Lecture 1: Introduction

[Disclaimer: These informal lecture notes are not intended to be comprehensive - there are some additional ideas in the lectures and lecture slides, textbook, tutorial materials etc. As always, the lectures themselves are the best guide for what is and is not examinable content. However, I hope they are useful in picking out the core content in each lecture.]

Definitions

The lecture began with a slightly long-winded way of introducing these two "definitions" for perception and cognition.

- **Perception** is concerned with how the mind interprets sensory data
- **Cognition** is concerned with knowledge, reasoning, memory, language, decision making, etc

To illustrate this definition, we used some empirical data (taken from De Deyne, Navarro & Storms 2013) to map out the words that people associate with "perception" and "cognition". The reason for doing this was partly because it's kind of neat to see the "definition" emerge from commonsense intuition (e.g., the word "perception" is associated with a cluster of words about "sensation" and another cluster of words about "mind"), but it's also a bit of foreshadowing - this kind of *semantic network* will appear later on in the cognition lecture stream in Prof. Taft's section.

Historical background

- Wilhelm Wundt founded the first experimental psychology lab in 1897 at the University of Leipzig. Used the method of **introspection** (subjective observation of one's own experiences) to investigate cognition

- In the early 20th century there was a backlash against this, mostly in the USA, leading to the rise of **behaviourism**, a school of thought that (led by researchers like John Watson and B.F. Skinner) disavowed any description of "mental processes" and insisted that psychology focus solely on describing the relationships between "objective" quanitites: stimuli and responses. Quote from J.B. Watson:

> “Psychology as the behaviorist views it is a purely objective experimental branch of natural science. Its theoretical goal is the prediction and control of behavior. Introspection forms no essential part of its methods, nor is the scientific value of its data dependent upon the readiness with which they lend themselves to interpretation in terms of consciousness. The behaviorist, in his efforts to get a unitary scheme of animal response, recognizes no dividing line between man and brute.”

- The behaviourist approach had many successes (beyond the scope of this class) but for our purposes the important point is to notice that there are two quite distinct claims that the behaviourists relied upon...

  1. **Methodological behaviourism** - psychological theories should be based solely on objective empirical data and not upon subjective, personal intuitions. Our goal is the successful prediction and description of *observed behaviour*. In modern times this seems a bit limiting - now that we have other technologies such as EEG, fMRI, etc, it does seem a little odd to restrict ourselves to behavioural data, but the spirit of methodological behaviorism is really that our theories must be based on objective, empirically observable data.
  2. **Theoretical behavioursm** (also referred to as radical behaviourism) makes a much stronger claim. It asserts that human behaviour is best understood in terms of "stimulus-response associations". That is, the reason why we should restrict ourselves to describing S-R associations is that these really are the fundamental building blocks of human behaviour. To the radical behaviourist, cognition, thoughts, feelings beliefs, etc, these are all fundamentally unscientific concepts and should form no part of psychology.

  These days, almost everyone accepts some (suitably extended) version of methodological behaviourism. Very few people still subscribe to radical behaviourism.

Some difficulties with behaviourism

One of the most influential criticisms of the radical behaviourist position occurred in the late 1950s when B.F. Skinner published *Verbal Behavior* in 1957. The key idea in the book was an attempt to describe how language might be "trained" via a process of operant conditioning. Leaving the particulars of Skinner's theory to one side, the main thing to note here is that he treated language as being no different to any other behaviour, a view that didn't go down well in linguistics. Noam Chomsky, who had published his own book *Syntactic Structures* in 1957 (which emphasised the importance of understanding the extraordinary amound of grammatical structure in human language), published a scathing review of *Verbal Behaviour* in 1959. There's quite a lot of ideas in those two publications by Chomsky, but several key points that he made were:

- children often blatantly ignore some kinds of parental "reinforcement" (e.g., correcting kids grammar rarely works)
- children understand the meanings of sentences they have never heard
- people can understand that completely new sentences can be grammatical while being totally nonsense (e.g., "colourless green ideas sleep furiously" is a novel sentence that is immediately recognisable as being meaningless and yet grammatical)

The overall message underpinning these kinds of observations is that human language (and by exension, perhaps, many other phenomena in human behaviour) is highly structured, and is not completely under the control of "the environment". Instead of thinking about the production of behaviour a "simple" mapping from *stimulus* to *response*, perhaps we should be thinking about what happens in betweebn the S and the R. Perhaps... we need to understand the thought processes that lie in between the two!

Cognitivism and the "computational metaphor"

To contrast the two positions:

- Radical behaviourism: the contents of mind are unobservable, and therefore off-limits to science
- Cognitivism: the mind is a kind of unknown biological machine, and scientists may propose theories about its structure

When we shift from a behaviourist to cognitivist perspective, the big question that we need to answer is "what kind of machine are we talking about here?" Our brains (and by extension, our minds) are terribly complicated things, and to make any progress in describing them in comprehensible ways, we'll need to rely on some simplifications, and maybe come up with a sensible "language" for describing what they're doing.

With that in mind, much of the cognitive literature relies on something known as the **computational metaphor**. The key idea is to analyse the behaviour of the mind by considering the "information processing steps" it goes through in order to produce a response to a particular stimulus. From the lecture:

- “Information processing”: The computational metaphor suggests that we can use the language of “computing” to build “models”

- Why do this?

  - Computational language is precise… we can generate empirically testable predictions (see: methodological behaviourism!)
  - Computational language is flexible... so we can postulate hidden mechanisms and structure to cognitive processes (goes beyond radical behaviourism)

[A toy example] In the lecture we gave a very simple example of an information processing theory, one that proposes that people do "mental multiplication" by repeatedly adding numbers together (e.g., 7 * 4 is solved by doing 7+7=14, then doing 14+14=28). Mental arithmetic is actually much more complicated than this, but the point we wanted to make is that although this theory is inconsistent with **radical behaviourism** (because it speculates about the internal mental processes that the mind uses to produce a response "28" to a stimulus "7 * 4"), it is consistent with **methodological behaviourism** because it yields testable empirical predictions. For instance, it predicts that people should respond faster to 7 * 2 than to 7 * 4, because it requires fewer steps. Etc.

Methodological examples

There are many different experimental methods used to investigate cognitive processes. Some common examples include:

- Accuracy of response
- Type of response
- Response time
- Neurological deficit
- Brain imaging
- Self-report

The lecture slides have a bit more detail on this.

Beware "the laptop fallacy"

Because cognitive theories tend to be "information processing" theories, and digital computers are a kind of "information processing machine", there's a bit of a tendency for people to interpret the computational metaphor in an overly literal way.

In lecture, I referred to this description by Robert Epstein in 2016:

> “Computers, quite literally, process information – numbers, letters, words, formulas, images. The information first has to be encoded into a format computers can use, which means patterns of ones and zeroes (‘bits’) organised into small chunks (‘bytes’). On my computer, each byte contains 8 bits, and a certain pattern of those bits stands for the letter d, another for the letter o, and another for the letter g. Side by side, those three bytes form the word dog.”

This is a very heavily simplified description of how modern digital computers encode the text "dog". However, there is a very extensive literature on how humans represent words, word spellings and word meanings that I won't go into here, but the short version of it is that the mind doesn't encode words the same way a computer does. On the basis of this, Epstein argues that the entirety of cognitive science is built on a terrible foundation:

> “Our shoddy thinking about the brain has deep historical roots, but the invention of computers in the 1940s got us especially confused. For more than half a century now, psychologists, linguists, neuroscientists and other experts on human behaviour have been asserting that the human brain works like a computer.”

In other words, he argues that the computational metaphor is wrong because (among other things) the specific way in which a digital computer encodes "dog" is different from the way that the mind encodes "dog". Is he right? I don't believe so, and I think Epstein is falling for something that I refer to as "the laptop fallacy".

To understand what's going on here, it's important to understand the distinction between the *theoretical idea of a computing machine*, and the very specific instantiation that your laptop embodies. In the abstract, a computing machine is simply anything that is able to process information in a sufficiently "complicated" way. There's a whole branch of mathematics that is devoted to making that idea precise, but for our purposes let's just note that there's such a thing as a "Turing machine", an abstract notion of "a machine that can do computations". You can build Turing machines out of lego, you can build them out of biologically inspired networks, you can even implement them as a tiling scheme in your bathroom floor (see lecture slides for pictures). Superficially, none of these "machines" are particularly similar to each other, and their real world behaviour can be very, very different to one another. But they *are* all Turing machines, they all process information, and they are all doing computation.

There's a serious point to all this:

- the computational metaphor doesn't mean "the brain is a lot like a laptop", because that's an absurd proposition and is easily falsified.
- what it does mean is, "we can use the language of computer science and information theory to build good theories of human cognition".

Marr's levels

All of that being said, there's a bit of a tension within the cognitive approach, between those who emphasise a "bottom up" view and those who emphasise a "top down" view:

- "Bottom up": Cognition is a thing performed by the brain, and our theories of human cognition should informed by the biology of the brain. People who take this perspective emphasise the biological bases of behaviour.
- "Top down": Cognition is a feature of intelligent agents, and our theories of cognition should be informed by understanding what intelligent agents do. People who take this perspective emphasise the connections between cognitive psychology and artificial intelligence.

The tension exists, but it's mostly amicable: I belong to the latter group, but many of the neuro folks around UNSW belong to the former. We get along pretty well.

One way of capturing the tension is via David Marr's (1980) three "levels of explanation"

1. **Abstract computation**: What problem does cognition solve?
2. **Algorithm**: What processing steps does it follow to do so?
3. **Implementation**: How is this instantiated as a physical entity?

The lecture slides have a bit more detail on this.