Labs: an introduction

Peter Bell

Automatic Speech Recognition— ASR Lecture 5
31 January 2021

Labs

Lab times

- Labs start this week – in person!
- Times:
  - Lab 01: Tuesdays 13.10 (Zeyu)
  - Lab 02: Wednesdays 10.00 (Jie)
  - Lab 02: Wednesdays 13.10 (Ramon)
  - Lab 03: Fridays 13.10 (Electra)
- Attend one lab per week
- You should attend the same lab as your partner
- Contact Timetabling if you need to swap.




How the labs will work

- Work on the exercises with a lab partner
- Ask the demonstrator for help whenever you need it
- If working outside scheduled lab times, ask for help on Piazza
- If you need to find a partner feel free to post on Piazza or post privately and we will try to pair you up
- It's normal to continue with the same partner for all labs and the coursework, but not obligatory
- Attendance will be taken by the demonstrator

Technical setup

Requirements

- Follow instructions at https://piazza.com/class/ky4bimyglss6tm?cid=10
- Ask for technical support on Piazza ahead of the labs
- The exercises have been tested on the Lab PCs.
- Instructions are available for running on your own machine via Remote Desktop (XRDP) or using SSH tunnels
- Setting up directly on your own computer is possible but not recommended

Lab exercises

- Lab exercises and solutions available from a GitHub repo: `https://github.com/ZhaoZeyu1995/asr_labs.git`
- Skeleton code is provided in a Jupyter notebook. You will write and run your code within a the notebook
  - there is also a Python template if you prefer to work with a different editor
- One notebook for each lab (except labs 3 and 4 share a notebook).
- Solutions will be available one week after the lab
- You will need to update the repo to receive subsequent weeks' exercises and the solutions

Asessment

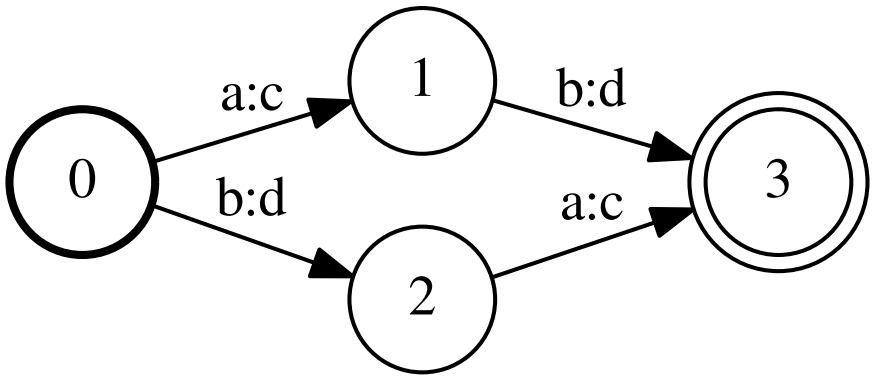
Lab solutions are worth 10% of the coursemark

- 2 marks available for each lab
  - No submission – 0 marks
  - A weak/partial submission – 1 mark
  - A good effort – 2 marks
- Marks over the first five labs will be aggregated and scaled in accordance with the University's common marking scheme
- You will need to submit your solutions by Monday morning in the *second week* following the lab
- Only one person from each pair need submit a solution – indicate your partner in the code
- Submission will use CodeGrade, linked from the Learn page

HMMs and WFSTs

- The labs will use OpenFst `http://openfst.org` to build and manipulate HMMs represented as weighted finite-state transducers (WFSTs)
- You will first build WFSTs for various phone and word structures, compute forward probabilities and implement your own Viterbi decoder
- WFST consist of states and directed arcs between them
- Each arc has an input label, and output label and (optionally) a weight (or cost)
- Labels can be blank (epsilon): written ε or `<eps>`
- The WFST *transduces* a sequence of input labels to a sequence of output labels (and optionally applies a weight to this)

WFST example

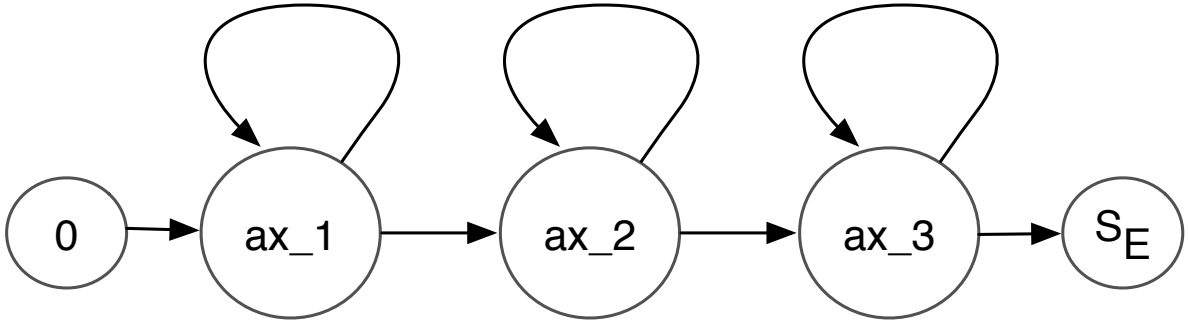


A very simple transducer mapping the string "ab" to "cd" and the string "ba" to "dc". The initial state is shown in bold; the final state is shown with a double circle.

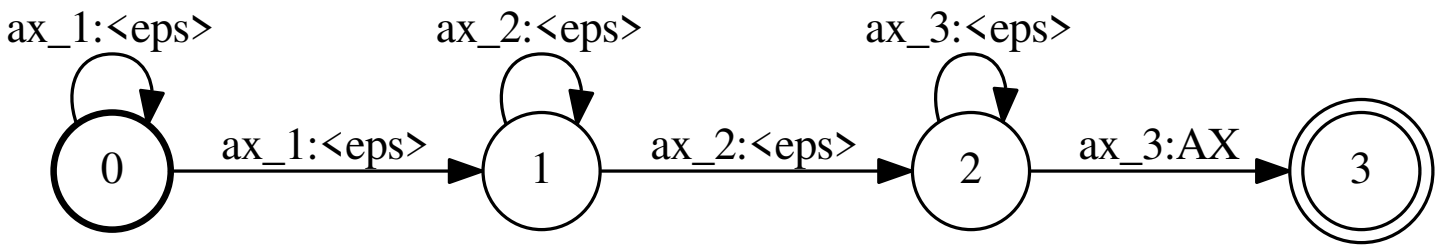
HMMs as WFSTs

- Easier to think of the HMM emitting states as being on the arcs
- Input labels used to denote state ID
- Output labels can be used to encode symbols to be output by the recogniser, such as words or phones
- We will cover WFSTs in much more detail later in the course

HMMs as WFSTs



Conventional HMM for phone "AX" with three emitting states



WFST representation of the HMM. Note the output label "AX"