# Probabilistic Methods for a Japanese Syllable Cipher

Sujith Ravi and Kevin Knight

Information Sciences Institute
Computer Science Department
University of Southern California
Marina del Rey, California 90292
{sravi,knight}@isi.edu

**Abstract.** This paper attacks a Japanese syllable-substitution cipher. We use a probabilistic, noisy-channel framework, exploiting various Japanese language models to drive the decipherment. We describe several innovations, including a new objective function for searching for the highest-scoring decipherment. We include empirical studies of the relevant phenomena, and we give improved decipherment accuracy rates.

**Keywords:** Substitution cipher, decipherment, language modeling

## 1 Introduction

In this paper, we use natural language processing techniques to attack a Japanese substitution cipher. In a substitution cipher, every token of a natural language (plaintext) sequence is replaced by a cipher token, according to some substitution key.

For example, an English plaintext:

"`HELLO WORLD ...`"

may be enciphered as:

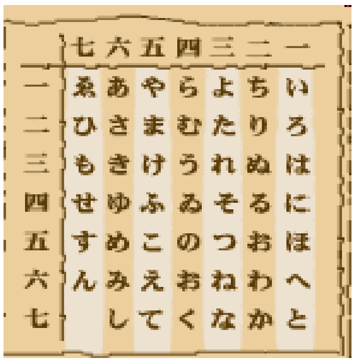
"`NOEEI TIMEL ...`"

according to the key:

```
P: ABCDEFGHIJKLMNOPQRSTUVWXYZ
C: XYZLOHANBCDEFGIJKMPQRSTUVW
```

If the recipients of the ciphertext message have the substitution key, they can use it (in reverse) to recover the original plaintext. Interestingly, a third party who intercepts the message may be able to guess the original plaintext by analyzing the repetition patterns in the ciphertext.

From a natural language perspective, we can view this cryptanalysis task as a kind of unsupervised tagging problem [1]. We must tag each ciphertext token with some element of the plaintext vocabulary. The resulting tag sequence must make sense, so we use language modeling (LM) techniques to rank proposed decipherments. We also need search techniques to find high-ranking decipherments quickly.



| | 七 | 六 | 五 | 四 | 三 | 二 | 一 |
|---|---|---|---|---|---|---|---|
| 一 | ゑ | あ | や | ら | よ | ち | い |
| 二 | ひ | さ | ま | む | た | り | ろ |
| 三 | も | き | け | う | れ | ぬ | は |
| 四 | せ | ゆ | ふ | ゐ | そ | る | に |
| 五 | す | め | こ | の | つ | を | ほ |
| 六 | ん | み | え | お | ね | わ | へ |
| 七 | | し | て | く | な | か | と |

**Fig. 1.** Original Uesugi cipher key in Japanese

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **1** | i | ro | ha | ni | ho | he | to |
| **2** | ti | ri | nu | ru | wo | wa | ka |
| **3** | yo | ta | re | so | tu | ne | na |
| **4** | ra | mu | u | <> | no | o | ku |
| **5** | ya | ma | ke | hu | ko | e | te |
| **6** | a | sa | ki | yu | me | mi | si |
| **7** | <> | hi | mo | se | su | n | <> |

**Fig. 2.** Transliterated version of the checkerboard-style key used for encrypting the Uesugi cipher

In this paper, we attack a particular Japanese syllable-substitution cipher from the Warring States Period, said to be employed by General Uesugi Kenshin.[1] The cipher employs the checkerboard-style key shown in Figures 1 and 2. The key is filled out according to the *i-ro-ha* alphabet, which uses each Japanese syllable exactly once. To encode a message, we look up each syllable in the key, and we replace it with a two-digit number. The first digit is the column index, and the second digit is the row index.

For example, the plaintext:

"wa ta ku si ..."

is enciphered as:

"62 23 74 76 ..."

Note that the key contains *ka* but not *ga*. In this system, plaintext *ga* is enciphered in the same way as *ka*, i.e., as 72. Likewise, *go* is enciphered the same as *ko*, and *gu* is enciphered the same as *ku*, and so forth. Note that this means the recipient may generate several readings from the cipher and must use common sense to determine which reading is most sensible. Alternatively, the sender can try to rephrase the original message in order to avoid unintended ambiguity.

Variations of the system exist. For example, instead of two digits, two characters can be used, so that the ciphertext has the outward appearance of written language. Also, the key can be scrambled periodically to increase security.

The goal of cryptanalysis is to take an intercepted number sequence and guess a plaintext for it, without the benefit of the key. In this paper, we investigate the number sequence in Figure 3.

[1] http://www.samurai-archives.com/kenshin.html
http://en.wikipedia.org/wiki/Cryptography_in_Japan
http://global.mitsubishielectric.com/misty/tour/stage2/index.html

```
11 53 33 54 64 51 67 71 36 41 72 67 13 34 55 72 34 11 16 25 23 26 45 14 27 23 25 27 35 42 73 72 41
11 71 11 16 67 55 71 73 36 36 31 41 31 16 14 32 72 57 74 33 75 71 36 56 36 23 25 45 16 22 35 22
31 76 56 13 22 62 33 31 71 64 37 27 16 72 22 23 25 61 42 64 51 67 72 23 72 23 56 26 25 76 36
37 54 41 64 71 76 56 43 63 66 23 25 45 64 73 76 51 71 43 33 13 22 35 14 45 54 72 34 11 23 12 31
25 76 75 11 16 57 72 14 57 16 26 46 45 54 66 11 16 53 72 61 41 53 35 75 37 27 71 54 55 55 21 52
54 66 34 55 72 76 34 14 66 52 64 45 53 37 22 41 11 16 16 22 35 67 11 71 16 53 76 74 73 22 46 36 37 54
55 55 21 51 43 35 41 26 71 72 12 73 42 52 11 13 11 13 16 72 57 16 31 33 73 42 37 54 41
64 37 51 76 75 27 71 54 43 76 22 52 37 65 31 31 72 14 47 23 25 31 57 13 54 23 56 76 41 37
73 22 32 61 36 64 51 67 37 75 73 76 73 22
72 67 23 12 56 34 61 27 71 73 71 37 16 11 73 74 56 52 43 31 56 53 53 11 71 25 31 46 36 27 71
54 64 51 37 51 11 65 73 22 37 21 55 76 41 37 72 72 42 55 71 54 64 55 22 41 55 43 13 37
66 23 33 16 76 72 22 35 33 71 11 16 34 11 16 34 16 56 54 76 23 41 37 16 12 36 73 34 27 71 54
37 75 73 11 16 66 74 26 41 73 22 75 11 16 34 36 27 54 23 56 76 37 27 36 11 75 53 61 74 73 22 46 74
41 11 71 31 76 23 73 36 55 71 64 51 72 33 71 72 23 76 35 73 36 66 55 55 21 31 61 54 23 74 27
73 36 52 23 54 66 41 75 25 76 14 27 23 25 45
12 12 54 23 11 73 55 67 31 73 74 73 22 75 31 31 36 23 54 72 23 73 67 11 41 76 61 54 13 76 16 42
41 75 64 11 16 34 12 74 76 26 76 16 23 22 75 13 54 64 51 11 65 31 73 11 16 72 73 42 64 51 67 72 23 72 23
41 37 11 23 34 64 71 14 57 73 41 55 71 54 36 76 36 52 37 37 75 73 76 23 25 27 35 33 71
71 22 23 75 75 31 72 31 72 76 36 34 76 21 66 76 73 35 33 31 55 71 16 42 71 36 31 73 51
13 22 71 55 21 73 74 55 55 21 51 43 35 73 22
```

**Fig. 3.** Encrypted Uesugi cipher sequence containing 577 syllables

## 2 Previous Work

In the past, researchers have explored many strategies for attacking decipherment problems [2, 3, 4]. We follow a probabilistic approach, as suggested by Knight et al. [5]. That paper proposes a noisy-channel model of ciphertext production. First, a plaintext $e$ is produced according to probability P($e$). Then, the plaintext is encoded as ciphertext $c$, according to probability P($c|e$). We estimate an n-gram model for P($e$) on separate plaintext data. We then adjust the P($c|e$) parameter values in order to maximize the probability of the observed (ciphertext) data. This probability can be written:

$$P(c) = \sum_{e} P(e) \cdot P(c|e) \tag{1}$$

The P($c|e$) quantity is the product of the probabilities of the individual token substitutions that transform $e$ into $c$, i.e., the substitution probabilities make up the guessed key. Knight et al. [5] propose the EM algorithm [6] as a method to guess the best probabilistic values for the key. Once the key is settled on, the Viterbi algorithm can search for the best decipherment:

$$argmax_e P(e|c) = argmax_e P(e) \cdot P(c|e) \tag{2}$$

Knight et al. [5] also develop a novel technique for improving decipherment accuracy. Prior to decoding, they stretch out the channel model probabilities, using the Viterbi algorithm to instead search for:

$$argmax_e P(e|c) = argmax_e P(e) \cdot P(c|e)^3 \tag{3}$$

Finally, they provide an evaluation metric (number of guessed plaintext tokens that match the original message), and they report results on English letter substitution decipherment.

## 3 Present Contributions

The novel contributions of this paper are:

- We attack a more difficult cipher system. The Uesugi cipher has more characters than English, it has no word boundaries, and even the correct key yields multiple decipherment candidates. (This last feature makes it unsuitable for methods such as [7]).
- We attack cipher lengths that are not solved by low-order language models. We present an empirical study of training sizes, perplexities, and memory requirements for n-gram language models, and we relate language-model perplexity to decipherment accuracy.
- We find that higher-order n-gram models paradoxically generate worse decipherments, according to the method of [5]. We invent an adjustment to the EM objective function that solves this problem. This method allows much more accurate decipherment of shorter ciphers.
- We study the impact of random restarts for EM (not employed by [5] for letter substitution ciphers), and we find additional significant effects on accuracy.

## 4 Experimental Set-up

In order to train language models over Japanese syllable sequences, we obtain a *roomaji* version[2] of the book *Tale of Genji* (c. 1021 AD). We process the roomaji into syllables and remove typographical errors. The final sequence contains approximately one million syllables and 65 unique syllable types. We split this data into three parts:

- LM training data (900,012 syllables).
- LM smoothing data (1419 syllables).
- Plaintext messages (various sizes).

We encipher the plaintext messages by first replacing *ga* with *ka*, and so forth, and then making substitutions according to the hidden i-ro-ha table 2. Our ciphertexts contain 46 unique types. Three of the cells in the $7 \times 7$ i-ro-ha table are unused in the cipher system.

Our evaluation metric for decipherment is the same as [5]—we count the number of matches between the guessed decipherment sequence and the original message. Note that a successful decipherment must not only re-create the correct

[2] http://etext.virginia.edu/japanese/genji/roman.html

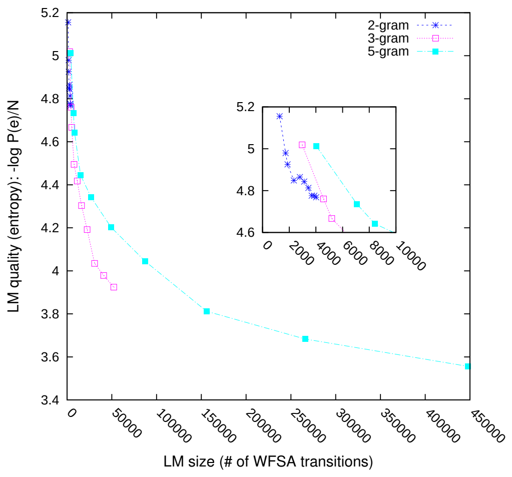


**Fig. 4.** Relationship between LM memory size and LM entropy. Plotted points represent language models trained on different amounts of data and different n-gram orders.

key, but it must determine (using plaintext context alone) whether a cipher token should be decoded as *ka* or *ga*, etc.

The most common syllable in our data is *to*. The baseline of replacing every ciphertext token with *to* yields an accuracy of 4%.

## 5 Experiments

To recap (from Equation 1), we search for values of the probabilistic substitution table P($c$|$e$) that maximize P($c$):

$$P(c) = \sum_{e} P(e) \cdot P(c|e)$$

P($c$|$e$) is a 65 x 46 table. We begin with uniform probabilities, so that each of the 65 plaintext characters maps to any of the 46 ciphertext numbers with probability 1/ 46.

### 5.1 Language Models

Decipherment requires knowledge about the general plaintext language (in our case, Japanese), and this is captured in the P(e) language model. In our experiments, we create n-gram models (at various orders) out of plaintext training data (of various sizes). In collecting n-gram counts, we drop singletons when n>3. We smooth by interpolating with lower-order n-grams. We estimate one smoothing $\lambda$ per n-gram order.

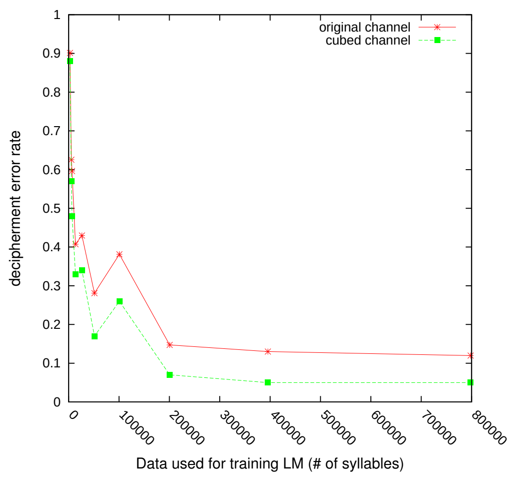


**Fig. 5.** Effect of LM training data size on decipherment error for the Uesugi cipher using 3-gram LM with (a) original channel decoding (b) cubing the channel probabilities before decoding (Knight et al. [5])

We represent LMs as weighted finite-state acceptors (WFSAs). For each LM, we measure its memory size (number of WFSA transitions) and its perplexity on held-out data. We measure perplexity on the plaintext messages, though of course, these messages are not used in the construction of any LMs. Figure 4 shows the relationship between memory requirements and LM entropy in bits/character (perplexity = $2^{entropy}$) of various LMs. Note that for any particular memory size a machine may have, we can select the LM order that gives the best perplexity.

We use EM to search for the best table P($c|e$). EM is an iterative algorithm that improves P($c$) from one iteration to the next, until convergence. Because exact convergence is typically not reached, we terminate when the iteration-over-iteration change in P($c$) falls below a set threshold. We terminate early if 200 iterations are reached.

Figure 5 shows decipherment results when we use a 3-gram LM trained on various amounts of data. We can see that more LM data (i.e., more knowledge about the Japanese language) leads to better decipherment. Figure 5 also confirms that maximizing Equation 3:

$$argmax_e P(e|c) = argmax_e P(e) \cdot P(c|e)^3$$

is better than maximizing Equation 2:

$$argmax_e P(e|c) = argmax_e P(e) \cdot P(c|e)$$

| | |
|---|---|
| **Ciphertext** | 11 53 33 54 64 51 67 71 36 41 72 67 13 34 55 72 34 11 16 25 23 26 45 14 27 23 25 27 35 42 73 72 41 11 71 11 16 67 55 71 73 36 36 31 41 31 16 14 32 72 57 74 33 75 71 36 56 36 23 ... |
| **Original message** | i du re no o ho n to ki ni ka n yo u go ka u i a ma ta sa bu ra hi ta ma hi ke ru na ka ni i to i a n go to na ki ki ha ni ha a ra nu ga su gu re te to ki me ki ta ... |
| **Best decipherment (3-gram LM)** | i du re no o ho n to ki ni ka n yo u go ka u i a ma ta sa bu ra hi ta ma hi ke ru na ka ni i to i a n go to na ki ki ha ni ha a ra $\underline{\text{zu}}$ $\underline{\text{ka}}$ su gu re te to ki me ki ta ... |

**Fig. 6.** Sample from the 577-syllable Uesugi cipher system showing (a) Ciphertext (b) Original message (c) Best decipherment using 3-gram LM (errors underlined)

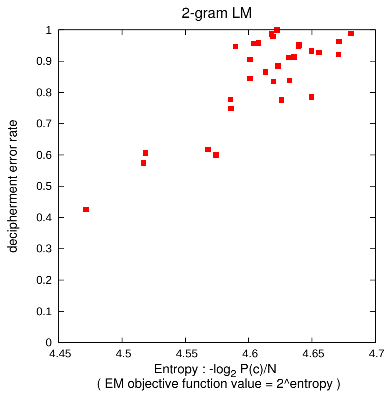


**Fig. 7.** Effect of random restarts on decipherment with a 2-gram model. Each point measures the result of a single EM run, in terms of EM's objective function (x-axis) and decipherment error (y-axis).

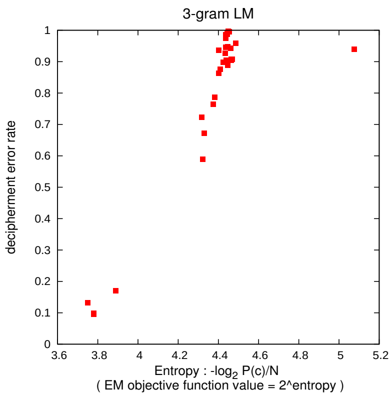


**Fig. 8.** Effect of random restarts on decipherment with a 3-gram model.

Figure 6 shows the ciphertext, original message, and best decipherment obtained from using the 3-gram LM.

## 5.2 Random Restarts

The results so far employ a uniform starting condition for $P(c|e)$. We were surprised to find that different starting points result in radically different decipherment strings and accuracies. Figure 7 and 8 shows the result of the uniform starting condition along with 29 random restarts using the 2-gram and 3-gram LMs, respectively. Each point in the scatter-plot represents the results of EM from one starting point. The x-axis gives the entropy in bits/character obtained at the end of an EM run, and the y-axis gives the accuracy of the resulting decipherment. We observe a general trend—when we locate a model with a good $P(c)$, that model also tends to generate a more accurate decipherment. This is

| | | |
|---|---|---|
| a | 16$^\star$ | 1.00 |
| ba | 31$^\star$ | 0.77 |
| | 45 | 0.16 |
| | 71 | 0.07 |
| be | 61$^\star$ | 0.67 |
| | 71 | 0.32 |
| bi | 76 | 0.47 |
| | 73 | 0.46 |
| | 13 | 0.06 |
| bo | 51$^\star$ | 0.83 |
| | 45 | 0.17 |
| bu | 45$^\star$ | 0.56 |
| | 53 | 0.44 |
| da | 23$^\star$ | 1.00 |
| de | 75$^\star$ | 0.54 |
| | 52 | 0.46 |
| di | 12$^\star$ | 0.79 |
| | 46 | 0.21 |
| do | 71$^\star$ | 0.66 |
| | 76 | 0.21 |
| | 36 | 0.12 |
| du | 31 | 0.53 |
| | 53$^\star$ | 0.24 |
| | 34 | 0.22 |
| e | 65$^\star$ | 1.00 |
| ga | 72$^\star$ | 1.00 |
| ge | 35$^\star$ | 0.80 |
| | 23 | 0.20 |
| gi | 13 | 1.00 |
| go | 55$^\star$ | 1.00 |
| gu | 74$^\star$ | 1.00 |
| ha | 31$^\star$ | 0.93 |
| | 76 | 0.07 |
| he | 61$^\star$ | 1.00 |
| hi | 27$^\star$ | 1.00 |
| ho | 51$^\star$ | 0.92 |
| | 31 | 0.08 |
| hu | 45$^\star$ | 0.91 |
| | 27 | 0.09 |
| i | 11$^\star$ | 1.00 |
| ka | 72$^\star$ | 1.00 |
| ke | 36 | 0.09 |
| ki | 36$^\star$ | 0.97 |
| | 34 | 0.03 |
| ko | 55$^\star$ | 0.97 |
| | 51 | 0.03 |
| ku | 74$^\star$ | 0.71 |
| | 36 | 0.15 |
| | 63 | 0.09 |
| | 67 | 0.05 |
| ma | 25$^\star$ | 0.87 |
| | 74 | 0.06 |
| | 53 | 0.04 |
| | 12 | 0.03 |
| me | 56$^\star$ | 1.00 |
| mi | 66$^\star$ | 0.82 |
| | 75 | 0.10 |
| | 22 | 0.08 |
| mo | 37$^\star$ | 1.00 |
| mu | 34 | 0.99 |
| | 31 | 0.01 |
| na | 73$^\star$ | 1.00 |
| n | 67$^\star$ | 0.94 |
| | 54 | 0.06 |
| ne | 56 | 1.00 |
| ni | 41$^\star$ | 1.00 |
| no | 54$^\star$ | 1.00 |
| nu | 32$^\star$ | 0.54 |
| | 62 | 0.45 |
| o | 64$^\star$ | 1.00 |
| ra | 14$^\star$ | 0.79 |
| | 61 | 0.12 |
| | 72 | 0.10 |
| re | 33$^\star$ | 0.91 |
| | 14 | 0.09 |
| ri | 22$^\star$ | 0.96 |
| | 56 | 0.04 |
| ro | 21$^\star$ | 1.00 |
| ru | 42$^\star$ | 0.90 |
| | 36 | 0.09 |
| sa | 26$^\star$ | 0.78 |
| | 72 | 0.22 |
| se | 27 | 0.39 |
| | 47$^\star$ | 0.31 |
| | 26 | 0.30 |
| si | 76$^\star$ | 1.00 |
| so | 43$^\star$ | 1.00 |
| su | 57$^\star$ | 0.34 |
| | 66 | 0.18 |
| | 53 | 0.17 |
| | 34 | 0.17 |
| | 71 | 0.14 |
| ta | 23$^\star$ | 0.90 |
| | 43 | 0.07 |
| | 41 | 0.03 |
| te | 75$^\star$ | 0.98 |
| | 65 | 0.01 |
| ti | 12$^\star$ | 1.00 |
| to | 71$^\star$ | 1.00 |
| tu | 53$^\star$ | 0.72 |
| | 76 | 0.15 |
| | 12 | 0.13 |
| u | 34$^\star$ | 0.93 |
| | 12 | 0.07 |
| wa | 31 | 0.76 |
| | 54 | 0.24 |
| wo | 52$^\star$ | 1.00 |
| yo | 13$^\star$ | 0.83 |
| | 66 | 0.09 |
| | 37 | 0.08 |
| yu | 46$^\star$ | 1.00 |
| za | 73 | 1.00 |
| ze | 61 | 1.00 |
| zi | $\star$ | 1.00 |
| zo | 51 | 0.86 |
| | 12 | 0.09 |
| | 37 | 0.05 |
| zu | 57$^\star$ | 0.70 |
| | 16 | 0.16 |
| | 32 | 0.15 |

**Fig. 9.** Substitution table learnt from best 3-gram decipherment showing P($c|e$) values. Correct entries are marked by $^\star$.

good, because it means that EM is maximizing something that is of extrinsic value. Figure 8 shows two distinct clusters. In one cluster, EM finds a very bad P($c$), and it returns a nonsense decipherment. In the other cluster, EM finds something much more reasonable. It is interesting that human cryptanalysts experience much the same phenomenon. When one has mentally committed to the correctness of a small set of substitutions (e.g.,"72 must be *ro*!"), it can be very hard to escape during subsequent analysis.

Figure 9 shows the learnt substitution table from the best 3-gram decipherment with correct entries marked with $^\star$.

When we use random restarts for decipherment, we cannot simply pick the string with the best accuracy, as this would entail looking at the answer. Instead, we pick the string that results from the model with the best P($c$). As seen in Figure 8, the best P($c$) does not guarantee the best accuracy. However, we are able to significantly improve on the uniform-start decipherment through random restarts. Because of this, the rest of the experimental runs in this paper employ 30 random restarts.

## 5.3 Objective Function

We intuitively feel that more knowledge of Japanese will lead to better decipherments. This is true for long ciphers like the one we have seen so far. What about shorter ciphers? How do the different n-gram LMs perform when used for deciphering shorter ciphers? We split the original 577-syllable cipher in half

**Table 1.** Decipherment error rates with different LMs on long/short ciphers

| **LM model** | **Long Cipher (577 syllables)** | **Short Cipher (298 syllables)** |
|---|---|---|
| 2-gram | 0.46 | 0.89 |
| 3-gram | 0.12 | 0.96 |
| 5-gram | 0.09 | 0.95 |

to create a shorter 298-syllable cipher, and apply the same decipherment strategy. Table 1 shows best decipherment error rates associated with different LM orders (trained on the same data) for the two ciphers (longer and shorter versions). We observe that we can get better accuracies with higher n-gram LMs for the longer cipher (577 syllables). However on the shorter cipher (298 syllables), when we increase the n-gram order of the LM, we find that decipherment surprisingly gets worse. Inspection reveals that the 5-gram decipherment contains many more actual Japanese words. While the text is nonsense, from a certain distance it appears more Japanese-like than the 2-gram decipherment. When we take measurements, we find that the 5-gram LM strongly prefers the (nonsense) string from its decipherment over the (sensible) correct answer, in terms of P($e$). It is therefore quite opinionated in the wrong direction. We find that the 2-gram LM is less opinionated—it still prefers its decipherment over the correct answer, but not to such a degree.

The consequence of the 5-gram model being so wrongly opinionated is that the substitution model probabilities learn to become more fuzzy than usual, in order to accommodate the LM's desire to produce certain strings. The learnt substitution table is much more non-deterministic than the true substitution table (key).

We remedy this. Recall from Equation 1 that EM's objective function is:

$$P(c) = \sum_e P(e) \cdot P(c|e)$$

Here, the P($e$) factor carries too much weight, so in order to reduce the "vote" that the LM contributes to EM's objective function, we create a new objective function:

$$P(c) = \sum_e P(e)^{0.5} \cdot P(c|e) \qquad (4)$$

Note that this is substantially different from the proposal of [5] to stretch out the substitution probabilities after decipherment has finished. Instead, we actually modify the objective function EM uses during decipherment itself.

Table 2 shows the improvements we get from the new objective function at various n-gram orders for the two ciphers. The results are much more in accord with what we believe should happen, and gets us to where better n-gram models give us better decipherment error on short ciphers. In addition, the new objective

**Table 2.** Decipherment error rates on two different Uesugi ciphers using (a) Original EM objective function (b) New objective function (square-rooting LM)

| LM model | Long Cipher (577 syllables) | | Short Cipher (298 syllables) | |
|---|---|---|---|---|
| | EM(original) | EM(new objective fn.) | EM(originial) | EM(new objective fn.) |
| 2-gram | 0.46 | 0.29 | 0.89 | 0.78 |
| 3-gram | 0.12 | 0.05 | 0.96 | 0.78 |
| 5-gram | 0.09 | 0.02 | 0.95 | 0.61 |

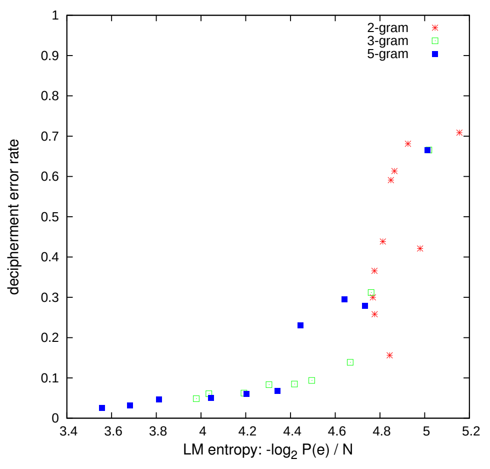


**Fig. 10.** LM entropy vs. decipherment error rate (using various n-gram order LMs on the 577-syllable Uesugi cipher)

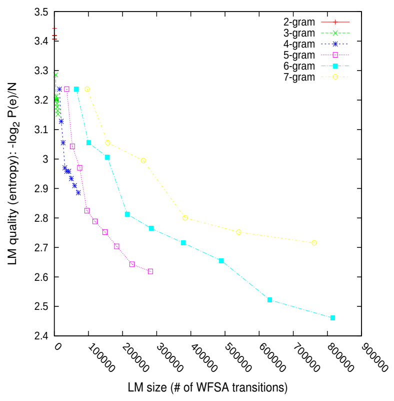
function substantially improves long-cipher decipherment error over the methods of [5].

To sum up our experiments on the Uesugi cipher, Figure 10 shows a nice correlation between LM entropy and end-to-end decipherment accuracy.

## 6 English Results

For direct comparison with [5], we include graphs with English letter-substitution results. Figures 11-13 give the results.

Table 3 shows improvements achieved on a 98-letter English cipher when using the new objective function introduced in Section 5.3. For the graphs in Figures 12 and 13, we also improve LM smoothing and thereby obtain further improvements in accuracy (down to an error-rate of 0.02 for the 7-gram model trained on 7 million letters).



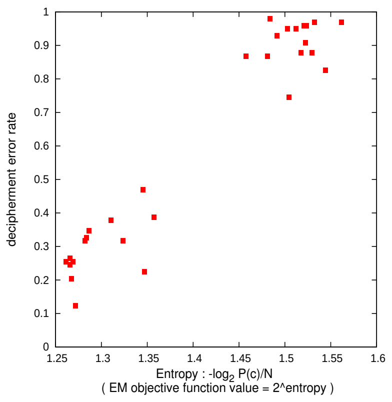


**Fig. 11.** Relationship between LM memory size and LM entropy for English letter-substitution decipherment

**Fig. 12.** Effect of random restarts on EM decipherment for a 98-letter English cipher using 2-gram model

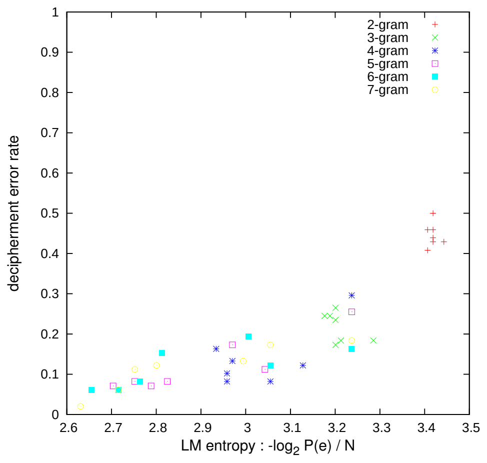


**Fig. 13.** LM entropy vs. decipherment error rate (using various n-gram order LMs on a 98-letter English cipher)

## 7 Conclusion

We have studied a particular Japanese syllable cipher. In doing so, we have made several novel improvements over previous probabilistic methods, and we report improved results. Further improvements in LMs may lead to accurate decipherment of shorter texts, and further algorithms may lead to accurate decipherment of more complex Japanese cipher systems, including translation to

**Table 3.** Decipherment error rates on a 98-letter English cipher using (a) Original EM objective function (Knight et al. [5]) (b) New objective function (square-rooting LM). All the LMs are trained on 1.4 million letters of English data, and use 10 random restarts per point

| LM model | EM(original) error-rate | EM(new objective fn.) error-rate |
|---|---|---|
| 2-gram | 0.41 | 0.43 |
| 3-gram | 0.59 | 0.16 |
| 5-gram | 0.53 | 0.11 |
| 7-gram | 0.65 | 0.11 |

other languages.

## 8 Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency under SRI International's prime Contract Number NBCHD040058.

## References

1. Merialdo, B. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.
2. Peleg, S. and Rosenfeld, A. 1979. Breaking substitution ciphers using a relaxation algorithm. *Comm. ACM*, 22(11):598–605.
3. Olson, E. 2007. Robust dictionary attack of short simple substitution ciphers. *Cryptologia*, 31(4):332–342.
4. Knight, K., and Yamada, K. 1999. A computational approach to deciphering unknown scripts. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing.*
5. Knight, K., Nair, A., Rathod, N. and Yamada, K. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL.*
6. Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series*, 39(4):1–38.
7. Ravi, S. and Knight, K. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the EMNLP.*