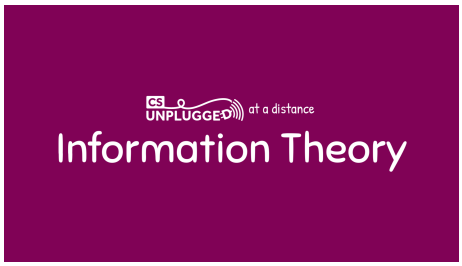
Information Theory (Speaker Notes)

Slide 1

No speaker notes for this slide.



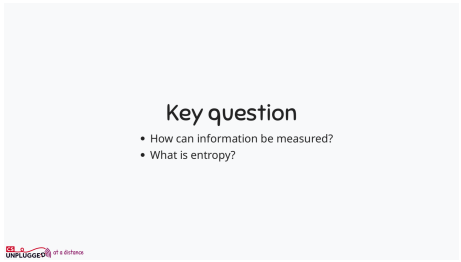
Slide 2

By the end of this session we'll be able to answer these questions:

- How can information be measured?
- What is entropy?

We are going to play a game to demonstrate one way of working out how much information a file holds. We know that information can be compressed, but do we know what the limitations are for each file? Computer Scientists consider the limitations and why that is. This helps us to understand what computers are capable of and what is impossible for them to do. This activity also demonstrates how important mathematics, in particular probability, is.

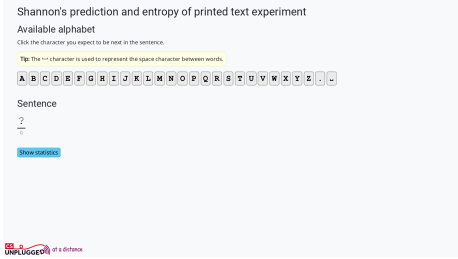
Let's look at the experiment that Claude Shannon used to explore this.



Slide 3

The method I'm going to share today was created by Claude Shannon, an American mathematician who published a paper on this theory in 1951. Siobhan Roberts wrote in The New Yorker magazine in 2016 “A hundred years after his birth, Claude Shannon's fingerprints are on every electronic device we own.” In this session we are going to see just how true this is.

Slide 4



We will be using this interactive to help us predict what the sentence is. Let's start! I've pre-set a sentence. You, as a group, are going to guess what my sentence is, letter by letter. What do you think the first letter is? Put your choice in the chat and I will enter the most popular one into the interactive. If we guess correctly, the letter will show in black. If we guess incorrectly it will show in red. I see several people have suggested __. Let's try it.

Click in the interactive on the character suggested most in the chat and choose the appropriate response:

It's black, we were right. What do you think the next letter is?

or

It's red. You can see a count has been added under the letter. We'll see why this is happening later. Let's have another guess.

The count under each letter shows the number of guesses it's taken to get the letter correct.

Continue guessing letters until the letter turns black. The number of guesses will automatically be recorded and the letter greyed out of the list when you enter it.

It is important to respond to guesses only by letting the audience know if they are correct or incorrect so that you don't give away any further information to help with the guessing. For example, don't say “S, you are close with that one”. Also, don't allow questions such as 'is it a vowel?' or 'is it before M in the alphabet?' Each guess must be for a specific letter.

Continue with participants guessing what they think the next letter is as you enter their choices into the interactive. If time permits continue for the whole sentence. If time is short, have them guess at least four words.

Suggested script when whole sentence has been deciphered:

> Well done! That took some perseverance!

Suggested script when stopping after the fourth word has been deciphered:

> To keep our session shorter I'm going to stop us here. If we had kept going we would have deciphered it to read `THERE IS NO REVERSE ON A MOTORCYCLE.`, but we have done enough now to illustrate the concept.

Suggested script for both options:

> Each guess can be represented as one “bit”. “Bit” is the technical term for the smallest piece of information a digital device can hold. 8 bits are called a “byte”. You have probably heard this term in relation to computers, particularly in the context of mega- and gigabytes of storage.

> Each bit can represent one of two states, in this case, either correct or incorrect. We can represent each correct answer as a 1 (one), and each incorrect answer as a 0 (zero). In our example, the first letter took [say how many guesses it took your group to find the correct letter, this is the number showing under the letter] bits of information to guess as each guess used one bit. If we measure it this way, effectively the first letter cost me [however many guesses it took them to guess it] bits to transfer the information to you.

> From this exercise we can see that some of the characters in a sentence took less bits to predict than others. Computer scientists use this information to determine how much information is needed to send a message accurately.

Highlight a letter that took more than one guess to get right.

We can see the __ [character] cost [the number of guesses it took to get it correct, the number showing under the letter] bits because we guessed incorrectly [one less than the total number of guesses] times before we got it right.

When we measure the information content of something in this manner it is based on how easy it is to predict what the content is going to be. This activity gave an idea of how many bits you might need to represent a sentence if you understand the English language as English speaking humans. If we count up how many guesses it took us as a group to guess this sentence we then have a number, a measurement, we can use.

Point to a letter that took a lot of guesses and seemed to take a long time to get correct.

Even this one that seemed to take a long time, still only took [number of guesses] bits.

In addition, if we look at the statistics we can see that we guessed __ [number shown in the left hand column] characters on the first guess.

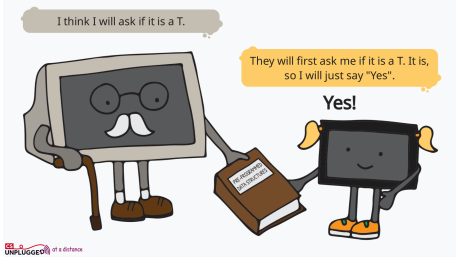
If you click on the "Show statistics" button it will open up two graphs to assist in understanding the data.

Often it is suggested that we can measure information on a computer by the size of the file. This is a good idea, but what if the file is a three minute recording of a blank, silent screen? Does this have as much information as a three minute instructional video on how to cook a roast meal? It may take the same number of megabytes to store, but I think you will agree it doesn't hold as much information.

Slide 5

So how does this fit with what computers do? Let's take the game we just played. If two devices are running the same prediction software and play this game, then they don't need to say the text to find the same sentence. Think about it this way: if I could read your mind and know what order you were going to guess letters in, then I could tell you “yes, no, no, yes, no, no, no yes...” until you guessed the right letter for each space. If I know what letters you are going to guess then you don't need to actually tell me those letters. This principle is what makes data compression possible (data compression is reducing the size of a file, such as using zip files.)

I have a file (information) I want to send to you from my device to your device. Because we are both running the same program, which tells us the most common letter at the start is T, my device knows T is the letter your device is going to guess first. So my device only needs to send “yes” to your device, in answer to the unasked question (“Is it T?”). Then our program would know your device is going to guess H next, so I say “yes” again. I can say “yes, yes” to you, and you will know that I mean “Th”.



Slide 6

Data compression relies on prediction. If you've looked at binary representation at all you might know that it typically takes 7 or 8 bits to store a single character. Shannon's experiment pointed towards being able to use fewer bits on average (compared to binary representation) to represent characters, if they are predictable. It also showed that random text, like "aZfoUs" will need the full number of bits since nothing is predictable.

The idea of having two people that think exactly the same is totally feasible with computers. You just use the same program on both computers. Data compression software uses variations of this idea. If you use a zip program to compress a file on your computer, and I use the same program on my computer to decompress it, the program is doing exactly the same thing to the same information. Or, if you use JPEG it follows exactly the same rules for how it is going to code and decode pictures, wherever that program is used.

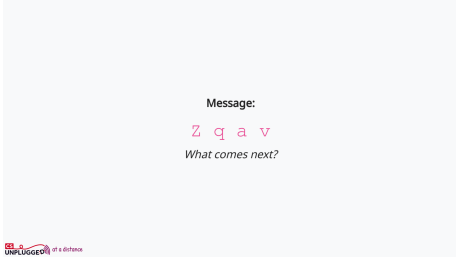
Slide 7

Shannon was really just interested in how much information content text has.

If I wanted to send you the message "aaaaaa", while it would probably take you a few guesses to work out what the first letter was, and probably several more to guess what the second letter was, it is likely you would guess 'a' fairly quickly for the third letter, and instantly for the ones after that. As this message has very little surprise value, it has low information content. After the first two letters, only 1 bit is needed to confirm every character.

Slide 8

If my message was completely random, such as "Z, q, a, v", it would cost on average 13 guesses (half the letters of the English alphabet) for every character. Each letter is a surprise and, as you have no way of predicting random information, there is no model for what might be next. So, that message is very surprising, it has a very high information content.

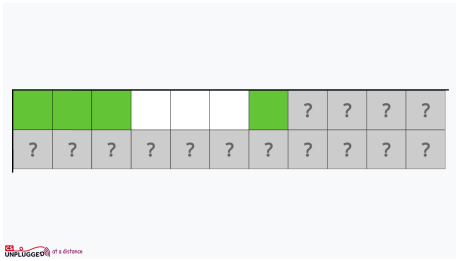


Slide 9

If my message had been, "a, b, c, a, b, c, a, b, c" it would also have been easy for you to predict, after the first 4 letters, and it can be transmitted efficiently.

Slide 10

Let's try it with something different. If I was describing this image as "green, green, green, white, white, white, green" what colour would you guess the next pixel is? Share your guesses in the chat.

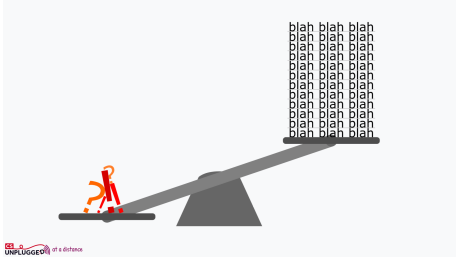


What for a few responses in the chat.

What is the probability we guessed the correct colour? 50%? What assumptions have we made? That there is only green and white as options to select from.

You would probably say “green” as the pattern doesn't have single coloured pixels standing alone. Or you might have said white. But you probably wouldn't guess red. This is an example of how images can be predictable - we can get a good idea of what a pixel's colour is likely to be based on the ones before it; as with guessing letters, we can't be sure, but if some things are more likely than others, there's an opportunity to represent it more efficiently than if all values were unpredictable.

Slide 11



So, how does this relate to what is happening inside the computer?

This experiment is used by computer scientists to measure the amount of "information" in a document. Being able to have a good guess at what is coming next is the basis of working out how to reduce the amount of data used for sending text, photos and videos. This idea is called 'compression', and is the underlying theory behind showing videos over the internet and storing large amounts of photos or songs on a mobile phone.

Understanding the predictability of text enables computer scientists to design good compression algorithms, and estimate how much compression files can take.

Predicting what will come next is also used for auto-completing text while you are typing. The computer is guessing what is most likely to come next based on what people usually type. This can save people time entering text, especially on an awkward interface like a phone or TV remote.

A technical term that comes up a lot around information theory is 'Entropy'. It's a word that is borrowed from thermodynamics, which loosely means the amount of disorder in a system. This term is used by computer scientists in information theory to describe the amount of information a file contains. The more information a file contains, the less predictable it is, and the more entropy, or disorder, it has. If data is very predictable (like the colours of the pixels in an image that is completely black) then it has a low information content, and the entropy is low.

As I mentioned earlier, the amount of information in a file can be measured by its 'surprise' factor. If the probability of a statement is 100%, then there is no new information in that file. For example, if you are standing with someone in the rain and they say “it is raining”, there is no new information in that statement, you did not learn anything new from them having said it. If they say an impossible statement, for example, “I am 1 month old”

(not many 1-month-olds can talk), then that sentence has a probability of 0%, it contains an infinite amount of information (it raises many questions: what does the person mean they are 'one month old', have you completely underestimated the ability of 1-month-old humans, is that 'person' human?...).

If the probability of the statement being true is 50% then that information uses 1 bit of computer memory to represent it. It would be like our previous example of green and white and asking if the next colour will be white. If the probability of the statement being true is 100% then it uses no bits to represent it as the computer already knows it. If probability is 0% then it theoretically uses an infinite number of bits; in other words, it can't be represented.

These ideas might just seem to be of theoretical interest, but the experiment we've done really does help to answer the question of how small a file can be compressed down to. In fact, Shannon's work has enabled people to spot some frauds over the years, when people tried to get investors for inventions that claimed to be able to compress files much more than the limits of Shannon's experiments. (For example, if you google “compression” with “Phil Whitley” or “Madison Priest” you'll see how millions of dollars have been lost by people who didn't know about Shannon's limits.)

In summary, Shannon's theory, which we have learnt about here, is fundamental to compression, encryption, error correction, and so on because it's all about how many bits we need to represent something.

Going back to our original questions: How can information be measured? As we have shown, it can be measured by its surprise value, by how many questions have to be asked to get the full information. What is entropy? Entropy is the amount of disorder in a system or file. The more disorder a file has, the less predictable it is, the more entropy it has.

Slide 12

Here are some supporting resources for you. I'll paste these links in the chat.

```
Original CS Unplugged activity
https://classic.csunplugged.org/activities/information-
theory/

Computer Science Field Guide: Shannon interactive
https://www.csfieldguide.org.nz/en/interactives/shannon-
experiment/

CS Unplugged at home: Guess the sentence
https://www.csunplugged.org/en/at-home/guess-the-
sentence/
```