Revisiting Unicity Distance: A Channel Transmission Perspective and Its Limits

Fangyuan Lin

Department of Statistics, Columbia University

Abstract

This paper revisits the classical notion of unicity distance for substitution ciphers from an information-theoretic perspective. We frame simple-substitution encryption as a noisy transmission channel, in which the plaintext is the message, the unknown key is the noise, and the attacker is the receiver. Using typical sets to motivate the notion of meaningful plaintexts, we apply the channel coding theorem to blocks encrypted with independent fresh keys. We show that a block length exceeding $H(K)/D$ permits reliable communication at the language's entropy rate using suitably chosen codebooks across many blocks. This provides a channel transmission interpretation of the classical unicity-distance formula. We then test how well this formula describes plaintext recovery when the message is a passage from six known novels. More than 99% of the passages are uniquely identifiable at 18 letters, well before the classical estimate of about 28. N-gram language models, based on the frequencies of short letter sequences, predict the recovery probabilities much more closely. In this experiment, the classical formula misses the recovery transition, while a more detailed model of the language captures it.

1 Introduction

The unicity distance represents the amount of ciphertext required, on average, for an attacker to uniquely determine the encryption key. In cryptography, encryption is a process that transforms a message, called plaintext, into a ciphertext. Encryption is represented as a function that maps the plaintext X and a key K to the ciphertext Y:

$$Y := \mathrm{E}_K(X).$$

We consider symmetric encryption, for which decryption with the same key recovers the original plaintext:

$$X = \mathrm{D}_K(Y).$$

The attacker observes the ciphertext and knows the encryption scheme, but not the key.

This paper studies the simple substitution cipher on the 26 English letters. Its key is a permutation of the alphabet, and encryption replaces every letter of the plaintext by its image under that permutation. Intuitively, a short ciphertext often admits several readings, while a longer one can distinguish its plaintext. One classical attack is frequency analysis: in English the letter `E` is the most common, so the most frequent symbol in a long ciphertext probably stands for `E`. With only a few letters, such counts are unreliable and decipherment may remain ambiguous. For example the ciphertext `XYZ` could represent the meaningful plaintexts `CAT` or `DOG`, and the ciphertext alone would not distinguish the two. On the other hand, additional letters can leave only one sensible plaintext, and we ask how the probability of this event depends on ciphertext length.

We revisit unicity distance [9] by viewing encryption as channel transmission. In this interpretation, an entire plaintext block is the input to a channel, the unknown random key supplies its noise, and the attacker attempts to decode its output. Independently encrypted blocks form uses of a discrete memoryless channel (DMC) [4, p. 193]; letters within one block share a key and are not independent channel uses. Shannon made this connection in his description of secrecy systems as noisy communication systems [9, pp. 685–686]. Throughout, the attacker's goal is to recover the plaintext and as we will see, this does not always require recovering the entire key.

In our formulation, the informal notion of a meaningful message is made precise using the *typical set* which has the Asymptotic Equipartition Property (AEP). Hellman [6] noted that the notion "meaningful" can almost be replaced by "typical" in an ergodic source model and we discuss in Section 3 the introduction of typicality in detail. The channel coding theorem [4] guarantees reliable communication using suitable codes across sufficiently many channel uses; we show that $ND > H(K)$ makes the rate NR achievable by suitable codes over independently encrypted blocks (Section 3.3). In Section 4, we compute the probability of unique decipherment for passages from six English novels and compare it with predictions from N-gram models using public letter-transition tables.

I began this project as an undergraduate, hoping that information and communication theory would give a single answer to the decipherment question. Working through the experiment changed my view. The classical formula does not describe the observed transition from multiple possible plaintexts to unique decipherment. Simple models of English letter sequences do much better. In this paper, we apply channel coding theorem to obtain an alternative derivation of the classical unicity distance formula and we run experiment to show why predicting recovery requires a closer look at the possible messages.

Previous work has studied spurious decryptions, uncertainty about the key, and the performance of decipherment algorithms [6, 2, 5, 7, 8]. Our experiment asks whether the plaintext is uniquely determined, regardless of the decoding algorithm. We treat passages from six known novels as the complete set of possible plaintexts. By checking every candidate, we obtain the exact probability of unique decipherment within this collection at each ciphertext length. This gives a benchmark against which to compare the classical unicity-distance estimate and predictions from N-gram language models.

2 Background: Entropy and Redundancy of Language

Let X be a discrete random variable taking values in a finite set $\mathcal{X}$, called the *alphabet* of X. The entropy of X, denoted $H(X)$, measures the uncertainty or information content of X [10]:

$$H(X) = -\sum_{x \in \mathcal{X}} \Pr(X = x) \log_2 \Pr(X = x).$$

An *information source* is a sequence of random variables $\{X_1, X_2, \dots\}$ taking values in a common alphabet $\mathcal{X}$. For plaintext, $\mathcal{X}$ is the set of letters, of size $G \geq 2$ ($G = 26$ for English). By the independence and uniform bounds [4, Theorems 2.6.4 and 2.6.6],

$$H(X^N) \leq \sum_{i=1}^{N} H(X_i) \leq N \log_2 G,$$

with equality throughout if and only if the letters are independent and uniformly distributed. We write $R_0 = \log_2 G$ for this largest possible entropy per letter, called the *absolute rate*.

For a stationary information source, the limits

$$R = \lim_{N\to\infty} \frac{1}{N} H(X^N) = \lim_{N\to\infty} H(X_N \mid X_{N-1}, \ldots, X_1)$$

exist and are equal [4, Theorems 4.2.1 and 4.2.2]. We call R the *entropy rate* and $D = R_0 - R$ the *redundancy* per letter. Linguistic conventions, such as "q" being usually followed by "u," make text predictable and create redundancy. We use the frequently quoted value $R = 1.5$ bits per letter for English [10, 6], which gives $D \approx 3.20$ bits per letter; it serves as a model parameter.

2.1 Classical Result: Unicity Distance

A *spurious key* is an incorrect key that also yields a meaningful decryption. For example, if the plaintext is *cat* and it is encrypted into ciphertext c, then the key that deciphers c into *dog* would be a spurious key, since *dog* is also a valid English word but it is not the actual plaintext. Intuitively, if the plaintext is "too short," then several meaningful decryptions may remain, and the question we ask is how short is "too short." To address this issue, we study the concept of unicity distance, which was first introduced by Shannon [9]. The classical approximate *unicity distance* is the crossover length at which the expected number of spurious keys falls to order one. The random-cipher calculation underlying the classical approximation uses the following assumptions [9, 6]:

1. There are $G^N = 2^{NR_0}$ strings of length N over the alphabet $\mathcal{X}$ of size $|\mathcal{X}| = G$.
2. There are M allowed plaintexts, sampled uniformly. At this fixed length, write $R = (\log_2 M)/N$ and $D = R_0 - R$, where D is called the redundancy of the language.
3. There are $|\mathcal{K}|$ keys, with key random variable K a priori distributed uniformly and independent of the plaintext variable. Each key independently specifies a uniformly random bijection of the entire space of length-N strings.

By [6], under these assumptions, the expected number $\widehat{n}_k$ of spurious keys is

$$\widehat{n}_k = (2^{H(K)} - 1)\frac{M}{G^N} = (2^{H(K)} - 1)2^{-ND} \approx 2^{H(K)-ND}. \tag{1}$$

The crossover where this expectation is approximately one gives the classical approximate formula

$$U \approx \frac{H(K)}{D}. \tag{2}$$

It does not assert that the expected number of spurious keys becomes exactly zero. Beauchemin and Brassard [1] extended Hellman's formula to nonuniform plaintext and key distributions.

For a uniform simple-substitution key, $H(K) = \log_2(26!) \approx 88.38$ bits. Inserting $R = 1.5$ into (2) gives approximately $27.62 \approx 28$ letters. We now recover this expression through a communication-theoretic approach.

3 Encryption as a Communication Channel

We use the language of channel coding. A *codebook* is a list $\mathcal{C} = (c_1, \ldots, c_M)$ of possible plaintexts of length N, known to both the sender and the attacker; its entries are called *codewords*. The sender chooses a message index J uniformly from $\{1, \ldots, M\}$ and encrypts the codeword c_J. The attacker observes the ciphertext and tries to identify the transmitted codeword. The *rate* of the codebook is $(\log_2 M)/N$ bits per letter. Thus meaningful plaintexts are codewords, encryption is transmission, and decipherment is decoding.

3.1 Typical Sets and Meaningful Plaintexts

We informally call a plaintext meaningful when it is plausible under a statistical model of language. Let $\{X_1, X_2, \dots\}$ be a stationary ergodic information source with alphabet $\mathcal{X}$ and entropy rate R.

For a stationary information source $\{X_1, X_2, \dots\}$ on a finite alphabet $\mathcal{X}$, the limits

$$R = \lim_{N\to\infty} \frac{1}{N} H(X_1, \dots, X_N) = \lim_{N\to\infty} H(X_N \mid X_{N-1}, \dots, X_1)$$

exist and are equal [4, Theorems 4.2.1–4.2.2]. We call R the *entropy rate* of the source and $D = \log_2 G - R$ its redundancy per letter, consistent with the notation of Section 2. Since $\frac{1}{N} H(X_1, \dots, X_N)$ is nonincreasing in N and converges to R, the per-letter entropy of a finite block may exceed R; the count 2^{NR} below is therefore an asymptotic description.

Definition 3.1 (Typical set)**.** For $\varepsilon > 0$, the ε-typical set of length N is

$$\mathcal{T}_N(\varepsilon) = \left\{ x^N \in \mathcal{X}^N : \left| -\frac{1}{N} \log_2 \Pr\{X^N = x^N\} - R \right| \le \varepsilon \right\}.$$

By the asymptotic equipartition property for stationary ergodic sources (the Shannon–McMillan–Breiman theorem) [4], $\Pr\{X^N \in \mathcal{T}_N(\varepsilon)\} \to 1$ as $N \to \infty$, and for large N

$$(1-\varepsilon) 2^{N(R-\varepsilon)} \le |\mathcal{T}_N(\varepsilon)| \le 2^{N(R+\varepsilon)}.$$

Thus about 2^{NR} texts carry nearly all the probability, which is a small fraction of the 2^{NR_0} strings when $R < R_0$. Each typical text has probability between $2^{-N(R+\varepsilon)}$ and $2^{-N(R-\varepsilon)}$, that is, about 2^{-NR} to first order in the exponent. Following [6], we use typicality to motivate a *codebook* of about 2^{NR} meaningful plaintexts, used with equal probability.

Although stationarity and ergodicity are idealizations of natural language, Shannon's prediction experiments provided an influential estimate of the entropy of English: between about 0.6 and 1.3 bits per letter when the preceding 100 letters are known [10]. The value $R = 1.5$ used in this paper is the conventional, slightly conservative figure corresponding to the redundancy $D \approx 3.2$ bits per letter quoted by [6].

3.2 The Substitution Channel

Having described the messages, we now turn to the channel that carries them. In communication theory, a *channel* is a random mechanism that turns an input into an output. It is specified by an input alphabet $\mathcal{A}$, an output alphabet $\mathcal{B}$, and a *transition law* $W(y \mid x)$, the probability that the output is $y \in \mathcal{B}$ when the input is $x \in \mathcal{A}$; for each x, $W(\cdot \mid x)$ is a probability distribution on $\mathcal{B}$. A noisy telephone line is the standard example: the speaker's words are the input, what the listener hears is the output, and noise on the line makes the output random. The receiver sees only the output and must infer the input.

The key observation of this paper is that encryption fits this description once we take the attacker's point of view. Shannon made the same observation: "From the point of view of the cryptanalyst, a secrecy system is almost identical with a noisy communication system," he wrote on page 685 of [9]. The rest of this section makes this precise for substitution cipher. Let $\mathcal{X}$ have $G \ge 2$ letters. Choose π_K uniformly among the $G!$ permutations of $\mathcal{X}$, independently of X^N, and encrypt letter by letter: $Y_i = \pi_K(X_i)$ for $i = 1, \dots, N$. For the intended recipient, who knows K, this map is deterministic and can be inverted. The attacker does not know K, so to the attacker

the key acts as random noise. The input is the whole plaintext $X^N \in \mathcal{X}^N$, the output is the whole ciphertext $Y^N \in \mathcal{X}^N$, and averaging over the unknown key gives the transition law of the channel

$$W_N(y^N \mid x^N) = \frac{1}{G!} \sum_k \mathbf{1}\{y^N = \pi_k(x^N)\}, \tag{3}$$

the probability that the plaintext x^N is encrypted to y^N. We call W_N the *substitution channel*. Because one key encrypts every letter, the letters of a ciphertext are not independent uses of a letter-by-letter channel. Instead, the input and output symbols of W_N are whole length-N blocks, and one use of W_N encrypts an entire plaintext.

What does this channel preserve? A substitution changes the names of letters but never which positions hold the same letter. Let $\mathrm{pat}(x^N)$ record which positions contain the same letter, labeling distinct letters in order of first appearance. For example, `EYE` and `DID` have pattern $(1, 2, 1)$, whereas `TOO` has pattern $(1, 2, 2)$. A substitution channel maps one plaintext to a ciphertext exactly when their patterns agree; we then say they are *compatible*. If x^N contains $d(x^N)$ distinct letters, the keys mapping x^N to a compatible y^N are those that agree on these letters, and there are $(G - d(x^N))!$ of them. Therefore

$$W_N(y^N \mid x^N) = \begin{cases} (G - d(x^N))!/G!, & \mathrm{pat}(y^N) = \mathrm{pat}(x^N), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

All plaintexts compatible with a ciphertext have the same likelihood. A decoder can therefore identify the compatible codewords but cannot distinguish their message indices by likelihood. We call plaintext decipherment *unique* when all compatible entries represent the same plaintext value, and *ambiguous* otherwise. For a list of distinct plaintexts this means that exactly one entry is compatible with the ciphertext.

To decide when the list shrinks to one entry, we need a measure of how much evidence a ciphertext provides. For an input law Q on $\mathcal{X}^N$, the output law and the *information density* are

$$Q_Y(y) = \sum_x Q(x) W_N(y \mid x), \qquad \imath_Q(x; y) = \log_2 \frac{W_N(y \mid x)}{Q_Y(y)}.$$

The information density measures how much more likely the observed ciphertext is under the transmitted plaintext than under the output law. An information density of b bits means that the ciphertext is 2^b times more probable given the true plaintext than it is in general. The average of the information density is the mutual information $I_Q(X^N; Y^N)$. The mutual information

$$I(X; Y) = \sum_{x,y} \Pr\{X = x, Y = y\} \log_2 \frac{\Pr\{X = x, Y = y\}}{\Pr\{X = x\} \Pr\{Y = y\}} = H(X) - H(X \mid Y)$$

is the reduction in uncertainty about X by observing Y [4]. It is symmetric, $I(X; Y) = I(Y; X) = H(Y) - H(Y \mid X)$, and it is zero exactly when X and Y are independent. When the distribution of X is a chosen input law Q, we write H_Q and I_Q to make the dependence explicit.

The following proposition makes precise the observation that the substitution channel transmits patterns and nothing else.

Proposition 3.2 (Information conveyed by substitution)**.** *For any input law Q independent of the key, let $S = \mathrm{pat}(X^N)$ have distribution Q_S. Then $\imath_Q(X^N; Y^N) = -\log_2 Q_S(S)$ almost surely, and*

$$I_Q(X^N; Y^N) = H_Q(S). \tag{5}$$

Proof. Suppose the plaintext x was sent and the ciphertext y was received, and let s be their common pattern.

If z has a different pattern than x, no key turns z into y, so $W_N(y \mid z) = 0$. If z has the same pattern s, then z uses the same number of distinct letters as x, say d_s, and by (4),

$$W_N(y \mid z) = w_s, \qquad \text{where } w_s = \frac{(G - d_s)!}{G!}.$$

This makes the probability of receiving y easy to compute. Only plaintexts with pattern s contribute, each with the same weight w_s, so

$$Q_Y(y) = \sum_z Q(z) W_N(y \mid z) = w_s \sum_{z:\, \mathrm{pat}(z) = s} Q(z) = w_s \, Q_S(s).$$

This is positive, because x itself has pattern s and $Q(x) > 0$.

Therefore,

$$\imath_Q(x; y) = \log_2 \frac{W_N(y \mid x)}{Q_Y(y)} = \log_2 \frac{w_s}{w_s \, Q_S(s)} = -\log_2 Q_S(s).$$

This holds whenever x can be sent and encrypted to y, so $\imath_Q(X^N; Y^N) = -\log_2 Q_S(S)$ almost surely. Mutual information is the average information density and since $\Pr\{X^N = x, Y^N = y\} = Q(x) W_N(y \mid x)$, we have

$$I_Q(X^N; Y^N) = \sum_{x,y} Q(x) W_N(y \mid x) \log_2 \frac{W_N(y \mid x)}{Q_Y(y)} = \mathbb{E}[-\log_2 Q_S(S)] = H_Q(S).$$

□

The other side of the picture is the noise, and here a subtlety appears. The key has $H(K) \approx 88.4$ bits of uncertainty for English, but not all of it affects a given ciphertext. The noise of the channel is the conditional entropy of the output given the input. Given $X^N = x$, the ciphertext is uniform over the $G!/(G - d(x))!$ strings compatible with x. We call

$$\nu(x) = H(Y^N \mid X^N = x) = \log_2 \frac{G!}{(G - d(x))!} \tag{6}$$

the *effective key noise* of the plaintext x. For example, `ATTACKATDAWN` uses only the seven letters `A, T, C, K, D, W, N`. Its ciphertext is determined by where the key sends those seven letters, so its effective key noise is $\log_2(26 \cdot 25 \cdots 20) \approx 31.6$ bits; the remaining 19! ways of arranging the other letters are not relevant.

Proposition 3.3 (Key noise that reaches the ciphertext)**.** *For every plaintext x,*

$$\nu(x) = H(K) - H(K \mid X^N = x, Y^N),$$

so $\nu(x) \le H(K)$, with equality if and only if x contains at least $G - 1$ distinct letters, in which case the plaintext and ciphertext determine the key.

Proof. Fix a plaintext x that uses $d = d(x)$ distinct letters. The ciphertext of x depends only on where the key sends these d letters. The d distinct letters of x can be sent to d different ciphertext letters in

$$G(G-1) \cdots (G - d + 1) = \frac{G!}{(G-d)!}$$

ways, and each choice gives a different ciphertext. Once that choice is fixed, the remaining $G - d$ letters can be arranged in $(G - d)!$ ways, and all of these keys give the same ciphertext. Since the key is uniform over all $G!$ keys and independent of the plaintext, each of the $G!/(G - d)!$ possible ciphertexts occurs with probability $(G - d)!/G!$. The ciphertext is therefore uniformly distributed over these possibilities, and

$$H(Y^N \mid X^N = x) = \log_2 \frac{G!}{(G-d)!} = \nu(x).$$

Now suppose we also observe a compatible ciphertext y. By Bayes' rule and the independence of key and plaintext, a key k has posterior probability

$$\begin{aligned}
\Pr\{K = k \mid X^N = x, Y^N = y\} &= \frac{\Pr\{K = k \mid X^N = x\}\, \Pr\{Y^N = y \mid X^N = x, K = k\}}{\Pr\{Y^N = y \mid X^N = x\}} \\
&= \frac{\Pr\{K = k\}\, \mathbf{1}\{\pi_k(x) = y\}}{\Pr\{Y^N = y \mid X^N = x\}} \\
&= \frac{1/G!}{(G-d)!/G!}\, \mathbf{1}\{\pi_k(x) = y\} \\
&= \frac{\mathbf{1}\{\pi_k(x) = y\}}{(G-d)!}.
\end{aligned}$$

So the key is uniform over the $(G - d)!$ keys that send x to y, and $H(K \mid X^N = x, Y^N = y) = \log_2((G - d)!)$. This value is the same for every compatible ciphertext y, so averaging over y gives $H(K \mid X^N = x, Y^N) = \log_2((G - d)!)$.

Subtracting from $H(K) = \log_2(G!)$,

$$H(K) - H(K \mid X^N = x, Y^N) = \log_2(G!) - \log_2((G-d)!) = \nu(x).$$

□

Proposition 3.3 splits the key's uncertainty into two parts:

$$H(K) = \underbrace{\nu(x)}_{\text{reaches the ciphertext}} + \underbrace{\log_2((G-d)!)}_{\text{stays hidden}}.$$

The first part concerns the letters that occur in the plaintext. Where the key sends them is visible in the ciphertext, so this part acts as noise on the channel. The second part concerns the letters that never occur. Nothing in the ciphertext depends on them, so they remain unknown even to someone who sees both the plaintext and the ciphertext.

The two parts trade off as the plaintext uses more of the alphabet. Writing out the factorials,

$$\nu(x) = \log_2\big(G(G-1)\cdots(G-d+1)\big) = \sum_{j=0}^{d-1} \log_2(G - j),$$

so each new letter moves some uncertainty from the hidden part to the noise. The first letter of the plaintext can be sent to any of G ciphertext letters, so it adds $\log_2 G \approx 4.70$ bits of noise. The second distinct letter can go to any of the remaining $G - 1$, adding $\log_2 25 \approx 4.64$ bits, and in general the $(j + 1)$st distinct letter adds $\log_2(G - j)$ bits. The increments shrink because each new letter has fewer places left to go. The last one adds nothing, since once $G - 1$ letters are placed, the final letter is determined.

3.3 Capacity and the Channel Coding Theorem

We now know what the substitution channel transmits (the pattern of repeated letters) and how much noise the key adds. The natural next question in communication theory is: how many different messages can the channel carry without the receiver confusing them?

The channel coding theorem answers this question when the noise is redrawn independently for every transmission. Suppose, then, that every new message of length N is encrypted with an independently chosen key, as when two parties agree to change the key for every message. The channel transitions across successive blocks are then conditionally independent given the inputs, so the encryption system is a discrete memoryless channel whose input is a whole plaintext of N letters and whose output is a whole ciphertext. Its *channel capacity*, in bits per ciphertext, is

$$C_N = \max_Q I_Q(X^N; Y^N),$$

where the maximum ranges over all input laws Q on $\mathcal{X}^N$.

To bound C_N from below, we need one specific input law, and the simplest is the uniform law on all sequences of N letters. Under this law the information density takes a very simple form.

Lemma 3.4 (Information density under the uniform law)**.** *Let Q be the uniform law on all G^N sequences of N letters. Then every compatible pair of plaintext and ciphertext (x, y) satisfies*

$$\imath_Q(x; y) = N \log_2 G - \nu(x). \tag{7}$$

Proof. By Proposition 3.2, $\imath_Q(x; y) = -\log_2 Q_S(s)$, where s is the pattern of x. A pattern s with d distinct letters is shared by exactly $G!/(G-d)!$ plaintexts. Under the uniform law, $Q_S(s) = \frac{G!/(G-d)!}{G^N} = 2^{\nu(x) - N \log_2 G}$, which gives (7). □

This gives a lower bound on channel capacity.

Proposition 3.5 (Capacity lower bound)**.**

$$C_N \geq N \log_2 G - H(K). \tag{8}$$

Proof. Take Q uniform on $\mathcal{X}^N$. By Lemma 3.4 and Proposition 3.3, every compatible pair of plaintext and ciphertext has $\imath_Q(x; y) = N \log_2 G - \nu(x) \geq N \log_2 G - H(K)$. Averaging gives $I_Q(X^N; Y^N) \geq N \log_2 G - H(K)$, and C_N is at least this value. □

To state the coding theorem, we need the notion of a code. A *code* over m channel uses is a list of M_m messages, agreed on in advance and known to everyone, including the attacker, in which each message has mN letters divided into m plaintexts of N letters. To send a message, each of its plaintexts is encrypted with an independent fresh key. The attacker knows the list and observes all m ciphertexts, but none of the keys, and must decide which plaintexts were sent. The *rate* of the code, $(\log_2 M_m)/m$, is the number of bits of plaintext carried per ciphertext.

The channel coding theorem [4] states that for every rate $0 \leq r < C_N$ there are codes whose rates tend to r and whose average decoding error tends to zero as $m \to \infty$, while rates above C_N cannot have vanishing error.

The typical-set approximation suggests a target rate of NR bits per ciphertext. This rate can be achieved by suitably chosen codes over many blocks when $NR < C_N$; the theorem does not guarantee recovery for a prescribed list of English texts. By Proposition 3.5, this holds as soon as $NR < N \log_2 G - H(K)$, that is, since $D = \log_2 G - R$,

$$ND > H(K) \quad \Longrightarrow \quad NR < C_N.$$

This recovers the classical scale $H(K)/D$ [6] as a sufficient capacity comparison.

In other words, if $ND > H(K)$, one can choose a list of approximately 2^{mNR} messages, each consisting of m length-N plaintext blocks, whose average decoding error tends to zero as m grows. Each of the m blocks is encrypted with its own independent key. The attacker knows the chosen list and jointly decodes the m ciphertexts without knowing the keys.

The channel coding theorem recovers the classical expression as a sufficient condition for reliable communication using suitably chosen codebooks. We now examine how well this expression describes unique decipherment in a fixed collection of English passages, and compare it with predictions from N-gram language models.

4 An Experiment with Six English Novels

4.1 Choosing a Passage from Six Novels

Imagine that the sender and the attacker own the same six novels. The sender chooses a passage and encrypts it; the attacker knows that the answer is somewhere in those books. These books contain every plaintext allowed in the experiment. We can therefore check every competing passage and compute the probability of a unique reading exactly.

The novels are *Alice's Adventures in Wonderland*, *Pride and Prejudice*, *Frankenstein*, *Moby Dick*, *The Adventures of Sherlock Holmes*, and *Adventures of Huckleberry Finn*. We use GITenberg copies of Project Gutenberg editions 11, 1342, 84, 2701, 1661, and 76. Removing the Gutenberg headers and footers, keeping only ASCII letters, and converting to uppercase leaves 2,791,233 letters.

For each $N = 10, \ldots, 24$, collect all passages of N consecutive letters lying within a single book and remove duplicate strings. The sender chooses uniformly from this set $\mathcal{C}_N$. Write $M_N = |\mathcal{C}_N|$. Grouping the passages by repetition pattern, with nonempty class sizes $t_1, \ldots, t_L$, gives the exact probability of unique decipherment:

$$u_N(\mathcal{C}_N) = \frac{\#\{j : t_j = 1\}}{M_N}. \tag{9}$$

Each singleton class contains one uniquely identifiable passage. This differs from an attack's success rate: a decoder may also guess correctly when several candidates remain. The source archive includes the text snapshots, their checksums, and the preprocessing code.

4.2 Models of English Letter Sequences

We compare this exact curve with predictions from Burkardt's public NGRAMS tables [3]. Monograms count single letters, bigrams pairs, trigrams triples, and quadgrams sequences of four letters. If c denotes the published counts, define

$$p_a = \frac{c(a)}{\sum_b c(b)}, \qquad T_{ab} = \frac{c(ab)}{\sum_c c(ac)},$$

$$A_{ab,c} = \frac{c(abc)}{\sum_d c(abd)}, \qquad B_{abc,d} = \frac{c(abcd)}{\sum_e c(abce)}.$$

We obtain a string's probability by multiplying the successive letter probabilities. The four models are

$$
\begin{aligned}
Q_0(x) &= \prod_{t=1}^{N} p_{x_t}, \\
Q_1(x) &= p_{x_1} \prod_{t=2}^{N} T_{x_{t-1},x_t}, \\
Q_2(x) &= p_{x_1} T_{x_1 x_2} \prod_{t=3}^{N} A_{x_{t-2}x_{t-1},x_t}, \\
Q_3(x) &= p_{x_1} T_{x_1 x_2} A_{x_1 x_2, x_3} \prod_{t=4}^{N} B_{x_{t-3}x_{t-2}x_{t-1},x_t}.
\end{aligned} \tag{10}
$$

The first model, Q_0, ignores the order of the letters. The others allow the next letter to depend on the preceding one, two, or three letters. In particular, predicting a letter from the three letters before it requires counts of four-letter sequences, which is why Q_3 uses quadgrams. All strings in the experiment are long enough for the initial factors shown above.

Unlisted N-grams have count zero. If the tables contain no continuation of a preceding sequence, we discard its first letter and try the shorter sequence, repeating if needed. This occurs for 23 three-letter sequences. We do not otherwise smooth the counts or fit them to the novels. The provider does not fully document the underlying corpus, so overlap with the novels is possible.

4.3 Computing the Predictions

For each length, we draw a list of $M = M_N$ candidates independently from a language model Q, give the list to the attacker, and choose the transmitted entry uniformly. Repeated entries retain their weight, but copies of the same plaintext count as one possible reading. Write $S(x) = \mathrm{pat}(x)$ and

$$
Q_S(s) = \sum_{x : S(x) = s} Q(x).
$$

The probability of unique plaintext recovery, averaged over the generated lists, is

$$
u_N(Q, M) = \sum_{x} Q(x) \big[1 - Q_S(S(x)) + Q(x)\big]^{M-1}. \tag{11}
$$

Given the transmitted plaintext x, a different compatible rival has probability $Q_S(S(x)) - Q(x)$. Unique recovery requires all $M - 1$ independent rivals to avoid this event; averaging over x gives (11).

The model uses the same number of candidates as the novels, but generates their text independently. Real passages can overlap and share phrases. Comparing the curves shows how much of the observed ambiguity can be explained by short letter sequences. We also include a baseline in which all letters are independent and equally likely.

For the tested lengths, evaluating the sum in (11) directly is impractical. Instead, for each model we generate five independent lists of M_N candidates and compute the fraction of entries whose pattern admits no different plaintext. Duplicate entries retain their weight: if all entries in a pattern class have the same value, every one counts as a success. Averaging these fractions gives an unbiased estimate of (11). Within each run, shorter strings are taken from the beginnings of the same generated strings. We calculate standard errors from the five independent runs, since

the recovery outcomes within a list are dependent. The bands in the figures show two estimated standard errors.

The uniform-letter model is simpler and can be evaluated without simulation. If $a_{n,d}$ is the probability that n uniform letters contain d distinct letters, then

$$a_{0,0} = 1, \qquad a_{n+1,d} = \frac{d}{G} a_{n,d} + \frac{G-d+1}{G} a_{n,d-1},$$

with out-of-range terms zero. There are $G!/(G-d)! - 1$ different compatible rivals for such a string, giving

$$u_N(Q_{\text{unif}}, M_N) = \sum_{d=1}^{\min(N,G)} a_{N,d} \left(1 - \frac{G!/(G-d)! - 1}{G^N}\right)^{M_N - 1}.$$

Here Q_{unif} is the uniform distribution on length-N strings. We use logarithms to evaluate the powers accurately. The English-model curves are simulation estimates; the corpus curve is an exact count. Seeds, raw results, and code are included in the source archive.

4.4 Results

Figure 1 shows how quickly the passages become distinguishable as their length increases. At 18 letters, 99.2075% of the 2,772,123 distinct corpus passages are uniquely identifiable. For all four English models, 18 is also the first tested length at which recovery exceeds 99%. Uniform letters reach this level at 21 letters.

The models improve as they take longer letter sequences into account. Over all tested lengths, the largest absolute discrepancies from the corpus are 12.91, 6.04, 2.52, and 1.56 percentage points for monograms, bigrams, trigrams, and quadgrams, respectively. The largest simulation standard error is only 0.0165 percentage points, so these differences reflect the models rather than simulation noise.

The first figure makes the curves look nearly identical once recovery is close to complete. The second figure shows where they still differ. Figure 2 plots the probability that at least one different passage is still compatible. At 18 letters, this probability is 0.7925% in the corpus and 0.3972% under the quadgram model. The best overall fit therefore still underestimates the remaining ambiguity by about a factor of two. Overlapping passages and recurring phrases are features of the novels that independent candidate generation does not preserve.

The classical formula does not describe this experiment. It gives about 28 letters, whereas the corpus already exceeds 99% unique recovery at 18. Even forcing the classical expression to equal 18 would require

$$R = \log_2 26 - \frac{\log_2(26!)}{18} \approx -0.210 \quad \text{bits per letter},$$

which cannot be an entropy rate. Changing the assumed rate of English therefore cannot make this expression match the observed 18-letter crossing. The N-gram models provide a fuller answer: they predict how often recovery is unique at each length, and their curves follow the experiment much more closely.

Identifying the passage does not identify the whole key. An 18-letter passage uses at most 18 letters of the alphabet, leaving at least $8! = 40{,}320$ keys that agree on every observed letter. The sufficient coding condition in Section 3.3 remains valid, but it does not locate the point at which the passages in this collection become distinguishable.

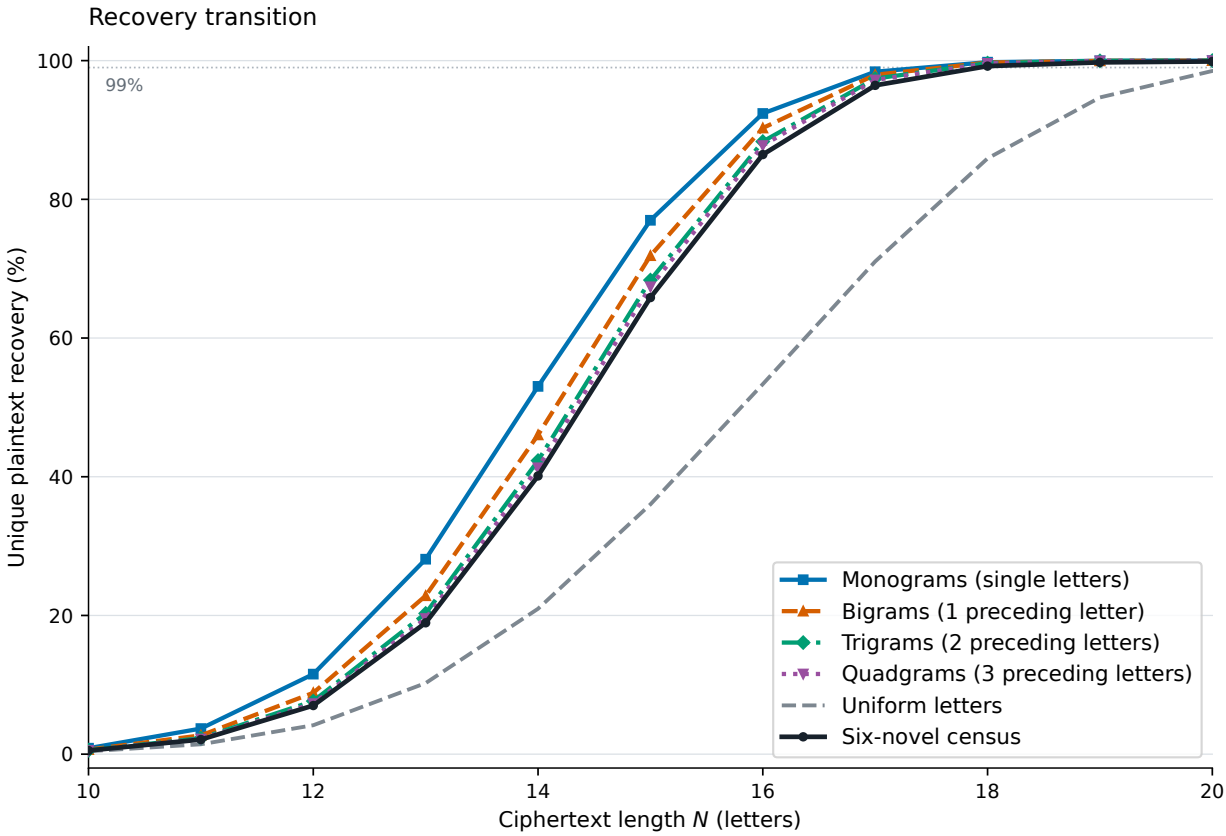

Figure 1: Probability of unique plaintext recovery in the six-novel universe and predictions using the same number of candidates. The corpus curve is exact. English-model curves average five independent simulated lists; the uniform-letter curve is evaluated directly. The classical estimate of about 28 letters lies beyond the displayed range.



Figure 2: Remaining ambiguity: the probability that a different candidate plaintext has the same repetition pattern. This is one minus unique recovery, not an attack's error probability. A logarithmic scale separates the curves near complete recovery. Shading shows two estimated standard errors. Zero simulation estimates are omitted; they do not establish zero ambiguity.

5 Conclusion

We started by treating encryption as transmission through a noisy channel. Applying the channel coding theorem led back to the classical expression $H(K)/D$. We then asked whether that expression describes the recovery of passages from six known novels. It does not: more than 99% of the passages are uniquely identifiable at 18 letters, well before the classical estimate of about 28. Models based on short English letter sequences predict the observed probabilities much more closely, though they still underestimate the ambiguity that remains.

Unique decipherment depends on whether another possible plaintext fits the same ciphertext. In our experiment, the language's entropy rate alone does not predict how often this happens. The channel interpretation explains the classical formula, while N-gram models give a more accurate account of the observed recovery probabilities.

Acknowledgments

The author thanks Professor Steven N. Evans for his guidance and support, Professor Andrew Gelman for helpful discussions, and the reviewers and editor of the PUMP Journal of Undergraduate Research for their patience and thoughtful suggestions.

References

[1] P. Beauchemin, G. Brassard, A Generalization of Hellman's Extension to Shannon's Approach to Cryptography, *J. Cryptology*, **1** (1988), 129–131. https://doi.org/10.1007/BF02252870.

[2] R. J. Blom, Bounds on Key Equivocation for Simple Substitution Ciphers, *IEEE Trans. Inf. Theory*, **25** (1979), 8–18.

[3] J. Burkardt, NGRAMS—String Frequency in English Text, public dataset collection, last revised February 12, 2016. https://people.math.sc.edu/Burkardt/datasets/ngrams/ngrams.html (accessed November 19, 2026).

[4] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, 2nd ed., Wiley-Interscience, 2006. https://doi.org/10.1002/047174882X.

[5] S. Hasinoff, Solving Substitution Ciphers, technical report, University of Toronto, 2003. https://people.csail.mit.edu/hasinoff/pubs/hasinoff-quipster-2003.pdf.

[6] M. E. Hellman, An Extension of the Shannon Theory Approach to Cryptography, *IEEE Trans. Inf. Theory*, **23** (1977), 289–294. https://doi.org/10.1109/TIT.1977.1055709.

[7] P. Norvig, Natural Language Corpus Data, in T. Segaran and J. Hammerbacher (eds.), *Beautiful Data*, O'Reilly Media, 2009, 219–242. https://www.norvig.com/ngrams/ch14.pdf.

[8] S. Ravi, K. Knight, Attacking Decipherment Problems Optimally with Low-Order N-gram Models, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2008, 812–819. https://aclanthology.org/D08-1085/.

[9] C. E. Shannon, Communication Theory of Secrecy Systems, *Bell Syst. Tech. J.*, **28** (1949), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x.

[10] C. E. Shannon, Prediction and Entropy of Printed English, *Bell Syst. Tech. J.*, **30** (1951), 50–64. https://doi.org/10.1002/j.1538-7305.1951.tb01366.x.