ACS Statistical Machine Translation

Lecture 7: Phase-based Translation with WFSTs

Department of Engineering
University of Cambridge

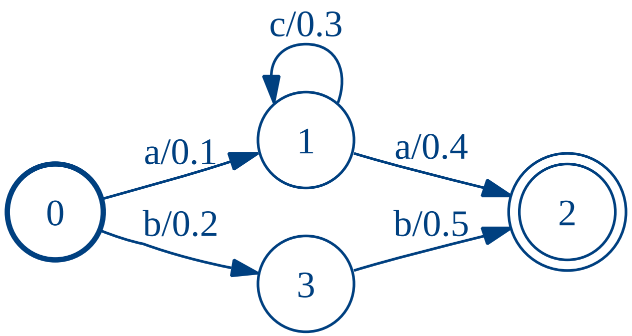
Bill Byrne – bill.byrne@eng.cam.ac.uk

Lent 2013

Introduction to WFSTs (review)

- General framework of structures/algorithms that are useful to encode and process conditional probability distributions
- Well-suited to carry out search procedures involving Markov processes and HMMs
- If a problem is cast in a WFSA framework, efficient standard algorithms can be applied directly

- Lecture 6: we saw **Weighted Acceptors**, which can implement **N-gram language models**
- Today we will see how a **full phrase-based translation system** can be implemented with Acceptors, Transducers and standard WFSA operations



Example of Weighted Acceptor

WFSA Operations

Basic operations can be performed over WFSAs

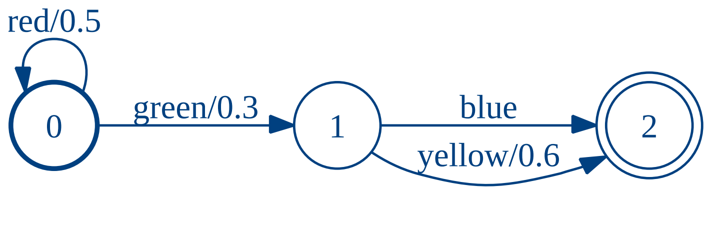
Some operations correspond to operations on the languages defined by WFSAs :
- Intersection
- Union
- Concatenation (or Product)
- ...

Other operations correspond to operations on the WFSA itself :
- Determinization
- Minimization
- Shortest distance calculations
- Pushing weights
- ...

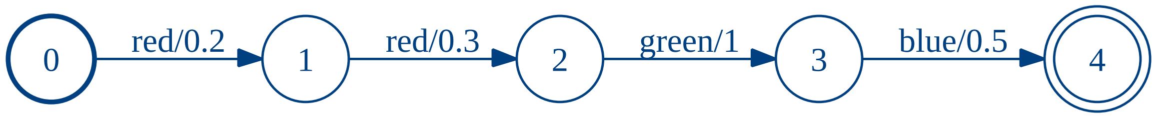
WFSA Operations - Intersection

A string x is accepted by $C = A \cap B$ if x is accepted by A and by B

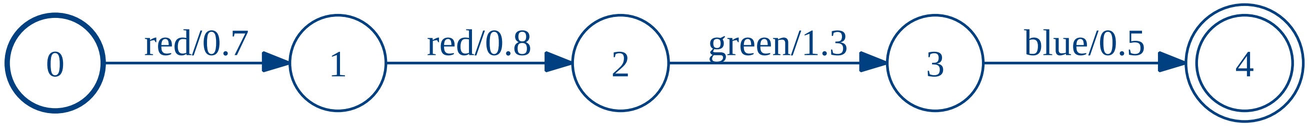
$$[\![C]\!](x) = [\![A]\!](x) \otimes [\![B]\!](x)$$



A



B



$C = A \cap B$

In this example $x =$ 'red red green blue' and $(\oplus, \otimes) = (\min, +)$.

Verify that $[\![A \cap B]\!](x) = [\![C]\!](x)$:

$[\![A]\!](x) = 0.5 + 0.5 + 0.3 + 0.0 = 1.3$

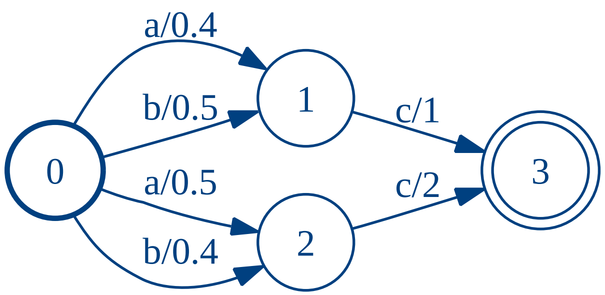
$[\![B]\!](x) = 0.2 + 0.3 + 1 + 0.5 = 2.0$

$[\![C]\!](x) = 0.7 + 0.8 + 1.3 + 0.5 = 3.3$

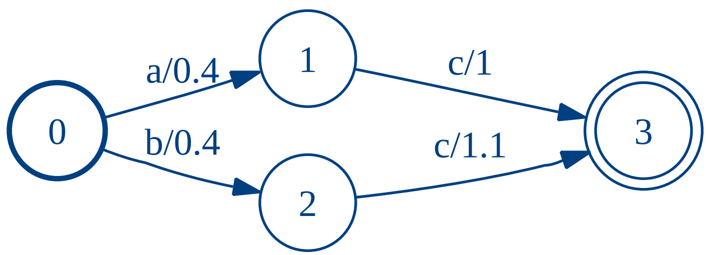
$[\![A \cap B]\!](x) = [\![A]\!](x) \otimes [\![B]\!](x) = [\![A]\!](x) + [\![B]\!](x) = 1.3 + 2.0 = 3.3$

WFSA Operations - Determinization

Some WFSAs (in some semirings) can be **determinized**. After determinization:
- there is a unique starting state
- no two transitions leaving a state share the same input label
- arc weights may change, but weights assigned to strings are unchanged
- there may be many new epsilon arcs



Before Determinization



After Determinization

- determinization can be followed by **minimization** which finds an equivalent machine with a minimal number of states and arcs

WFSA Operations - Single Shortest Distance Algorithms

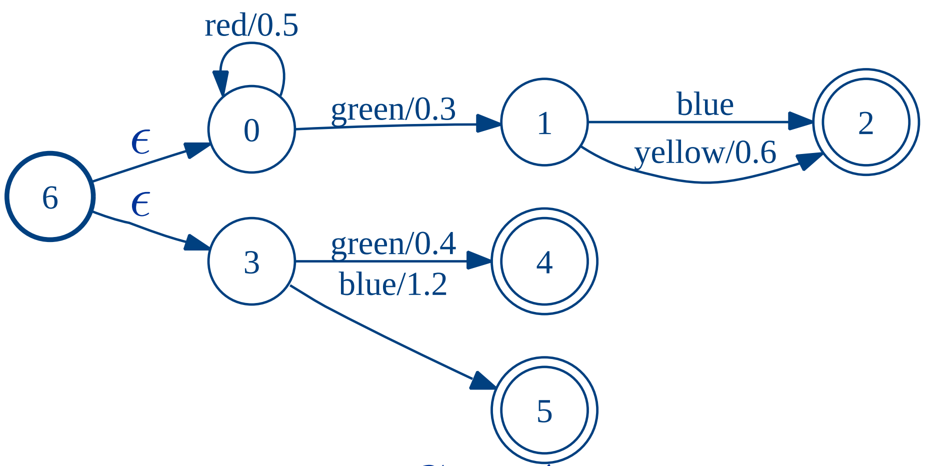
Let F be the set of final states (in case there's more than one)

Let $P(q,F)$ be the set of paths from any state q to any final state in F

- $d[q]$ is the sum of the weights of all paths from q to any final state in F

$$d[q] = \bigoplus_{p \in P(q,F)} w(p)$$

- the costs $d[q]$ can be computed efficiently (e.g. recursively), and trace-back can be added to reconstruct shortest-distance paths



$C = A \cup B$

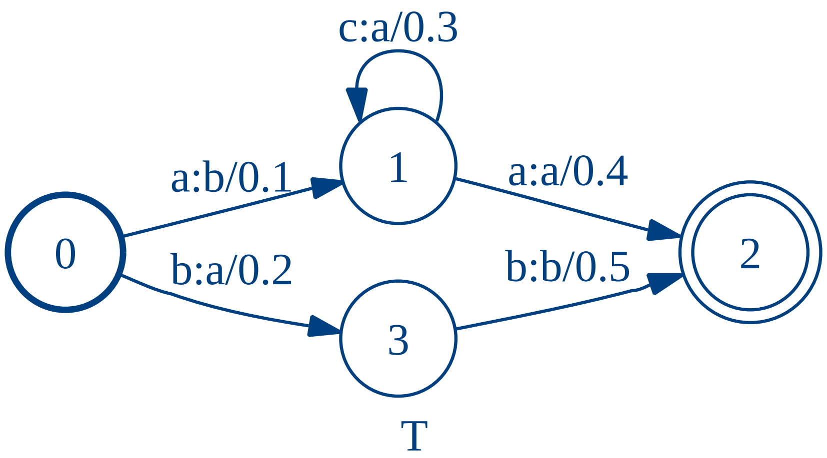
```
> fstshortestdistance --reverse C.fst
0 0.30
1 0
2 0
3 0.40
4 0
5 0
6 0.30
```

Leads easily to a **least cost calculation** procedure

- e.g. the weight of the shortest complete path is 0.30 .

Weighted Finite State Transducers

WFSTs can be used to transform one string to another string
- this is done via symbol-to-symbol mappings
- arcs are modified to have an 'output' symbol
- the interpretation is 'read a symbol x , write a symbol y'
- weights are applied analogously to weighted acceptors



Input String x	Output String y	Cost $[\![T]\!](x,y)$
'b b'	'a b'	0.7
'a a'	'b a'	0.5
'a c a'	'b a a'	0.8
'a c c a'	'b a a a'	1.1
'a c c c a'	'b a a a a'	1.4
⋮	⋮	

In a weighted transducer, arcs have the form: $q \overset{x:y/k}{\rightarrow} q'$
- e.g. the WFST T has an arc with $q = 0$, $q' = 3$, $x = b$, $y = a$, $k = 0.2$

Weighted Finite State Transducer – Definition

The definition of the acceptor is extended to support output operations:
- Two alphabets: Input alphabet: Σ , Output alphabet: Δ
- Each arc (edge) e has an output symbol $o(e) \in \Delta$
- Each arc e has an input symbol $i(e) \in \Sigma$
- For strings $x \in \Sigma^*$ and $y \in \Delta^*$, define $P(x, y)$ to be the set of all complete paths $p = e_1 \cdots e_{n_p}$ which have x as an input sequence and y as an output sequence

$$p \in P(x, y) : x = i(e_1) \cdots i(e_{n_p}) \, , \; y = o(e_1) \cdots o(e_{n_p})$$

- Path weights are computed as in acceptors: $w(p) = \otimes_{j=1}^{n_p} w(e_j)$

The transducer T implements a **weighted mapping** of string x to string y :
- the weight is the sum of all path weights along which x is mapped to y

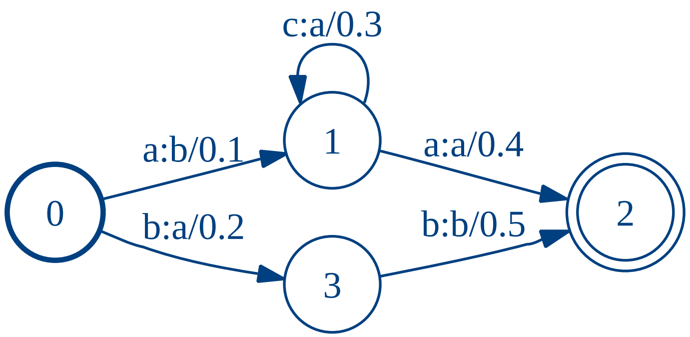
$$[\![T]\!](x, y) = \bigoplus_{p \in P(x,y)} w(p)$$

WFST Operations – Composition

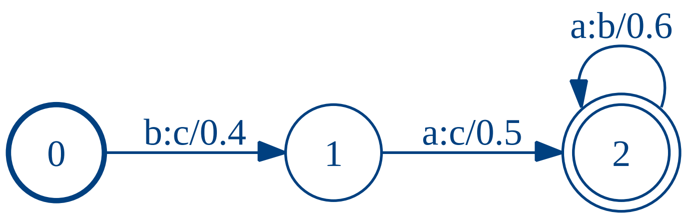
Suppose A and B are two WFSTs: A maps x to y ; B maps y to z.

$A \circ B$ is the composition of A with B which maps x to z

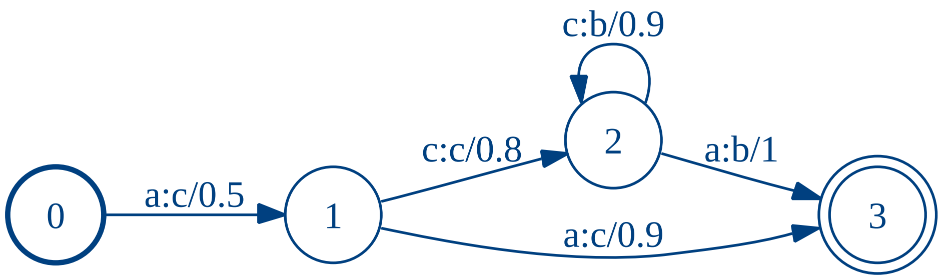
$$[\![A \circ B]\!](x,z) = \bigoplus_y [\![A]\!](x,y) \otimes [\![B]\!](y,z)$$



A



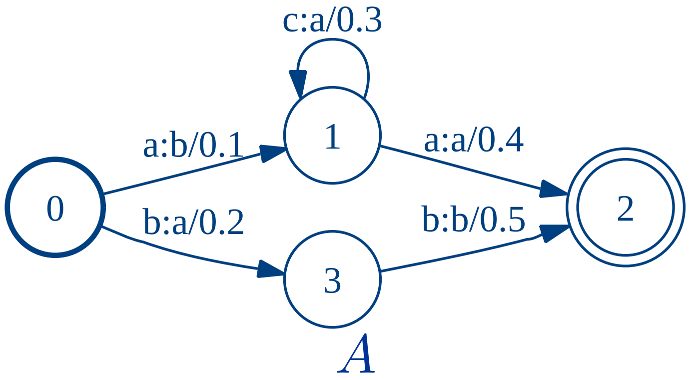
B

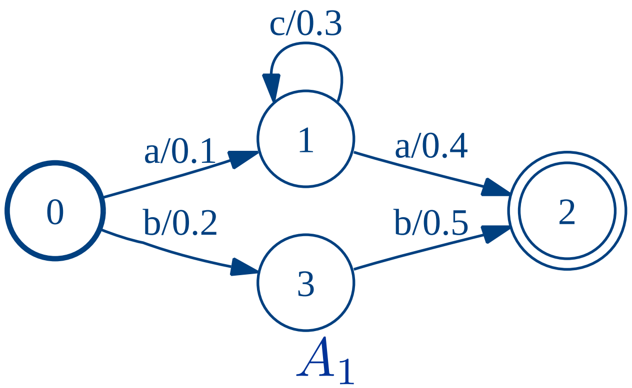


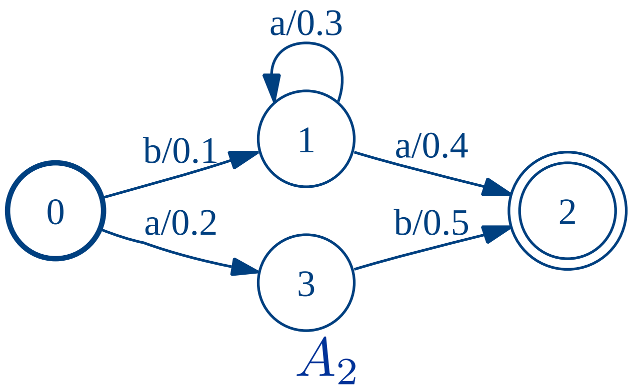
$A \circ B$

WFST Operations – Projection

Transforms a transducer to an acceptor by projecting either onto the input arcs or the output arcs.







Create A_1 by **input projection** of A : $[\![A_1]\!](x) = \bigoplus_y [\![A]\!](x, y)$

Create A_2 by **output projection** of A : $[\![A_2]\!](y) = \bigoplus_x [\![A]\!](x, y)$

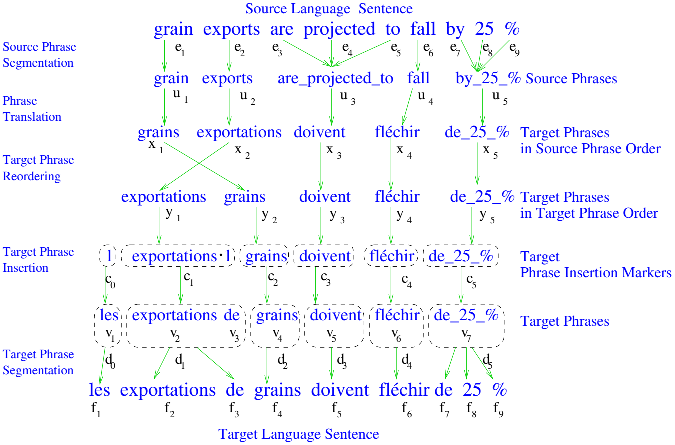
Transducer Translation Model (TTM)

- Transducer Translation Model (TTM): phrase-based SMT system [1]
- Generative model of translation
- Implemented with Weighted Finite State Transducers (WFST)
  - WFSTs used for word alignment, language model, word-to-phrase segmentation, phrase translation and reordering
  - Translation is performed using libraries of standard FST operations
  - No special-purpose decoder required
  - Modularity. Easy to work on translation components in isolation
  - Open Source WFST Toolkit [2] – www.openfst.org/
- Incorporates various second-pass lattice rescoring stages

[1] G. Blackwood et al. (2008), Large-scale statistical machine translation with weighted finite state transducers. FSMNLP.

[2] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut , and M. Mohri (2007), OpenFst: A General and Efficient Weighted Finite-State Transducer Library. CIAA.

Transducer Translation Model (TTM)



- Transformations via stochastic models implemented as WFSTs
- Built with standard WFST operations such as composition and best-path search

TTM Component Models

Compose the following models:

- Source phrase segmentation (unweighted) Ω
- Phrase translation Y
- Phrase reordering R
- Source phrase insertion Φ (optional)
- Target phrase segmentation (unweighted) W
- Target language model acceptor G
- Word penalty is included in language model acceptor

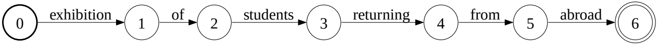
Decoding: A translation lattice is obtained through the series of compositions:

$$L = \mathbf{S} \circ [\Omega \circ Y \circ R \circ \Phi \circ W \circ G]$$

where $\mathbf{S}$ is the source sentence to translate.

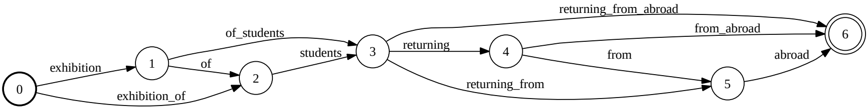
$\Rightarrow$ the most likely translation $\widehat{\mathbf{T}}$ is the path in L with least cost (i.e. minimum negative log-likelihood in tropical semiring). This is found via the standard shortestpath operation.

Phrase Segmentation Transducers



Sentence Acceptor

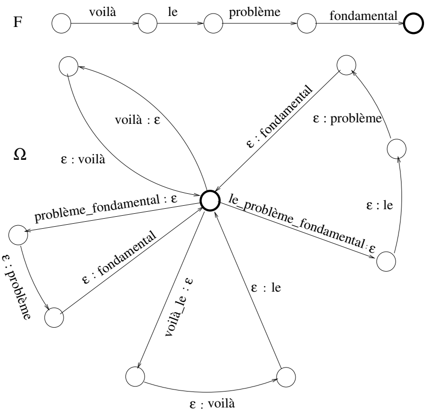
$\Downarrow$ Phrase Segmentation Transducer W or Ω



Phrase Sequence Lattice

- Phrase Segmentation Transducers convert word sequences (or lattices) into phrase lattices according to Phrase Pair Inventory
- lattice is unweighted $\Rightarrow$ all segmentations equally likely in first-pass decoding

Phrase Segmentation Transducers , Ω and W



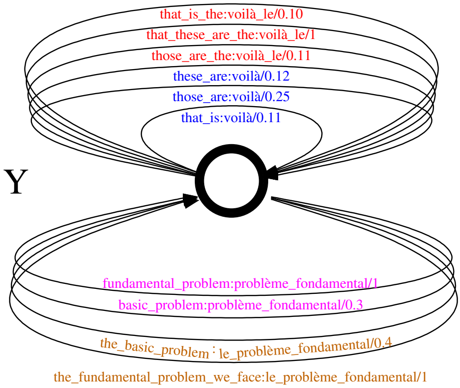
In translation of text, this transducer implements a degenerate distribution:

$$P(T|v_1^K) = \begin{cases} 1 & T \sim v_1^K \\ 0 & \text{other} \end{cases}$$

where v_1^K is any phrase sequence

Phrase Translation Transducer , Y

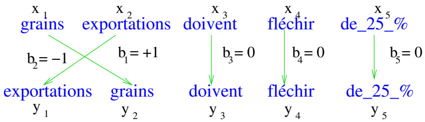
Single state, trivial transducer to implement phrase sequence translation



- Maps English phrases into French
- Based on the Phrase pair Inventory
- Phrase sequences are translated phrase–by–phrase

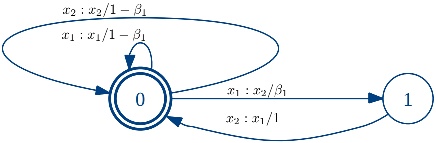
$$P(v_1^K|u_1^K) = \prod_{k=1}^{K} p(v_k|u_k)$$

Phrase Reordering Transducer , R



Associate a jump sequence b_1^K with each sequence y_1^K

$$P(b_1^K | x_1^K, u_1^K, K, s_1^I) = \prod_{k=1}^{K} \underbrace{P(b_k | b_{k-1}, x_{k-1}, x_k, u_{k-1}, u_k)}_{\text{orientation prob., estimated from alignments}}$$



b_k specify relative offsets

MJ-1 : maximum jump of 1

$$b \in \{0, +1, -1\}$$

Extremely simple, but [3]

→ Properly parameterized

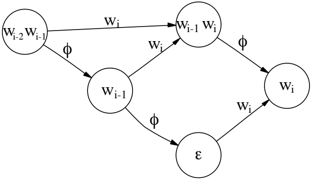
→ Can be extended to MJ2

[3] Kumar , Byrne 2005. Local phrase reordering models for statistical machine translation. HLT-EMNLP.

Target Language Model (word n-Gram) , G

Backoff n-gram approximation : $P(s_1^I) \approx \prod_i P(s_i|s_{i-n+1}^{i-1})$

$$P(s_i|s_{i-n+1}^{i-1}) = \begin{cases} \rho(s_{i-k+1}^{i}) & \text{if } c(s_{i-k+1}^{i}) > \tau \\ \lambda(s_{i-k+1}^{i-1})\, P(s_i|s_{i-n+2}^{i-1}) & \text{otherwise} \end{cases}$$



WFSA Trigram [4]

- each probability and back-off weight is encoded as a cost on an arc in the grammar WFST
- ρ and λ can be pre-computed and stored for reasonable sized language models
- WFST implements backoff n-gram exactly (ϕ is a failure arc) or approximately

For 'reasonable' sized LM training sets, WFST implementations work well

[4] C. Allauzen et al. 2003. Generalized Algorithms for Constructing Statistical Language Models. Proc. ACL

Grammar constraints as LM acceptor

- often certain input word sequences are to be passed through the translation system intact

  此外，大约 三十 个 摊位 也 以 各类 行动 电视 手机 如 t-dmb (terrestrial digital media broadcasting) , s-dmb (satellite digital multimedia broadcasting) 及 dvb-h (digital video broadcasting-handhelds) , 提供 杜林 冬 运 现场 实况 转播 的 画面，藉 以 吸引 参观 者 注意 .

- Separate translation of Foreign-language sequences is not ideal, as it prevents long-span translation, reordering and language models from looking accross boundaries

- Solution: Compose source language model with an additional constrained grammar
  - $G' = G \circ C$, where C accepts sequences $V^* \cdot u_1 \cdot V^* \cdot u_2 \cdot V^*$
    (V is the source language vocabulary)

- Useful to impose constraints on output and keep scores based on long-span models
  - parentheses or quotes properly matched
  - names correctly transliterated

Practical 2/3

- Weighted Finite-State Transducers
- Handout available at:

  http://www.cl.cam.ac.uk/teaching/1213/L102/materials.html
  http://www.cl.cam.ac.uk/teaching/1213/L102/practicals//handout-2.pdf
- Demonstrated Session: 18th February
- Answers to practical questions should be included in a single practical report to be handed at the end of term