Entropy Model in Application to Natural Languages

Yue Zeng

Thesis submitted to the Faculty of Science in partial fulfillment of the requirements
for the degree of
Master of Science in Mathematics and Statistics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

In this thesis, we study the mathematics behind the information theory. The focus is on Shannon's notion of the asymptotic entropy (entropy rate) and its applications to natural languages. We discuss various approaches to estimation of the entropy rate and their ramifications.

This thesis is dedicated to my beloved parents and family, for their unconditional love, continued support and the low interest rates on everything I owe.

Acknowledgement

I must express my sincere gratitude to my supervisor, Professor Vadim Kaimanovich, for his persistent support of my study and this research, for his patience and immense knowledge. He encouraged me to keep trying during all the difficult times. His positive attitude kept me engaged and motivated to learn. Throughout my time as his student he cared about my work and answered my questions promptly. This work would not have been possible without his guidance.

Contents

List of Figures vii

List of Tables viii

Introduction 1

1 Background on probability and entropy 5
- 1.1 Probability spaces and partitions . . . 5
- 1.2 Independence . . . 7
- 1.3 Amount of information . . . 7
- 1.4 Average amount of information . . . 9
- 1.5 Properties of entropy . . . 13
  - 1.5.1 Entropy function . . . 13
  - 1.5.2 Concavity and convexity . . . 13
  - 1.5.3 Entropy inequalities . . . 14
  - 1.5.4 Comparison of entropies . . . 16
- 1.6 Conditional entropy of finite distributions . . . 17

2 Shannon's model and asymptotic entropy 21
- 2.1 Prefix code and binary tree . . . 21
- 2.2 Shannon's source coding theorem . . . 27
- 2.3 Shannon's model . . . 30
- 2.4 Asymptotic entropy . . . 31
- 2.5 Shannon-McMillan-Breiman theorem . . . 34

3 Shannon's approach to natural languages 35
- 3.1 Complexity of language . . . 35
- 3.2 Shannon's entropy estimates . . . 37
  - 3.2.1 Frequency tables . . . 37
  - 3.2.2 Word rank . . . 39
- 3.3 Shannon's cognitive experiment . . . 44
  - 3.3.1 Identical twins thought experiment I . . . 44

3.3.2 Identical twins thought experiment II 45
3.3.3 Shannon's practicable experiment 46
3.3.4 Reversed experiment 48

4 Further developments **50**
4.1 Same methodology 50
4.1.1 Objective method 50
4.1.2 Subjective method 51
4.2 Different methodology 53
4.2.1 Data compression and optimal encoding 53
4.2.2 Simulation experiment 53
4.3 Other languages 59
4.3.1 Alphabet sizes across languages 60
4.3.2 Entropy across similar languages 60
4.3.3 Why are English books so much thicker than Chinese ones? 62
4.4 Modern views on applicability of Shannon's model 65
4.4.1 Difference analysis 65
4.4.2 Hilberg's Ansatz 66

Bibliography **67**

List of Figures

1.1 Graphical representation of a random variable 11
1.2 The associated information function 12
1.3 Graph of the entropy function $\varphi(t) = -t \log t$ 13

2.1 Binary tree 22
2.2 Prefix tree for Example 2.1.2 23
2.3 Illustration of the proof of Kraft's inequality 26

3.1 Rank frequency graph 41
3.2 Rank frequency graph (logarithmic scale) 41
3.3 Shannon's rank frequency graph 42
3.4 Shannon's experiment results 48

4.1 Compression ratio for GZIP 58
4.2 Compression ratio for ZIP 59
4.3 Chinese and English characters 63

List of Tables

3.1 Frequency distribution in the extended English alphabet 38
3.2 h_n 38
3.3 h'_n 38
3.4 h'_n for the standard English alphabet 43
3.5 h_n for the extended English alphabet 43
3.6 Upper and lower bounds for h_n 47
3.7 Shannon's reversed experiment 49

4.1 Data cleaning results 54
4.2 Shakespeare: file sizes (in bits) and the corresponding compression ratios 55
4.3 Compression ratios: Shakespearean works 57
4.4 Compression ratios: different genres of literature 58
4.5 Body segment length measurements 61
4.6 The Bible: file sizes (in bits) and the corresponding compression ratios 64

Introduction

This thesis is a survey of some of the mathematics behind the information theory. The underlying idea is that one can evaluate the *amount of information* in a communicated message based on its probabilistic properties. The resulting quantitative measure is called *entropy*. The entropy was first introduced as a measurable physical quantity in classical thermodynamics, and later was defined in purely mathematical context. We begin by reviewing two articles by the father of information theory, Claude Shannon, *A mathematical theory of communication* [Sha48] and *Prediction and entropy of printed English* [Sha51]. Then we look at the ensuing development of his ideas. In more technical terms, we outline the basics of the entropy theory, including the definition of the asymptotic entropy of an information source and overview the applications of this notion to the study of natural languages.

We begin Chapter I with the basic definitions. After first setting up the standard probability background, we give several definitions of the entropy in various guises. More specifically, we talk about the entropy of a partition, formula (1.4.1), the entropy of a discrete distribution, equation (1.4.2), and the entropy of a discrete random variable, equation (1.4.3); their relationship is illustrated by various examples. Further, we discuss the concavity and convexity properties of entropy, which lead to the fundamental inequalities involving the entropy; in particular, the maximality of the entropy of the uniform distribution. In a more technical Section 1.5.4, we establish some entropy bounds for distributions arising from renormalization. An example of this situation is provided by the frequency distributions in a given alphabet depending whether the space (or various punctuation marks) is included or not. The final section of this chapter is devoted to a discussion of the notion of conditional entropy. In Proposition 1.6.3, we establish the key inequalities: the conditional entropy is always sandwiched between zero and the unconditional entropy. Moreover, one can explicitly characterize the situations when these bounds are attained. A consequence of this is the subadditivity of entropy: the joint entropy of two random variables does not exceed the sum of their entropies, and the equality holds if and only if they are independent (Corollary 1.6.5).

We begin the following Chapter II with a discussion of an interpretation of entropy in terms of prefix codes. By using Kraft's inequality (Theorem 2.1.1), we establish Shannon's source coding theorem (Theorem 2.2.1), which states that for

any prefix encoding of a finite alphabet, the entropy of any distribution on this alphabet does not exceed the expected length of the code words. We define the code efficiency of a source to be the ratio of the entropy of a distribution and the expected length of the code words with respect to this distribution. According to Shannon's source coding theorem, the code efficiency does not exceed one. If the equality is attained, then such code is optimal. It turns out that there are always codes that have code efficiency close to one. This interpretation is an inspiration for the notion of asymptotic entropy, which is based on Shannon's assumption that all streams of symbols are stationary sequences (in probabilistic sense): Section 2.3.

Finally, in Section 2.4, we define the asymptotic entropy. One definition is based on the Fekete Lemma and subadditivity of entropy which implies that for a stationary source (X_i), the entropy of the first n symbols divided by n converges to a limit, which is naturally called the *asymptotic entropy*. More precisely,

$$\begin{aligned} H_n = H(X_1^n) &= H(X_1, X_2, \dots, X_n) \\ &= -\sum_{a_1,\dots,a_n} P(X_1 = a_1, \dots, X_n = a_n) \log P(X_1 = a_1, \dots, X_n = a_n) \ , \end{aligned}$$

and

$$h = \lim_{n\to\infty} \frac{H_n}{n} \ .$$

Another definition can be given in terms of the conditional entropy, namely, the conditional entropy of the time 0 symbol conditioned by the future $n-1$ symbols,

$$h_n = H(X_0|X_1^{n-1}) = H_n - H_{n-1} \ ,$$

and

$$h = \lim_{n\to\infty} h_n \ .$$

The second definition implies that the sequence H_n behaves more regularly than what is prescribed by the subadditivity. Namely, not only H_n/n converge, but the *entropy increments* $h_n = H_n - H_{n-1}$ also converge. Moreover, this definition is symmetric with respect to the time reversal as

$$h_n = H(X_n|X_1^{n-1}) = H\big(X_0|X_{-n+1}^{-1}\big) \ .$$

In Chapter III, we discuss the ramifications of the definition of asymptotic entropy. According to Shannon's model, the English text can be considered as a stationary sequence, therefore one can talk about its asymptotic entropy. If we assume all letters appear with equal probability, and they are independent, then asymptotic entropy h_0 is the logarithm of the alphabet size, which is the theoretical maximal

value of the asymptotic entropy. Shannon used the frequency tables of letters, *digrams* (pairs of letters) and *trigrams* (triples of letters) in English — the only ones available at this time — to calculate the entropies H_1, H_2, H_3 and entropy increments h_1, h_2 and h_3 for both the original English alphabet and the extended English alphabet (with added space symbol). After, he attempted to extrapolate the missing data by using more physical and less rigorous argumentation based on Zipf's law. It states that the frequency f_r of the r-th most frequent word of English language is approximately

$$f_r = \frac{0.1}{r} \ .$$

The data for this calculation was made up from two parts. According to BRILLOUIN [Bri13], the frequencies for the first 100 most frequent words were collected from Table 3 in DEWEY's book *Relative frequency of English speech sounds*, and the rest (words with rank from 101 to 8727) was obtained using Zipf's law. Namely, the fact that the sum of all frequencies of these words is equal to one imposes the cut-off value 8727 for the total number of words. Taking into account that the average length of an English word $L_{\text{word}} = 4.5$ letters, Shannon was able to estimate

$$h'_w = \frac{H_{\text{word}}}{L_{\text{word}}} = \frac{-\sum_1^{8727} f_r \log_2 f_r}{4.5} \ .$$

This value was recorded as h_5 or h_6 in Shannon's work.

These values h_1, h_2, h_3 and $h_{5 \text{ or } 6}$ being outright and insufficient for making any conclusions about the asymptotic entropy, Shannon resorted to a different *cognitive* "subjective" approach. In his experiment, a subject was asked to predict the next symbol of a text given several previous ones. The result of this prediction was used to obtain upper and lower bounds of the entropy increment h_n by using the inequalities

$$\underline{h_n} = \sum_{i=1}^{k} i(q_i^n - q_{i+1}^n) \log i \leq h_n \leq -\sum_{i=1}^{k} q_i^n \log q_i^n = \overline{h_n} \ ,$$

where q_i^n is the empirical frequency that the subject uses i guesses to arrive at the correct symbol given the previous $n-1$ symbols, and k, the maximal number of guesses that is the alphabet size. Shannon plotted the upper and lower bounds for $h_1, \ldots, h_{15}, h_{100}$ from the experiment and estimated the entropy rate to be $h = 1.3$.

We begin Chapter IV with a discussion of both the objective and subjective methods described in Shannon's original work. This section mainly focuses on the development of these ideas. With powerful computers, researchers are now able to generate and analyze bigger and more reliable data. In the first part of this chapter we discuss the experiments that were conducted using the same methods as Shannon's. We have mentioned that one of the limitations Shannon had was not being able to find n-gram frequencies for $n > 3$. Now, this is no longer an issue, and this is the

approach used in a number of publications. However, there are several major issues with this type of calculation, which we address in Section 4.1.1.

Following this, in Section 4.2.1, we also discuss the estimation of the entropy rate through compression algorithms which is a method that was not mentioned in Shannon's original work since it was unavailable at the time. Using the fact that natural language has redundancy, applying a compression algorithm on text can convert the original text into a more efficient encoded form to reduce length. Therefore, the entropy rate can be estimated as

$$\text{Entropy rate of the text} = \frac{\text{length of the compressed text}}{\text{length of the original text}} .$$

We ran a simulation experiment, the results of which are presented in Section 4.2.2. We selected a collection of Shakespeare's plays and poetry [Sha20], and ran four compression algorithms *GZIP,ZIP,BZIP2* and *LZMA* on the cleaned data and estimated the entropy rate to be $h \approx 2.18$ bits per symbol. In addition, we also examined the possible influence of the difference in literature style on its estimated entropy rate in the situation when all work was produced by the same author. The collection of Shakespearean work was divided into six types, and no major discrepancy was found (Table 4.4).

The following Section 4.3 focuses on the entropy rates across different languages. One could use the entropy rate as a parameter to study the similarities among languages, and it should be expected that the languages from the same family should have relatively close entropy rates. On the contrary, in Section 4.3.3, we look at the English — Chinese language pair, where the languages belong to completely different families. As a sample, we use the book *Steve Jobs* by Walter Isaacson both in its original English version and the translated Chinese version. We discuss the nature of translation, average word length, formatting, and most pertinently, the entropy rate per page, which appears to be more intuitive in this context. In our estimation, we found that the Chinese text contains twice as much information per page as English. This explains why it is common to see much thicker books in English than those in Chinese.

The last part of this thesis (Section 4.4) is devoted to the modern views on the applicability of Shannon's model. We discuss possible improvements in the various aforementioned experiments. The basic models of Hilberg's vanishing entropy conjecture are also included in our discussion.

Chapter 1

Background on probability and entropy

The main sources for this short introduction are *Probability, random processes, and ergodic properties* [Gra09] and *Entropy and information theory* [Gra11] by Robert Gray.

1.1 Probability spaces and partitions

There are plenty of events in life that one can not predict with certainty. This is what we call randomness in everyday life. Probability theory helps to build mathematical models describing these phenomena, so that one could further analyze them. The flip of a coin, the roll of a dice and the content of Elon Musk's next tweet are examples of randomness. The basis of the modern probability theory is the notion of a probability space.

Definition 1.1.1. A sample space Ω is a collection of sample points ω (also called elementary outcomes of an "experiment" described by Ω). Events are subsets of Ω. Usually one specifies a collection $\mathcal{F}$ of "nice" subsets required to be a σ-algebra. It is only the subsets from the collection $\mathcal{F}$ that are called events. The final (and most important ingredient) is a probability measure (= probability distribution or just probability) denoted by P. This is a function from $\mathcal{F}$ to real numbers, so that each event A has a probability that is denoted by $P(A)$. The measure P satisfies the following conditions:

- $\forall\ A$ in $\mathcal{F},\ 0 \leq P(A) \leq 1$
- $P(\emptyset) = 0$ and $P(\Omega) = 1$.

- If there is a sequence of disjoint events $A_1, A_2, A_3, \ldots$ then:

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

(this property is called σ-additivity).

The triple $(\Omega, \mathcal{F}, P)$ is called a probability space.

All mathematically precise models of a random experiment or a collection of experiments must be of this kind.

Definition 1.1.2. A finite (or, more generally, countable) collection of events $\beta = \{B_1, \ldots, B_n\}$ is called a partition of Ω if all the sets $B_i,\ 1 \leq i \leq n$ are disjoint and make up Ω together. In other words, we have $B_i \cap B_j = \emptyset$ for $i \neq j$, and $\bigcup_{i=1}^{n} B_i = \Omega$.

We can see a partition β as a way to divide the probability space into a few smaller subsets B_i called the elements of the partition β. The simplest non-trivial partitions are the ones with just two elements, in which case the partition consists just of a single subset $B \subset \Omega$ and its complement B^c.

For the sake of completeness, let us remind ourselves of the standard facts (going back to Bayes) about the *conditional probabilities* with respect to a partition. If $(B_1, \ldots, B_n)$ is a partition of Ω with $P(B_i) > 0$ for all $1 \leq i \leq n$, then any event A is partitioned into its intersections with the sets B_i, whence one has the full probability formula:

$$P(A) = \sum_{i=1}^{n} P(A \cap B_i) = \sum_{i=1}^{n} P(A|B_i)P(B_i)\ .$$

For a two-element partition $\{B, B^c\}$, it takes the form:

$$P(A) = P(A \cap B) + P(A \cap B^c)\ .$$

The probabilities $P(B_k)$ are often referred to as *prior* probabilities of the elements of the partition β, whereas their conditional probabilities with respect to A are called *posterior* probabilities. They can be expressed by the *generalized Bayes' formula*:

$$P(B_k|A) = \frac{P(A \cap B_k)}{P(A)} = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)} \quad \forall\ 1 \leq k \leq n\ .$$

In the particular case of a two-element partition, this expression takes the form of the simple Bayes' formula:

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B^c)P(B^c)}\ .$$

1.2 Independence

Another important concept related to conditional probability is that of independence. Two events A and B from the same probability space are called independent if the *conditional probability* $P(A|B)$ is equal to the *unconditional probability* $P(A)$. This condition is actually symmetric with respect to A and B:

$$P(A|B) = P(A) \iff \frac{P(A \cap B)}{P(B)} = P(A) \iff P(A \cap B) = P(A)P(B) \ .$$

Similarly, one defines the mutual independence of more than two events: events $A_1, \ldots, A_n$ are called mutually independent if for every collection of indices: $1 \leq i_1 < i_2 < \cdots < i_k \leq n$,

$$P(A_{i_1} A_{i_2} \ldots A_{i_k}) = P(A_{i_k}) \times P(A_{i_2}) \times \cdots \times P(A_{i_k}) \ .$$

Definition 1.2.1. An event A is independent of partition $\beta = (B_i)$ if it is independent of any element B_i of β; two partitions $\alpha = (A_i)$ and $\beta = (B_j)$ are called independent if their elements are pairwise independent, i.e., A_i and B_j are independent for any i and j.

1.3 Amount of information

We want to assign to any event A in a probability space $(\Omega, \mathcal{F}, P)$ a number $I(A)$, which is a quantitative measure of the amount of information obtained when this event is realized. It must depend on the probability of A: when the probability is high, the uncertainty is low, therefore the amount of information is also low.

First of all, let's consider the case of flipping an unbiased coin, which means the probability of obtaining a head is the same as the probability of obtaining a tail after one flip. Here we use the notation in the usual way by writing $P(H)$ instead of $P(\{H\})$, the corresponding probability space being:

$$\Omega = \{H, T\} \quad \text{with} \quad P(H) = P(T) = \tfrac{1}{2}$$

Let us assume that:

- The amount of information that is obtained from one coin flip is 1 bit. We can assume that both outcomes H and T bring the same amount of information, because their probabilities are equal.
- Since the flips are independent, it is natural to assume that the amount of information that is realized from two independent coin flips is 2 bits. On the other hand, the probability of each of these 4 outcomes is:
  $P(HH) = P(HT) = P(TH) = P(TT) = \frac{1}{4}$.

- In the same way, the amount of information that is realized from three coin flips is 3 bits, whereas the probability of each of the possible 8 outcomes:
  $P(HHH) = P(HTH) = P(HTT) = P(HHT) = P(TTT) = P(THT) = P(THH) = P(THH) = \frac{1}{8}$.

Therefore, in all these cases we have the relation

$$I(A) = -\log_2 P(A) \ ,$$

which we can take as the definition of the amount of information obtained in the result of realizing an event A.

When talking about coin flips, we have actually used the following general principle. When two events are independent, the uncertainty of the joint event (i.e., of their intersection) should be the sum of the measures of uncertainty of each single event:

$$I(A \cap B) = I(A) + I(B) \ ,$$

and our definition of uncertainty matches this observation, because the independence of A and B means precisely that:

$$P(A \cap B) = P(A) \times P(B) \ .$$

Since the key idea of the definition of uncertainty is the additivity for intersections of independent events, the choice of the logarithm base is a matter of convenience. One could choose the logarithm base according to the number of outcomes of a given experiment. For instance, if one started with an experiment that has 10 outcomes, one would have arrived at base 10 logarithm $I(A) = -\log_{10} P(A)$, and the corresponding *decimal unit* is called *dit* or *hartley*. Comparing

$$-\log_2\left(\frac{1}{2}\right) = 1 \text{ and } -\log_{10}\left(\frac{1}{2}\right) \approx 0.30 \ ,$$

the *decimal unit* is roughly 3.32 times greater than the *binary unit (bit)*. It is also quite important that we keep natural logarithm in mind, $I(A) = -\ln P(A)$, and the corresponding unit is *nat*. However, to keep things simple most researchers choose to omit the base of the system of logarithms and simply use *binary logarithm* when discussing information theory, which is $I(A) = -\log_2 P(A)$.

Remark 1. The reason why the binary unit is so convenient is the overwhelming use of the binary system in computers and telecommunication (almost entirely digital nowadays), due to the physical nature of the elements used in the corresponding devices. This consideration was also at the origin of the Morse code (contrary to the codes used in earlier optical telegraphs). It encodes text characters as standardized

sequences of two different signals (*dot* and *dash* or *dit* and *dat*). "A" is "$*-$" and there are no distinctions between uppercase and lowercase letters in Morse code. The letters in Morse code are not equal length. More commonly used letters are defined by shorter sequences to optimize the efficiency in which communication can happen (cf. the discussion on optimal codes in Section 4.2.1).

1.4 Average amount of information

In the previous section, we described a measure of uncertainty contained just in a single event, or in an elementary outcome of an experiment (if it has positive probability). Now, we are interested in finding the measure of uncertainty for a whole experiment.

Consider a probability space (Ω, P); an experiment is presented by its partition α into n subsets A_i. Each element A_i of the partition α has probability $p_i = P(A_i)$; all the probabilities combined give us a discrete distribution $p = (p_i)$.

Definition 1.4.1. The entropy $H(\alpha)$ of a finite partition α is the average amount of information of its elements, i.e.,

$$H(\alpha) = \sum_{i=1}^{n} I(A_i)P(A_i) = -\sum_{i=1}^{n} \log P(A_i)P(A_i) \ , \tag{1.4.1}$$

or, in terms of the distribution $p = (p_i)$,

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i \ . \tag{1.4.2}$$

In the context of information theory this notion is first introduced by Claude Shannon, the "father" of information theory, in his famous book *Mathematical Theory of Communication* [Sha48]. However, the concept of entropy can be tracked back to the much earlier works of Boltzmann, Gibbs, Planck and von Neumann in other fields.

It is easy to see that the numerical measure of uncertainty for an experiment with one possible outcome would be zero, since there is no uncertainty on the outcome. If an experiment has a large number of outcomes, then the prediction of an outcome becomes quite difficult, therefore the uncertainty of the result or we can say the entropy of this experiment would be relatively high. We will return to this observation when we rigorously formulate the main properties of entropy in section 1.5.

So far we have only defined the entropy of a partition and the entropy of a distribution. Now let's consider the entropy from a random variable perspective. The entropy of a random variable is the entropy of its distribution, or equivalently, the

entropy of its pre-image partition. More precisely, let X be a discrete real valued random variable, i.e, a map from the probability space to real numbers

$$X : \Omega \to \mathbb{R} \, .$$

Then the elements of the associated preimage partition α^X are

$$A_i = X^{-1}(x_i) = \{\omega : X(\omega) = x_i\} \, ,$$

where x_i are the values that X takes with positive probability. Then, the entropy of the random variable X is the entropy of its distribution p^X, i.e., the entropy of the preimage partition

$$H(X) = H(p^X) = H(\alpha^X) \, . \tag{1.4.3}$$

Remark 2. By passing to the preimage partition, one loses the information about the values of the random variable, and only retains the information about whether the values of the random variable at two different points are the same or not.

Example 1.4.1. Let the probability space be

$$\Omega = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

endowed with the uniform distribution $P = Unif(\Omega)$, and let X be the following random variable:

$$X\,(1) = X\,(2) = X\,(3) = X\,(4) = 5, \; X\,(5) = 3, \; X\,(6) = X\,(7) = 0, \; X\,(8) = -2 \, .$$

The preimage partition is

$$\begin{aligned} \{1, 2, 3, 4\} &= X^{-1}(5) = \{\omega : X(\omega) = 5\} \, , \\ \{5\} &= X^{-1}(3) = \{\omega : X(\omega) = 3\} \, , \\ \{6, 7\} &= X^{-1}(0) = \{\omega : X(\omega) = 0\} \, , \\ \{8\} &= X^{-1}(-2) = \{\omega : X(\omega) = -2\} \, . \end{aligned}$$

The distribution of X is

$$p^X\,(5) = \tfrac{1}{2}, \; p^X\,(3) = \tfrac{1}{8}, \; p^X\,(0) = \tfrac{1}{4}, \; p^X\,(-2) = \tfrac{1}{8} \, .$$

Then the entropy of the partition becomes

$$\begin{aligned} H(X) =& H\left(p^X\right) \\ =& -\tfrac{1}{2} \times \log_2\left(\tfrac{1}{2}\right) - \tfrac{1}{4} \times \log_2\left(\tfrac{1}{4}\right) - \tfrac{1}{8} \times \log_2\left(\tfrac{1}{8}\right) - \tfrac{1}{8} \times \log_2\left(\tfrac{1}{8}\right) = \tfrac{7}{4} \, . \end{aligned}$$

The entropy formula can also be rewritten as:

$$H(\alpha) = -\int \log P(\alpha(\omega))dP(\omega) \ ,$$

where $\alpha(\omega)$ is the element of the partition α that contains ω. The advantage of this definition is that it expresses the entropy of a partition as an integral over the whole probability space.

Example 1.4.2. Let now Ω be the unit interval endowed with the uniform distribution $P = Unif(0,1)$, and let Y be the random variable defined as

$$Y(\omega) = \begin{cases} 5, \omega \in (0, \frac{1}{2}) \\ 3, \omega \in (\frac{1}{2}, \frac{5}{8}) \\ 0, \omega \in (\frac{5}{8}, \frac{7}{8}) \\ -2, \omega \in (\frac{7}{8}, 1) \end{cases}$$

The graph of Y is presented in Figure 1.1, and the distribution p^Y of Y is

$$p^Y(5) = \tfrac{1}{2},\ p^Y(3) = \tfrac{1}{8},\ p^Y(0) = \tfrac{1}{4},\ p^Y(-2) = \tfrac{1}{8}$$

(the same as p^X from Example 1.4.1), and the preimage partition is

$$\begin{aligned} (0, \tfrac{1}{2}) &= Y^{-1}(5) = \{\omega : Y(\omega) = 5\} \ , \\ (\tfrac{1}{2}, \tfrac{5}{8}) &= Y^{-1}(3) = \{\omega : Y(\omega) = 3\} \ , \\ (\tfrac{5}{8}, \tfrac{7}{8}) &= Y^{-1}(0) = \{\omega : Y(\omega) = 0\} \ , \\ (\tfrac{7}{8}, 1) &= Y^{-1}(-2) = \{\omega : Y(\omega) = -2\} \ . \end{aligned}$$

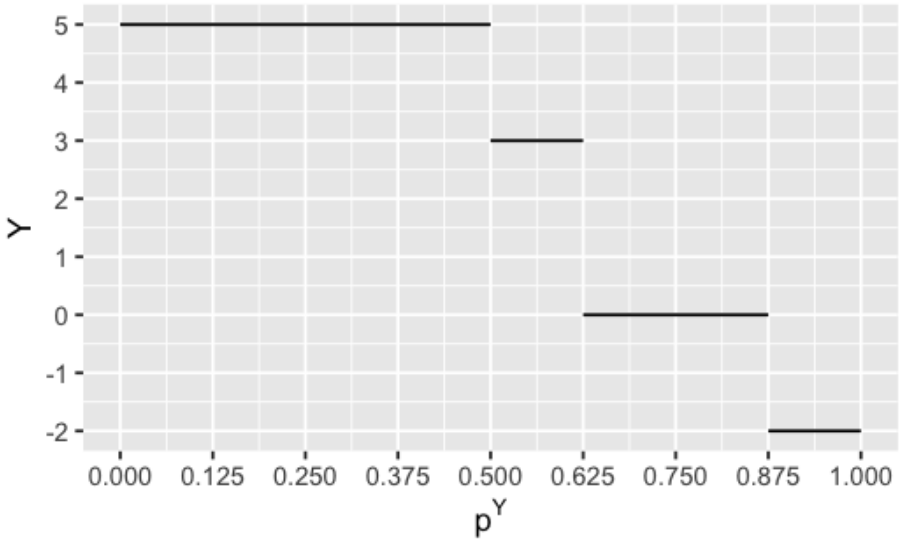

Figure 1.1: Graphical representation of a random variable

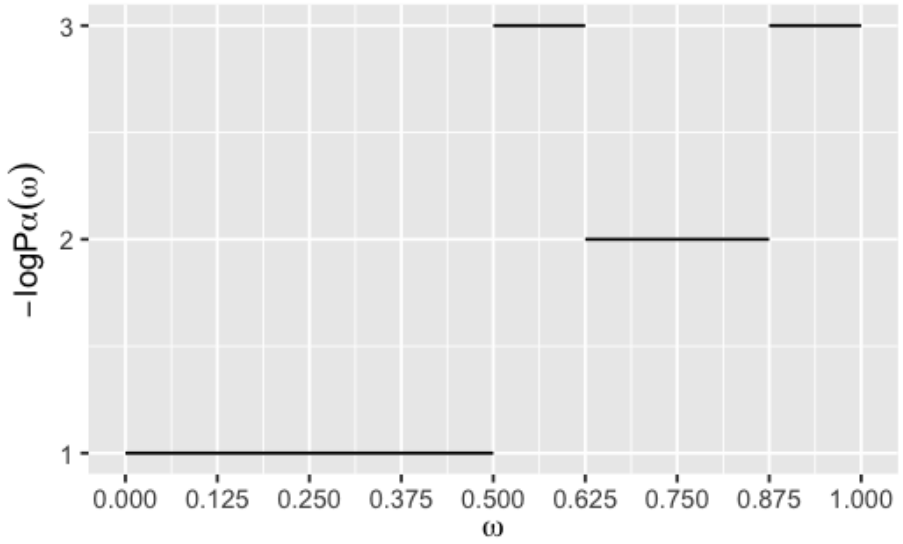

Figure 1.2: The associated information function

The graph of the resulting function

$$\omega \mapsto -\log P\left(\alpha^Y(\omega)\right)$$

is presented in Figure 1.2. From this graph of the information function, for instance,

$$P(\alpha(0.3)) = \frac{1}{2^1} = \frac{1}{2} \text{ or } P(\alpha(0.51)) = \frac{1}{2^3} = \frac{1}{8}\,.$$

Using Figure 1.2 to find the value of the integral, the entropy of the partition becomes,

$$\begin{aligned} H(\alpha) &= \int_0^1 (-\log P(\alpha(\omega)))\, d\omega \\ &= \int_0^{1/2} (-\log P(\alpha(\omega)))\, d\omega + \int_{1/2}^{5/8} (-\log P(\alpha(\omega)))\, d\omega \\ &\quad + \int_{5/8}^{7/8} (-\log P(\alpha(\omega)))\, d\omega + \int_{7/8}^{1} (-\log P(\alpha(\omega)))\, d\omega \\ &= 1 \times \tfrac{1}{2} + 3 \times \tfrac{1}{8} + 2 \times \tfrac{1}{4} + 3 \times \tfrac{1}{8} = \tfrac{7}{4} \end{aligned}$$

which is the same result as $H(p^X)$ from the previous example.

1.5 Properties of entropy

1.5.1 Entropy function

Let's first introduce the entropy function

$$\varphi(t) = -t \log t \ .$$

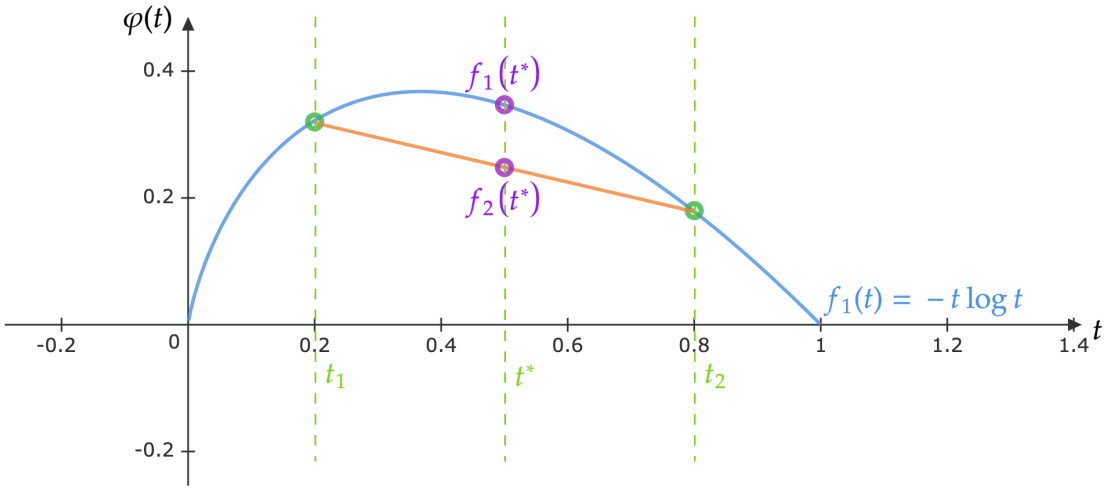

Figure 1.3: Graph of the entropy function $\varphi(t) = -t \log t$

We are interested in this function on the interval $(0, 1]$, and extended it by continuity to the point 0 by putting $\varphi(0) = 0$, so that φ is defined on the whole closed interval $[0, 1]$, vanishes at its endpoints, and is strictly positive otherwise. The entropy of a discrete (not necessarily finitely supported) distribution $p = (p_i)$ can be rewritten as:

$$H(p) = -\sum_{i=1}^{n} p_i \log p_i = \sum_{i=1}^{n} \varphi(p_i) \ .$$

1.5.2 Concavity and convexity

Below we need the fact that the function φ is concave.

A function f defined on an interval X is called *convex* if and only if for all $0 < t < 1$, $x_1, x_2 \in X$, and $x_1 \neq x_2$:

$$f\left(tx_1 + (1-t)x_2\right) \leq tf(x_1) + (1-t)f(x_2) \ .$$

The function is said to be *concave* if $-f$ is convex,

$$f\left(tx_1 + (1-t)x_2\right) \geq tf(x_1) + (1-t)f(x_2) \ .$$

These inequalities are often called *Jensen's Inequality*.

If the above inequality is strict, then the function is called strictly convex (resp., concave). If f is twice differentiable, then it is convex if and only if $f'' \geq 0$ on X (resp., strictly convex if $f'' > 0$ on the interior of X).

Then we calculate the second derivative of the entropy function,

$$\begin{aligned}(-t\log t)' &= -\log t - 1\\ (-t\log t)'' &= -\tfrac{1}{t}\ ,\end{aligned}$$

of which the result is always negative. One can also immediately see that φ is concave from looking at its graph:

In Figure 2.3, define the original function to be $f_1(t)$,

$$f_1(t) = \varphi(t) = -t\log t\ ,$$

then pick any two points t_1 and t_2 where $t_1 < t_2$. Define $f_2(t)$ to be the straight line connecting two points on $f_1(t)$, $-t_1\log(t_1)$ and $-t_2\log(t_2)$. For any t^*, where $t_1 < t^* < t_2$, we have

$$f_1(t^*) > f_2(t^*)\ .$$

Therefore, according to the Figure 2.3, we can conclude that $\varphi(t)$ is strictly concave *(convex to the top)*.

The first property of the entropy (Proposition 1.5.1) immediately follows from the fact that the function φ is non-negative.

1.5.3 Entropy inequalities

Proposition 1.5.1. The entropy of any finite partition α is non-negative and

$$H(\alpha) = 0$$

if and only if the partition α consists of an atom of probability one.

As mentioned previously, we will now use the concavity of the function φ.

Proposition 1.5.2. The entropy of a partition with a fixed number of elements is maximal when their probabilities are equal. In terms of distributions

$$H(p_1,\ldots,p_n) \leq \log n\ ,$$

and the equality is attained if and only if $p_i = \frac{1}{n}$.

Proof. Let us notice that for the uniform distribution

$$H\left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right) = n\varphi\left(\frac{1}{n}\right)$$

is indeed $\log n$. In the case when the probabilities p_i are not all equal, we apply the concave Jensen's inequality with equal weights $\frac{1}{n}$, which gives

$$\varphi\left(\frac{p_1 + \cdots + p_n}{n}\right) > \frac{1}{n}\sum_{i=1}^{n} \varphi(p_i) \ ,$$

The inequality here is strict, because the function φ is strictly concave, and the probabilities p_i are not equal. Since

$$p_1 + \cdots + p_n = 1 \ ,$$

in terms of entropy, this inequality becomes

$$H(p) = \sum_{i=1}^{n} \varphi(p_i) < n\varphi\left(\frac{1}{n}\right) = \log n \ .$$

□

Another natural way to prove this property is by using the concavity of the function

$$p \mapsto H(p)$$

defined on the space of distributions instead of just the concavity of φ.

Proposition 1.5.3. The entropy is strictly concave, i.e., for any $0 < t < 1$,

$$H(tp + (1-t)p') \geq tH(p) + (1-t)H(p') \ , \tag{1.5.4}$$

and the equality holds if and only if $p = p'$. In other words, the entropy of a *convex combination (weighted average)* of the two distributions is greater than the convex combination of the entropies.

Proof. By the definition of the entropy,

$$\begin{aligned} H(tp + (1-t)p') &= \sum_i \varphi(tp_i + (1-t)p'_i) \\ &\geq \sum_i [t\varphi(p_i) + (1-t)\varphi(p'_i)] \text{ by concavity of the function} \\ &= tH(p) + (1-t)H(p') \ . \end{aligned}$$

Since the function φ is strictly concave, equality holds for all i, i.e., if and only if $p_i = p'_i$.

□

Second Proof of Proposition 1.5.3 (maximality). Now we consider a total of n distributions, defined in the following way: the first distribution p^1 is the original one, we have

$$(p_1, p_2, \ldots, p_n) = p^1 = p \ ,$$

the second distribution p^2 has the same weights cyclically shifted by 1, i.e.,

$$(p_2, p_3, \ldots, p_1) = p^2 \ ,$$

and so on. And finally, the n^{th} distribution p^n is

$$(p_n, p_1, \ldots, p_{n-1}) = p^n \ .$$

For all these distributions p^i are obtained by cyclic permutation of the same weights, and therefore each of them still has the same entropy $H(p^i) = H(p)$. Then the average of these distributions is the uniform distribution

$$q = (\tfrac{1}{n}, \tfrac{1}{n}, \ldots, \tfrac{1}{n}) \ .$$

The concavity inequality from Formula (1.5.4) tells us that the entropy of this convex combination that is of the uniform distribution will be greater than or equal to the convex combination of the entropies, which are all the same and equal to the entropy of the original distribution, i.e.,

$$H(p) = \sum \frac{1}{n} H\left(p^i\right) \leq H\left(\sum \frac{1}{n} p^i\right) = H(q) \ .$$

Since entropy is strictly concave, the inequality is strict unless all p^i are the same, which can only be the case when p is uniform.

□

1.5.4 Comparison of entropies

Given a discrete probability distribution $p = (p_0, \ldots, p_n)$, let $p' = (p'_1, \ldots, p'_n)$ be a new distribution obtained from p by removing the atom p_0 and then renormalizing the remaining weights, so that

$$p'_i = \frac{p_i}{1 - p_0} \ .$$

An example of this situation is provided by the frequency distributions in the English language. Here $n = 26$ (the number of letters in the English alphabet), p is the frequency distribution of the "extended" alphabet (the space is considered as a symbol; its frequency is denoted by p_0), and p' is the frequency distribution of the 26 "genuine" letters.

In order to compare the entropies $H = H(p)$ and $H' = H(p')$ of these distributions, it is convenient to use the language of partitions. We denote by α the partition of a probability space (Ω, P) into sets $A_0, \ldots, A_n$ such that $P(A_i) = p_i$, so that

$$H = H(\alpha) \ .$$

Let β be the 2-element partition of Ω with the elements A_0 and

$$\overline{A}_0 = \Omega \setminus A_0 = A_1 \cup \cdots \cup A_n \ ,$$

so that α is a refinement of β. Then

$$H(p) = H(\alpha) = H(\alpha \vee \beta) = H(\beta) + H(\alpha|\beta) \ ,$$

where

$$H(\alpha|\beta) = P(A_0)H(\alpha|A_0) + P\left(\overline{A}_0\right) H\left(\alpha|\overline{A}_0\right) = (1 - P(A_0))H(p') \ ,$$

because the conditional distribution of the partition α under the condition $\overline{A}_0$ is precisely p'. Thus, finally

$$H = \psi(t) + (1 - t)H' \ , \tag{1.5.5}$$

where

$$\psi(t) = -t \log t - (1 - t) \log(1 - t) = H(\beta)$$

is the *binary entropy function* of the parameter $t = P(A_0)$, and in our example t is the frequency of the "space" as a symbol of the extended English alphabet.

Therefore, $H < H'$ if and only if $\psi(t) < tH'$, i.e.,

$$H' > \frac{\psi(t)}{t} \ . \tag{1.5.6}$$

1.6 Conditional entropy of finite distributions

In the following section, we switch from the language of partitions to the language of variables. we are interested in defining the entropy for finite alphabet processes, which provides us with an essential tool for studying information theory.

Conditional probability is the probability with respective to a certain algebra (more generally, σ-algebra) of events. Later on, when talking, for instance, about

conditioning by a certain letter, we actually mean conditioning by the algebra of the events described by this letter. In the same way, conditioning by 2 letters, means conditioning by the algebra of the events generated by these letters. There are more events described by 2 letters than by 1 letter. The finer the partition is, the more information it provides.

For example, a simple partition is: "if it rains or not". A finer partition would be when more specific information is provided about "how much it rains". If you condition by this finer partition, it means you probably already know the answer to the first question. The more you know, i.e., the more information is contained in the condition, the less information this observation brings to you.

Definition 1.6.1. The conditional entropy of a discrete random variable X given another discrete random variable Y is:

$$H(X|Y) \equiv H(X,Y) - H(Y) \ . \tag{1.6.1}$$

Therefore,

$$\begin{aligned} H(X|Y) =& H(X,Y) - H(Y) \\ =& -\sum p_{X,Y}(x,y) \log p_{X,Y}(x,y) + \sum_y p_Y(y) \log p_Y(y) \\ =& -\sum p_{X,Y}(x,y) \log p_{X,Y}(x,y) + \sum_{x,y} p_{X,Y}(x,y) \log p_Y(y) \\ =& -\sum_{x,y} P_{X,Y}(x,y) \log \left(\frac{p_{X,Y}(x,y)}{p_Y(y)} \right) \\ =& -\sum_{x,y} p_{X,Y}(x,y) \log p_{X|Y}(x|y) \ , \end{aligned}$$

where $p_{X,Y}(x,y)$ is the joint probability mass function for (X,Y) and

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$

is the conditional probability mass function of X conditioned by Y. In yet another interpretation the entropy of the conditional distribution of X conditioned by a value y of Y is

$$\begin{aligned} H(X|Y=y) =& -\sum_x p_{X|Y}(x|y) \ln p_{X|Y}(x|y) \\ =& -\sum_x \frac{p_{X,Y}(x|y)}{p_Y(y)} \ln p_{X|Y}(x|y) \ . \end{aligned}$$

Whence the conditional entropy is the average of the entropies of conditional distributions

$$H(X|Y) = \sum_y p_Y(y) H(X|Y=y) \ . \tag{1.6.2}$$

Proposition 1.6.3. Given two discrete random variables X, Y, the conditional entropy of X with respect to Y satisfies the inequalities

$$0 \leq H(X|Y) \leq H(X) \ ,$$

(i) The left inequality holds as equality if and only if X is a function of Y.

(ii) The right inequality holds as equality if and only if X and Y are independent.

Proof. (i) In view of Formula (1.6.2),

$$H(X|Y) = 0$$

if and only if all

$$H(X|Y=y) = 0 \ ,$$

which happens if and only if the conditional distributions of X given Y are all at one point, which means that X is, indeed, a function of Y.

(ii) We have

$$p_X = \sum p_Y(y)\, p_{X|Y=y} \ ,$$

whence by concavity of entropy (see Formula (1.5.4)),

$$H(X) = H(p_X) = H\left(\sum p_Y(y)\, p_{X|Y=y}\right) \geq \sum p_Y(y) H\left(p_{X|Y=y}\right) \tag{1.6.4}$$

and the equality holds if and only if

$$p_{X|Y=y}$$

are all the same, i.e., X and Y are independent.

□

Corollary 1.6.5. (Subadditivity of entropy)

$$H(X,Y) \leq H(X) + H(Y) \ ,$$

and the equality holds if and only if X and Y are independent.

Proof. By Definition 1.6.1,

$$H(X,Y) = H(X|Y) + H(Y) \ ,$$

whereas,

$$H(X|Y) \leq H(X)$$

with the equality if and only if X and Y are independent by Proposition 1.6.3.

□

Proposition 1.6.6. If $Z = f(Y)$, then $H(X|Y) \leq H(X|Z)$.

Proof. We use the formula

$$H(X|Y) = \sum p_Y(y) H(X|Y=y) \ ,$$

and in the same way, we also have

$$H(X|Z) = \sum p_Z(z) H(X|Z=z) \ .$$

Using the fact that

$$p_{X|Z=z} = \sum_y p_{Y|Z}(y|z) p_{X|Y=y} \ ,$$

therefore, by Formula (1.6.4) for each z,

$$H(p_{X|Z=z}) = H\left(\sum_y p_{Y|Z}(y|z)) p_{X|Y=y}\right) \geq \sum_y p_{Y|Z}(y|z) H(p_{X|Y=y}) \ .$$

Then summing by z,

$$\begin{aligned} H(X|Z) = \sum_z p_Z(z) H(p_{X|Z=z}) &\geq \sum_z p_Z(z) \sum_y p_{Y|Z}(y|z) H(p_{X|Y=y}) \\ &= \sum_y p_Y(y) H(p_{X|Y=y}) \\ &= H(X|Y) \ . \end{aligned}$$

Hence,

$$H(X|Y) \leq H(X|Z) \ .$$

□

Chapter 2

Shannon's model and asymptotic entropy

2.1 Prefix code and binary tree

Definition 2.1.1. A **code** of a finite set Σ is a map

$$S : \Sigma \mapsto A^* ,$$

where A is the coding alphabet and A^* is the set of all words in this coding alphabet, including the empty one.

For instance, when talking about English language texts, one can take Σ to be the set of all Latin alphabet letters (without distinguishing upper and lower case) and the space, i.e., altogether 27 symbols. As for the coding alphabet A, usually one takes the two symbol alphabet $A = \{0, 1\}$. $S(\Sigma) \subset A^*$ is the set of all code words, and the words $S(\sigma)$ for $\sigma \in \Sigma$ is are called *code words*.

The first few consecutive bits (symbols) of a code word is a *prefix*. If there are two code words with different lengths, the shorter word is said to be a prefix of the longer one if the former coincides with an initial segment of the latter one. For example, let's say $u = 01, v = 010, w = 0$, then "u" is a prefix of "v", "w"is a prefix of both "u" and "v".

One can also add extra symbols, for example "/", in the encoded message to make the decoding process easier, for example "010010", then it is clear to us that "01/0/010" gives us "uwv". However, this method is not as efficient as one extra symbol being added at every spot. If there is only one way to decode the encoded message, then such a code has *unique decodability*,

Definition 2.1.2. A prefix code is a code for which there is no code word that is a prefix of another code word. It is a code that can be uniquely deciphered by reading the message consecutively, character by character.

Remark 3. All fixed length codes (for example, Unicode's UTF-8 and ASCII) are prefix binary code.

Example 2.1.1. A commonly recognized example of code is Morse code. It is a type of code, but not a prefix code. For Morse code, $A = \{\cdot, -\}$, Σ is the English alphabet.

- The code word for "E" is "$\cdot$"
- The code word for "S" is "$\cdot\cdot\cdot$"
- The code word for "H" is "$\cdot\cdot\cdot\cdot$"

One has to use their knowledge of language to decode the given code word. For example, "$\cdot\cdot\cdot\cdot$" can be "H", "ES","SE" or "SSSS". Morse code is not a type of prefix code, and it is not uniquely decodable.

Prefix codes are a subset of all uniquely decodable codes. However, one can not conclude that such code is not uniquely decodable solely based on that fact that it is not a prefix code. For example, one can encode (a,b,c,d) with (0,01,011,0111), this is not a prefix code, however it is uniquely decodable.

A tree data structure in which each vertex has at precisely two offspring, often represented with "0" and "1", is called a *binary tree*. An illustration of this can be found in Figure 2.1. One can visualize a prefix binary code as a subset of vertices (code words) of a binary tree such that the subtree based at each of these vertices does not contain any other vertices from this subset. The length of a code word is the depth of the corresponding vertex. This type of tree is often referred to as a *code tree* or *prefix tree*. A code tree is the subtree of a binary tree spanned by the code words.

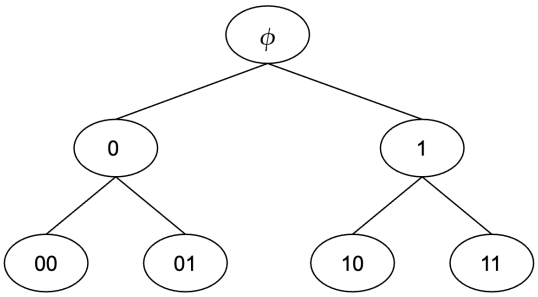

Figure 2.1: Binary tree

In 1952, David Huffman developed an algorithm which was later called *Huffman coding* [Huf52]. This algorithm assumes that the set to be encoded is endowed with a probability distribution, and the algorithm consists of recursive pooling together of the two least frequent elements. One wants to make the code words corresponding to the most frequent elements as short as possible.

Example 2.1.2. There are 4 characters a, b, c, d with the weights

$$P(a) = \frac{2}{5},\ P(b) = \frac{3}{10},\ P(c) = \frac{1}{5},\ P(d) = \frac{1}{10}.$$

We start with a, b, c, d. c and d have the least two frequencies in my case, therefore these two characters would be merged together into one tree, a, b, cd.

Then this merged tree will be combined into another tree with the next least frequent character, which is b in this case, receiving a, bcd.

Now we only have one character a, so we can stop the tree at this point after merging everything we have created above.

Therefore, we can create the following prefix code,

$$a = 0, b = 10, c = 110, d = 111\ .$$

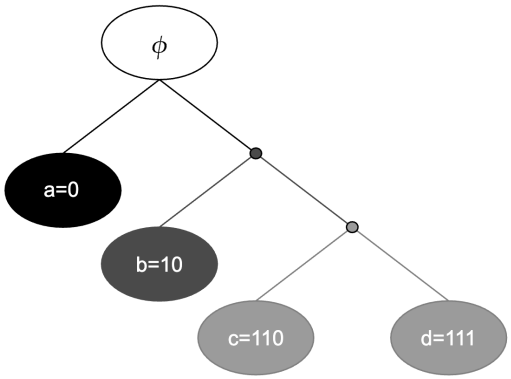

Figure 2.2: Prefix tree for Example 2.1.2

Theorem 2.1.1. (Kraft's inequality) A prefix binary code

$$S : \Sigma \to \{0,1\}^*$$

with a prescribed set of code word lengths $|S(\sigma)|\ , \sigma \in \Sigma$ exists if and only if

$$K(S) = \sum_{\sigma \in \Sigma} 2^{-|S(\sigma)|} \leq 1\ . \tag{2.1.1}$$

Proof. We denote the elements of Σ by σ_i with $1 \leq i \leq q$, where q is the number of elements of Σ, and denote by $l_i = |S(\sigma_i)|$ the lengths of the corresponding code words. The proof of the theorem consists of two parts.

$\Longrightarrow$) If there exists a prefix code with the code word lengths l_i, then inequality (2.1.1) is satisfied. We consider two cases:

1. When $q \leq 2$, we have at most 2 code words that have length of at least 1,

$$K(S) \leq 2^{-1} + 2^{-1} = 1 \, .$$

2. When $q > 2$.

   - We have n_1 code words with length of 1, so that $n_1 \leq 2$.
   - We have n_2 code words with length of 2, so that $n_2 \leq 2(2 - n_1)$.
   - We have n_3 code words with length of 3, so that

$$n_3 \leq 2\left(2(2 - n_1) - n_2\right) = 2^3 - 2^2 n_1 - 2n_2 \, .$$

   - ......
   - For the code words with length of j, we have

$$n_j \leq 2^j - 2^{j-1} n_1 - 2^{j-2} n_2 - \cdots - 2n_{j-1} \, .$$

In the above inequality, move all the negative terms to the left hand side,

$$2^{j-1} n_1 + 2^{j-2} n_2 + \cdots + 2n_{j-1}^1 + 2^0 n_j \leq 2^j \, .$$

Multiplying by 2^{-j} on both sides,

$$2^{-1} n_1 + 2^{-2} n_2 + \cdots + 2^{-(j-1)} n_{j-1} + 2^{-j} n_j \leq 1 \, .$$

As n_j is the total number of code words with length of j and $n_1 + n_2 + \cdots + n_j = q$,

$$\sum_{k=1}^{n_1} 2^{-1} + \sum_{k=1}^{n_2} 2^{-2} + \cdots + \sum_{k=1}^{n_j} 2^{-j} \leq 1 \, .$$

Therefore,

$$\sum_{k=1}^{q} 2^{-l_k} \leq 1 \, ,$$

which is the claim.

$\Longleftarrow$) If the code word lengths satisfy Inequality (2.1.1), then there exists a prefix code with these lengths.

Let us rewrite the Kraft inequality (2.1.1),

$$\sum_{i=1}^{q} 2^{l_q - l_i} \leq 2^{l_q} \tag{2.1.2}$$

.

Firstly, one can choose any vertex at the depth l_1, which corresponds to the first word of the code. As the goal is to create a prefix code, all the descendants of this vertex have to be removed (i.e. there are no other words that would have this first word as a prefix). Now, if we consider the number of descendants at level l_q, a total of $2^{l_q - l_1}$ descendants are removed from consideration.

Secondly, pick one of the surviving vertices at depth l_2 to be the second word of the code. Then one has to remove a total of $2^{l_q - l_2}$ descendants at the l_q level.

After $q-1$ iterations of this procedure, the total number of vertices that are removed at depth l_q is

$$\sum_{i=1}^{q-1} 2^{l_q - l_i} \ .$$

Since there are only $q-1$ summands, and one has a strict inequality, Formula (2.1.2) becomes,

$$\sum_{i=1}^{q-1} 2^{l_q - l_i} < 2^{l_q} \ .$$

As 2^{l_q} is the total number of vertices at depth l_q and strict inequality holds, one can always find an available vertex at the l_q level. Therefore, at each step of the procedure there is still an available vertex at the bottom level, and then all the upward lineage of this vertex is also available, so that one can always choose an available vertex on any intermediate level.

Hence, a prefix code can be built. However, the choice of vertices at each iteration is largely arbitrary. In general, many different prefix codes can be built through the procedure described above. Figure 2.3 is an illustration of this proof.

□

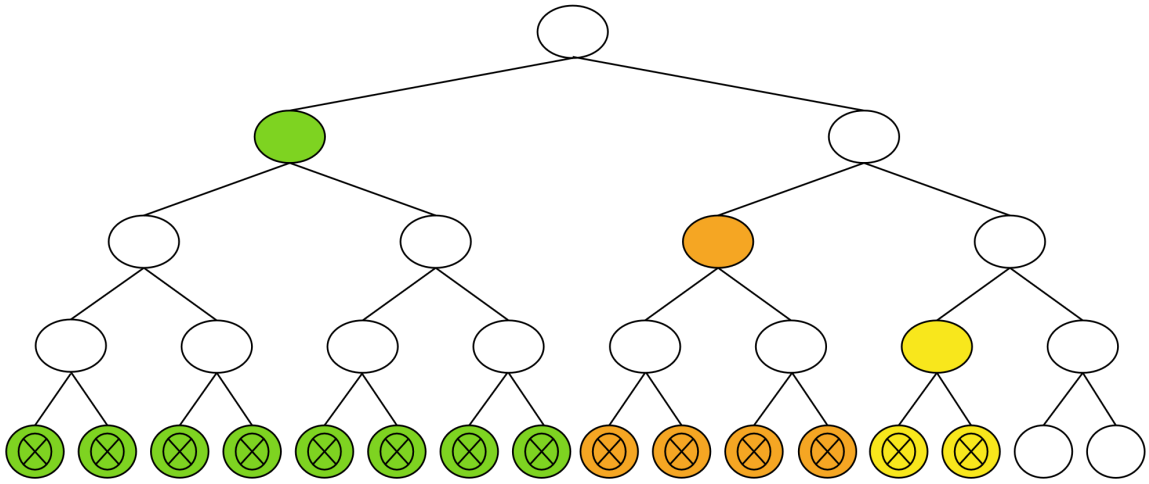

Figure 2.3: Illustration of the proof of Kraft's inequality

Remark 4. If a given set of code word lengths do satisfy the Kraft inequality, then one can conclude that it is possible to find a prefix code with these lengths. However, the fact that the code word lengths satisfy the Kraft inequality alone does not imply the prefix property. For example, (0, 00, 10) is not a prefix code, although its lengths satisfy the Kraft inequality.

Remark 5. Given two sets of code word lengths (l_i) and (l'_i), we say that (l'_i) is a reduction of (l_i) if $l'_i \leq l_i$ and the inequality is strict for at least one i. If the Kraft inequality is strict, i.e.,

$$\sum 2^{-l_i} < 1 \, .$$

The difference

$$1 - \sum 2^{-l_i}$$

is of the form $\frac{n}{2^{l_q}}$. Since

$$2^{-l_q+1} - 2^{-l_q} = 2^{-l_q} \, ,$$

if l_q is replaced with $l_q - 1$, then the Kraft inequality is still satisfied.

A length reduction is possible if and only if the Kraft inequality is strict. In another words, the Kraft inequality holds equality if and only if the code word lengths can not be reduced.

Example 2.1.3. let's say we have $l_1 = 1, l_2 = 3$ and $l_3 = 4$.

- $2^{-2} + 2^{-3} + 2^{-4} = \frac{11}{16}$, then the difference is $1 - \sum_{i=1}^{3} 2^{-l_i} = \frac{5}{16}$. Therefore, we can replace $\frac{1}{16}$ with $\frac{1}{8}$ (because the gain is only $\frac{1}{16}$), so that (1,3,3) is also admissible.

- $2^{-1}+2^{-3}+2^{-3}=\frac{3}{4}$, $1-\sum_{i=1}^{3}2^{-l_i}=\frac{1}{4}$. We can replace $\frac{1}{8}$ with $\frac{1}{4}$ and the Kraft inequality is still satisfied, so that (1,2,3) is also admissible.
- $2^{-1}+2^{-2}+2^{-3}=\frac{7}{8}$, $1-\sum_{i=1}^{3}2^{-l_i}=\frac{1}{8}$. We can replace $\frac{1}{8}$ with $\frac{1}{4}$, since the gain is $\frac{1}{8}$. Now we receive equality for the Kraft inequality, so that we have $l_1=1, l_2=2, l_3=2$, and lengths can not be further reduced.

Let's consider the following

$$(l_i)\to(l_i')$$

such that $l_i'\leq l_i$ and the inequality is strict for at least one i, this is what is called *length reduction*. For $\sum 2^{-l_i}<1$, the difference $1-\sum 2^{-l_i}$ is of the form $\frac{n}{2^{l_q}}$. Therefore, if l_q is replaced with l_{q-1}, then the Kraft inequality is still satisfied.

2.2 Shannon's source coding theorem

Now let us assume that the original set Σ in the definition of a code is additionally endowed with a probability distribution p. For instance, if Σ is the English alphabet, then p is the frequency distribution of letters.

Shannon's source coding theorem states that as the length of a string sampled from a stationary stream of symbols goes to infinity, it is possible to compress the string without losing any information in such a way that the average bit per character is asymptotically the same as the Shannon's entropy of the source.

Remark 6. Let us remind ourselves of the standing assumption that the logarithm base in the definition of H is equal to 2.

Theorem 2.2.1. For any prefix code $S:\Sigma\to\{0,1\}^*$ and any distribution p on Σ,

$$H(p)\leq\mathbb{E}[S]\ ,$$

where $\mathbb{E}[S]=\sum_{i=1}^{n}p_il_i$ is the expected length of the code words with respect to the distribution p and $l_i=|S(\sigma_i)|$ are the lengths of the code words representing the elements σ_i of the set Σ.

Proof. Here we want to show that

$$H(p)-\mathbb{E}[S]\leq 0\ .$$

Indeed,

$$H(p) - \mathbb{E}[S] = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} - \sum_{i=1}^{n} p_i l_i$$

$$= \sum_{i=1}^{n} p_i \left(\log \frac{1}{p_i} - l_i \log 2 \right)$$
$$= \sum_{i=1}^{n} p_i \left(\log \frac{1}{p_i} - \log 2^{l_i} \right)$$
$$= \sum_{i=1}^{n} p_i \log \frac{1}{p_i 2^{l_i}}$$
$$= \log e \sum_{i=1}^{n} p_i \ln \frac{1}{p_i 2^{l_i}} \ .$$

For all $x > 0$, we have $\ln x \leq x - 1$, then we get

$$\log e \sum_{i=1}^{n} p_i \ln \frac{1}{p_i 2^{l_i}} \leq \log e \sum_{i=1}^{n} p_i \left(\frac{1}{p_i 2^{l_i}} - 1 \right)$$
$$= \log e \left(\sum_{i=1}^{n} \frac{1}{2^{l_i}} - \sum_{i=1}^{n} p_i \right)$$
$$= \log e \left(\sum_{i=1}^{n} 2^{-l_i} - 1 \right) \ .$$

From Theorem 2.1.1,

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1 \ ,$$

then,

$$\sum_{i=1}^{n} 2^{-l_i} - 1 \leq 0 \ .$$

Since $\log e > 0$, the inequality becomes

$$H(p) - \mathbb{E}[S] \leq \log e \left(\sum_{i=1}^{n} 2^{-l_i} - 1 \right) \leq 0 \ .$$

□

Definition 2.2.1. The **code efficiency** of a source is the ratio of the entropy of the text to the expected code word length of encoded text,

$$E(p, S) = \frac{H(p)}{\mathbb{E}[S]} \ .$$

By Theorem 2.1.1, the code efficiency is always less than or equal to 1. If the equality is attained, i.e., the code efficiency is equal to 1, then the code is called *optimal*.

Theorem 2.2.2. Under the same conditions as in Theorem 2.1.1, there exists a code with

$$\mathbb{E}[S] < H(p) + 1 \ .$$

Proof. For each weight p_i of the distribution p, $1 \leq i \leq n$, let

$$l_i = \log \frac{1}{p_i} + \varepsilon_i \ , 0 \leq \varepsilon_i < 1 \ ,$$

so that,

$$l_i < \log \frac{1}{p_i} + 1 \ .$$

Multiplying by p_i on both sides,

$$p_i l_i < p_i \left(\log \frac{1}{p_i} + 1 \right) \ .$$

Whence,

$$\begin{aligned} \mathbb{E}[S] = \sum_{i=1}^{n} p_i l_i &< \sum_{i=1}^{n} p_i \left(\log \frac{1}{p_i} + 1 \right) \\ &= \sum_{i=1}^{n} p_i \log \frac{1}{p_i} + \sum_{i=1}^{n} p_i \\ &= H(p) + 1 \ . \end{aligned}$$

From our definition, it is easy to obtain that

$$l_i \geq \log \frac{1}{p_i} \ .$$

This implies,

$$2^{l_i} \geq \frac{1}{p_i} \ ,$$

which gives us

$$2^{-l_i} \leq p_i \ .$$

Since $\sum_{i=1}^{n} p_i = 1$, we have that

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1 \, .$$

On the other hand, by the Kraft inequality in Theorem 2.1.1 a prefix code with the lengths of the code words l_i is realizable precisely when the latter inequality is satisfied.

□

2.3 Shannon's model

In 1948, American engineer and Mathematician Claude Shannon published an article "A Mathematical Theory of Communication" in the *Bell System Technical Journal*. Shannon first proposed a linear model of communication that provides a framework for analyzing the sending and receiving of messages [Sha51], which later is known as the foundation of "information theory".

Shannon said in the article "The fundamental problem of communication is that of reproducing a message sent from one point, either exactly or approximately, to another point", and this is so as the goal of the model. With the help of the mathematical theory of communication Shannon proposed, one could have a better understanding of the mixed up or misinterpreted message in the transmission process.

Shannon proposed a linear model to describe the communication between the message sender to receiver. It includes five important concepts: sender, encoder, channel, decoder and finally the receiver.

Sender is the initial message creator, it can be a person or an object; any information source. The message can be sent through oral words, written sentences, body language or even music.

The next key step in Shannon's model is the *encoder*. The encoder can be a machine or a person which has the ability to convert the initial message into signals which can be easily delivered between sender and receiver. In today's communication system, it is often our computers or telephone which can encode the message with *1-0* binary code or radio waves.

The following step is the *channel* of communication, which gets the message from sender and encoder and passes it to the decoder and receiver. The channel acts as a message medium, and it is where noise occurs. Internet plays the role of channel in email communication, or it could be sound waves and air in face-to-face conversation.

Channel has *noise*, and noise might interrupt the message on the way from sender to receiver. Internal noise and external noise are the two types of noise that a piece

of message might encounter. *Internal noise* happens either when the sender makes a mistake when encoding the message or when the receiver makes a mistake when decoding the message. *External noise* is the noise that are not caused by sender or receiver, it could be white noise in the working environment or other uncontrollable factors that have the potential to cause misunderstanding from the initial message.

The next step in Shannon's model is *decoder*. It plays a role which is the exact opposite of encoder. For example, the message is send with *1-0* binary code or radio waves, then the decoder changes this back into a format that the final receiver can easily understand. An example of an encoder in everyday life is a translator, someone who is tasked with reading information in one language and outputting another.

The final step in Shannon's model is *receiver*, which is the end point of this model. The receiver gets the message or potentially an obscured version of the message, after it moves through the noise.

This model explains the barriers in communication quite well, as it breaks down the whole process into small parts and accounts for the idea of "noise". The model can be applied to many forms of communication, from simple (face-to-face communication) to complex (email exchange).

The key ingredient of Shannon's model is the idea that all streams of symbols in the above description are stationary, or, in other words, are described by probability measures on the corresponding spaces of sequences of symbols that are shift-invariant. In this situation one can talk about their asymptotic entropies (entropy rates).

In this paper, we are most interested in Shannon's explanation of the language model as a stationary sequence of symbols when describing the outbound message of the sender. This is applicable to any stationary sequence, however, since we are interested in application to language models, we will explain only the subject around language.

2.4 Asymptotic entropy

In a stationary model, the language is described by a probability measure P on the space of infinite sequences of letters from a finite alphabet A, and this measure is assumed to be shift invariant ($\equiv$ stationary).

In other words, we have a sequence of A-valued random variables X_i, which is stationary in the sense that the joint distribution $P_{i,d}$ of the random variables

$$X_i^{i+d} = (X_i, X_{i+1}, \ldots, X_{i+d})$$

does not depend on i, i.e., $P(X_i = a_0, \ldots, X_{i+d} = a_d)$ does not depend on i for any finite sequence of letters $a_0, \ldots, a_d \in A$. Such a sequence is called n-gram, where $n = d + 1$ is its length. Therefore, stationarity means that the distribution of n-grams which we see in any "window" of length n positioned over the infinite

time parameterized string of random symbols does not depend on the position of the window.

We define the n-gram entropies of the stationary process (X_n), or, equivalently, of the corresponding shift-invariant measure P as

$$\begin{aligned} H_n &= H(X_1^n) \\ &= -\sum_{a_1,\ldots,a_n} P(X_1 = a_1, \ldots, X_n = a_n) \log P(X_1 = a_1, \ldots, X_n = a_n) \ . \end{aligned} \tag{2.4.1}$$

Proposition 2.4.2. The sequence H_n is subadditive, i.e.,

$$H_{n+m} \leq H_n + H_m \ .$$

Proof. By definition in Formula (2.4.1), we can rewrite H_{n+m},

$$H_{n+m} = H(X_1^{n+m}) = H(X_1^n, X_{n+1}^{n+m}) \ .$$

By Corollary 1.6.5,

$$H(X_1^n, X_{n+1}^{n+m}) \leq H(X_1^n) + H(X_{n+1}^{n+m}) \ .$$

By stationarity,

$$H(X_1^n) + H(X_{n+1}^{n+m}) = H(X_1^n) + H(X_1^m) = H_n + H_m \ .$$

□

According to the general *Fekete lemma* [Fek23], for any non-negative subadditive sequence (a_n) there exists the limit

$$\lim_{n\to\infty} \frac{a_n}{n} = \inf \frac{a_n}{n} \ ,$$

and therefore Proposition 2.4.2 implies that there exists the *entropy rate* (or, *asymptotic entropy*)

$$h = \lim_{n\to\infty} \frac{H_n}{n} = \inf_n \frac{H_n}{n} \ . \tag{2.4.3}$$

Since

$$H_n = H(X_1, \ldots, X_n) \ ,$$

entropy rate can be interpreted as the time density of the average information in a stochastic process.

Simply, we can take entropy rate as the average information per symbol in the entire sequence. For example, if we have a sequence of independent identically distributed (i.i.d) random variables $X_1, X_2, \ldots, X_n$ then the entropy rate is

$$h = \lim_{n\to\infty} \frac{H(X_1, X_2, \ldots, X_n)}{n} = \lim_{n\to\infty} \frac{nH(X_1)}{n} = H(X_1) \ .$$

i.e., it coincides just the entropy of the common distribution of X_i. The reason for this is that in the independent case, the information contained in an n-gram is the sum of the information about each of its n symbols; on the other hand, if the sequence is not independent, then $h < H(X_1)$.

It is rather difficult to estimate the distribution of X_1^n, especially when n is a relatively large number. There is another equivalent definition of the entropy rate which is more appropriate in these situations.

By Definition 1.6.1 and Formula (2.4.1), for $n \geq 1$,

$$H(X_0|X_1^{n-1}) = H(X_0, X_1^{n-1}) - H(X_1^{n-1}) = H_n - H_{n-1} \ .$$

By Proposition 1.6.6, for $n \geq 1$, the sequence of differences $h_n = H_n - H_{n-1}$ is monotonically decreasing and therefore we also have

$$h = \lim_{n\to\infty} h_n \ ,$$

or, in other words,

$$h = \lim_{n\to\infty} H(X_0|X_1^{n-1}) \ . \tag{2.4.4}$$

Alternatively, one can also express h_n as

$$h_n = H(X_n|X_1^{n-1}) = H(X_1^n) - H(X_1^{n-1}) = H_n - H_{n-1} \ . \tag{2.4.5}$$

Here, h_n is the conditional entropy of the next symbol when the preceding $(n-1)$ symbols are known. For $n = 0$, we put

$$h_0 = \log(|\Sigma|) \ , \tag{2.4.6}$$

where Σ is the alphabet.

In other words, h_n naturally quantifies the difficulty in guessing a letter when the preceding $(n-1)$ letters are known. As n increases, h_n involves bigger ranged statistics.

Remark 7. A *unilateral* stationary sequence $(X_0, X_1, \ldots)$ can always be considered as a restriction of a *bilateral* stationary sequence $(\ldots, X_{-1}, X_0, X_1, \ldots)$ to non-negative times. Since

$$H(X_{-n}^{-1}) = H(X_1^n)$$

as both represent the entropies of n-gram distributions along a stationary sequence, the entropy rate of the time reversal of a stationary sequence coincides with the entropy rate of the original sequence. Therefore, in the same way as in Formula (2.4.4), we also have

$$h = \lim_{n\to\infty} H(X_0|X_{-n}^{-1}) = \lim_{n\to\infty} H(X_n|X_0^{n-1}) \ ,$$

so that the asymptotic entropy is the limit of the entropy of predicting the symbol X_n based on the knowledge of the previous symbols $X_0, \ldots, X_{n-1}$.

2.5 Shannon-McMillan-Breiman theorem

This theorem consists of three parts, it is also known as the ergodic theorem of information theory. In terms of stationary ergodic Markov sources, SHANNON is the first one to develop the result for convergence in probability. MCMILLAN proved that L^1 convergence for stationary ergodic source. Last but not least, BREIMAN proved that there is convergences for stationary and ergodic sources almost everywhere.

In the context of estimating entropy, with the Shannon-McMillan-Breiman Theorem, the statistical approach could be a bit simpler in application. Instead of looking at all n-grams, one can just look at the frequency of a particular n-gram. The concept is the following; take a random book and open it at a random page, take a string of n symbols (let's assume $n = 10$ in this example), then you look at all 10-gram of the books that are available and find the frequency of the particular 10-gram that you have selected. Then take

$$\frac{\log \text{frequency}}{10} \ .$$

It is not the entropy, but based on the theorem, this will be close to the entropy asymptotically either in probability or in the average or almost everywhere.

Chapter 3

Shannon's approach to natural languages

3.1 Complexity of language

To measure the complexity of a sequence of symbols from an alphabet (set of letters), one can use a number of options. The constant sequences have the lowest possible complexity. The more variety a sequence shows, the more complex it is.

Consider a binary sequence of length n, i.e. the alphabet consists of two symbols denoted for simplicity; 0 and 1. The total number of possible sequences is then the product of n 2s, i.e. 2^n. There are a lot of different sequences, each of them must have its own description different from the descriptions of other sequence - in total you need 2^n descriptions.

We want to study English texts in the same manner. The extended English alphabet consists of 26 characters and a space if capitals are disregarded. Then the possible total number of sequences of length n is 27^n, which is, if we put it into binary terms with base of 2,

$$2^{\log_2 27^n} = 2^{n \log_2 27} = 2^{nh_0} \approx 2^{4.75n} ,$$

where

$$h_0 = \log_2 27 \approx 4.75 .$$

However, not all combinations of characters in English are admissible. For example, although there is a word that contains *za*, there is no word that contains *zb* or *zc*. Therefore the number of admissible sequences (i.e., the ones that can really occur in the language) has to be a lot smaller than 27^n. If we continue the study with the base 2, then we can assume that there is constant $\widetilde{h} < h_0$ such that the total number of possibilities is approximately $2^{\widetilde{h}n}$. Then, we are interested in estimating the precise value of $\widetilde{h}$.

More rigorously, let W_n be the set of all meaningful (admissible) sequences of length n, then the *growth rate (or topological entropy)* of the language is

$$\widetilde{h} = \lim_{n\to\infty} \frac{\log W_n}{n} .$$

If we consider a probability distribution, it may so happen that a large part of the probability mass is concentrated on a small set of states. For example, the 2-gram "te" happens a lot more often than the 2-gram "za" in English language.

Returning to the setup of the previous chapter, by *Shannon's equidistribution theorem*, if we fix $\varepsilon > 0$, and look at the minimal size $S_{\varepsilon,n}$ of the set of n-grams which has probability $> 1-\varepsilon$, then

$$\frac{(\log S_{\varepsilon,n})}{n} \to h ,$$

where

$$h = \lim_{n\to\infty} \frac{H(X_1^n)}{n} \tag{3.1.1}$$

is Shannon's entropy rate. Since

$$H(X_1^n) \le \log W_n \le h_0 n ,$$

we have

$$h \le \widetilde{h} < h_0 \approx 4.75 .$$

Taking a large piece of text written in a certain language and calculating the empirical frequencies of individual letters (and, more generally, of n-grams) would yield the probabilities needed for the calculation of the n-gram entropies $H(X_1^n)$, see Formula (2.4.1). It is worth noting that the writing style of the text would affect these numbers to a certain degree. Yaglom mentioned that scientific books often have some deviations on frequencies of individual letters due to the presence of special names or foreign words. Deviations in word frequencies can also be easily observed in poetry and some refined fiction work. The novel *Gadsby* published in 1939 by American writer Ernest Vincent Wright is one of the extreme examples, it does not contain any instances of the letter "E" which is a letter that is rather common in English text [YY83].

Therefore, one should not really talk about the entropy of English as the variety of styles or even of individual authors is reflected in the potential variation of the entropy. As a result, entropy estimates can be used in many applications, such as identification of the authorship of texts [Khm00], [KT01], [Kje94], [KPK02].

Lastly, the idea of Kolmogorov consists in defining the complexity of an *individual* sequence as the length of its shortest description [LV$^+$08]. The more "constant" the sequence, the simpler the description. For example, in a sequence of length 100, the sequence containing 100 0s is a lot simpler to describe than a random sequence of

0s and 1s. The concept of entropy on the other hand, is statistical and characterizes a distribution of sequences rather than an individual one.

From Theorem 2.2.1, we can see that the entropy gives a theoretical bound of the minimum number of bits for the original sequence. The difference between the average length of the translated code and entropy of the source is commonly referred to as *redundancy*. Distributions in natural language are often not optimal. When one is trying to deliver a message, to make sure it is understandable, words are often repeated to emphasize the key points. We could use redundancy to explain the reason that the calculated entropy and theoretical entropy are not always the same. Therefore, one can also see entropy as a measure of redundancy.

Here is another example to explain the fact that language has redundancy. You are trying to remember a poem for a presentation, all you need are just a few key words for reference, and you will be able to fill in the blanks and reproduce the whole text. In this case, the key words on your cheat sheet are called *reduced text*. We could see the reduced text as an encoded form of the original. Putting the original text through a reversible transducer will provide a reduced text for communication, and both texts contain the same amount of information. We will further explore this topic in the following sections.

3.2 Shannon's entropy estimates

The idea of entropy rate was introduced by Shannon in his 1948 paper [Sha48] as a key measure in studying communications. This foundational work mainly focuses on the theoretical aspects of the theory and entropy is used as a tool for maximizing the rate of actual transmission between encoding and decoding.

In the 1951 publication *Prediction and Entropy of Printed English* [Sha51], Shannon returns to the idea that the entropy as statistical measure evaluates the amount of information that is contained on average in an individual letter of an English text by conducting more experiments. His study provides three different approaches and the ultimate conclusion is that the entropy rate of English is approximately 1.3 bits per character.

3.2.1 Frequency tables

One way Shannon used to estimate the entropy is through using the frequency tables following Formula (3.1.1), which is the Definition (2.4.3) of asymptotic entropy from Section 4.2. This definition requires knowing the entropies

$$H_n = H(X_1^n)$$

of n-grams, and, ultimately, the n-gram distributions. Shannon based his calculation on the frequency tables from Fletcher Pratt's book *Secret and Urgent. The story*

of Codes and Ciphers [Pra39], which contains the frequency tables of letters, digrams and trigrams. For example, below is the table for $n = 1$.

Letter	Empirical frequency	Letter	Empirical frequency
_	0.182	m	0.021
e	0.107	u	0.020
t	0.086	g	0.016
a	0.067	y	0.016
o	0.065	p	0.016
n	0.058	w	0.013
r	0.056	b	0.012
i	0.052	v	0.007
s	0.050	k	0.003
h	0.043	x	0.001
d	0.031	j	0.001
l	0.028	q	0.001
f	0.024	z	0.001
c	0.023		

Table 3.1: Frequency distribution in the extended English alphabet

By using these tables, Shannon was able to calculate the *entropy increment* h_n using the following formula,

$$h_n = H(X_n|X_1^{n-1}) = H_n - H_{n-1}$$

for $1 \leq n \leq 3$. He obtained the following results.

h_0	h_1	h_2	h_3
4.75	4.03	3.32	3.10

Table 3.2: h_n

He also did the calculation for the 26 letter alphabet (the space is discarded), which gave the following values:

h'_0	h'_1	h'_2	h'_3
4.7	4.14	3.56	3.3

Table 3.3: h'_n

Comparing the two tables above, we can see that $h_n < h'_n$, see Section 1.5.4 for a theoretical discussion of this inequality. As for the numerical values, in what concerns h_1 and h'_1, for $t = p(\_) = 0.182$ and $H' = h'_1 = 4.14$, for a predicted value $\hat{H}$, Formula (1.5.5) yields,

$$\begin{aligned}\hat{H} &= \psi(0.182) + (1 - 0.182) \times H' \\ &= -0.182 \log_2 0.182 - 0.818 \log_2 0.818 + 0.818 \times 4.14 = 4.07 \ .\end{aligned}$$

which is quite close to Shannon's value $H = 4.14$, the discrepancy possibly due to rounding errors.

3.2.2 Word rank

The frequency tables were not available at the time for $n > 3$, and therefore Shannon could not find the entropies for bigger n using the same method.

The second approach Shannon mentions in his work is to estimate the English language entropy by involving the notion of the word rank. In the same way as we talked about theoretical frequencies of letters, digrams and trigrams, one can also talk about frequencies of words. They form a probability distribution on the set of all words, therefore, we can also refer to them as probabilities. The ordered sequence of frequencies $f_1 \geq f_2 \geq \dots$ is usually referred to as the rank distribution, so that r is the rank of the word whose frequency f_r is the r-th entry of the above sequence.

We say that the rank distribution obeys the power law of degree $\eta > 1$ if

$$f_r = a r^{-\eta} \ , \tag{3.2.1}$$

where a is a normalizing constant (the inverse sum of the series).

In 1949, Zipf noticed that there are a lot of rank distributions which satisfy Formula (3.2.1) with $\eta = 1$ [Zip65]. The harmonic series arising for $\eta = 1$ is divergent, and therefore in this situation the total number N of frequencies has to be finite, and satisfy the relation

$$a \sum_{r=1}^{N} \frac{1}{r} = 1 \ ,$$

or

$$\frac{1}{a} = \log N + \gamma \ , \tag{3.2.2}$$

where $\gamma \approx 0.5772$ is the *Euler-Mascheroni constant*. This distribution (nowadays called *Zipf's Law*) is quite universal and appears in numerous situations of varying nature. For example, the distribution of cities by population, the distribution of hockey players by goals and the distribution of citizens by salary. Of course, this law is approximate, and real life distributions deviate from it, especially for the highest ranks and at the tail of the distribution.

For instance, "the" is the word that occurs the most often with the frequency 0.071 and "of" is the second most frequent word with the frequency 0.034 in English [Dew50]. Zipf suggested $a = 0.1$ to be a reasonable value for describing the rank distribution of the English vocabulary, which yields,

$$f_r = \frac{0.1}{r} \ . \tag{3.2.3}$$

Examples of Rank Distribution

In order to verify the simulation model, I considered the enhanced Shakespearean corpus from *The Folger Shakespeare Library* in my own experiment [Sha20]. I used *Text4* generated from the collection of Shakespearean work which is later described in Table 4.1. There are a total of 979,576 words in this sample, and I graphed the first 600 most frequent words from my result in Figure 3.1 and Figure 3.2. We plot the relative frequency f_r against the rank r in Figure 3.1, and having double-logarithmic axes with $\log(f_r)$ and $\log(r)$ in Figure 3.2. The red lines in these graphs are the theoretical projection from Formula (3.2.3), and they appear to be a good fit for our sample.

Figure 3.3 shows Dewey's result [Dew50] used in Shannon's estimation [Sha51]. In DEWEY's work, 100,000 words were manually collected from 38 sources that were representative of English as written and spoken in his time, no single source contributed more than 5,000 words. Dewey provided a total of four tables for the analysis on "words", and Shannon used *Table 3* from Dewey. Dewey discovered 10,119 different words from this selected corpus, and recorded the 1,027 most common words (any that occurred more than 10 times) in this table. These listed words took up 78,633 words in his sample. Dewey also extended his research with consideration of all variant forms of the common words where he grouped together all common variants of one root, and the result was recorded in *Table 4*. Table 4 recorded a total of 1,131 common words (any that appeared more than 10 times), and these listed words add up is 87,358 words. The first six words are the same in both Table 3 and Table 4, but discrepancy appear afterwards. Hence, the estimated entropy rate would be slightly different if Shannon used Table 4 instead of Table 3 in his calculation.

The top ten most frequent words from my experiment are "the", "and", "i", "to", "of", "a", "you", "my", "that" and "in". Although different samples were used, we notice that five of the top ten most frequent words indicated by Shannon, "the", "of", "and", "to" and "i" are also presented in my Shakespearean sample but with different ranks.

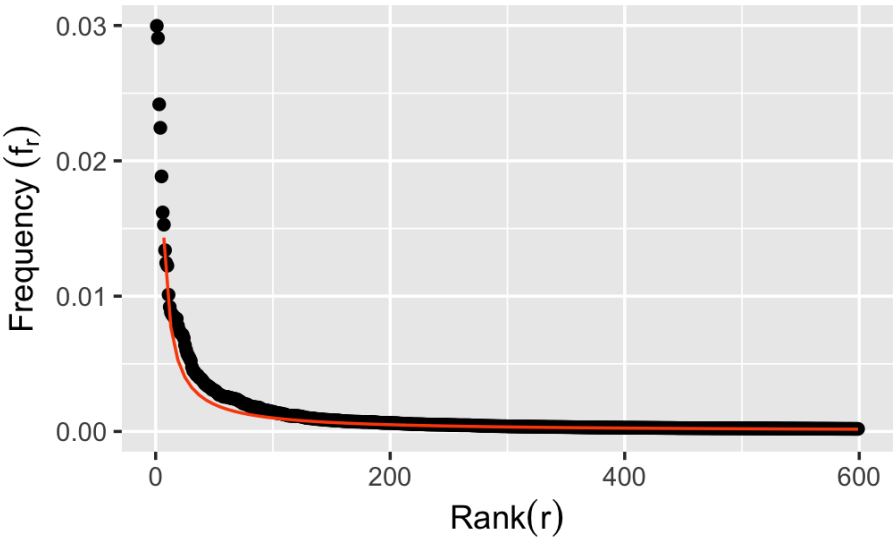

Figure 3.1: Rank frequency graph

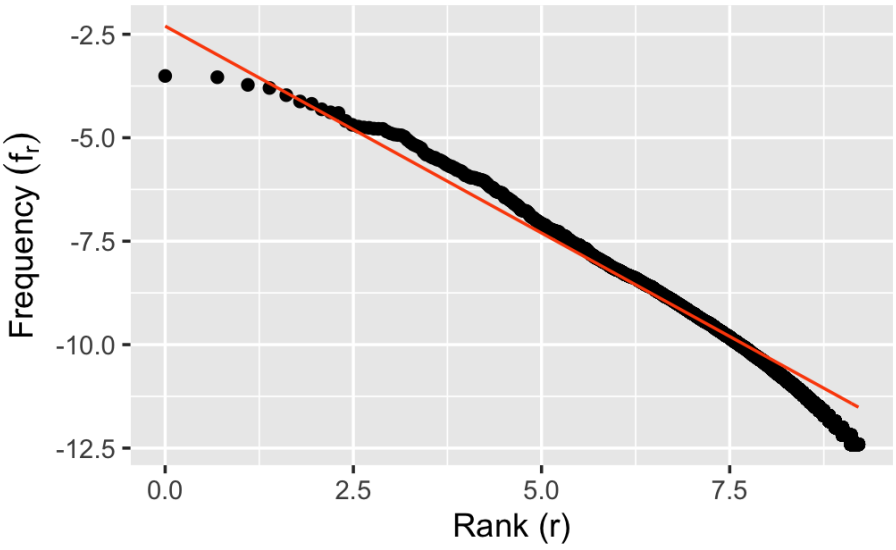

Figure 3.2: Rank frequency graph (logarithmic scale)

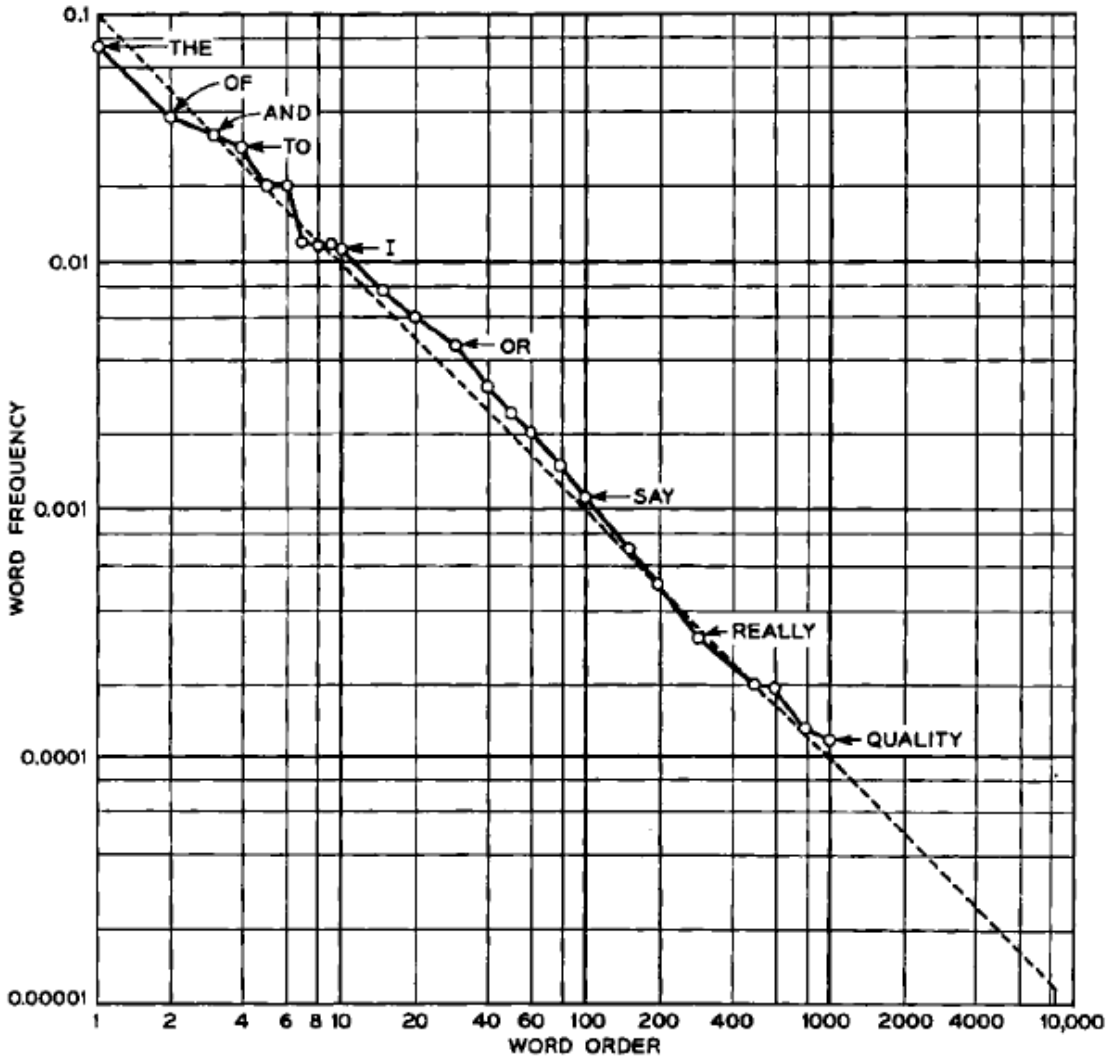

Figure 3.3: Shannon's rank frequency graph

Application to entropy

Applying $a = 0.1$ and $\gamma \approx 0.5772$ to Formula (3.2.2), $N = e^{-\gamma} \approx 12367$ for the total number of words. However, Shannon uses a different value of $N = 8727$ in his calculation without specifying its origin. According to Brillouin [Bri13], Shannon took the table frequency values for the 100 most frequent words, and then extrapolated the rest by Formula (3.2.3), which gave,

$$H_{\text{word}} = -\sum_{1}^{8727} f_r \log_2 f_r = 11.82 \text{ bits per word} .$$

Shannon took the average word length L_{word} to be 4.5 letters, and we denote h'_w to be the entropy rate estimated from word frequency for English with 26 letters. This leads to,

$$h'_w = \frac{H_{\text{word}}}{L_{\text{word}}} = \frac{11.82}{4.5} = 2.62 \text{ bits per symbol} . \tag{3.2.4}$$

The average word length is 4.5 letters, therefore the average length of the corresponding blocks (ending with a space) is $4.5 + 1 = 5.5$. We denote the entropy rate for English with extended alphabet estimated from word frequency to be h_w, when the sample is relatively large, for a rough estimation,

$$h'_w = \frac{4.5}{5.5} h_w .$$

The estimate in equation (3.2.4) is not rigorous mathematically, but rather refers to a "physical" or "engineering" way of thinking. However, the resulting value is consistent with the estimates obtained by other methods.

Below are the augmented tables obtained by adding the above values of h_w and h'_w to the Table 3.2 and Table 3.3.

h'_0	h'_1	h'_2	h'_3	h'_w
4.70	4.14	3.56	3.30	≈ 2.62

Table 3.4: h'_n for the standard English alphabet

h_0	h_1	h_2	h_3	h_w
4.76	4.03	3.32	3.10	≈ 2.14

Table 3.5: h_n for the extended English alphabet

Remark 8. The average word length of 4.5 used by Shannon is the one traditionally used for calculations in the book publishing industry. There are studies that consider the average word length dynamics as an indicator of cultural changes in society. For instance, it is found that the average word length slightly increased in the nineteenth century, then it grew rapidly over most of the twentieth century and started a decreasing trend around the end of the twentieth and the beginning of the twenty-first century [BSS15], which means that shorter words have become more frequent with time. Most of these studies give the values of the average word length in English to be between 4.5 and 5.5.

3.3 Shannon's cognitive experiment

As we have seen above, the scope of the objective method based on empirical frequencies is quite limited due to the sheer volume of the required data. Because of that, Shannon came up with another "subjective" approach based on an interaction with a human subject, who is fluent in the language and familiar with its words, cliches, idioms and grammar to be able to fill in the missing letters while proof-reading. The passages in the experiment are selected randomly and are unfamiliar to the subject.

3.3.1 Identical twins thought experiment I

This experiment is based on the assumption that one can have an identical copy (clone in today's language, or twin in the original language of Shannon) of the subject. In this envisioned communication system, one of these copies (A) serves as the encoder at one end and the other one (B) serves as the decoder at the other end.

A starts guessing the first symbol (from the extended English alphabet) in the selected passage and only one guess is allowed at every instance. If the guess of the first letter is correct, then the subject will be informed and proceed to the second letter; if A made an incorrect guess, the correct letter of this spot will be told and the subject can proceed to the next letter. As the experiment goes, A always knows the entire $(n-1)$ block of text when guessing the n-th letter.

The following example is from Shannon's description, the first line is the original text, the second line contains dashes for the letters that have been correctly guessed and the correct letters at the positions where an incorrect guess was made.

Example 3.3.1.

```
(1) THE ROOM WAS NOT VERY LIGHT A SMALL OBLONG
(2) ----ROO------NOT-V-----I------SM----OBL----
(1) READING LAMP ON THE DESK SHED GLOW ON
(2) REA----------O------D----SHED-GLO--O--
(1) POLISHED WOOD BUT LESS ON THE SHABBY RED CARPET
(2) P-L-S-----O---BU--L-S--O------SH-----RE--C------
```

The errors often occur either at the beginning of the words or where it might have various appropriate subsequent letters. The second line is referred to *reduced text*, however it does not contain much less information comparing to the first line, since one can possibly recover the first line with it.

Indeed, as B is able to process information exactly the same way as A by our assumption, this gives him the ability to make correct guesses precisely at all of the "-" spots in line `(2)` in Example 3.3.1. B would also make incorrect guesses at the spots A made mistakes. Therefore, if one provide the reduced text from A's attempts to B, then B would be able to recover the original text in `(1)`. Thus, we can conclude that both the original and reduced text contains same information.

3.3.2 Identical twins thought experiment II

The experiment in Section 3.3.1 illustrates the idea that the language is indeed not optimal, so that one can compress a message and make it much shorter in the communication system. However, it is still not clear how one can quantify this experiment in order to produce actual numerical values. Because of that, Shannon suggested a modification of the experiment in Section 3.3.1, which we describe below.

Just like the thought experiment in the last section, A is given a string of symbols, and is asked to guess the sequence one symbol at a time. A will not be informed of the correct symbol until he guess it himself, if the guess is wrong, one has to try again until the right symbol appears. Then the total number of guesses at each instance is recorded. As an illustration, below are the results of a sample experiment from Shannon's publication.

Example 3.3.2.

(1) THERE IS NO R E V ERSE ON A MOTORCYCLE A
(2) 11151 1 21 1 21 1 151171112 1 32 1 2 2 7111141111 1 3 1
(1) FRIEND OF MINE FOUND THIS OUT
(2) 861311 1 11 1 1111 1 62111 1 1121 1 111 1
(1) RATHER D RAMATICALLY THE OTHER DAY
(2) 411111 1 11 51111111111 1 611 1 11111 1 111 1

The guessing process briefly goes as the following:

- At the first trial ($n = 1$), A is given no information (block length is $n - 1 = 0$). She needs to keep trying until she gets "t" in the first position, then the number of attempts to arrive at "t" will be recorded.
- At the second dash ($n = 2$), A could use the information received from the first dash - "t" (block $n - 1 = 1$) in this example. As the design in the experiment, A does not know where this phrase is located (beginning of the sentence or just somewhere in a word), so "t" can be followed by a space or any possible letters. The total number of guesses A uses to arrive at "th" is recorded.
- Knowing "th", A is expected to find "the".
- .....

Out of 102 symbols, a total of 79 letters were guessed correctly at the first trial and 90/102 symbols (88.2%) were guessed with no more than three guesses. This prediction shows that the selected subject is familiar with the ordinary English language. A change in the literary genres of the string might result in a poorer result, for example, poetic language, scientific books or newspaper writing are harder to make prediction in general.

In this thought experiment, the second line can be seen as the encoded version (or the reduced text) of the first line. A guesses the letters in the decreasing order of the conditional probabilities of the n-th symbol based on the given $(n-1)$ block of text. For each instance, the minimum number of guesses is 1 and the maximum is 27 since there are 27 symbols in the extended alphabet. Hence, a "1" in the encoded string means that the letter has the highest probability that could occur at this spot given the $(n-1)$ proceeding block, and in the same way, 27 means that it is the least probable letter to occur at this place. Therefore, A maps the sequence of letters into a string of numbers $1, 2, \ldots, 27$ through this encoding (guessing) process.

Such a translation largely simplifies the statistical structure of the text. The redundancy due to the complex constraints on admissible combinations of letters is reflected in the rather unequal probabilities of the new symbols $(1, \ldots, 27)$ through the experiment process.

When recovering the original text using the string of numbers, B can map the numbers in the reduced text to letters since A and B share an identical thinking process and, in particular, identical conditional probabilities of letters. For example, the corresponding numbers of "`FRIEND OF MINE...`" in `(1)` is "`86131111111111...`" in `(2)`. When B is processing the given preceding block "`THERE IS NO REVERSE ON A MOTORCYCLE A `", the letter "`F`" will also be B's 8th guess at this position just like A. Therefore, B will map "`F`" to the number "`8`" and give out "`F`" in the recovered sequence. The process is repeated until the entire 102 symbols have been reproduced. Therefore, similar to Example 3.3.1, the second line contains the same information as in the first line.

3.3.3 Shannon's practicable experiment

As the identical twin experiment is impossible to realize. Shannon suggested yet another technique for estimating entropy based on its definition as the limit

$$h = \lim_{n\to\infty} h_n = \lim_{n\to\infty} \left(H_n - H_{n-1}\right)$$

of conditional entropies. Its design is established on the twin experiment described in the Section 3.3.2. In this experiment, a total of 100 strings from the biography of the US President Thomas Jefferson written by historian Dumas Malone *Jefferson the Virginian* was randomly selected. Shannon's spouse Mary was the only human *subject* in this cognitive experiment.

For each string, Mary was asked to predict the next symbol (n-th symbol) given a sequence of length $(n-1)$, she keeps guessing until the right symbol appears. Shannon repeated this step for each of the 100 strings with $n = 1, 2, \ldots, 15$. Then this process was repeated for $n = 100$. Therefore he obtained a total of 16 observations for each string and 1,600 observations in total for the whole experiment. For each length n,

100 observations were used in the calculation when finding the probabilities which is a reasonable sample size for model validation.

For example, Mary is given a sequence of dashes, each representing either one of the 26 English letters or a space. She has to guess the following symbols of the targeted sequence based on the preceding block. The number of guesses to reach the correct symbol at each dash is recorded.

Example 3.3.3. The subject is given:

"----------------" ,

where the sequence of dashes are viewed as sequences of characters, therefore Mary is asked to guess one dash at a time. She is expected to arrive at the following sentence after many trials:

"there is no reve" ,

which is the first 15 letters of the phrase "there is no reverse on a motorcycle".

Let q_i^n denote the empirical probability that the subject requires i guesses to locate the correct symbol at the n-th position following a block of length $(n-1)$ symbols. Obviously, i does not exceed k which is the total number of letters in the alphabet.

For example, throughout the experiment, the total number of guessing the correct letter at position one with exactly two trials is 10 out of 100 observations, then the empirical probability of a subject who can find the correct X_1 at the second trial is $\frac{10}{100} = 10\%$, then $q_2^1 = 0.1$.

Shannon proved the following inequalities linking the conditional entropies h_n with the empirical probabilities q_i^n in formula (17) from *Prediction and Entropy of Printed English* [Sha51]. One can estimate h_n,

$$\underline{h_n} = \sum_{i=1}^{k} i(q_i^n - q_{i+1}^n) \log i \leq h_n \leq -\sum_{i=1}^{k} q_i^n \log q_i^n = \overline{h_n} \, . \tag{3.3.1}$$

Shannon obtains the following results from his experiment with Mary:

n	1	2	3	4	5	6	7	8
$\overline{h_n}$	4.03	3.42	3.0	2.6	2.7	2.2	2.8	1.8
$\underline{h_n}$	3.19	2.50	2.1	1.7	1.7	1.3	1.8	1.0
n	9	10	11	12	13	14	15	100
$\overline{h_n}$	1.9	2.1	2.2	2.3	2.1	1.7	2.1	1.3
$\underline{h_n}$	1.0	1.0	1.3	1.3	1.2	0.9	1.2	0.6

Table 3.6: Upper and lower bounds for h_n

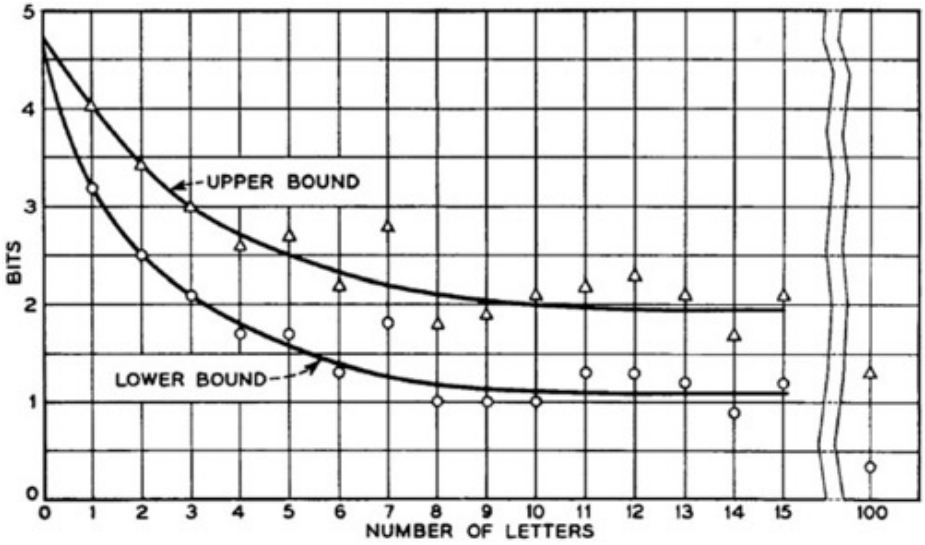

Figure 3.4: Shannon's experiment results

As an engineer, Shannon made his prediction based on Figure 3.4, using the upper bounds and lower bounds he obtained from the experiment. He used the 16 entries from his calculations and observed a decreasing trend in both the upper and lower bounds as n gets bigger. He concluded that there exists a long range statistical effect which would bring down the entropies h_n consistent with the theoretical prediction. An upper bound of 1.3 was found for $n = 100$, which he took as an estimate of the limit entropy rate. It can be interpreted as saying that there are $2^{1.3} = 2.46$ possible successive letters on average following a character [TI21].

There is limitation to Shannon's model in his original objective experiment. We still have to assume certain homogeneity of the text. More explicitly, if we look at text with different nature, the entropy may be different, See Section 3.1.

3.3.4 Reversed experiment

Similar to the original experiment, Shannon also conducted a "reversed" version of the experiment where the subject is required to guess the letter preceding what is already given. For instance, using Example 3.3.3, in the reversed experiment, one is given "`o reve`", the subject needs to give out "`n`" as a correct response. This task seems to add difficulty to the experiment as it likely contrasts the way human brain prefers to work. Though Shannon indicates that the result is not dramatically different to the original experiment. Using two 101 letter samples, one is able to find the correct letter within two guesses in the forward direction is 80 times ($p = 80/101$) and in the backward direction is 73 times ($p = 73/101$). In the stationary model, Formula (2.4.5) states that the forward entropy rate coincides with the backward entropy rate.

No. of guess(es)	1	2	3	4	5	6	7	8	> 8
Forward	70	10	7	2	2	3	3	3	4
Reverse	66	7	4	4	6	2	1	2	9

Table 3.7: Shannon's reversed experiment

Chapter 4

Further developments

After the foundational paper Shannon published in 1950, there were further experiments using his objective and subjective methods in order to obtain a more precise estimate of the entropy rate h, while some proposed new approaches. As far as the objective methods are concerned, in Shannon's time, the computational equipment for finding the frequencies of n-grams in a given text was not available. Therefore, Shannon conducted a subjective experiment and used the resulting data for estimating the entropy rate of English text. Nowadays, with more powerful computers, estimating the entropies using n-gram frequencies is achievable.

4.1 Same methodology

4.1.1 Objective method

Guerrero uses twenty-one classic literary works from the list "100 novels everyone should read" in the *Daily Telegraph British newspaper* [Gue09]. His work uses direct computer calculations and evaluates n-gram frequencies for n up to 500. Then he finds the entropies and looks at their differences with Formula (2.4.5). Guerrero finds an entropy rate $h = 1.58$ based on his analysis using around 20.3 million printable characters in English text.

The twenty-one literary works Guerrero used all have different lengths (number of total characters). According to his finding, the entropy value for n-gram H_n decreases as the size of the selected sample decreases for $n \geq 6$. Moreover, since the values of h_n are received through differences (Formula 2.4.5), therefore negative values are found for h_n, and this contradicts with our base model which asserts that the sequence of n-gram differences (h_n) is monotone. Guerrero found the block length where the function of h_n crosses zero and denoted with N_z, which is the point that the conditional entropy becomes zero as it is the root of the continuous function h_n. Instead of $\lim_{n\to\infty} h_n$, $\lim_{n\to N_z} h_n$ is used in his work, and this is the reason that the

n in his conclusion that the maximal entropy rate occurs when $8.31 \leq n \leq 11.43$ is not a whole number.

Another objective method is described in JUHLIN's thesis [Juh17]. In this work, a total of 154 English text of different nature, novel or twitter, was used as a collection of the data sample. For standard English, non-filtered English text contain 72 symbols as the filtered one has 27 symbols. For Twitter English, it contains the same 27 filtered symbols and 68 for the non-filtered. As the amount of various combinations of n-grams is huge, Juhlin was only able to make estimation up to $n = 6$ for nonfiltered data and $n = 8$ for filtered data. The entropy rates are estimated to be in the range from 2.71 to 2.99. It is easy to notice that the values for non-filtered data are higher than the filtered data. The character limit for a tweet is 140 symbols as of 2017 which means its language is more concise compared to what is in the novel. From another perspective, abbreviations are common in twitter language. Therefore, he observes a greater entropy rate for Twitter English than standard English.

There are several major issues with this type of calculation. First, the amount of distinct n-grams is huge as the number of all possible combination of n-gram grows exponentially which requires a lot of time and storage in the estimating process. The second problem is that a lot of admissible n-grams only appear a few times or not at all in the selected content, which leads to the issue that the chosen text is not a representative sample, and the code becomes inefficient with big ns. The third issue comes in when taking into account the fact that the computers are able to run program on a big text. The length of the selected big text is still fixed, the longest n-gram is the length of the text. The n-gram entropy H_n is bounded by the logarithm of the text length, this is universal. This shows dependency on the length of the text, the entropy rate generated in this calculation become some artefacts about maths. Perhaps one improvement in Guerrero's experiment could be selecting a fixed number of chapters in these books to make sure the size of total n-gram is relatively close to one another. Therefore, other methods can also be considered.

4.1.2 Subjective method

In 1954, BURTON and LICKLIDER modeled Shannon's experiment to further explore the dependency between entropy and length of the preceding block known to the subject [BL55]. Each of the ten participants was given a different book. Following Shannon's design, $n - 1 = 0, 1, 2, 4, 8, 16, 32, 64, 128$ and $n - 1 \approx 10000$ were used. The experiment was conducted as instructed in Shannon's original design where individuals are given blocks of text in these lengths and are asked to guess the next letter. Burton and Licklider focused their analysis more on the redundancy perspective of the study. They define *relative redundancy* to be

$$1 - \frac{\text{actual amount of information per letter of the sequence}}{\text{the amount of information per letter if letters were independent}} . \qquad (4.1.1)$$

In other words, the denominator in formula (4.1.1) is just H_1. They found that the estimate of relative redundancy increases as n gets bigger in the range from 0 to 32. The upper and lower bounds for $n - 1 = 64$, 128 and 10,000 are not much different from the bounds for $n - 1 = 32$. Their finding agrees with Shannon's conclusion when n is relatively small ($n \leq 15$) that the long range statistical effects reduces the entropy of English text. However, since Shannon made his conclusion only using $n = 1$ to $n = 15$ and $n = 100$, Burton and Licklider used a finer breakdown in their experiment and concluded that for $n > 33$ there is no noticeable change in redundancy as n increases.

In 1976, Cover and King approached the problem of estimating the entropy of printed English with a different subjective method. A total of 12 people participated in this study, and the given text is the same as in Shannon's original work, *Jefferson the Virginian*. The experiment was designed using a gambling approach by having the subjects place sequential bets on the next symbol of text [CK78]. The underlying probability distribution for the process gives guidelines to each subject to place bets with the amount of money to be determined by the subject at each position. With a total of 900 samples, Cover and King came to the conclusion that the English text has an entropy of approximately 1.3 bits per symbol, which is well in line with Shannon's estimate.

In 1998, Moradi *et al.* conducted Shannon's experiment under a similar setting. Two books were included in their first experiment, one subject was used for each book. In total, 100 phrases of length $n = 64$ were randomly selected from each book. For subject 1, a romance novel written by Judith Krantz *Scruples II* was used and an engineering book, *Digital Signal Processing* by William D. Stanley was used for subject 2. A result of $h = 2.05$ bits per character was found at length $n = 32$, and it does not vary by much with greater n values.

Later, they conducted their second experiment with a different setting, four different books were used with eight participants, four males and four females. 100 samples of length 32 were selected from each book, and the participants were asked to guess the 32nd symbol based on the first 31 characters. The total number of guesses was recorded. At length $n = 32$, it was calculated that h_n were ranged from 1.62 to 3.00 depending on the books and individuals [MGBR98].

With today's computational development, one is able to conduct experiment with a much bigger sample size. In 2019, Ren, Takahashi and Tanaka-Ishii conducted this experiment with the help of a crowd-sourcing service *Amazon's Mechanical Turk*. This service makes gathering a large number of participants in a short time possible within reasonable budget [RTTI19]. This experiment focused on estimating the entropy rate in English. 225 sentences were randomly selected from the *Wall Street Journal* for this experiment. A total of 683 subjects participated in this survey. They used a bootstrap technique on these samples and obtained an estimate of $h \approx 1.22$, which is fairly close to Shannon's result.

4.2 Different methodology

4.2.1 Data compression and optimal encoding

It is easy to notice that when we are compressing text files of comparable length, the zip file for a children's book is likely to be smaller than the one for adult literature. From the wording point of view, the vocabulary for children's book is a lot smaller than adult's book. From the content point of view, the content in children's book is simpler which means that sentence structure is also less complicated in comparison. Therefore, children's literature is more homogeneous at the scale in comparison.

For example, start with *"cat"*, we can get *"catholic"* or *"cat␣"* (␣ is a space). If it is in a children's book, it has a bigger chance which there is just a space following *"cat"*. Hence if we consider the idea of entropy, the amount of information is associated with the model for the data stream, with a sequence of symbols that is easier to predict, the less information it has. Therefore the less information you have, the shorter the length of the compressed file will be.

All natural languages have significant redundancy. Using compression algorithm, one can remove the *redundancy* which converts the original text to the most efficient encoded form in order to reduce length. If the encoding matches with the entropy, then this piece of work is at its optimal and it has no redundancy. Using the length of the compressed data, we can find the entropy rate of original data simply through calculation. As the unit for compressed text is *bit* and the unit for the original text is *letter*:

$$\text{Entropy rate of the text for English} = \frac{\text{length of the compressed text}}{\text{length of the original text}} = \frac{\#\ bit}{\#\ letter} \tag{4.2.1}$$

One of the popular compression algorithm used today is prediction by partial matching (PPM) compression. By using this algorithm, Takahira *el at.* One found an entropy rate of English to be 1.134 bpc in 2016 [TTID16]. With variation of PPM, an entropy rate of 1.46 bpc was found by Teahan and Cleary [TC96] on the same text which was used in Shannon's original human cognitive experiment.

4.2.2 Simulation experiment

Different compression methods

A collection of Shakespeare's plays and poetry from *The Folger Shakespeare Library* [Sha20] was used in my experiment. The original downloaded file is a 2,183,684 bytes ZIP archive file, and it generated 42 TXT files. The TXT files were then merged into one TXT file of 5,513,117 bytes in size, and it is used as my sample in this experiment.

The data processing was done by using a Python 3 code. The original data is stored in *Text1*, which contains various format symbols, for example, "\n" for a new line or \r" for carriage return, punctuation marks, both upper and lower cases of English alphabet and digits. This original file is cleaned with different criteria and generated 3 texts, *Text2, Text3* and *Text4*, which is summarized in Table 4.1. In Text2, all of the non-alphabetic characters including space were removed and contains only upper and lower cases letters of English alphabet, which making it a string of letters. In Text3, all of the format symbols and punctuation marks are replaced with a space, however multiple spaces in a row are allowed. Finally Text4 is based on Text3, where all upper cases are converted to lower cases and strings of multiple spaces are replaced with just one space. Text4 is a string of words in lower cases, which is in the same format as used by Shannon in his experiment.

File Name	Format	Note
Text1	"\n", "\r", space, punctuation, A-Z, a-z, 0-9	the format was still there, e.g. extra line
Text2	A-Z, a-z	no space, a string of letters
Text3	space, A-Z, a-z, 0-9	many spaces in a row
Text4	space, a-z	string of words

Table 4.1: Data cleaning results

Apply different compression algorithm and compare the arising result

Natural languages have redundancy and we define the compression ratio to be

$$\text{Compression ratio } \rho = \frac{\text{The size of uncompressed file in bytes}}{\text{The size of the compressed file in bytes}} . \tag{4.2.2}$$

We have unit conversion: 8 bits = 1 byte and 1 English symbol = 1 bit in *ASCII* (American Standard Code for Information Interchange) as a character encoding standard for electronic communication. In order to obtain an entropy for English from Formula (4.2.1), we can use

$$\begin{aligned} \text{Entropy Rate of the text} &= \frac{\#\ bit}{\#\ character} \\ &= \frac{\text{Compressed file in \# byte}}{\text{Uncompressed file in \# byte}} \times \frac{\text{8 bits/1 byte}}{\text{1 letter/1 byte}} \\ &= \frac{1}{\rho} \times \frac{\text{8 bits/1 byte}}{\text{1 letter/1 byte}} . \end{aligned} \tag{4.2.3}$$

This yields,

$$\text{Entropy of the text in English} = \frac{8}{\rho}\,. \tag{4.2.4}$$

Altogether, four compression algorithms were used in this experiment, and the results are presented in Table 4.2. The compression algorithms used in the experiment are GZIP, ZIP, BZIP2 and LZMA. The detailed descriptions for these algorithms can be found [Deu96], [Sta01], [Sew96] and [LS13].

Using GZIP as the compression algorithm, the average compression ratio for our sample is 2.56. Using the method of ZIP gives us the same average compression ratio of 2.56. With BZIP2, the average compression ratio for these four samples is 3.32. Finally, with LZMA, the average compression ratio is 3.34. We can observe that Text4 performs the best among the selected sample.

Text4 with the BZIP2 method produces the biggest compression ratio, 3.67, which gives us an entropy rate of $h \approx 2.18$ bits per symbol by using Formula (4.2.4). The value for entropy rate we obtained is not as low as the result using the same method BZIP2 on text from the Bible in an experiment by Behr *et al* [BFMX03] where it is at $h = 1.70$ bits per symbol, however it is relatively close. This discrepancy could have been caused by the difference in language styles of the authors.

Text#	Original	GZIP	ZIP	BZIP2	LZMA
1	5,330,389	2,106,832 (2.53)	2,106,946 (2.53)	1,578,929 (3.38)	1,426,872 (3.74)
2	4,056,616	1,774,062 (2.29)	1,774,176 (2.29)	1,473,320 (2.75)	1,484,863 (2.73)
3	5,169,972	1,954,054 (2.65)	1,954,168 (2.65)	1,490,204 (3.47)	1,537,946 (3.36)
4	5,034,540	1,804,502 (2.79)	1,804,616 (2.79)	1,373,005 (3.67)	1,426,872 (3.53)

Table 4.2: Shakespeare: file sizes (in bits) and the corresponding compression ratios

Dependence of compression ratio on literature style

The following experiment investigates the possible influence of the difference in literature style on compression ratio. Two of the most common algorithms used today,

GZIP and ZIP, were applied onto the original 42 TXT files downloaded files as described in the previous section. We divided all of the 42 files into three types, plays, sonnets and other poetry. Shakespearean sonnets are relatively short, they all have the same format containing fourteen lines each. Therefore all of his 154 sonnets are stored in one single TXT file. We ran this merged file through GZIP and ZIP, and we had one compression rate for each algorithm.

Using GZIP as the compression algorithm, we found a compression ratio of 2.56 for plays, 2.47 for sonnets and 2.21 for poetry. Using ZIP as the compression algorithm, we estimated a compression rate of 2.51 for plays, 2.43 for sonnet and 1.96 for poetry.

According to Dubrow, the Shakespearean poetry can be further divided into narrative poems and allegorical poems [Dub08]. We did another comparison between these two categories. Narrative poetry has a compression rate of 2.37 using GZIP and 2.32 using ZIP. The values are not close to those found for allegorical poetry, which has 1.90 using GZIP and 1.25 using ZIP. As indicated in Table 4.3, there is only one allegorical poetry with size 2,466 bytes, which is very small file in comparison, it is not statistically representative.

Title	Type	Subtype	Original	GZIP	ZIP
The Comedy of Errors	Play	Comedy	93,250	34,893	35,550
a midsummer night's dream	Play	Comedy	100,554	40,568	41,282
The Tempest	Play	Comedy	103,322	42,052	42,673
Two Gentlemen of Verona	Play	Comedy	105,727	40,052	40,737
Pericles, Prince of Tyre	Play	Comedy	113,339	45,294	45,903
Twelfth Night	Play	Comedy	119,215	46,921	47,550
The Merchant of Venice	Play	Comedy	125,304	48,906	49,571
Much Ado About Nothing	Play	Comedy	127,784	48,873	49,538
The Taming of the Shrew	Play	Comedy	130,330	49,793	50,462
As You Like It	Play	Comedy	131,862	50,257	50,890
Measure for Measure	Play	Comedy	132,764	51,231	51,884
Love's Labour's Lost	Play	Comedy	132,828	52,781	53,426
The Merry Wives of Windsor	Play	Comedy	136,592	51,261	51,942
All's Well That Ends Well	Play	Comedy	138,456	54,274	54,988
The Two Noble Kinsmen	Play	Comedy	147,304	59,690	60,351
The Winter's Tale	Play	Comedy	149,062	59,996	60,637
Troilus and Cressida	Play	Comedy	166,143	65,637	66,524
Cymbeline	Play	Comedy	169,072	67,584	68,197
Macbeth	Play	Tragedy	106,538	43,112	43,717
Timon of Athens	Play	Tragedy	115,166	46,048	46,685
Julius Caesar	Play	Tragedy	119,985	46,045	46,674
Titus Andronicus	Play	Tragedy	124,759	48,974	49,615
Romeo and Juliet	Play	Tragedy	146,706	57,960	58,601
Antony and Cleopatra	Play	Tragedy	156,343	61,803	62,689
Othello	Play	Tragedy	159,411	62,311	62,916
King Lear	Play	Tragedy	159,461	64,046	64,659
Coriolanus	Play	Tragedy	171,215	66,908	67,525
Hamlet	Play	Tragedy	182,866	73,088	73,920
King John	Play	Histories	125,575	49,849	50,462
Richard II	Play	Histories	135,081	53,402	54,019
Henry VI, Part I	Play	Histories	136,156	54,235	54,872
Henry IV, Part I	Play	Histories	145,915	58,058	58,695
Henry VIII	Play	Histories	149,163	59,378	59,995
Henry VI, Part III	Play	Histories	152,246	57,762	58,399
Henry V	Play	Histories	157,464	63,110	63,946
Henry VI, Part II	Play	Histories	158,182	62,495	63,132
Henry IV, Part II	Play	Histories	159,858	63,135	63,772
Richard III	Play	Histories	181,588	69,574	70,195
The Phoenix and Turtle	Poetry	Allegorical	2,466	1,300	1,965
Venus and Adonis	Poetry	Narrative	56,790	24,159	24,800
The Rape of Lucrece	Poetry	Narrative	87,921	36,830	37,435
shakespeares-sonnets	Sonnet	Sonnet	99,272	40,122	40,820

Table 4.3: Compression ratios: Shakespearean works

We then divided all of the plays into three categories, comedy, tragedy and history [Jam18]. There is no noticeable difference among these groups, which indicates this feasibility of using relative entropy for authorship attribution. It can be interpreted that the work in this sample collection all belong to the same Shakespearean English style.

Literature	Type	GZIP Compression ratio		ZIP Compression ratio	
Play	Comedy	2.56	2.54	2.52	2.51
	Tragedy	2.53		2.50	
	History	2.54		2.51	
Poetry	Narrative	2.37	2.21	2.32	1.96
	Allegorical	1.90		1.25	
Sonnet	-	2.47		2.43	

Table 4.4: Compression ratios: different genres of literature

We removed the data from allegorical poetry since it is an outlier. Plotting the compression rate against the original file size, we can see from Figure 4.1 and Figure 4.2 that the size of original file does not effect the compression rate like when estimating with n-gram frequencies. Therefore, estimation of entropy through compression rate is relatively reliable.

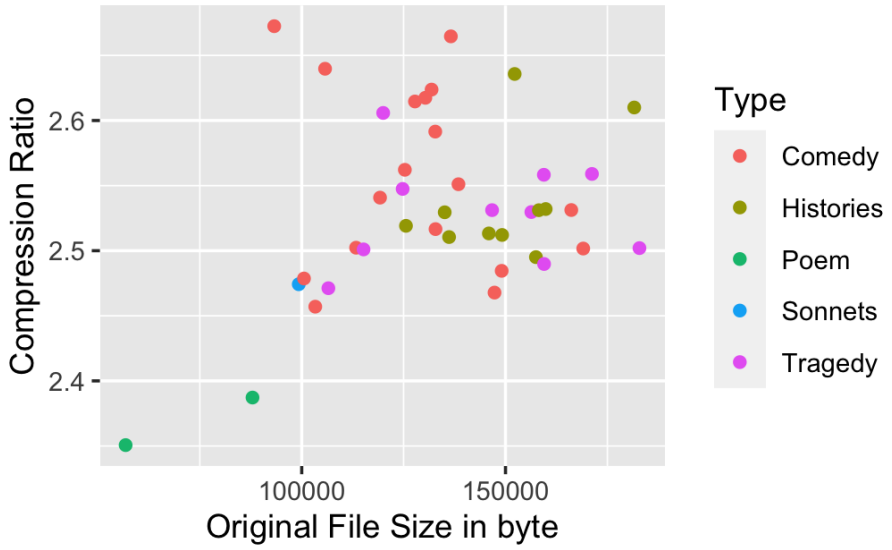

Figure 4.1: Compression ratio for GZIP

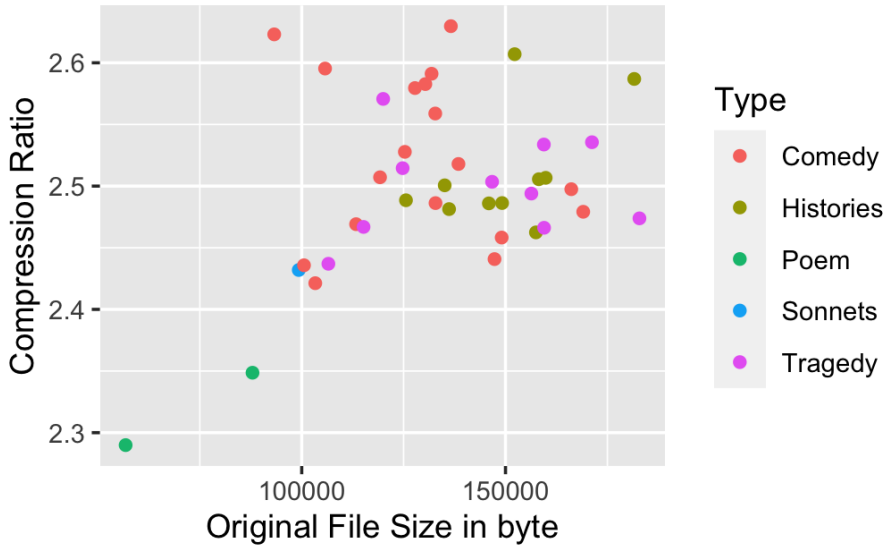

Figure 4.2: Compression ratio for ZIP

4.3 Other languages

As discussed earlier in Remark 8, varying characteristics such as the time dynamics of one language will bring different entropy values. We can also conclude that letter frequencies, word lengths as well as the sizes of the alphabet would vary for different languages, resulting different values of h_n in formula (2.4.5). It can also be speculated that regional variation of the same language would result different values. In this section, we will further investigate the difference in entropy values among various languages.

Yaglom mentioned in 1983 that due to the fact that the average word-length is longer in *English* or *French* compared to *German*, the appearance of "space" exists more often in *German*, resulting a greater probability of "space" in a text with the same length [YY83]. There are unique common combinations in every language. For example, "the" is one of the most frequent combination of symbols in *English*, and so is "sch" in *German*, but these two combinations are not frequently in use in other languages. The rank distribution is different in different languages.

We arrange the symbols in the order of a descending probability, the first symbol being the most frequent. For *English*, we receive a sequence "_ETAONRI...", where "_" stands for the space. For *French*, the corresponding sequence is "_ESANITUR...", and it becomes "_ENISTRAD..." for *German*.

4.3.1 Alphabet sizes across languages

For English language, there are 26 letters plus a space, therefore $k = 27$ in Formula (3.3.1) that appears in our description of Shannon's experiment. The value of k changes depending on the language. Other Latin alphabet based languages have roughly the same number of characters as English, the number varies due to the presence or absence of certain letters and/or diacritic signs.

We consider the case of *hanyu pinyin* for Chinese, the first word "hanyu" means "Chinese language" and the second term "pinyin" means "spelled sound". It is used to describe the *phonemes* (unit of sound) in Mandarin. Pinyin was developed in the 1950s as a phonetic sound system using Latin alphabet which contains various combinations of these 26 letters, its syllables contain 23 initials, 39 finals and 4 diacritics denoting the tones [Liu08]. Each Chinese syllable in spoken mandarin can be spelled as a combination of one/zero initial, one final and/or one diacritic tone mark. Spaces can be inserted between every syllable (which corresponds to one character) or every word. Pinyin is commonly used as an computer input method to enter Chinese characters, diacritic tone marks are not needed in an everyday typing script which yields a value of $k = 27$ in hanyu pinyin.

For writing system using other alphabets, the value of k could be a lot bigger. Now we consider the case of writing Chinese by using characters. Pinyin defines the phonemes, it gives no literal meaning, (like the idea of "omni"). Each Chinese character in the written language consists one pinyin syllable. One syllable can give multiple characters, for example, there are 370 characters that have "yi" as its pinyin. Pinyin is only a vehicle to help with the pronunciation of Chinese language, it is not a replacement for the Chinese writing system which uses Chinese characters. Examples of Pinyin and Chinese characters can be found in Figure 4.3.

The current standard, which is currently in use in Mainland China contains 8105 simplified Chinese characters. There is a smaller list of *Commonly Used Characters in Modern Chinese* which contains about 3,500 characters [GBM13] published by the General Office of the State Council of the People's Republic of China. *Traditional Chinese* is still in practice today in Taiwan, Hong Kong and Macau, it contains more complicated writing characters. Over 4,000 (4,762) characters were collected in the *List of Graphemes of Commonly-used Chinese Characters* by Hong Kong Education Bureau [CLESCDIEB12].

4.3.2 Entropy across similar languages

In other languages that share the same alphabets (or symbols) as English, the maximum information that can be conveyed by just one symbol of text, h_0, would have the same value,

$$h_0 = \log_2 27 \approx 4.75 \ .$$

Researchers proposed linguistic entropy can also be used as a measure to study the differences and similarities among languages. Consider the case where we follow Shannon's objective guessing design and the chosen subject only knows English. We randomly select a passage in another Latin script language, for example, German or French, and we ask the subject to guess as instructed in Shannon's experiment, from left to the right one letter at a time. If the subject knows absolutely nothing about German, then the subject can only select randomized succeeding letters with equal chances and the results should be rather close to $h_0 = 4.75$.

However, German and English share lots of similarities that would assist the subject in this task and perform slightly better in this partially known or even unknown language. In the scenario where the length of the preceding block is short, it is likely that the subject would only perform slightly worse in a foreign language than his mother tongue.

In 1968, D. Jamison and K. Jamison invited two subjects to conduct their experiment, one being an university graduate who spent some time studying in Italy in the past, and the other one being a junior in high school [JJ68]. Total of 50 random phrases were selected for subject one and 40 for subject two. In this experiment $n = 4, 8, 12$ and 100 from unspecified source was used.

	Entropies of					
n	English		French		Italian	
	Subject1	Subject2	Subject1	Subject2	Subject1	Subject2
4	2.18	2.25	3.66	3.34	2.89	3.43
8	1.56	1.98	2.87	2.81	2.57	3.11
12	1.69	1.98	2.64	3.24	2.78	3.28
100	1.63	1.67	3.16	3.20	3.00	3.78

Table 4.5: Body segment length measurements

From the results in Table 4.5 above, we can see that there are similarities among English, French and Italian. Looking at results for French and Italian, we can see that the entropies for French are quite similar to Italian's for both subjects. Subject two's result indicates that French and Italian have similar guessing difficulty for English speakers. Subject one's result suggests that the knowledge of Italian can be applied to French. Hence, entropy can be used as a quantitative measure in studying the similarities among languages. Comparing the numbers in Italian for both subjects, we can also see that the entropies for subject one are smaller than the ones for subject two, which suggests that the entropy decreases as familiarity of the language increases.

4.3.3 Why are English books so much thicker than Chinese ones?

Students often complain how heavy the physical copy of textbooks are in North America. However, this is less of an issue for students in China. This observation is also applicable to other books. For example, the textbook *Introduction to Algorithms, Third Edition* by CORMEN *et al*, contains 1312 pages for the English version [CLR+09] and only 780 pages for the Chinese version. Another example; the original English version of *Steve Jobs* by WALTER ISAACSON published by *Simon & Schuster* has 656 pages in length, and the Chinese translated version published by *CITIC Press Corporation* only has 560 pages.

There are several aspects we need to consider before answering the question from the title of this section. 1. Nature of translation, 2. average word length, font size and format, 3. entropy rate per page.

Original vs translation

English is widely used around the world, therefore there are a lot more books that were written in English originally before getting translated into other languages. I briefly went over these two versions of the aforementioned book *Steve Jobs* and I noticed some differences in the content due to formatting structure and the nature of translation, which potentially leads to a greater amount of information in the Chinese version. There are a lot more footnotes and explanations in order to help readers better understand the background of the context in the translated version. For example, explaining "CNN" as "The Cable News Network" in the Chinese version whereas the original work only contains the abbreviation. Another example here would be that both English and Chinese versions of the Apple's slogan, "Think Different" were included in the translated version. That being said, if the translators completely followed the English text structure, the size of this book could be further reduced. In spite of all this, although there is more information in the Chinese version, it is still significantly shorter.

Average word length, font size and format

There are words that appear to be shorter in English than Chinese. For example, "we" only contains 2 letters and the equivalent Chinese word also contains two characters, which generally take up more space. However, the average word length in English is greater than the average word length in Chinese. For instance, the word "mathematics", in English it contains 11 letters and the equivalent term in Chinese contains two Chinese characters.

With a one inch margin for all sides, one can fit about 3,500 English (Calibri, font size 12) letters on 44 lines or 1,500 Chinese (DengXian Regular, font size 12)

characters on 39 lines on one standard US letter page. Chinese characters all have the same width and height under the same font size, and punctuation marks take up less space. There are mono-spaced font and variable-width font in English, which the later one is more commonly in use. See *Standard alphabets for traffic control devices introduction* for detailed descriptions on font sizes in English. These values are for rough estimations.

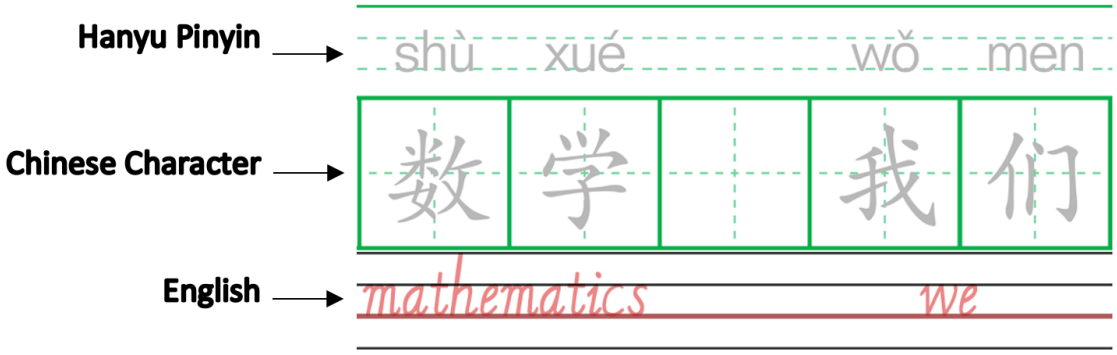

Figure 4.3: Chinese and English characters

$L_{\text{word}} = 4.5$ is the average word length in English Shannon used, since there are spaces in between words, then about 640 words can be fitted on one page. Meanwhile, Chinese is unspaced, some words are one character in length but the majority of words are two characters in length, about 750 words can fit on one page.

If we look at the TXT versions of the book Steve Jobs, the English version contains 1,360,343 bytes and the Chinese version has 1,314,393 bytes. Using *ASCII* (American Standard Code for Information Interchange), English text only requires one byte per symbol, whereas Chinese text requires three bytes per character and 2 bytes per punctuation mark. If we use the font and page standard from the previous paragraph, we have roughly 380 pages in English and 290 pages in Chinese (disregarding the format). Of course this is a lot smaller than the values from the actual book, since there are illustrations and further formatting in the printed versions.

Results from PPM experiment with the Bible

Table 4.6 contains the result from Behr *et al.* [BFMX03], which uses the Bible as a sample. We selected four compression algorithms used in Behr's experiment, PPMD, PPMZ, BZIP2 and GZIP, their detailed description can be found in [BFMX03]. Unlike the book by Issacson we previously considered, the different translations do not contain additional background and comments. Theoretically speaking, contents in the Bible should contain the same amount of information. The only difference between versions is the format. As the "Original" column indicates, the file size of Chinese

Bible is less than half size of the English Bible, using the symbols and page conversion we obtain earlier, it leads to about 190 pages in Chinese and about 540 pages in English. It can be deduced from this that Chinese characters are more expressive.

Language	Original	PPMD	PPMZ	BZIP2	GZIP
English	1,936,473	390,846 (4.95)	363,288 (5.33)	411,732 (4.70)	562,720 (3.44)
French	1,896,459	393,805 (4.82)	359,903 (5.27)	422,532 (4.61)	572,904 (3.37)
Chinese	884,860	337,505 (2.62)	341,850 (2.59)	370,168 (2.15)	438,738 (1.57)

Table 4.6: The Bible: file sizes (in bits) and the corresponding compression ratios

In theory, using an efficient algorithm can remove all the redundancy which would result in the compressed files containing only pure information. Therefore, the sizes of the compressed Bible should be the same. Table 4.6 shows that the range for compressed file sizes reduces significantly in comparison to the range for the original ones. However, the differences in compressed files among languages are still not negligible. We speculate that the remaining difference is caused by the nature of the language like grammar.

For the compressed data, we can see that PPMD gives the smallest compressed file in size for Chinese where PPMZ gives out the smallest compressed size for English. The biggest compression ratio for English is 5.33 with PPMZ and 2.62 for Chinese with PPMD which is not as efficient as when applying to English. Therefore, we can see that for the language with bigger entropy rate, the compression ratio is relatively small. This part is easy to understand, as it would be harder for one to compress a "compressed" file. This is the reason why the Bible in English is so much thicker than the Chinese version.

Entropy per page

Similar to formula (4.2.3), we could also use the unit conversion: 8 bits = 1 byte and 1 Chinese character = 3 bytes to obtain the entropy rate per character of Chinese

text from formula (4.2.2), we have

$$\begin{aligned}\text{Entropy Rate} &= \frac{\#\ bit}{\#\ character} \\ &= \frac{\text{Compressed file in \# byte}}{\text{Uncompressed file in \# byte}} \times \frac{\text{8 bits/1 byte}}{\text{1 character/3 bytes}} \\ &= \frac{1}{\rho} \times \frac{\text{8 bits/1 byte}}{\text{1 character/3 bytes}}\ .\end{aligned}$$

This yields,

$$\text{Entropy of the text in Chinese} = \frac{24}{\rho}\ . \tag{4.3.1}$$

Continuing our discussion on the Bible experiment, (results are recorded in Table 4.6), we pick the most efficient algorithm for both languages, PPMZ for English and PPMD for Chinese. We get an entropy rate of 1.5 bits per symbol for English using formula (4.2.4), and 9.2 bits per character for Chinese using formula (4.3.1). However, it is not intuitive to draw a conclusion based on these two values as they have different units. In order to answer the raised question, we decide to use a consistent unit like *bits per page*. Based on the standard font size and format we used earlier, we get an entropy rate of 5.3×10^3 bits per page for English and an entropy rate of 13.9×10^3 bits per page for Chinese. This implies that Chinese text contains double amount of information compared to English on the same page. This is precisely the reason that it is common to see much thicker textbooks in English compared to those in Chinese.

4.4 Modern views on applicability of Shannon's model

4.4.1 Difference analysis

The above experiments all show different results of entropy rate h. Some of them seem to be close to Shannon's foundational result, however the others reflect a bigger range of this value. All the experiments conducted above have bias. For Shannon's experiment [Sha51], only one subject was used with one literary work. Many trials could improve the subject's skill at predicting and the statistical accuracy of the results. In Jamisons' experiment [JJ68], the sample size is also really small. In Moradi et al.'s work [MGBR98], max n for a session is only 64 and 32 for his first and second experiment respectively. Small n might lead to a less accurate h since one is interested in the result when $n \to \infty$. The result from the experiment conducted in 2019 by Ren, Takahashi and Tanaka-Ishii [RTTI19] seems to be a fairly appropriate estimate. However, even though these subjects were limited to residences in the

United States, Canada, Great Britain and Australia, their native tongues can not be controlled. Discrepancy might occur since all participants remain anonymous. Guerrero's calculation [Gue09] is based on a reasonably large sample, however only novels were selected and this method requires a considerably long computing time.

In my own calculation, an entropy rate of $h \approx 2.18$ was found using collection of Shakespearean works. This value was slightly bigger than results from other researchers'. There are a few improvements for future study. First of all, We can choose our data using articles with modern English, which could lead us to a smaller entropy rate due to the changes in English over time. Secondly, a combination of various works written by different authors can also be used to study the connection between authorship and entropy rate. Thirdly, it might also be interesting to extend Jamisons' cognitive experiment described in [JJ68] from different language families. In this case subjects would be given strings in languages that are not from the same family as the ones they know. For example, a Chinese subject could be given a string of words in English and vice versa.

4.4.2 Hilberg's Ansatz

As more and more people get into the field of studying entropy, more variables have been taken into consideration. With computational technique, scholars are only able to use a finite data set, and the true entropy rate is formulated as a limit for infinite data. Considering the convergence is rather slow for natural language, modifications to Shannon's formula have been proposed.

In 1990, Wolfgang Hilberg proposed a function with more functional parameters, α and β [Hil90]. For some $\beta < 1$, we define a new function to be $\hat{h}_0(n)$, where

$$\hat{h}_0(n) = \alpha n^{\beta-1} \ .$$

Then we take h as another functional variable, the above formula was modified as

$$\hat{h}_1(n) = \alpha n^{\beta-1} + h \ .$$

Hilberg believes the entropy rate vanishes, and $h = 0$.

However, there are plenty of works that have been done on the topic of entropy and many researchers have shown that the entropy rate should be positive with various experiment results [JL06],[GC02].

Bibliography

[AA65] Robert B. Ash and Robert F. Ash, *Information theory*, Interscience tracts in pure and applied mathematics, Interscience Publishers, 1965.

[BFMX03] Frederic Behr, Victoria Fossum, Michael Mitzenmacher, and David Xiao, *Estimating and comparing entropies across written natural languages using PPM compression*, Data Compression Conference, 2003. Proceedings. DCC 2003, IEEE, 2003, pp. 416–424.

[BL55] N. G. Burton and J. C. R. Licklider, *Long-range constraints in the statistical structure of printed English*, The American Journal of Psychology **68** (1955), no. 4, 650–653.

[Bri13] Leon Brillouin, *Science and information theory*, Courier Corporation, 2013.

[BSS15] Vladimir V. Bochkarev, Anna V. Shevlyakova, and Valery D. Solovyev, *The average word length dynamics as an indicator of cultural changes in society*, Social Evolution and History **14** (2015), no. 2, 153–175.

[CK78] Thomas Cover and Roger King, *A convergent gambling estimate of the entropy of English*, IEEE Transactions on Information Theory **24** (1978), no. 4, 413–421.

[CLESCDIEB12] Chinese Language Education Section Curriculum Development Institute Education Bureau, *List of graphemes of commonly-used Chinese characters*, The Government of the Hong Kong Special Administrative Region, 2012.

[CLR+09] Tohams H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Stein Clifford, et al., *Introduction to algorithms, third edition*, MIT Press, 2009.

[Dęb15] Łukasz Dębowski, *A preadapted universal switch distribution for testing Hilberg's conjecture*, IEEE Transactions on Information Theory **61** (2015), no. 10, 5708–5715.

[Deu96] Peter Deutsch, *GZIP file format specification version 4.3*, Tech. report, 1996.

[Dew50] Godfrey Dewey, *Relative frequency of English speech sounds*, Cambridge Harvard University Press, 1950.

[Dub08] Heather Dubrow, *The Arden Shakespeare. Shakespeare's Poems*, Shakespeare Quarterly **59** (2008), no. 4, 488–491.

[Fek23] Michael Fekete, *Über die Verteilung der Würzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten*, Mathematische Zeitschrift **17** (1923), no. 1, 228–249.

[GBM13] General Office of the State Council of the People's Republic of China Guowuyuan Bangongting Mishuju, *Tongyong guifan hanzi biao, table of general standard Chinese characters*, Guojia Yuyan Wenzi Gongzuo Weiyuanhui, State Language Commission, 2013.

[GC02] Dmitriy Genzel and Eugene Charniak, *Entropy rate constancy in text*, Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 199–206.

[Gra09] Robert M. Gray, *Probability, random processes, and ergodic properties*, vol. 1, Springer, 2009.

[Gra11] ———, *Entropy and information theory*, Springer Science & Business Media, 2011.

[Gue09] Fabio G. Guerrero, *A new look at the classical entropy of written English*, arXiv preprint arXiv:0911.2284 (2009).

[Hil90] Wolfgang Hilberg, *Der bekannte Grenzwert der redundanzfreien Information in Texten-eine Fehlinterpretation der Shannonschen Experimente?*, Frequenz **44** (1990), no. 9-10, 243–248.

[Huf52] David A. Huffman, *A method for the construction of minimum-redundancy codes*, Proceedings of the IRE **40** (1952), no. 9, 1098–1101.

[Jam18] Lee Jamieson, *Tragedy, comedy, history?*, Jul 2018, https://www.thoughtco.com/tragedy-comedy-history-plays-2985253.

[JJ68] Dean Jamison and Kay Jamison, *A note on the entropy of partially-known languages*, Information and control **12** (1968), no. 2, 164–167.

[JL06] T. Florian Jaeger and Roger Levy, *Speakers optimize information density through syntactic reduction*, Advances in neural information processing systems **19** (2006).

[Juh17] Sanna Juhlin, *An entropy estimate of written language and twitter language: A comparison between English and Swedish*.

[Khm00] Dmitry V. Khmelev, *Disputed authorship resolution through using relative empirical entropy for markov chains of letters in human language texts*, Journal of quantitative linguistics **7** (2000), no. 3, 201–207.

[Kje94] Bradley Kjell, *Authorship determination using letter pair frequency features with neural network classifiers*, Literary and Linguistic Computing **9** (1994), no. 2, 119–124.

[KPK02] O. V. Kukushkina, A. A. Polikarpov, and D. V. Khmelev, *Using literal and grammatical statistics for authorship attribution*, Problemy Peredachi Informatsii **37** (2002), no. 2, 96–108.

[KT01] Dmitri V. Khmelev and Fiona J. Tweedie, *Using markov chains for identification of writer*, Literary and linguistic computing **16** (2001), no. 3, 299–307.

[Liu08] Yuehua Liu, *Integrated Chinese: Simplified characters textbook, level 1, part 1 (English and Chinese edition)*, Cheng & Tsui, 2008.

[LS13] E Jebamalar Leavline and DAAG Singh, *Hardware implementation of LZMA data compression algorithm*, International Journal of Applied Information Systems (IJAIS) **5** (2013), no. 4, 51–56.

[LV+08] Ming Li, Paul Vitányi, et al., *An introduction to Kolmogorov complexity and its applications*, vol. 3, Springer, 2008.

[ME11] Nathaniel F. G. Martin and James W. England, *Mathematical theory of entropy*, vol. 12, Cambridge University Press, 2011.

[MGBR98] Hamid Moradi, Jerzy W. Grzymala-Busse, and James A. Roberts, *Entropy of English text: Experiments with humans and a machine learning system based on rough sets*, Information Sciences **104** (1998), no. 1-2, 31–47.

[Pra39] Fletcher Pratt, *Secret and urgent. the story of codes and ciphers*, no. 12, Robert Hale Limited, 1939.

[Pu05] Ida Mengyi Pu, *Fundamental data compression*, Butterworth-Heinemann, 2005.

[RTTI19] Geng Ren, Shuntaro Takahashi, and Kumiko Tanaka-Ishii, *Entropy rate estimation for English via a large cognitive experiment using mechanical turk*, Entropy **21** (2019), no. 12, 1201.

[Sew96] Julian Seward, *bzip2 and libbzip2*, avaliable at http://www. bzip. org (1996).

[Sha48] Claude Elwood Shannon, *A mathematical theory of communication*, The Bell system technical journal **27** (1948), no. 3, 379–423.

[Sha51] ———, *Prediction and entropy of printed English*, Bell system technical journal **30** (1951), no. 1, 50–64.

[Sha20] Folger Shakespeare, *Complete set*, May 2020, https://shakespeare.folger.edu/download-the-folger-shakespeare-complete-set/.

[Sta01] Michael Stay, *ZIP attacks with reduced known plaintext*, International Workshop on Fast Software Encryption, Springer, 2001, pp. 125–134.

[TC96] William J. Teahan and John G. Cleary, *The entropy of English using PPM-based models*, Proceedings of Data Compression Conference-DCC'96, IEEE, 1996, pp. 53–62.

[TI21] Kumiko Tanaka-Ishii, *Statistical universals of language: Mathematical chance vs. human choice*, Springer Nature, 2021.

[TTID16] Ryosuke Takahira, Kumiko Tanaka-Ishii, and Łukasz Debowski, *Entropy rate estimates for natural language—a new extrapolation of compressed large-scale corpora*, Entropy **18** (2016), no. 10, 364.

[YY83] Akira Moiseevich Yaglom and Isaak Moiseevich Yaglom, *Probability and information*, vol. 35, Springer Science & Business Media, 1983.

[Zip65] George Kingsley Zipf, *Human behavior and the principle of least effort; an introduction to human ecology.*, facsim. of 1949 ed., Hafner Pub. Co., New York, 1965.