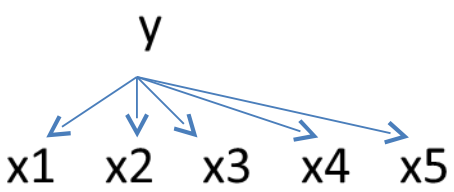
Semi-Supervised Learning and Domain Adaptation in Natural Language Processing

2.8Part-of-speech Tagging
2.9 Dependency Parsing

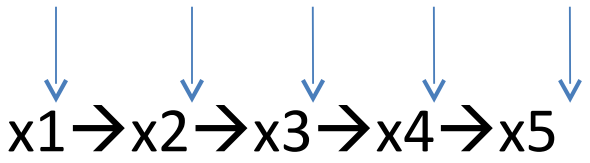
肖 麗穎

2.8 PART-OF-SPEECH TAGGING

- The canonical approach to POS tagging is using *hidden Markov models*(HMMs).
- Difference between Naïve Bayes and (simple) HMMs.
  - Formulate them as Bayesian network.
  - In a Bayesian network:
    - $P(x1, \dots xm) = \prod_{i\geq 1}^{m} P(xi|\{xj| < j, i > \in E\})$
  - Recall that in Naïve bayes:
    - $P(y, x1, \dots, xm) = P(y) \prod_{i\geq 1}^{m} P(xi|y)$
    - 

As for hidden Markov models

- Classification methods rely on the assumption that data is independently drawn, but this assumption is violated in POS tagging. Two famous example:
  - 1. Time flies like an arrow.
    - N-V-ADV-DET-N ○
    - N-N-V-DET-N ○
    - N-N-ADV-DET-N ×



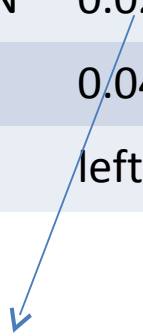
  - 2. The British left waffles on Falklands
    - Left→noun or verb?→*garden path effect* in psycholinguistics

Here we compute the most probable sequence of values for the output variables.

given/	ADJ	DET	NOUN	VEB
ADJ	0.2	0.0	0.7	0.1
DET	0.5	0.0	0.5	0.0
NOUN	0.0	0.1	0.3	0.6
VERB	0.0	0.4	0.2	0.4

given/	The	British	left	waffles
ADJ	0.0	0.4	0.6	0.0
DET	0.1	0.0	0.0	0.0
NOUN	0.0	0.2	0.2	0.6
VERB	0.0	0.0	0.7	0.3

0.0:ADJ	0.2:DET-ADJ	0.024:DET-ADJ-ADJ	0.0:DET-NOUN-VERB-ADJ
0.1:DET	0.0:DET-DET	0.0:DET-ADJ-DET	0.0:DET-NOUN-VERB-DET
0.0:NOUN	0.1:DET-NOUN	0.028:DET-ADJ-NOUN	0.00504:DET-NOUN-VERB-NOUN
0.0:VERB	0.0:DET-VERB	0.042:DET-NOUN-VERB	0.00504:DET-ADJ-NOUN-VERB
The	British	left	waffles



0.028=0.2*0.2*0.7

2.9 Dependency Parsing

- There is no guarantee that a classification-based model does not predict all words to be subjects of the main verb, which would be linguistically impossible.
- For example:
  - “John saw the man on the mountain with a telescope.”
- Two approaches to the struture prediction problem of dependency parsing
  - *Transition-based* dependeny parsing
  - *Graph-based* dependency parsing

2.9.1 Transition-based dependency parsing

A configuration c=(σ, β, A) consists of a stack of words σ, a buffer of words β, and a set of dependency arcs A.(triples of a head word, a dependency label, and a dependent word.)

SHIFT	… John smokes Lebanese		
Lrft-arc	…, John	smokes Lebanese	
Shift	…	smokes Lebanese	John←smokes
Right-arc	…, smokes	Lebanese	
Shift	…	…smokes	smokes→Lebanese
Left-arc	…	…	root→smokes

2.9.2 Graph-based Dependency parsing

- Dependency tree *G=(V,E)*
- Score(G)=$\sum_{(\omega i,r,\omega j)\epsilon A} \lambda_{(\omega i,r,\omega j)}$
- Two challenges:
  - Find the best dependency tree given a model
  - Learn a good model from data.
- Minimum spanning tree
- Comparison of transition- based and graph-based dependency parsing.
  - Transition-based parsers allow us to condition our choice on derivation history.
  - Graph-based parsers are usually better at predicting better at predicting long dependencies and at predicting the syntactic heads of verbs and conjunctions.