Simulating Language
Lecture 1: introduction

Simon Kirby

simon@ling.ed.ac.uk

What have computers got to do with linguistics?

- For several decades, a lot of money has been put into building better interfaces to computers that use natural language

- Significant investment into:
    - Speech synthesis
    - Speech recognition
    - Dialogue generation
    - Natural language understanding
    - Machine translation
    - etc.

Other uses of computing in linguistics

- All these areas of research are topics from engineering
- The problem is to build better machines
  - Language just happens to have something to do with it
  - Linguistics may help, but not necessarily
  - E.g. much speech recognition now largely to do with statistical analysis
- Can computers “give something back” to linguistics?

Computers as research tools

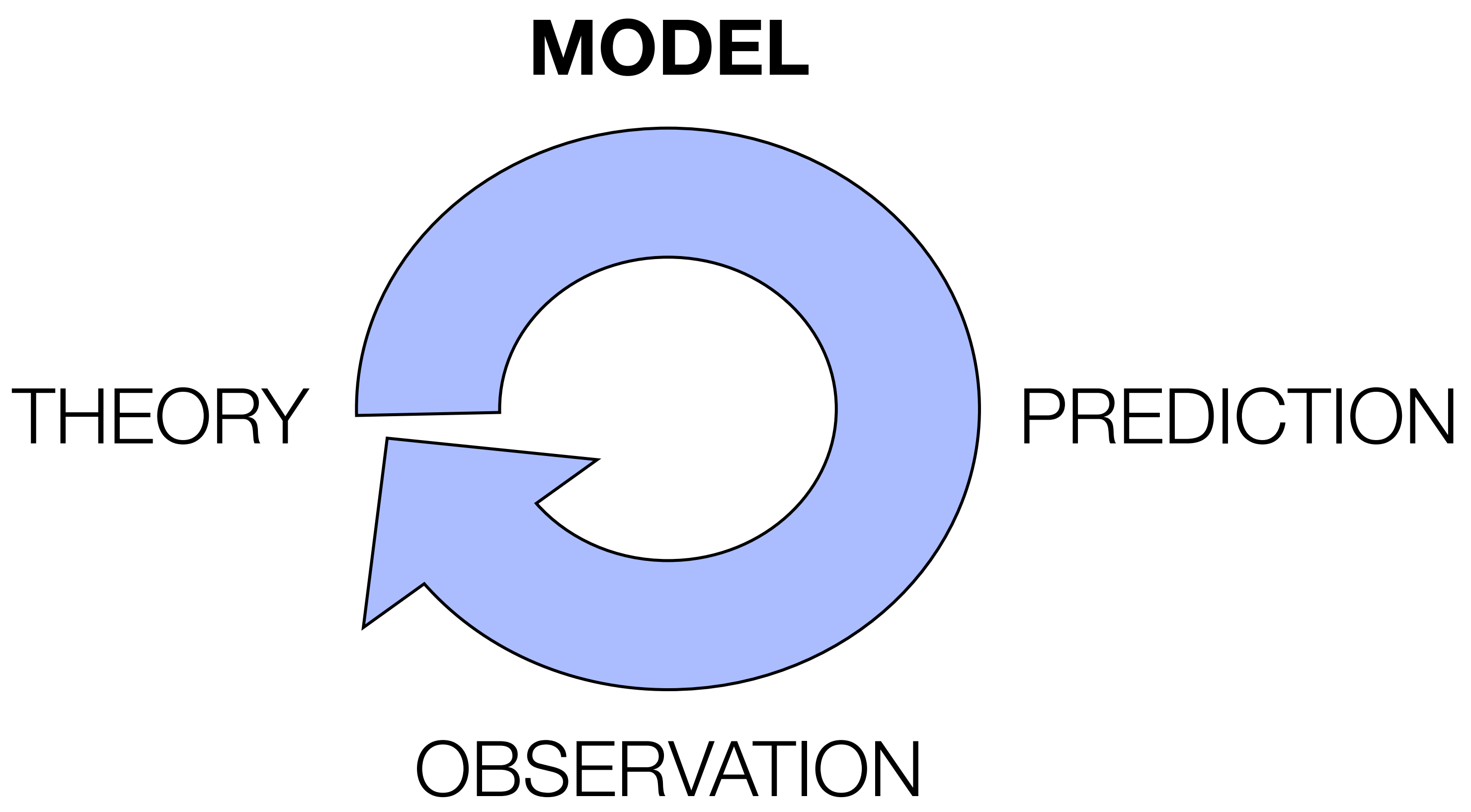
- Aside from using linguistics to build better computer interfaces, we use computers as tools for linguistics
- For example:
    - Speech analysis (e.g. spectrograms)
    - Psycholinguistics (e.g. resynthesis of speech, displaying stimuli)
    - Corpus analysis (e.g. counting words, discovering patterns)

Computers as platforms for *modelling*

- This course is *not* about engineering, or research toolkits
- Instead, we will be looking at **model building**
- Key questions:
  - Why would we want to build models in linguistics?
  - Why would computers help?
  - What is a model anyway?

What is a model?

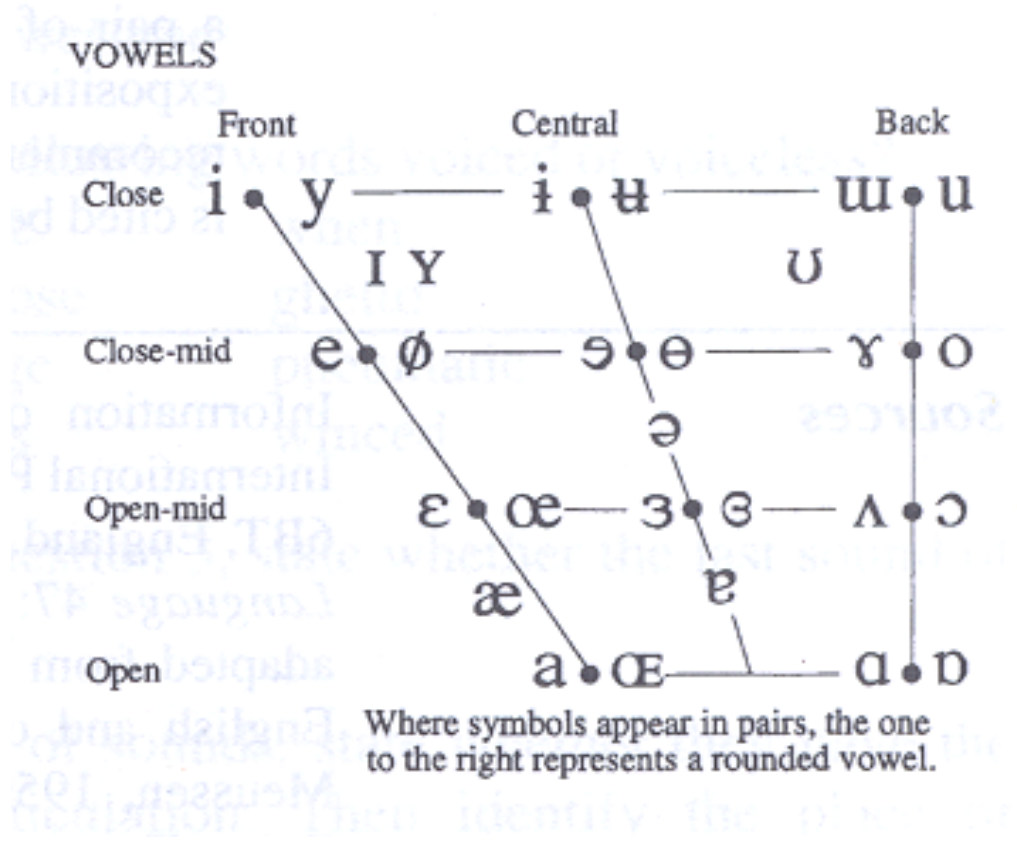
- One view:



- We use models when we can’t be sure what our theories predict
- Especially useful when dealing with *complex systems*

A simple example

- Vowels exist in a “space”



- Only some patterns arise cross linguistically.
  e.g. Vowel space seems to be symmetrically filled.

  Why?

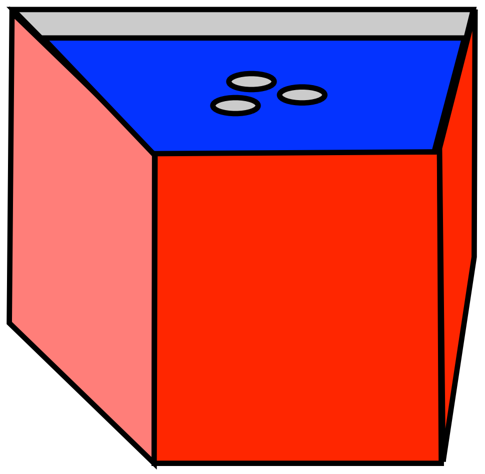
The need for a theory

- Why might such universal patterns exist?
- This is where we need a theory
- Possible theory:
    - **Vowels tend to avoid being close to each other in order to maintain perceptual distinctiveness.**
- How do we tell if our theory is correct?

The need for a model

- We need some way of generating predictions from the theory which can be compared with the real data.

- A physical model (Liljencrants and Lindblom 1972): a tank of water, some corks with magnets attached.



- From this model, predicted patterns can be compared with cross-linguistic data

How close to reality should this model be?

- One view is that the model should leave nothing out
- But is it sensible to build the real thing in miniature?
- Will we actually learn anything from this?
- If not, then where do we draw the line?

- Build your model to have as little extra in it that isn’t part of your theory.

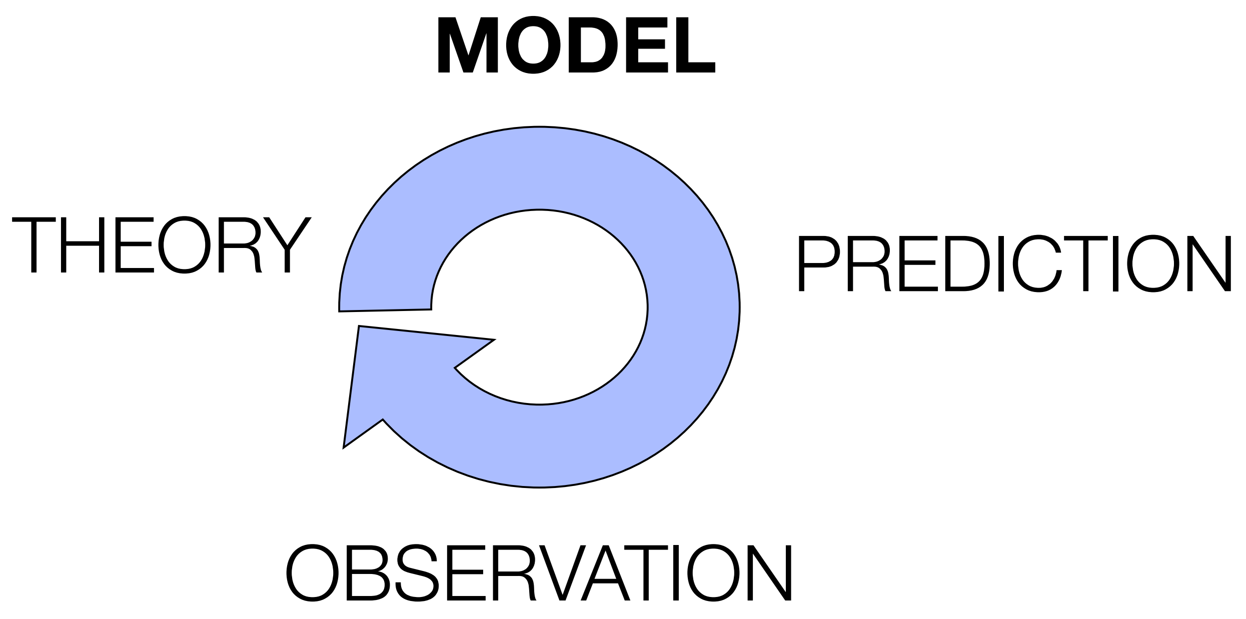
Why use computers for modelling?

- What if:
  - your theory is too difficult to understand simply through verbal argument, or introspection?
  - or a physical model cannot be constructed simply?
  - or a mathematical model is too difficult (or impossible) to construct?
- Particularly difficult problems involve dynamic interactions. For example:
  - a child's knowledge changing as she responds to hearing thousands of words
  - people interacting in groups over thousands of years
  - communicating organisms evolving over millenia

Computational modelling is the solution

- Computers are very good for models of many interacting components
- This has proved particularly valuable in allowing us to build a fundamentally *evolutionary* approach to understanding linguistics
- In this course, we will be building and playing with models to tackle questions like:
    - How do innate signalling systems evolve?
    - How are vocabularies shaped by cultural evolution?
    - How can we learn meaning in the face of uncertainty?
    - Where do grammatical generalisations come from?
    - What do we mean when we say language is innate?

Revisited: What is a model?



- An alternative / complementary approach: models as tools for understanding

"Predictions are not the pinnacle of science. They are useful, especially for falsifying theories. However, predicting can't be a model's *only* purpose. ... surely the *insights* offered by a model are at least as important as its *predictions*: they help in understanding things by playing with them." (Sigmund, 1995, *Games of Life*, p. 4)

This is a **practical** course

- We will be spending a lot of our time in the lab, working with simulations

- You do not need to know how to program, but you do need not to be scared of computers, and willing to try things out

- We will be working in a simplified subset of **Python**

  We will supply the code for the practicals, but you will need to modify it to carry out the tasks on the worksheets.

  This isn't a programming course: we aren't going to teach you how to program, but we will teach you just enough to understand and use some simple models we provide. You will have to meet us half way: you'll get on much better if you get your hands dirty and try things out.

Outline of the course

- About half labs, half lectures. Details on WebCT.
- Labs are the primary source of feedback throughout the course. Lots of opportunity for one-to-one help and discussion.
- Rough order of topics:

  Innate vocabularies
  Modelling populations
  Evolution
  Learning
  Cultural transmission
  Cross-situational learning
  Innateness and generalisation
  Co-evolution of genes and culture

Lab classes: 2pm, 3pm or 4pm

- Labs take place in DSB 1.16
- 18 PCs
- We need to split up into three groups!
- Please mark your preferences on the forms, but note that we really need to spread the class over the three sessions as evenly as possible
- You need to always go to the same slot irrespective of each day (otherwise it’ll be just too confusing!)

Readings

- Readings required, specified on WebCT and at the end of lectures
    - No readings today
    - On average, 1 journal article per lecture
    - PDFs of all readings on WebCT

Assessment

- HONOURS STUDENTS
    - 3,500 word essay or project write-up on topic selected from list of approved topics.
- POSTGRADUATE STUDENTS
    - 4,000 word essay or project write-up on topic selected from list of approved topics or agreed with me.
- DEADLINE (FOR ALL): 26th April

Next up

- Thursday: your first lab
    - Python basics: running python, variables, lists, conditionals, loops, functions, ...
- Friday: lecture on signalling
- Finally, my top tip for this course:

Don’t miss any lectures or labs! It’ll be very tricky to keep up if you skip any sessions because everything builds on the previous step.