Structured prediction: practical

Leon Derczynski

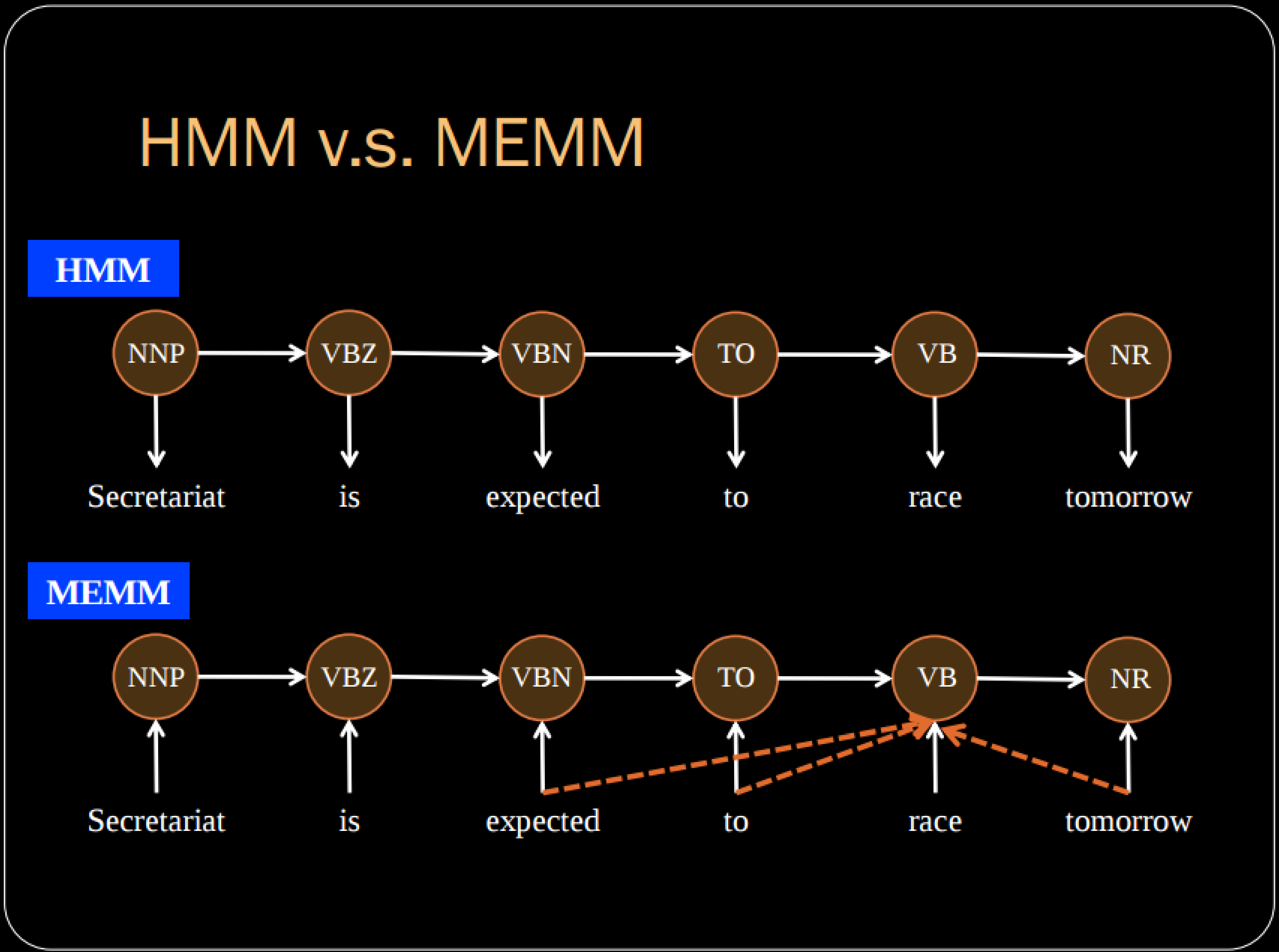
Innopolis University

Beyond HMMs

- HMM: generative
  - States emit words
  - Joint probability: p(words, tags)
  - No long-distance features

- MEMM: discriminative
  - Words emit states
  - Conditional probability: p(tags | words)
  - Long distance features are a go

Beyond HMMs


HMM v.s. MEMM
HMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow
MEMM
NNP
VBZ
VBN
TO
VB
NR
Secretariat
is
expected
to
race
tomorrow

HMM, MEMM, CRF

- MEMM is a discriminative HMM
  - It chooses labels
- HMM and MEMM are similar
- Super-class of these: CRF
  - Conditional Random Fields
  - Plenty of software for these
  - Flexible!
- Kind of problems..?
- Bias..?

CRFsuite

- Let's do some PoS tagging!
- First step: get the tools
  - Data from website
  - CRFsuite
    - OSX:
      - Install Homebrew
      - brew tap homebrew/science
      - brew install crfsuite
    - Linux, Windows: binaries at http://www.chokkan.org/software/crfsuite/
- Second step: feature extraction
- Remaining steps: getting a good score

Feature extraction

- Data format is "CoNLL":
  - One example per line
  - Tab-separated
  - Reference label in the first position
  - Other columns are features

```
VB Have  prefix_Ha  suffix_ve  lower_have
DT a     prefix_a   suffix_a   lower_a
JJ Good  prefix_Go  suffix_od  lower_good
NN day   prefix_da  suffix_ay  lower_day
UH :)    prefix_:)  suffix_:)  lower_:)
```

- Sequences separated by a blank line

Write a feature extractor

- Read in examples, and convert to features
- Input is a label/data pair
- Write a feature extraction function
  - Prefix, suffix (what lengths?)
  - Lower case version
  - All numbers / uppercase / lowercase?
  - Is it a capital?
  - Is it a monetary amount?
- Output the data with features to a file

Learn and view a classifier

- crfsuite learn -h
- crfsuite learn -m my.model train.input
- crfsuite dump my.model | less

- You will see:
  - Transition probabilities
  - State probabilities

Test a classifier

- crfsuite tag -h
- Convert the evaluation data to features, too
- crfsuite tag -q -m my.model -t test.input

- How well did it do?
- Which tags are hardest?

Tuning the model

- With NER, we had three states; i.e. $\Lambda = \{$

  O (outside)

  B (begin)

  I (inside) $\}$

- $|\Lambda|^2 = 9$ transitions
- With PoS data, $|\Lambda|$ is often 11 – 60
- We won't see all the transitions!
  - crfsuite learn -m extra-transitions.model -p feature.possible_transitions=1 train.input
- Does this make a difference?

Too many features

- Many things will occur only once in the data
- The more features, the more likely this is to happen
- Low-frequency features can be distracting
  - Why?
- Trim them out!

Graphical models

1d: 1st-order Markov CRF wit

The 1st-order Markov CRF
combinations of attributes

feature.minfreq=VALUE
Cut-off threshold for
occurrences in the tr

Exercise

- Play with feature extraction, CRFsuite
- Keep a note of what you try and what works
- Write it up, and submit your feature extraction code and model
  - Describe what you tried and, if applicable, why

- Opportunities:
  - Google's Universal PoS tag set
  - Data for many languages
  - Suggests features

Reading

- http://universaldependencies.org/u/pos/
- Petrov, Das, McDonald; “A Universal Part-of-Speech Tagset” - Proc. LREC, 2012