c \equiv k + m \pmod{26}$. As an example consider::

Message:	ATTACKATDAWN
Key:	LEMONLEMONLE
Cipher:	LXFOPVEFRNHR

A heuristic decryption method for a Vigenère cipher with periodic key consists of estimating or guessing the period of the key and then comparing frequencies of unigram or bigram letter sequences to find how far each letter is shifted. This method requires a lot of text to enable decryption since we need to be able to compare frequencies. This method of decryption does not use or require the key to be a word – it could indeed be a random letter of sequences. A common mistake is to take the key from a book or newspaper. The problem is then simpler and less text is needed to decipher.

If the key is non-repeating or very long the approach of comparing frequencies does not work. Such a cipher is known as a running key cipher or a one-time pad. It was first introduced by Gilbert Vernam of AT&T in 1917 and was used throughout World War II. The original heuristic approach consisted of placing common

words like "the" at different positions in the key and probing the corresponding message for word segments. No systematic method existed until Griffing's 2006 paper, (Griffing, 2006), which applied letter language models to the problem.
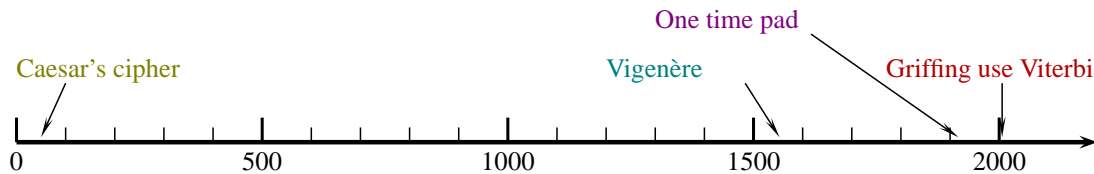

Figure 1: Time line for events significant to the paper.

We describe the Viterbi algorithm, (Viterbi, 1967; Rabiner and Juang, 1993), applied to this problem. Next we show that it does not suffice to decipher the running key cipher as claimed in (Griffing, 2006). An example of a running key cipher is

Message:	ATTACKATDAWNWITHTHREEBATTALIONS
Key:	THEFUTUREBELONGSTOTHOSEWHODARET
Cipher:	TAXFWDUKHBAYKVZZMVKLSTEPAOOIFRL

This example also happens to be one of the cases where our non-repeating Viterbi algorithm works and gives the exact message and key text. We extend the Viterbi approach to repeating keys, where it works quite well.

2   Letter Sequence Models

The basic approach is to define a probabilistic model of letter sequences. To find the secret message we look for the most likely message letter sequence $\mathbf{m}$ and key letter sequence $\mathbf{k}$ that satisfy the constraint $\mathbf{m} + \mathbf{k} \equiv \mathbf{c} (\mathrm{mod}\ 26)$, where $\mathbf{c}$ is the cipher text. To compute the likelihood of a letter sequence $\mathbf{l} = l_{1:N} = (l_1, l_2, \ldots, l_N)$ we use the n-gram decomposition

$$
\begin{aligned}
P(l_{1:N}) &= P(l_1)P(l_2|l_1)\cdots P(l_N|l_{1:N-1}) \\
&\approx P(l_1)P(l_2|l_1)\cdots P(l_{n+1}|l_{1:n}) \cdot \\
&\quad P(l_{n+2}|l_{2:n+1})\cdots P(l_N|l_{N-n:N-1})
\end{aligned}
\tag{1}
$$

2.1   N-gram letter models

A letter n-gram probability can be estimated from a training corpus by corresponding n-gram frequencies. Let the training corpus contain N letters and denote the number of occurences of the letter n-gram $l_{1:n} = (l_1, l_2, \ldots, l_n)$ by $c(l_{1:n})$. The n-gram frequency estimators,

$$
r_n(l_{1:n}) = c(l_{1:n})/N
\tag{2}
$$

$$
q_k(l_{n+1}|l_{1:n}) = \frac{c(l_{1:n})}{c(l_{1:n-1})}
\tag{3}
$$

are then consistent estimators for the probabilities $P(l_{1:n})$ and $P(l_{n+1}|l_{1:n})$.

For large n it is well known that such a model does not give good estimates on test data due to the existence of previously unseen, n-grams. There are a number of choices for smoothing techniques that address this problem. Witten–Bell smoothing is used in (Griffing, 2006). We chose to use interpolated ngrams. Kneser–Ney is also a very good smoothing choice. An excellent survey of smoothing techniques can be found in (Chen and Goodman, 1998).

2

2.2 Interpolated Letter n-grams

An interpolated n-gram model is a mixture model of letter n-gram models where the interpolation weights are trained on some held out data. Thus the model can be written

$$P(l_{n+1}|l_{1:n}) \approx \sum_{k=1}^{n} \lambda_k q_k(l_{n+1}|l_{n-k+1:n}), \tag{4}$$

where the weights λ_k are chosen to maximize the likelihood on some held out data. This is achieved using the EM algorithm or the Baum Welch algorithm, (Baum and Eagon, 1967). In the expectation step we compute the posterior counts

$$\gamma_k(l_{i:i+n+1}) = \frac{\lambda_k q_k(l_{i+n+1}|l_{i+n+1-k:i+n})}{\sum_{j=1}^{n} \lambda_j q_j(l_{i+n+1}|l_{i+n+1-j:i+n})}, \tag{5}$$

and in the maximization step we update the parameters

$$\lambda_k = \sum_{i=1}^{N-n-1} \gamma_k(l_{i:i+n+1}). \tag{6}$$

The iteration constitutes the EM algorithm, and each step increases the log likelihood. We initialized the weights λ_k to 1.

3 Viterbi Algorithm

There are 26 possible letter pairs consistent with the cipher letter. Given a hypothesized key and message letter sequence of length k *and* the $k + 1$th cipher letter, there are 26 ways to choose the next letter for the message and key. We compute the sum of the log likelihoods of the message and key for each choice. Thus for each length k message and key hypothesis we get 26 new length $k + 1$ hypotheses along with their log likelihood scores. We keep the top scoring n_{beam} new hypotheses, and iterate until the cipher letters are exhausted.

Consider a hypothesized message letter sequence "ATTA", key letter sequence "THEF" and corresponding cipher sequence "TAXF". If the next cipher letter is W the possible message and key letter pairs are:

Message:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Key:	WVUTSRQPONMLKJIHGFEBCBAZYX.

The procedure is shown in Table 1. For the first 3 letter pairs the score for the cipher is the sum of the log likelihoods for both of the sequences: The best next letter pair was CU, but we keep the best n_{beam} extensions, since they may lead to higher likelihood in the end.

4 Periodic Assumption

So far we have addressed the running key ciphers. For repeating keys there is additional redundancy that can be exploited should we know the length of the key. We modify the Viterbi algorithm to take this into account. In the periodic key example we split the cipher into segments:

Message:	ATTAC	KATDA	WN
Key:	LEMON	LEMON	LE
Cipher:	LXFOP	VEFRN	HR

Given the key there is a unique message matching the cipher. Thus we only need to search key letter sequences. In stage one, the left context for each segment is unknown, so we compute the log likelihood within each segment independently. In stage two we compute the log likelihood across segment boundaries.

In stage one the Viterbi algorithm uses the sum of the log likelihoods of each message segment and that of the key. For example with a decoding of the letter L in the key we can compute the score for next letter being E as follows:

3

Decoded	Guess	Probability	log likelihood
ATTA	A?	$P(\texttt{A}|\texttt{ATTA})$	-3.8
THEF	W?	$P(\texttt{W}|\texttt{THEF})$	-5.4
TAXF	W	total	-9.2
ATTA	B?	$P(\texttt{B}|\texttt{ATTA})$	-1.7
THEF	V?	$P(\texttt{V}|\texttt{THEF})$	-6.1
TAXF	W	total	-7.8
ATTA	C?	$P(\texttt{C}|\texttt{ATTA})$	-0.6
THEF	U?	$P(\texttt{U}|\texttt{THEF})$	-0.5
TAXF	W	total	**-1.1**!
⋮	⋮	⋮	⋮

Table 1: Viterbi algorithm in action. With a decoded message key pair of (`THEF`, `ATTA`) and a cipher `TAXF` with next cipher letter `W` we see how new letters are tried and the correct letter pair detected.

Segment	Source	Decoded	Guess	loglik
	key	L	E?	-2.6
1	message	A	T?	-2.2
	cipher	L	X FOP	
2	message	K	A?	-2.2
	cipher	V	E FRN	
3	message	W	N?	-3.7
	cipher	H	R	
Score: $\log(P(\texttt{E}|\texttt{L})P(\texttt{T}|\texttt{A})P(\texttt{A}|\texttt{K})P(\texttt{N}|\texttt{W}))$				-10.7

For the next letter of the key there will only be two segments and the score of the correct path will be $\log(P(\texttt{M}|\texttt{LE})P(\texttt{T}|\texttt{AT})P(\texttt{T}|\texttt{KA}))$ Stage one is complete at the last letter of the key. With a complete message hypothesis we have complete contextual information for all segments, but have not yet taken into account cross segment context. In stage two we re-evaluate the log likelihood for the entire message for each hypothesis and choose the top scoring hypothesis. For a trigram letter model the score for the middle segment of the correct path changes from $\log(P(\texttt{K})P(\texttt{A}|\texttt{K})P(\texttt{T}|\texttt{KA})P(\texttt{D}|\texttt{AT})P(\texttt{A}|\texttt{TD}))$ to $\log(P(\texttt{K}|\texttt{AC})P(\texttt{A}|\texttt{CK})P(\texttt{T}|\texttt{KA})P(\texttt{D}|\texttt{AT})P(\texttt{A}|\texttt{TD}))$ in stage 2.

4.1 Symmetry breaking property

In the case where the key is not periodic, no information in the model can distinguish the key from the message. We refer to this problem as the symmetry breaking problem. There is a tendency for the deciphering algorithm to switch from the key to the message in the middle of letter sequences. When the key is periodic the model for the key is different from the model of the message and we do not suffer the symmetry breaking problem.

5 Experiments

The experiments consisted of training letter language models and using these to extract key, message pairs from ciphers. We measure the error as the number of letter errors in the message.

5.1 Training Data

The training data used 10 classic books downloaded from the Gutenberg project. See Table 2 for the list. For held out data we used two additional books, "Alice's Adventures in Wonderland" by Charles Lutwidge Dodgson (aka Lewis Carroll) and "Adventures of Huckleberry Finn" by Samuel Langhorne Clemens (using pen name Mark Twain).

4

Book	Author
Animal Farm	George Orwell
A Tale of two Cities	Charles Dickens
Call of the Wild	Jack London
Crime and Punishment	Fyodor Dostoevsky
Gone with the Wind	Margaret Mitchell
Howards End	Edward Morgan Forster
The Great Gatsby	Francis Scott Key Fitzgerald
Tarzan of the Apes	Edgar Rice Burroughs
The Three Musketeers	Alexandre Dumas
Ulysses	James Joyce

Table 2: Books used as training data for building letter ngrams

In cryptography it is common to use only the plain alphabet without spacing. Spacing and punctuation give additional clues and are not recommended for encryption. Each of the books downloaded contains punctuation and spaces between words. We removed all but the English letters and capitalized all letters before making letter ngram models.

5.2 Test Data

For test data we carefully chose 7 quotes that did not appear in the training data. In (Griffing, 2006) the test set consisted of keys and messages from the training data! This gives an unfair impression that the algorithm works well. We do not repeat our experiments on the data used in (Griffing, 2006) for this reason. We used 7 idioms as keys and 7 cryptography like texts as messages. The messages and keys are displayed in Table 3 and 4. All pairs of keys and messages were used to form the test set. The perplexity on the held-out data set was roughly 4, so we can see that quite a few of the key and message texts are somewhat harder.

	Key text	len	pp
1:	ACTIONSSPEAKLOUDERTHANWORDS	27	3.2
2:	BETWEENAROCKANDAHARDPLACE	25	4.8
3:	CURIOSITYKILLEDTHECAT	21	5.3
4:	DRASTICTIMESCALLFORDRASTIC MEASURES	34	7.4
5:	PICKUPYOUREARS	14	4.7
6:	GREATMINDSTHINKALIKE	20	4.7
7:	HITTHENAILONTHEHEAD	19	5.8

Table 3: The 7 key texts along with number of letters (len) and perplexity (pp) for each key phrase.

5.3 Results for Running Key Ciphers

As mentioned earlier the problem with Griffing's paper was not in the Viterbi inference method, but rather in the evaluation of the deciphering algorithm. As we can see from Table 5 the results are not sufficiently good to be understood. For example message number 5 and key number 1 gave the deciphered message CHWITHTRANSFERONEITHERINALWAYSITWASDARKPERHANDCANDABABYOUWAR and key NGBANKSS-PEAKLOUDEVILHECOUGHLESSOFEYEARSOLDJOESHOUGHILOVKNEWT. It is difficult to make any sense of the deciphered message — the deciphering method makes everything look like words. Although "transfer one" is deciphered correctly, it is not enough to make sense out of the message. We chose to use $n = 7$ for the ngram length and $n_{\mathrm{beam}} = 1000$ for the example decodings. This gave a good result in a reasonable amount of time. As seen in the table the key with the lowest perplexity had the lowest Letter Error Rate (LER),

5

	Message text	len	pp
1:	ATTACKATDAWN	12	3.9
2:	THEBRITISHHAVEFIFTYTANKS	24	4.2
3:	FIRSTSOLVERWINSONEHUNDRED DOLLARS	32	4.5
4:	MOVEARMYACROSSDELAWAREAT MIDNIGHT	32	8.1
5:	PLEASETRANSFERONEMILLION DOLLARSTOMYSWISSBANK ACCOUNTSIXTWOSIX	60	5.9
6:	PRESIDENTSEMBARGORULINGSHOULD HAVEIMMEDIATENOTICEGRAVESITUA TIONAFFECTINGINTERNATIONALLAW STATEMENTFORESHADOWSRUINOFMAN YNEUTRALSYELLOWJOURNALSUNIFYI NGNATIONALEXCITEMENTIMMENSELY	172	5.5
7:	IHAVEADREAMTHATONEDAYTHISNATI ONWILLRISEUPANDLIVEOUTTHETRUE MEANINGOFITSCREEDWEHOLDTHESET RUTHSTOBESELFEVIDENTTHATALLME NARECREATEDEQUAL	132	4.1

Table 4: The 7 messages along with number of letters (len) and perplexity (pp) for each. Three of the messages are long enough that all the keys will have to repeat to encrypt the messages

and the key with the highest perplexity had the highest LER. In other words the more the text looks like the training data the easier it is to recover.

Table 6 shows that there was no clear benefit to going to ngram lengths beyond 7 and that a beam of 1000 is not far behind the beam of 100,000.

5.4 Results for Periodic Key Ciphers

Introducing the additional information that the key is periodic should give better reconstruction. The results of using the periodic Viterbi algorithm are displayed in Table 7. These results are dramatically better than before when there is significant repetition of the key. The two longest sequences are in fact reconstructed perfectly for all the keys! These results where obtained using the actual lengths of the keys. This could be considered cheating, but in practice it's a reasonable approach to try deciphering using different periods and

m \ k	1	2	3	4	5	6	7	LER
1	12	8	11	10	11	11	9	86
2	4	22	22	22	12	20	21	73
3	29	12	27	29	27	28	27	80
4	26	20	29	30	29	28	27	84
5	46	47	54	53	53	51	45	83
6	122	123	136	157	123	142	129	77
7	84	115	112	117	103	95	90	77
LER:	70	75	84	90	77	81	75	79

Table 5: Results of Viterbi deciphering using a search-beam of 1000 and ngram length of 7. The numbers in the interior of the table are raw number of letter errors. LER stands for letter error rate percentage. m/k stands for message/key.

6

ngram\beam	1000	10000	100000
5	88.05%	87.99%	87.87%
6	83.65%	83.68%	83.44%
7	78.82%	77.89%	76.72%
8	77.99%	77.19%	76.79%
9	78.97%	77.31%	76.91%

Table 6: Letter error rates for different ngram lengths and beam-sizes

attempt to read the resulting messages for each of the periods.

message\key	1	2	3	4	5	6	7	LER
1								
2			23		0	10	20	55
3	8	30	31		0	9	12	47
4	31	30	26		0	27	28	74
5	0	0	0	34	0	9	0	10
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
LER:	9	14	18	9	0	12	13	11

Table 7: Results of periodic Viterbi deciphering using known periods and a search-beam of 1000 and ngram length of 7. Only message key pairs where the message was longer than the key were deciphered.

5.5 Random periodic key

We tested our periodic algorithm with the random key WCHBO and message THEFUTUREBELONGSTOTHOSEWHODARE. The result was perfect reconstruction when given the key length of 5. This despite the mismatch in the model used to generate the key and to decipher the message.

5.6 An example deciphering

We found an example Vigenère cipher on the web;
            http://synapse.daiict.ac.in/perplex_problems_round2.pdf.

The problem was given with punctuation and white space in the message, which makes the deciphering simpler. We removed the punctuation and white space. The first paragraph of the cipher is:

```
BYIRLBFMVG SXFEJFJLXA MSVZIQHENK FIFCYJJRIF SEXRVCICDT EITHCBQVXS
GWEXFPZHHT JGSPLHUHRP FDBPXNLMFV TFMIGRBZJT XIGHTJDAMW VMSFXLHFMS
UXSDGEZDIE PCZLKLISCI JIWSIHTJVE VWVFMVWISO DFKIEQRQVL EPVHMYZSRW
CIMZGLWVQQ RAWRTZFKYV HOZIFJRDHG WVWKRRQSKM XOSFMVQEGS OJEXVHGBJT
XXRHTJFTMQ WASJSJPOZP ZRHUSCZZVI VHTFKXLHME MFYPGRQHCE VHHTJTEYVS
                     EBYMGKWYUV PXKSYYFXLH GQURVEWWAS
```

Viterbi deciphering algorithm on this problem gives the message

```
NGEVENTFRE EDOMFREEDO MFREEZEAND TERINGINFR OMFREEDOMF REEDOMFREE
DOMFREEDOM FREEDOMFRE EDOMFREEDO MFREEDOMFR EEDOMFREED OMFREEDOMF
REEDOMFREE DOMFREEDOM FREEDOMFRE EDOMFREEDO MFREEDOMFR EEDOMFREED
OMFREEDOMF REEDOMFREE DOMFREEDOM FISHANDSTH EHOBANDKNO WHEIRACQUI
TENATGREED OMFREEDOMF REEDOMFREE DOMFREEDOM FREEDOMFRE EDOMFREEDO
                     MTHISFARMT INGBUCCEED INGDELIRIU
```

and key

```
OSEWHOMHEC OURSEOFHUM ANEVERDEAH MEOULDBEDO ESSARYFORO NEPEOPLETO
DISSOLVETH EPOLITICAL BANDSWHICH HAVECONNEC TEDTHEMWIT HANOTHERAN
DTOASSUMEA MONGTHEPOW ERSOFTHEEA RTHTHESEPA RATEANDEQU ALSTATIONT
OWHICHTHEL AWSOFNATUR EANDOFNATU RNEDRENARF THEEMINUTE SCAPEHELPL
ETEHADOPIN IONSOFMANK INDREQUIRE STHATTHEYS HOULDDECLA RETHECAUSE
                  SIREOFWHIC HKEREWDTHE YDOORTOFSY.
```

Obviously the key and message have been swapped. We can plainly read the key "freedom" from the message. Using the periodic Viterbi algorithm with period of 7 we get the exact text for the message:

```
WHENINTHEC OURSEOFHUM ANEVENTSIT BECOMESNEC ESSARYFORO NEPEOPLETO
DISSOLVETH EPOLITICAL BANDSWHICH HAVECONNEC TEDTHEMWIT HANOTHERAN
DTOASSUMEA MONGTHEPOW ERSOFTHEEA RTHTHESEPA RATEANDEQU ALSTATIONT
OWHICHTHEL AWSOFNATUR EANDOFNATU RESGODENTI TLETHEMADE CENTRESPEC
TTOTHEOPIN IONSOFMANK INDREQUIRE STHATTHEYS HOULDDECLA RETHECAUSE
                  SWHICHIMPE LTHEMTOTHE SEPARATION
```

and the key FREEDOM. We have succesfully deciphered the cipher — It is simply an excerpt from the decla-ration of independence. We could have also found this by searching in google using the legible sections from the non-periodic Viterbi deciphered texts or by reading off the key word.

References

L. E. Baum and J. A. Eagon. 1967. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73:360–363.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University, August.

Alexander Griffing. 2006. Solving the running key cipher with the Viterbi algorithm. *Cryptologia*, 30:361–367.

Lawrence Rabiner and Bing-Hwang Juang. 1993. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Andrew J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–267, April.