The Second Law and Computer Science

Advanced

- The essence of the second law is entropy, and that is also the essence of part of computer or information science. If we want to grasp the relation between **Entropy** and **Information**, we must first define the precise meaning of information in a mathematical way. For entropy we did that already.
  - The first insight we need to have is that it can't be done as one naively assumes. Let's look at an example to make that very clear. So which one of the following *strings* of *symbols* carries more information?

    1. **YOU PASSED THE EXAM**
    2. **ADE SPUX AMTEY ESHO**

    Well, you're wrong. It is not No.1! It isn't No. 2 either. Both strings contain exactly the same amount of information—if you are a computer doing the judgement. Being a computer, you only go for the **syntax**—the pattern or structure—of a statement. You never go for the **semantics**, the meaning, because for you there is no meaning. The syntax of both strings is the same. Same number of symbols, same symbols. The way they are arranged is meaningless for the kind of information theory we are after.
  - If you think that is pointless, you are wrong. Because both strings contain the same number of **bits** that need to be processed, and that is all computers care for. If you *transmit* those bits on some data channel, they need exactly the same resources. Their susceptibility to getting distorted, mutilated, decoded or whatever, is exactly the same. And it would be the same for all the other strings you could form with the set of symbols given.
    Now look at those two strings:

    1. What?
    2. Q'est-ce que?

    Same *semantics* or meaning here but the French "what" carries more information in the technical sense explained above. Thinking about that a bit you realize that no computer can ever *know* all the various codes there are (e.g. other languages, including those of aliens; Morse code, ..) that contain the same semantics! Syntax is all we can go for.
- Since slime bags like you and me are not computers, we have that habit of looking for some *meaning* in strings of symbols. This is often difficult and always a bit ambiguous. Three lawyers find at least four different meanings in any paragraph of some law - if they wouldn't, we would not need courts of law to decide which meaning is the "true" one, and a lot of people would be out of jobs plus a lot of (different) people would be happier.
  It would be a major break-through in computer science if we could come up with a clear way to teach computers to find meaning in a string of symbols like: "Georgia businessman Herman Cain told the debate audience on the CNBC television network that the United States needs to concentrate on issues at home if it wants to avoid the massive debt that is plaguing Italy." (Voice of America; Nov. 10, 2011; picked one at random).
  - What does that mean? Do the United States have to take over the Italian deficit if they don't concentrate? Is is enough if one state doesn't concentrate or do all of them have to be inattentive? How does a state concentrate anyway? On issues?
    Computers won't get what that could mean; witness the the quick ascend and inglorious end of "artificial intelligence".
- Let's be honest. There is no good definition of what constitutes *meaning* in some string of symbols. Even the definition of what constitutes information isn't so great—we only have a working one that **Shannon** introduced in 1948.
  For that we first look at a group of symbols, e.g. an alphabet with ***N*** symbols (most of them called letters). If one forms *random* strings with the symbols of the alphabet, all symbols will occur with equal average frequency, in stark contrast to the formation of *words* in languages. Some symbols / letters are found far more frequently in "meaningful" words of some language than others. For example, we have a probability of 12.702 % for an "e" occurring in an English word, and only a probability of 0.074 % for a "z".
  - But let's look at *random* strings first. The probability p_i for the occurrence of the symbol No. **i** is thus the same as for all the other symbols and simply given by $p_i = p = 1/N$.
- Now let's ask the big question: How much *information* can be carried in just *one* of the ***N*** different symbols?
  To assess this, we imagine that we we are receiving a string of symbols via some information channel and are waiting for the next one to be transferred. We *define* the amount of information ***I*** contained in that symbol as

$$N = 2^I$$

- The meaning of that equation is that ***I*** gives the *smallest* possible number of "yes" / "no" questions you need to ask in order to find out which symbol it is.
  Imagine your friend has already received that symbol, and you have to find out what it is by questioning her. You don't ask: "Is it an "A"? "No." Is it a "B"? "No", ... You ask: Is it in the first half of the alphabet" "No". "Is it in the first half of the second half"?... Much faster; fewest amounts of questions.
  Rewriting the equation above for ***I*** we get:

$$I = \mathrm{ld}\, N = \mathrm{ld}\,(1/p)$$

- "**ld** " is short for the "logarithm dualis", the logarithm for base **2**.

Simple but rather useless. So let's go for the next step. Let's allow that the probabilities for the occurrence of the symbols can be different, just like in the alphabet of a real language. The information $I(z_i)$ that is contained in the ***i***-th symbol that we now call z_i will then be given by

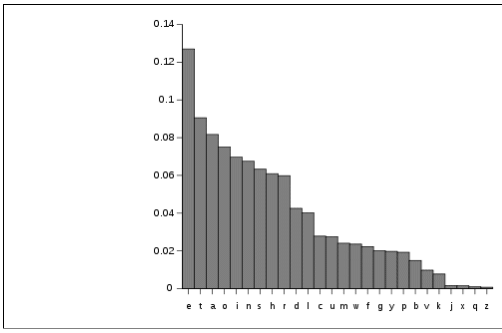
$$I(z_i) = \mathrm{ld}\,(1/p_i) = -\mathrm{ld}\,(p_i)$$

- This has an unexpected consequence: Symbols that only occur *rarely* contain *more* information than the ubiquitous ones. Look up the logarithm module if you can't see that.
  If you think about that, this is as it should be. If you just get another "e", you just know less in comparison to a "z" or "x" or "j" coming down the line. It's harder to guess that one oft the latter will appear, and therefore more surprising.

Seen like that, we can define information in a qualitative way as follows:

The information content in some symbol (or string of symbols) is proportional to the degree of our surprise upon receiving it

Now let's look at the *mean* or average information ***H*** that is contained in an alphabet with ***N*** symbols that is specified by defined p_i values. For example, the English alphabet is characterized as follows:



Probability of letters in English

Source: Wikipedia

- In order to get the *average* information ***H*** contained in strings made with this alphabet, we must sum over the information contained in the symbols multiplied by their probability of occurrence and get

$$H = \sum_{i=1}^{N} p_i \cdot I(p_i) = -\sum_{i=1}^{N} p_i \cdot \mathrm{ld}\,(p_i)$$

- That is Shannon's classical equation (worth a Noble prize). Shannon named the quantity ***H*** "**Entropy** ". That shouldn't come as a surprise anymore. It is quite similar to Boltzmann's formula for the "real" entropy; it contains the same kind of thinking.
  - Shannon's entropy ***H*** is important. For example, if you want to transfer a message with ***Z*** symbols in the chosen alphabet, you need at least ***H · Z*** digital **bits** for that. In other words, Shannon's equation gives the "*bandwidth* " that a channel must have and that is, of course, of overwhelming importance to real signal transmission.
- So what are the relations between Shannon's information entropy and the thermodynamic Boltzmann entropy? Are they identical?
  - Not exactly. But they are close enough. ***H*** is different from the thermodynamic entropy (always written as ***S***) in two minor points:
    - The proportionality constants are different. But that is truly trivial.
    - The proper thermodynamic entropy ***S*** is only well-defined for equilibrium ("nirvana"). It corresponds directly to the *maximal* ***H*** that we would get if all symbols have equal probability.
- If one takes that into account and does proper work, a quite interesting relation for the thermodynamic entropy ***S*** contained in *1 bit of information* emerges:

$$S(1\ \mathrm{bit}) = -k \cdot \ln 2$$

  - This means that an entropy increase of $\mathbf{0{,}957 \cdot 10^{-23}\ JK^{-1}}$ in a given system destroys exactly 1 bit of information. Since entropy in a closed system can never spontaneously decrease, information cannot spontaneously come into being.
    That's not just an "esoteric" theory. This solved an old and deep puzzle of thermodynamics. **Leo Szilard** was the first one who used this insight to exorcise "**Maxwell's demon** forever from the world of science. Google it yourself!
- Whatever way you look at it, there is some connection between information theory and thermodynamics. Will computer science, information science, and so on, one day become just another subgroup of physics, like chemistry (or possibly biology soon)?
  - Nobody knows for sure. Some famous scientists like Peter **Atkins** contain that this is all a crock of ~~sh~~ nonsense. Some equally famous scientists like Roger **Penrose** , however, believe that here we have the key for the new physics of the future.
    So non-famous guys like you and me can only wait and see, preferably while relaxing (for having peace and quiet you may need to shut down some information channels; send her shopping) and having a beer or two.