Introduction to Information Theory

Information must not be confused with meaning. "The semantic aspects of communications are irrelevant to the engineering aspects". [Sh48].

Information is a measure of one's freedom of choice and is measured by the **logarithm of the number of choices.** Tossing of a coin gives two choices . If the logarithm is with respect to base 2, we have unit information called a "bit". With doubling of choices you have an extra bit of information. Thus 4,8,16 choices lead to 2, 3, 4 bits, respectively, of information. In general if you have *N* choices, the information content of the situation is$\lceil \log_2 N \rceil$, which is the number of binary digits to encode the number *N.*

The above situation can be **captured by using probability**. Since each event is assumed to be independent, the probability of the *i*th ($1 \leq i \leq N$) event is $p_i = 1/N$ (All events are assumed to be equally probable) and the amount of information associated with the occurrence of this event or **self-information** is given by $-\log p_i$. If p_i=1 then the information is zero (certainty) and if p_i=0 , it is infinity; if p_i equals 0.5, it is one bit corresponding to N=2. If N=4, p_i =0.25 and the information is 2 bits and so on.

Note in the case of tossing of a coin there are two possible events: head or tail If you consider the tossing of the coin to be an "experiment", the question is **how much total information will this experiment have?** This can be quantified if we can describe the outcome of the experiment in some reasonable fashion. Lets "encode" the outcome 'head' to be represented by the bit 1 and outcome 'tail' by the bit 0. Thus, a minimal description of this experiment needs only one bit. **Note the experiment is the sum total of all the events. If we take the self-information of each event, multiply this by its probability and sum it up over all the events, intuitively that gives a measure of information content or *average information* of the experiment**. It just so happens that this entity is also just one bit for the tossing event since the probability of either head or tail is 0.5 and self information for each event is also 1 bit. ***This 1 bit also expresses how uncertain we are of the outcome.*** How do you generalize the definition?

Suppose, we have a set of *N* events whose probabilities of occurrence are $p_1, p_2, \ldots, p_N$. Can we measure how much "choice" is involved or how much uncertain we are of the outcome? Such a measure is precisely the **entropy** of the experiment or "source" denoted as $H(p_1, p_2, \ldots, p_N)$. [More precisely, it is called the **first order entropy**. Higher order entropies depend on contextual information. The true entropy is infinite order entropy. But,by popular use, entropy most often refers to first order entropy unless stated otherwise. Read the discussion from Sayood pp.14-16].

It is reasonable to require the following properties of *H*.

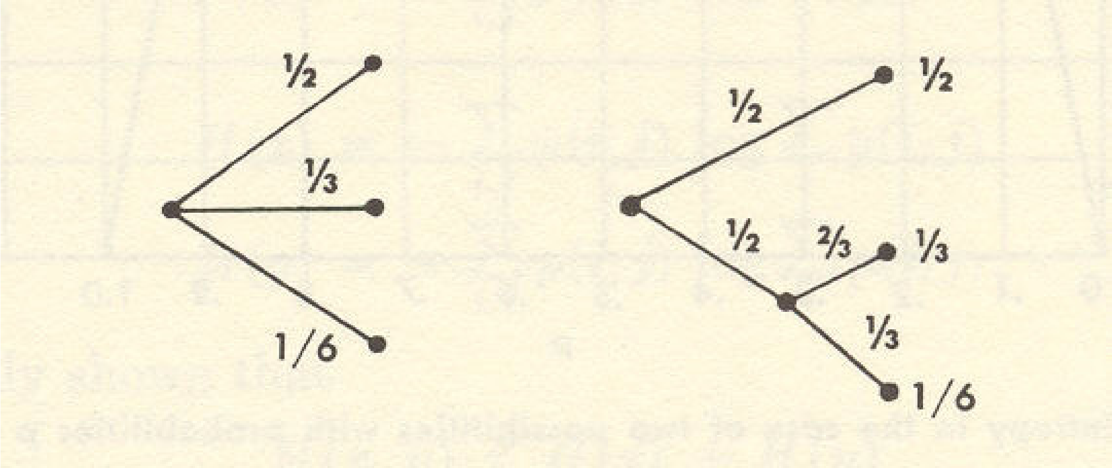
1. *H* should be continuous in p, that is, a small change in the value of p_i should cause small change in the value of *H*.

2. If $p_i=1/N$, then H should be a monotonic increasing function of N. That is, with equally likely events, there is more choice or uncertainty when there are more possible events.

3. If a choice is broken down into two successive choices, the original H should be weighted sum of the individual values of H. Thus , we require

$$H(1/2, 1/3, 1/6) = H(1/2, 1/2) + 1/2H(2/3, 1/3)$$

(Figure taken from [ShW98])



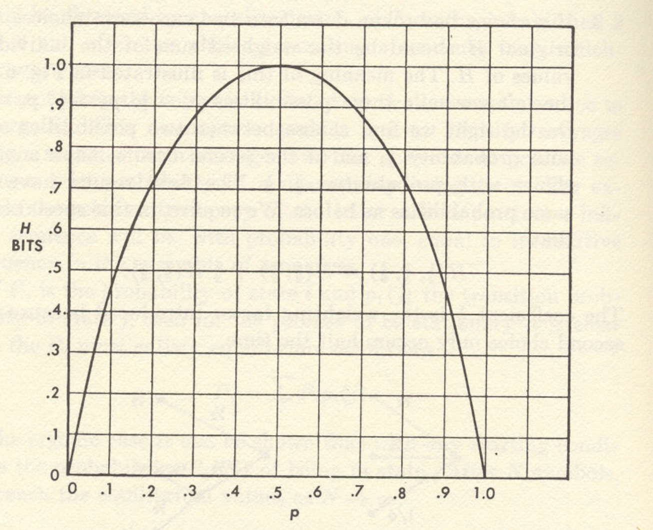
- Decomposition of a choice from three possibilities.

Theorem 1: **The only H satisfying the above assumptions is $H= -\sum p_i \log p_i$**

(Proof: See [ShWe00] Appendix 2 or Saywood, pp.18-22.)

The form of H is recognized as that of entropy in statistical mechanics and thermodynamics and the H is the Boltzmann's famous H theorem. The entropy in case of an experiment with two possibilities with probabilities p and $q=1-p$ is

$$H= - (p\log p + (1-p) \log (1-p)) \qquad (1)$$



g. 7. — Entropy in the case of two possibilities with probabilities p and $(1—p)$.

(Figure taken from [ShW98]

The quantity H has several interesting properties:

1. H=0 iff one of the p_i is 1 and the rest are 0. Thus, only when we are certain of the outcome , H will vanish or become 0.
2. For a given N, H is maximum and equals $\log N$ when all p_i 's are equal , that is, 1/N. This is the most uncertain situation.

The above two situations are special situations of the following theorem.

Theorem2: **The entropy *H* of {*p_i*} , *1≤i≤N*, satisfies *0≤ H ≤ log N***

Proof: Points 1) and 2) above prove the left and right equality. We need the following inequalities to prove the inequalities.

$$\begin{aligned}
&\ln(x) \le x-1 \\
&\ln(x) \ge 1-\tfrac{1}{x} \\
&\log(x) \le \log(e)[(x-1)] \\
&\log(x) \ge \log(e)[(1-\tfrac{1}{x})]
\end{aligned} \qquad (2)$$

To obtain the left inequality, note $-p\log p \ge 0$ for $0 \le p \le 1$ with equality if p=1. Hence $H \ge 0$. To obtain the right inequality, note $\sum_i p_i = 1$, so we can write:

$$\begin{aligned}
&\log N - H = \sum_i p_i \log N + \sum_i p_i \log p_i \\
&= \sum_i p_i (\log N + \log p_i) \\
&= \sum_i p_i (\log Np_i) \qquad (3)\\
&\geq \sum_i p_i (\log e(1 - \tfrac{1}{Np_i})
\end{aligned}$$

and equality holds iff $p_i = \frac{1}{N}, \forall i.$ Thus, we can write

$$\log N - H \geq k(\sum_i p_i - \sum_i \tfrac{1}{N}) = k(1-1) = 0$$

where $k = \log_2 e$ is a constant. Thus we have the result:

$$0 \leq H \leq \log N \qquad (4)$$

3. Any change toward equalization of probabilities $p_1, p_2, \ldots, p_n.$ increases *H.* Thus if $p_1 < p_2$ and we increase p_1 decreasing p_2 by the same amount so that these two probabilities are nearly equal, then *H* increases. **In general, any averaging operation will increase *H*.**

Joint Probability

Suppose there are two discrete events, *X* and *Y*, with *N* possibilities for X and *M* possibilities for Y. Let *p(i,j)* be the probability of the **joint occurrence** of i $(\mathbf{1 \leq i \leq N})$ for *X* and *j* $(\mathbf{1 \leq i \leq M})$ for *Y.* The entropy of the joint event is

$$H(X,Y) = -\sum_{i \ni X} \sum_{j \ni Y} p(i,j) \log p(i,j) \qquad (5)$$

which is also sometimes written as

$$H(X,Y) = -\sum_{i,j} p(i,j) \log p(i,j) \qquad (6)$$

Given the joint probabilities, the entropy *H*(*X*) and *H*(*Y*) can be easily obtained as

$$H(X) = -\sum_i p_i \log p_i = -\sum_i [\sum_j (p(i,j)) \log(\sum_j p(i,j))] \qquad (7)$$

$$H(Y) = -\sum_j p_j \log p_j = -\sum_j [\sum_i (p(i,j)) \log(\sum_i p(i,j))] \qquad (8)$$

since $p(i) = \sum_j p(i,j)$ and $p(j) = \sum_i p(i,j)$.

Homework Assignment: Show that $H(X)+H(Y)\geq H(X,Y)$. **The uncertainty of the joint event is less than equal to the sum of individual uncertainties. The equality holds when the two events are independent, that is,** $p(i,j)=p(i)p(j)$.

Conditional Probability

Suppose there are two chance events *X* and *Y*, not necessarily independent. If $p(i)$ is the *a priori* probability of the event *X=i* and if *p(i/j)* is *a posteriori* probability of the event *X=i*, given the event *Y=j* has occurred, then we can express $p(i,j)$ as

$$p(i,j)=p(j)p(i/j)$$

which gives the **Bayer's rule** stated normally as

$$p(i/j)=\frac{p(i,j)}{p(j)} \qquad (9)$$

Similarly we have,

$$p(j/i)=\frac{p(i,j)}{p(i)} \qquad (10)$$

which gives

$$p(i,j)=p(j)p(i/j)=p(i)p(j/i) \qquad (11)$$

and

$$p(i)=\sum_{j}p(i,j)=\sum_{j}p(j)p(i/j) \qquad (12)$$

and substituting the value of *p(i,j)* in equation (9), we get

$$p(i/j)=\frac{p(i)p(j/i)}{p(j)} \qquad (13)$$

and so on.

Conditional Entropy

The entropy *H(X/Y)* is defined to be the average of entropy of *X* for all values of *Y*.

$$H(X/Y)=\sum_{j\ni Y}p(j)H(X/Y=j) \qquad (14)$$

$$=-\sum_{j}p(j)(\sum_{i\exists X}p(i/j)\log p(i/j)) \qquad (15)$$

$$=-\sum_{i}\sum_{j}p(j)p(i/j)\log p(i/j) \qquad (16)$$

$$=-\sum_{i}\sum_{j}p(i,j)\log p(i/j) \qquad (17)$$

Similarly, we can write

$$\text{H(Y/X)} = -\sum_j \sum_i p(i) p(j/i) \log p(j/i) \quad (18)$$

$$= -\sum_j \sum_i p(i,j) \log p(j/i) \quad (19)$$

The above definition of the joint and conditional entropies is naturally justified by the fact that the entropy of two discrete random variables is the entropy of one variable plus the conditional entropy of the other. This is expressed by the theorem:

Theorem3: $H(X,Y) = H(X) + H(Y/X)$ (20)

Proof**:** From Eqn.(5),we have

$$\text{H(X,Y)} = -\sum_i \sum_j p(i,j) \log p(i,j)$$

$$= -\sum_i \sum_j p(i,j) \log p(i) p(j/i) \qquad \text{[By Eqn.(10)]}$$

$$= -\sum_i \sum_j p(i,j) \log p(i) - \sum_i \sum_j p(i,j) \log p(j/i)$$

$$= -\sum_i p(i) \log p(i) - \sum_i \sum_j p(i,j) \log p(j/i)$$

$$= H(X) + H(Y/X) \qquad \text{[By Theorem1 and Eqn.(19)}$$

We can similarly prove that

$$H(X,Y) = H(Y) + H(X/Y) \quad (21)$$

Combining Eqns(20 and (21), we have

$$H(X,Y) = H(X) + H(Y/X) \quad = H(Y) + H(X/Y) \quad (22)$$

Or

$$H(X) - H(X/Y) = H(Y) - H(Y/X) \quad (23)$$

The quantity expressed by Eqn.(23) is called the **mutual information ,** denoted as *I(X:Y)* or *I(Y:X).* **It can also be defined as the relative entropy between the joint distribution *p(i,j)* and the product distribution *p(i)p(j)*. That is,**

$$I(X:Y) = \sum_i \sum_j p(i,j) \log \frac{p(i,j)}{p(i)p(j)}$$

$$= \sum_i \sum_j p(i,j) \log \frac{p(i/j)}{p(i)} \qquad \text{[By Eqn.(9)]}$$

$$= -\sum_i \sum_j p(i,j) \log p(i) - [-\sum_i \sum_j p(i,j) \log p(i/j)]$$

$$= H(X) - H(X/Y) = H(Y) - H(Y/X) \quad (24)$$

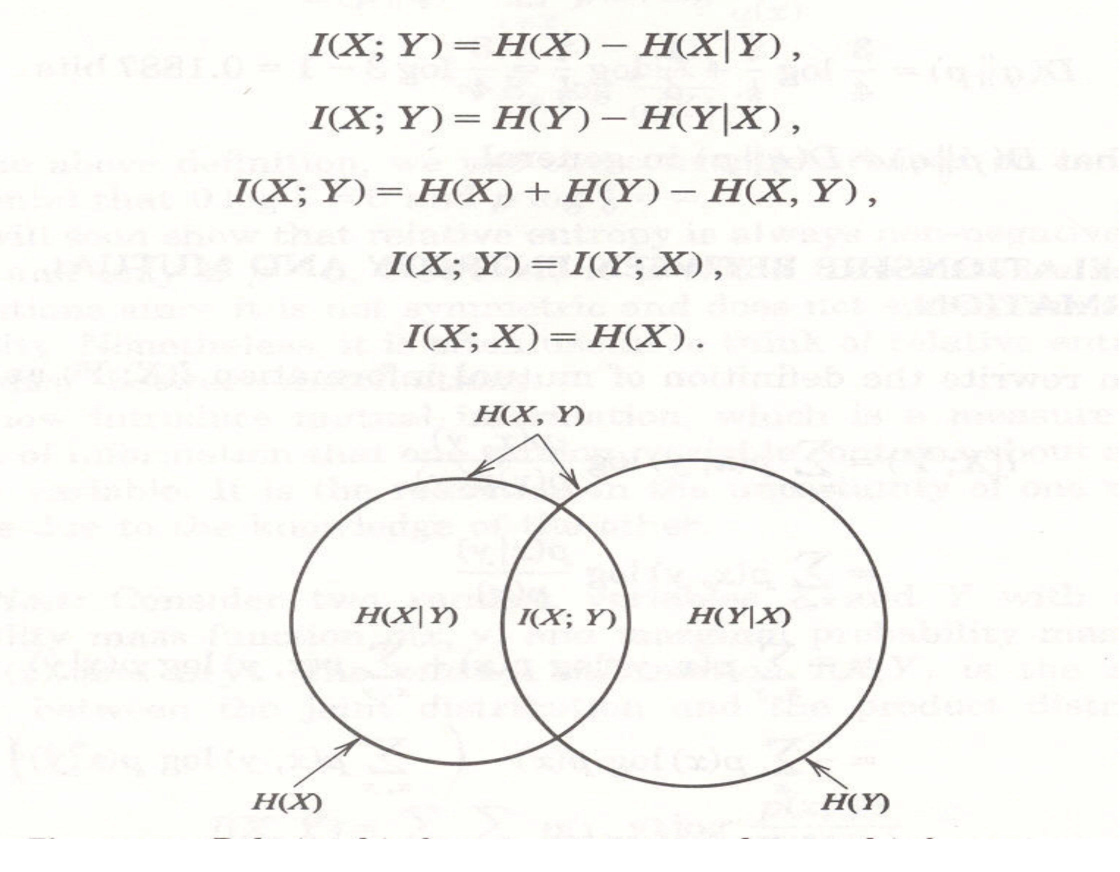
The mutual information is the reduction in the uncertainty of X due to knowledge of Y and vice versa. Thus X says as much about Y as Y says about X.
Also,

$$I(X:X) = H(X) - H(X/X) = H(X) - 0 = H(X) \qquad (25)$$

This is the reason why the entropy is sometimes referred to as self-information.
Similarly,

$$I(Y:Y) = H(Y) \qquad (26)$$

The relationship between entropy and mutual information can be depicted by the following Venn diagram.



$$I(X;Y) = H(X) - H(X|Y)\,,$$

$$I(X;Y) = H(Y) - H(Y|X)\,,$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y)\,,$$

$$I(X;Y) = I(Y;X)\,,$$

$$I(X;X) = H(X)\,.$$

(Figure taken from CoT91]}

Information Sources

An “event” in the above discussion could be a “message” in the context of communication application. Thus the above discussion is applicable to an artificial situation when the information source is free to choose only between several definite messages. A more natural situation is when the information source makes a sequence of choices from a set of elementary symbols (letters of an alphabet or words) or musical notes or web pages. The successive sequences are governed by probabilities which are not independent but at any stage, depend on the preceding choices. For example, if the source is English language text, not all letters are equiprobable; there are no words starting with the letter "j" and followed by 'b, c, d, f, g, j, k, l, q, r, t, v, w, x, z'. Thus, in real-life data compression applications, context plays a critical role

A system which produces a sequence of symbols according to certain probabilities is called a **stochastic process** and the special case of a stochastic process in which probabilities depend on the previous events, is called a **Markov process**. We will be concerned with only discrete sources in the following discussion.

We will first discuss the simplest case of a stochastic process where the successive symbols are **independent and identically distributed (iid) leading** to what is called the first order entropy model of a source. Some abstract examples are:

1. Alphabet $A=(a,b,c,d,e)$ and all probabilities are equal 0.2. A typical sequence may look like *bdcbecbabdcedbae….*

2. Same alphabet but $p(a)=0.4$, $p(b)=0.1$, $p(c)=0.2$, $p(d)=0.2$ and $p(e)=0.1$. A typical sequnce may look like *aaacdcdbdceaadada…* Note there are more *a*'s, *c*'s and *d*'s and only a few *b*'s and *e*'s.

Note in the examples above, the probabilities are pre-specified or *a priori probabilities.* How can these values be determined? Only, if we examine all possible sequences of arbitrary lengths which is not practical. A simplifying assumption is that the Markov process is **ergodic,** that is, every sequence produced by the process has the same statistical properties. Obviously, for short sequences this property is not valid as in the case of the above two examples. We will assume that all the sequences that we are dealing with are ergodic. For such a sequence, if the successive symbols are independent, we can write an expression for the first order entropy of the source to be:

$$H = -\sum p_{\mathrm{i}}\, log\, p_{\mathrm{i}}$$

where $\mathrm{p_i}$ is the probability of symbol *i.* Suppose in this case, *we* consider now a long message of N symbols. It will contain with high probability about p_1N occurrences of the first symbol x_1, p_2N occurrences of the second symbol x_2, etc. where the alphabet $A=(x_1, x_2, \ldots, x_n)$. Hence the probability of this particular message will be roughly

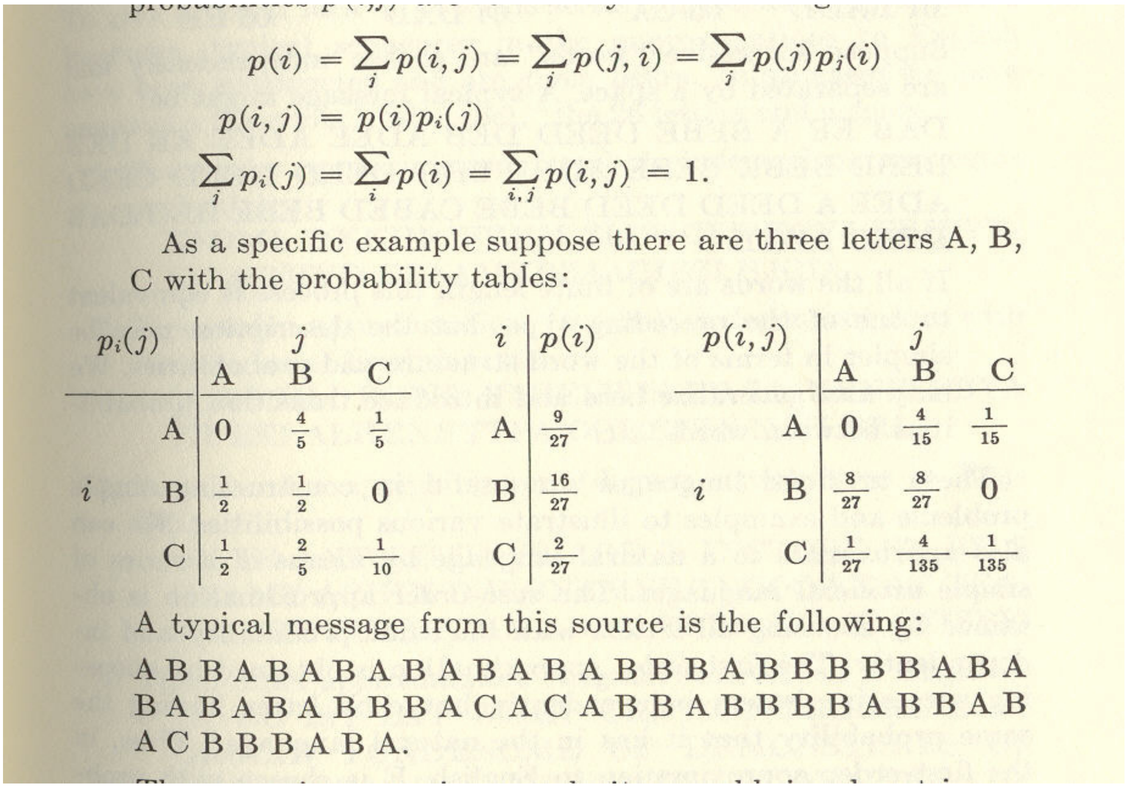
$$p = p_1^{p_1 N} p_2^{p_2 N} \ldots\ldots p_n^{p_{1n} N} \tag{27}$$

or

$$\begin{gathered} \log p \approx N \sum_i p_i \log p_i \\ \log p \approx -NH \\ H \approx \frac{\log 1/p}{N} \end{gathered} \tag{28}$$

H is thus approximately the logarithm of the reciprocal of the probability of the typical long sequence divided by the length of the sequence. It can be rigorously proved that when *N* is large, the right hand side is very close to *H*.

3. Consider now the case when successive symbols are not chosen independently but their probabilities depend on the preceding letters (and not one before that). This is called the Order(1) model which can be described by a set of *transition probabilities* $p_i(j)$, the probability that letter *i* is followed by letter *j* [the bigram *ij*]. The indices *i* and *j* run over all symbols. An equivalent way is to specify all the diagram frequencies *p(i,j)*. We have already encountered these probabilities as conditional probability and joint probability of two independent events. As we know these probabilities are related by the following equations:



$$p(i) = \sum_j p(i,j) = \sum_j p(j,i) = \sum_j p(j)p_j(i)$$

$$p(i,j) = p(i)p_i(j)$$

$$\sum_j p_i(j) = \sum_i p(i) = \sum_{i,j} p(i,j) = 1.$$

As a specific example suppose there are three letters A, B, C with the probability tables:

$p_i(j)$		j A	B	C
	A	0	$\frac{4}{5}$	$\frac{1}{5}$
i	B	$\frac{1}{2}$	$\frac{1}{2}$	0
	C	$\frac{1}{2}$	$\frac{2}{5}$	$\frac{1}{10}$

i	$p(i)$
A	$\frac{9}{27}$
B	$\frac{16}{27}$
C	$\frac{2}{27}$

$p(i,j)$		j A	B	C
	A	0	$\frac{4}{15}$	$\frac{1}{15}$
i	B	$\frac{8}{27}$	$\frac{8}{27}$	0
	C	$\frac{1}{27}$	$\frac{4}{135}$	$\frac{1}{135}$

A typical message from this source is the following:

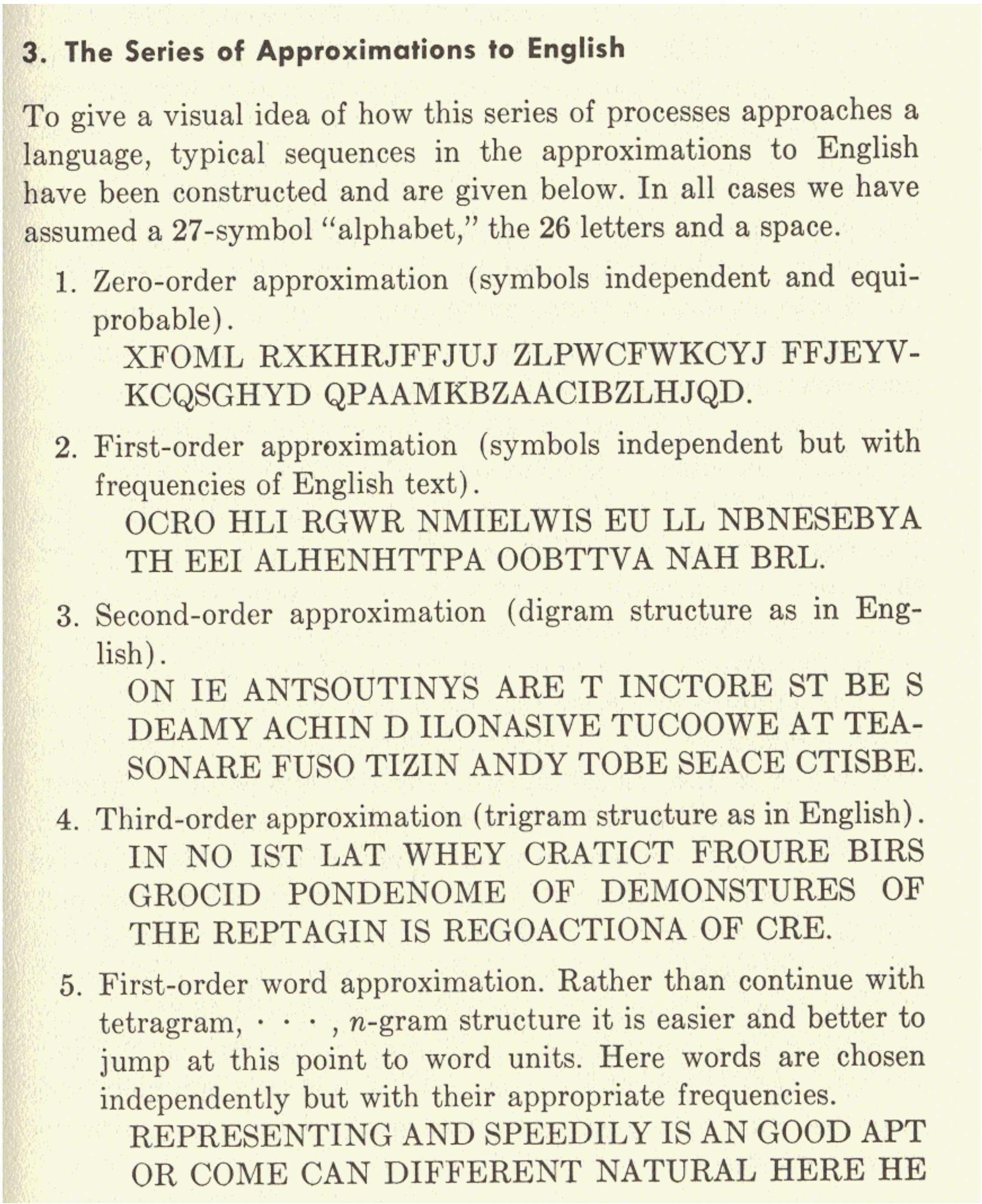
A B B A B A B A B A B A B A B B B A B B B B B A B A
B A B A B A B B B A C A C A B B A B B B B A B B A B
A C B B B A B A.

(Example taken from [ShW98])

Note the probabilities *p(i)* for any letter can be obtained by adding the probabilities in the corresponding row or column in the joint probability *p(i,j)* table. The conditional probabilities can be obtained by following the Bayer's rule :

$p_i(j)$= Given *i* as preceding the letter, what is the probability of *j* occurring as a succeeding letter which we denoted earlier as *p(j/i)= [p(i,j)] / p(i)*

The next increase in complexity will involve trigram probabilities p(i,j,k) or transition probabilities $p_{ij}(k)$. This model is called Order (2) model. And in general we can talk about an Order(k) model. Read from Shannon's original paper the dramatic effect of context as a source goes through a series of approximations to English as the context order is increased (Taken from [ShW98]).



3. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

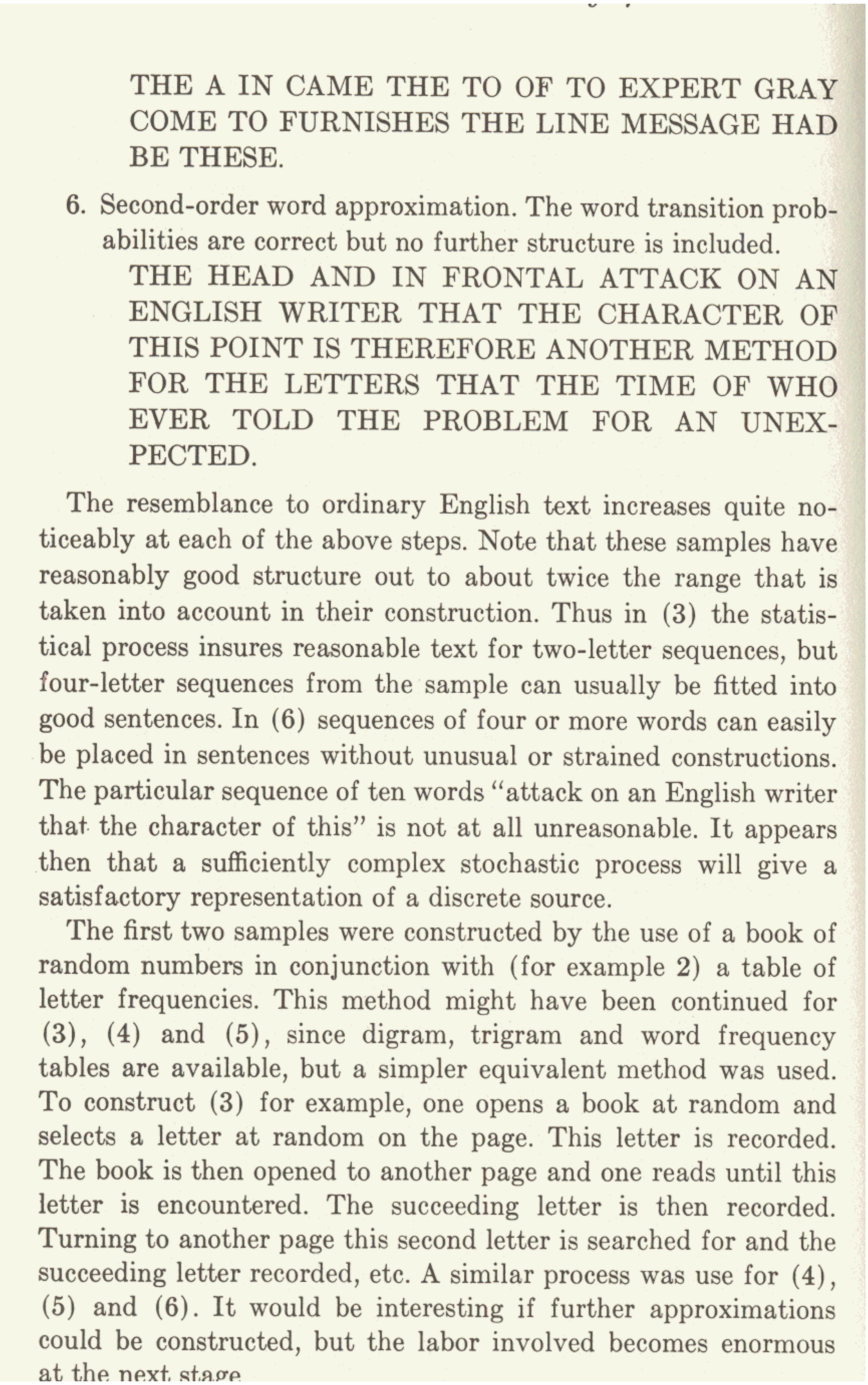
   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, · · · , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE



THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. The particular sequence of ten words "attack on an English writer that the character of this" is not at all unreasonable. It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source.

The first two samples were constructed by the use of a book of random numbers in conjunction with (for example 2) a table of letter frequencies. This method might have been continued for (3), (4) and (5), since digram, trigram and word frequency tables are available, but a simpler equivalent method was used. To construct (3) for example, one opens a book at random and selects a letter at random on the page. This letter is recorded. The book is then opened to another page and one reads until this letter is encountered. The succeeding letter is then recorded. Turning to another page this second letter is searched for and the succeeding letter recorded, etc. A similar process was use for (4), (5) and (6). It would be interesting if further approximations could be constructed, but the labor involved becomes enormous at the next stage

Markov Process Represented by a Graph

A finite number of states $S_1, S_2, \ldots, S_n$ and a set of transition probabilities $p_i(j)$, the probability that the system in state S_i will go to state S_j define a Markov graph. If each transition is now associated with an "output" symbol, it becomes an information source controlled by a Markov process. (Ignore figure number below. Example B and C are examples 2 and 3 on p.8, respectively, above. Figures taken from [ShW98])

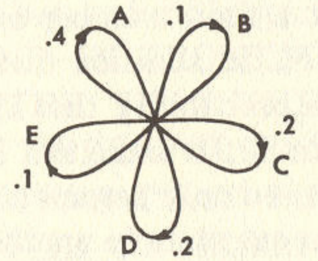


Fig. 3. — A graph corresponding to the source in example B.

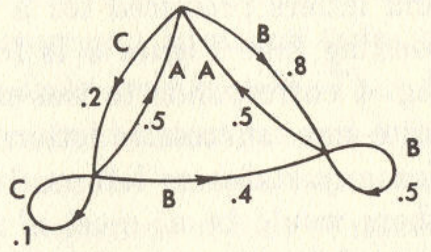


Fig. 4. — A graph corresponding to the source in example C.

(Read now Section 2.3, pp.22-26 from K. saywood and go through the example of entropy calculation of an image using first the probability model and then the Markov model)

Ergodic Process

If every sequence produced by the process is the same in statistical properties, then it is called an ergodic source. For English text, if the length of the sequence is very large, this assumption is approximately true. For an ergodic sequence:

1. The markov graph is connected.
2. The greatest common divisor of lengths of all cycles in the graph is 1.

Text Compression and Information Theory

The general approach to text compression is to find a representation of the text requiring less number of binary digits. In its uncompressed form each character in the text is represented by an 8-bit ASCII code[1]. It is common knowledge that such a representation is not very efficient because it treats frequent and less frequent characters equally. It makes intuitive sense to encode frequent characters with a smaller (less than 8) number of bits and less frequent characters with larger number of bits (possibly more than 8 bits) in order to reduce the *average* number of *bits per character* (BPC). In fact this principle was the basis of the invention of the so-called Morse code and the famous Huffman code developed in the early 50's. Huffman code typically reduces the size of the text file by about 50-60% or provides compression rate of 4-5 BPC. The entropy *H* can be looked upon as defining the *average number of BPC* required to represent or encode the symbols of the alphabet. Depending on how the probabilities are computed or modeled, the value of entropy may vary. If the probability of a symbol is computed as the ratio of the number of times it appears in the text to the total number of symbols in the text, the so-called *static* probability, it is called an Order(0) model. Under this model, it is also possible to compute the *dynamic* probabilities which can be roughly described as follows. At the beginning when no text symbol has emerged out of the source, assume that every symbol is equiprobable[2]. As new symbols of the text emerge out of the source, revise the probability values according to the actual frequency distribution of symbols at that time. In general, an Order(*k*) model can be defined where the probabilities are computed based on the probability of distribution of the (*k+1*)-grams of symbols or equivalently, by taking into account the context of the preceding *k* symbols. A value of *k=-1* is allowed and is reserved for the situation when all symbols are considered equiprobable, that is, $p_i = \frac{1}{|\mathrm{A}|}$, where | *A* | is the size of the alphabet *A*. When *k=1* the probabilities are based on *bigram* statistics or equivalently on the context of just one preceding symbol and similarly for higher values of *k*. For each value of *k*, there are two possibilities, the static and dynamic model as explained above. For practical reasons, a static model is usually built by collecting statistics over a test *corpus* which is a collection of text samples representing a particular domain of application (viz. English literature, physical sciences, life sciences, etc.). If one is interested in a more precise static model for a given text, a *semi-static* model is developed in a two-pass process; in the first pass the text is read to collect statistics to compute the model and in the second pass an encoding scheme is developed. Another variation of the model is to use a specific text to *prime* or seed the model at the beginning and then build the model on top of it as new text files come in.

Independent of what the model is, there is an entropy associated with each file under that model. **Shannon's fundamental noiseless source coding theorem says that entropy**

[1] Most text files do not use more than 128 symbols which include the alphanumerics, punctuation marks and some special symbols. Thus, a 7-bit ASCII code should be enough .

[2] This situation gives rise to what is called the zero-frequency problem. One cannot assume the probabilities to be zero because that will imply an infinite number of bits to encode the first few symbols since –log o is infinity. There are many different methods of handling this problem but the equiprobabilty assumption is a fair and practical one.

defines a lower limit of the average number of bits needed to encode the source symbols [ShW98]. The "worst" model from information theoretic point of view is the order(-1) model, the equiprobable model, giving the maximum value H_m of the entropy. Thus, for the 8-bit ASCII code, the value of this entropy is 8 bits. The **redundancy *R*** is defined to be the difference[3] between the maximum entropy H_m and the actual entropy H. As we build better and better models by going to higher order k, lower will be the value of entropy yielding a higher value of redundancy. The crux of lossless compression research boils down to developing compression algorithms that can find an encoding of the source using a model with minimum possible entropy and exploiting maximum amount of redundancy. But incorporating a higher order model is computationally expensive and the designer must be aware of other performance metrics such as decoding or decompression complexity (the process of decoding is the reverse of the encoding process in which the redundancy is restored so that the text is again human readable), speed of execution of compression and decompression algorithms and use of additional memory.

Good compression means less storage space to store or archive the data, and it also means less bandwidth requirement to transmit data from source to destination. This is achieved with the use of a *channel* which may be a simple point-to-point connection or a complex entity like the Internet. For the purpose of discussion, assume that the channel is noiseless, that is, it does not introduce error during transmission and it has a *channel capacity C* which is the maximum number of bits that can be transmitted per second. Since entropy H denotes the average number of bits required to encode a symbol, C/H denotes the average number of symbols that can be transmitted over the channel per second [ShW98]. **A second fundamental theorem of Shannon says that however clever you may get developing a compression scheme, you will never be able to transmit on average more than C/H symbols per second [ShW98]. In other words, to use the available bandwidth effectively, *H* should be as low as possible, which means employing a compression scheme that yields minimum BPC.**

References

[ShW98] C. E. Shannon, W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, 1998.

[Sh48] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, Vol. 27, pp.379-423,623-656, 1948.

[CoT91] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. John Wiley, 1991.

[3] Shannon's original definition is R/H_m which is the fraction of the structure of the text message determined by the inherent property of the language that governs the generation of specific sequence or words in the text.