Entropia to wielkość określająca liczbę bitów informacji zawartej w danej wiadomości lub źródle. Spełnia ona trzy naturalne warunki:

- $I(s)$ jest malejącą funkcją prawdopodobieństwa zajścia zdarzenia s.
- $I(s) = 0$ gdy s jest pewne, czyli $\mathbb{P}(S = s) = 1$.
- $I(s,t) = I(s) + I(t)$, gdy s i t są zdarzeniami niezależnymi, np. kolejnymi sygnałami emitowanymi przez źródło bez pamięci.

Dla pojedynczego zdarzenia s_i o prawdopodobieństwie p_i definiujemy entropię jako

$$I(s_i) = -\log p_i.$$

Podstawa logarytmu jest dowolna, bo

$$\log_s x = \log_s r \cdot \log_s x,$$

czyli w wyniku zmiany podstawy następuje tylko przeskalowanie entropii o stałą. Zazwyczaj przyjmuje się jako podstawę liczbę znaków w alfabecie, w naszym wypadku będzie to 2.

Dla źródła S, które emituje znak s_i z prawdopodobieństwem p_i możemy określić średnią entropię

$$H(S) = \sum_i p_i I(s_i) = \sum_i p_i \log \frac{1}{p_i}.$$

Dziś zajmiemy się sytuacją, w której informacja jest przekazywana od źródła do odbiorcy przez kanał, który niekoniecznie jest wolny od zakłóceń. Zakładamy istnienie źródła A emitującego znaki s_i z prawdopodobieństwem p_i (jak wyżej), kanału Γ, przez który przepływa informacja i który być może ją deformuje deformuje oraz wyjścia B.

$$\text{INFORMACJA} \xrightarrow{\text{kodowanie}}$$
$$\xrightarrow{\text{kodowanie}} \text{WIADOMOŚĆ} \xrightarrow{\text{przekaz}}$$
$$\xrightarrow{\text{przekaz}} \text{ZNIEKSZTAŁCONA WIADOMOŚĆ} \xrightarrow{\text{dekodowanie}}$$
$$\xrightarrow{\text{dekodowanie}} \text{ZNIEKSZTAŁCONA INFORMACJA}$$

Przykłady:

Binarny kanał symetryczny (BSC) przyjmuje bit zero lub jeden i z prawdopodobieństwem P przekazuje go wiernie, a z prawdopodobieństwem $\bar{P} = 1 - P$ przekazuje bit przeciwny.

Binarny kanał z kasowaniem (BEC) przyjmuje bit zero lub jeden i z prawdopodobieństwem P przekazuje go wiernie, a z prawdopodobieństwem $\bar{P} = 1 - P$ przekazuje informację o błędzie.

Aby opisać działanie kanału używamy macierzy $P_{i,j}$. Liczba $P_{i,j}$ (prawdopodobieństwo w przód) to prawdopodobieństwo tego, że zostanie przekazany znak b_j, jeśli nadany został znak a_i. Obrazuje to sytuację, w której znajduje się nadawca, który wie, co nadał, a nie wie, co wyjdzie.

Używa się też prawdopodobieństw w tył, czyli macierzy $Q_{i,j}$. Liczba $Q_{i,j}$ oznacza prawdopodobieństwo, że nadany został znak a_i, jeśli odebrany został znak b_j. Obrazuje to sytuację, w której znajduje się odbiorca — wiemy, co przyszło i zastanawiamy się, skąd to się wzięło.

Trzecim rodzajem prawdopodobieństw są prawdopodobieństwa łączne, opisane w macierzy $R_{i,j}$. Liczba $R_{i,j}$ opisuje prawdopodobieństwo tego, że zostanie wysłany bit a_i, a odebrany b_j. Obrazuje to sytuację niezależnego obserwatora, który nic nie wie.

Liczby $P_{i,j}$ są własnościami wewnętrznymi samego kanału, natomiast $Q_{i,j}$ oraz $R_{i,j}$ zależą również od źródła.

Przykładowo rozważmy kanał BSC z parametrem $P = 0.8$, w którym nadawca nadaje 0 z prawdopodobieństwem $p = 0.9$. Wtedy:

$$P_{00} = P_{11} = 0.8, P_{01} = P_{10} = 0.2.$$

$$Q_{00} \simeq 0.973, Q_{10} \simeq 0.027, Q_{01} \simeq 0.692, Q_{11} \simeq 0.308.$$

$$R_{00} = 0.72, R_{01} = 0.18, R_{10} = 0.02, R_{11} = 0.08.$$

Interesuje nas, ile informacji traci się przy przekazie. Możemy policzyć entropię wejścia i entropię wyjścia. Mogłoby się wydawać, że wystarczy odjąć od pierwszej liczby drugą i dostaniemy wynik. Niestety — nie jest tak łatwo.

Przykładowo dla BEC widać wyraźnie, że wśród informacji, które otrzymujemy na wyjściu nie są tylko te związane ze źródłem, ale też nowa informacja, czy nastąpił błąd, która wpływa na entropię (zwiększa ją), natomiast nie daje nam informacji o źródle.

Definiujemy zatem entropię warunkową. Mówi ona, ile nieznanej nam jeszcze informacji zawiera jeszcze źródło A, jeżeli już odczytaliśmy wyjście b_j.

$$H(A|b_j) = \sum_i Q_{ij} \log \frac{1}{Q_{ij}}.$$

Definiujemy teraz średnią entropię warunkową, czyli to, co średnio jeszcze zawiera nowego A, jeżeli znam B:

$$H(A|B) = \sum_j q_j H(A|b_j) = \sum_{i,j} R_{ij} \log \frac{1}{Q_{ij}}.$$

Analogicznie definiujemy też

$$H(B|A) = \sum_{i,j} R_{ij} \log \frac{1}{P_{ij}}.$$

Definiujemy też łączną wiedzę zawartą w wejściu oraz wyjściu:

$$H(A, B) = \sum_{i,j} R_{ij} \log \frac{1}{R_{ij}}.$$

Obejrzmy to na kilku przykładach:

Dla kanału pewnego (np. BSC z $P = 1$, czyli jeśli wyślę a_i, to prawie na pewno dojdzie b_i) będziemy mieli

$$H(A|B) = \sum_{i,j} R_{ij} \log \frac{1}{Q_{ij}} = \sum_{i=j} p_i \log 1 + \sum_{i \neq j} 0 \log \frac{1}{0} = 0,$$

analogicznie $H(B|A) = 0$ i $H(A, B) = H(A) = H(B)$.

Dla kanału zerującego (np. BEC z $P = 0$, czyli niezależnie od tego, co wyślę, dojdzie 0 będziemy mieli

$$H(A|B) = \sum_{i,j} R_{ij} \log \frac{1}{Q_{ij}} = \sum_{i} p_i \log \frac{1}{p_i} = H(A),$$

$$H(B|A) = \sum_{i,j} R_{ij} \log \frac{1}{P_{ij}} = \sum_{i} p_i \log 1 = 0$$

i $H(A, B) = H(A)$.

Dla kanału losowego (czyli to, co wyjdzie jest niezależne od tego, co wejdzie, np. BSC z parametrem $P = \frac{1}{2}$) mamy $H(A|B) = H(A)$, $H(B|A) = H(B)$ i $H(A, B) = H(A) + H(B)$.

W ogólności zachodzą następujące równości:

$$H(A,B) = H(B) + H(A|B)$$

oraz

$$H(A,B) = H(A) + H(B|A).$$

Umiemy już zatem policzyć, ile informacji zawiera jeszcze A, gdy już poznaliśmy B i na odwrót. Interesuje nas jednak raczej, ile informacji o A zawiera B. Na dedukcję, wydawałoby się, że powinno być to $H(B) - H(B|A)$ — informacje zawarte w B pomniejszone o to, co w B jest niezależne od A. Definiujemy zatem wzajemną informację:

$$I(A, B) = H(B) - H(B|A).$$

Korzystając z udowodnionych wcześniej równości opisujących $H(B|A)$ stwierdzamy, że

$$I(A, B) = H(B) + H(A) - H(A, B),$$

co jest niezależne od kolejności argumentów. Otrzymujemy zatem fakt, że wejście mówi nam tyle samo o wyjściu, co wyjście o wejściu.

Dla dowolnego kanału możemy zdefiniować jego pojemność jako ilość informacji, którą możemy przez niego przepuścić, jeżeli prawidłowo dobierzemy źródło. Zatem

$$C(\Gamma) = \sup_A I(A, B).$$

Uwaga: supremum w definicje $C(\Gamma)$ zawsze jest skończone, oraz osiągane dla pewnego rozkładu prawdopodobieństwa A.

Przypomnijmy sobie, jak wyglądał proces przekazywania informacji.

$$\text{INFORMACJA} \xrightarrow{\text{kodowanie}}$$
$$\xrightarrow{\text{kodowanie}} \text{WIADOMOŚĆ} \xrightarrow{\text{przekaz}}$$
$$\xrightarrow{\text{przekaz}} \text{ZNIEKSZTAŁCONA WIADOMOŚĆ} \xrightarrow{\text{dekodowanie}}$$
$$\xrightarrow{\text{dekodowanie}} \text{ZNIEKSZTAŁCONA INFORMACJA}$$

Zbadaliśmy już całkiem nieźle drugą strzałkę, pora zająć się trzecią.

Dla danego wyjścia B oraz wejścia A potrzebujemy reguły decyzyjnej $\Delta : B \to A$, która każdemu wyjściu przypisywałaby to wejście, od którego naszym zdaniem pochodzi.

Wydaje się naturalne branie takiej reguły decyzyjnej, która każdemu b_j przypisuje najbardziej prawdopodobne a_i, czyli to a_i, dla którego Q_{ij} jest największe. Oznaczmy to i przez j^*. Wtedy prawdopodobieństwo trafienia wynosi

$$P_{\Gamma,C} = \sum_j Q_{j^*j},$$

zaś prawdopodobieństwo chybienia —

$$P_{\Gamma,E} = \sum_{j, i \neq j^*} Q_{ij}.$$

Jeżeli nie znamy rozkładu prawdopodobieństw p_i określających A, a tylko prawdpodobieństwa P_{ij} określające Γ, to naturalna wydaje się chęć, by minimalizować średnie prawdopodobieństwo błędu po wszystkich możliwych rozkładach A. Okazuje się, że to to samo, co brać optymalną regułę decyzyjną dla równo prawdopodobnych sygnałów z A, czyli innymi słowy jako j^* brać to i, dla którego P_{ij} jest największe. To nazywa się regułą największej wiarygodności.

Wydaje się, że problem został rozstrzygnięty, a cały skomplikowany aparat okazał się bezużyteczny. Ale spójrzmy na już rozważany przykład BSC z $p = 0.9$ i $P = 0.8$. Tam mieliśmy $Q_{00} = 0.973$ i $Q_{01} = 0.692$, czyli niezależnie od tego, co weszło, zwracamy zero. Widać, że powinno dać się zrobić coś lepiej...

Pomysłem, podobnie jak w wypadku kodowania bez szumu, wydaje się być branie dłuższych słów kodowych.

Na rozgrzewkę rozważmy kod powtórzeniowy. Każdy bit przesyłamy trzy razy. Teraz nawet dla powyższego przykładu jeżeli przyszło 111, to odczytamy 1 ($Q_{1,111} \simeq 0.877 > 0.123 \simeq Q_{0,111}$), co daje już pewne szansę na poprawne przekazanie treści.

Jeżeli jesteśmy zainteresowani regułą największej wiarygodności, to otrzymamy tzw. dekodowanie większościowe. Dla danego wyjścia u jako $\Delta(u)$ bierzemy ten znak, który w u powtarza się najwięcej razy.

Oczywiście tę zasadę łatwo uogólnić na większą liczbę powtórzeń. Rozważmy przykładowo BSC z $p = 0.5$ i $P = 0.99$. Wtedy mamy następujące wartości błędu:

Liczba powtórzeń	1	3	5	7	9
P–stwo błędu	10^{-2}	$3 \cdot 10^{-4}$	10^{-5}	$3.5 \cdot 10^{-7}$	1.3×10^{-8}

Wadą tego rozwiązania jest jednak dramatycznie malejąca wydajność — by przesłać 1 bit informacji, potrzebujemy wysłać n bitów.

To, co było istotne w kodzie powtórzeniowym to to, że dwa różne kody nie stają się takie same nawet po kilku błędach (zamianach jednego znaku na drugi). Tę cechę pomoże uchwycić odległość Hamminga, zdefiniowana jako:

$$d(u, v) = |\{i : u_i \neq v_i\}|.$$

Przykładowo, jeżeli Γ =BSC z parametrem $P > \frac{1}{2}$ to dla dowolnych słów u i v prawdopodobieństwo w przód, czyli prawdopodobieństwo otrzymania u, jeśli wyślemy v wynosi

$$P(u|v) = P^n \left(\frac{\bar{P}}{P}\right)^{d(u,v)}.$$

Jako, że $P(u|v)$ jest malejącą funkcją $d(u, v)$, to regułą największej wiarygodności będzie reguła najbliższego sąsiada, czyli zakładanie, że dane słowo u pochodzi od słowa v, które należy do kodu i jest mu najbliższe w sensie odległości Hamminga.

Twierdzenie Shannona: Jeżeli Γ to BSC o paramterze $P > \frac{1}{2}$, zaś δ i ε to dowolne liczby rzeczywiste większe od zera, to dla wystarczająco dużych n istnieje kod, który do zakodowania $(1-\varepsilon)nC(\Gamma)$ bitów używa n bitów, a odczyt następuje błędnie z prawdopodobieństwem mniejszym niż δ.

Bardzo szkic idei dowodu: Jako nasz kod losujemy $2^{nC(1-\varepsilon)}$ słów. Ma on wydajność $(1-\varepsilon)C$, bo używa n bitów do przekazania $nC(1-\varepsilon)$ bitów informacji.

Przy wysyłaniu zazwyczaj (prawie zawsze dla wystarczająco dużych n) zdarzy się nie więcej niż $n\bar{P}(1+\sigma)$ błędów dla dowolnie małego σ.

To, co pozostaje do zrobienia, to oszacowanie prawdopodobieństwa, że jakieś spośród $2^{nC(1-\varepsilon)}$ słów znajduje się bliżej niż $n\bar{P}$ od danego słowa kodowego v, które usiłujemy przesłać. Jeśli tak nie jest, to uda nam się przesłać i odczytać poprawnie słowo v.

Okazuje się, że to prawdopodobieństwo jest wystarczająco małe — jest to lemat kombinatoryczny, którego dowodzić nie będziemy.

Uwagi do twierdzenia Shannona.

1. Twierdzenie to zachodzi też dla innych kanałów niż BSC. Dobierając wystarczająco wysokie n jesteśmy w stanie osiągnąć prawdopodobieństwo błędu mniejsze niż δ oraz wydajność conajmniej $C(\Gamma)(1-\varepsilon)$.
2. Niestety, $n \to \infty$, i jest to problem podobnie jak w wypadku kodowania bez szumu — żeby dostać efektywne kodowanie, trzeba bardzo wiele stracić z własności natychmiastowego odkodowywania.
3. Co najgorsze, jak widać nawet z przedstawionej, bardzo zgrubnej idei dowodu, twierdzenie nie daje nam żadnego pomysłu, jak konstruować dobre kody. Losowy kod bardzo często jest dobry, ale niestety nie ma łatwego sposobu, by to sprawdzić.
4. Co więcej, wolelibyśmy nie mieć kodów losowych, a raczej takie, które mają pewną strukturę ułatwiającą szybkie kodowanie i dekodowanie. O tym, jak sobie radzić z tym problemem (zapewne) będą następne referaty z teorii kodów.