# COMP9414: Artificial Intelligence

# Lecture 5b: Language Models

Wayne Wobcke

e-mail:w.wobcke@unsw.edu.au



## Probabilistic Language Models

- Based on statistics derived from large corpus of text/speech
  - Brown Corpus (1960s) – 1 million words
  - Penn Treebank (1980s) – 7 million words
  - North American News (1990s) – 350 million words
  - IBM – 1 billion words
  - Google & Facebook – Trillions of words
- Contrary to view that language ability based on (innate) knowledge
- Idea is language ability can be learnt … with enough data …



## This Lecture

- Part of Speech Tagging
  - n-gram Models
  - Hidden Markov Models
  - Viterbi Algorithm
- Word Sense Disambiguation
  - Mutual Information
  - Class-Based Models



## Penn Treebank Tagset

| Tag | Description | Example | Tag | Description | Example |
|---|---|---|---|---|---|
| CC | coord. conjunction | *and, or* | RB | adverb | *extremely* |
| CD | cardinal number | *one, two* | RBR | adverb, comparative | *never* |
| DT | determiner | *a, the* | RBS | adverb, superlative | *fastest* |
| EX | existential there | *there* | RP | particle | *up, off* |
| FW | foreign word | *noire* | SYM | symbol | *+, %* |
| IN | preposition or sub-conjunction | *of, in* | TO | “to” | *to* |
| JJ | adjective | *small* | UH | interjection | *oops, oh* |
| JJR | adject., comparative | *smaller* | VB | verb, base form | *fly* |
| JJS | adject., superlative | *smallest* | VBD | verb, past tense | *flew* |
| LS | list item marker | *1, one* | VBG | verb, gerund | *flying* |
| MD | modal | *can, could* | VBN | verb, past participle | *flown* |
| NN | noun, singular or mass | *dog* | VBP | verb, non-3sg pres | *fly* |
| NNS | noun, plural | *dogs* | VBZ | verb, 3sg pres | *flies* |
| NNP | proper noun, sing. | *London* | WDT | wh-determiner | *which, that* |
| NNPS | proper noun, plural | *Azores* | WP | wh-pronoun | *who, what* |
| PDT | predeterminer | *both, lot of* | WP$ | possessive wh- | *whose* |
| POS | possessive ending | *'s* | WRB | wh-adverb | *where, how* |
| PRP | personal pronoun | *he, she* | | | |

# Part of Speech Tagging

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.
- There/EX are/VBP 70/CD children/NNS there/RB
- Preliminary/JJ findings/NNS were/VBD reported/VBN in/IN today/NN s/POS New/NNP England/NNP Journal/NNP of/IN Medicine/NNP ./.






# Why is this Hard?

Ambiguity, e.g. back

- earnings growth took a back/JJ seat
- a small building in the back/NN
- a clear majority of senators back/VBP the bill
- Dave began to back/VB toward the door
- enable the country to buy back/RP about debt
- I was twenty-one back/RB then






# Probabilistic Formulation

- Events: Occurence of word $w$, occurrence of a word with tag $t$
- Given sequence of words $w_1, \cdots, w_n$, choose $t_1, \cdots, t_n$ so that – $P(t_1, \cdots, t_n | w_1, \cdots, w_n)$ is maximixed
- Apply Bayes' Rule
  - $P(t_1, \cdots, t_n | w_1, \cdots, w_n) = \frac{P(w_1, \cdots, w_n | t_1, \cdots, t_n).P(t_1, \cdots, t_n)}{P(w_1, \cdots, w_n)}$
  - Therefore maximize $P(w_1, \cdots, w_n | t_1, \cdots, t_n).P(t_1, \cdots, t_n)$



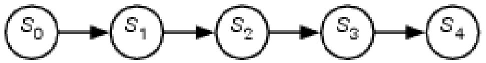



# Unigram Model

Maximize $P(w_1, \cdots, w_n | t_1, \cdots, t_n).P(t_1, \cdots, t_n)$

- Apply independence assumptions
  - $P(w_1, \cdots, w_n | t_1, \cdots, t_n) = P(w_1 | t_1). \cdots .P(w_n | t_n)$
  - Probability of word $w$ generated by $t$ independent of context
  - $P(t_1, \cdots, t_n) = P(t_1). \cdots .P(t_n)$
  - Probability of tag sequence independent of order
- Estimate probabilities
  - $P(w|t)$ = #($w$ occurs with tag $t$)/#(words with tag $t$)
  - $P(t)$ = #(words with tag $t$)/#words
  - Choose tag sequence that maximizes $\Pi P(t_i | w_i)$
  - Chooses most common tag for each word
- Accuracy around 90% – but still ≈1 word wrong in every sentence!

# Markov Chain

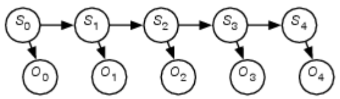



- Bayesian network
  - $P(S_0)$ specifies initial conditions
  - $P(S_{i+1}|S_i)$ specifies dynamics (stationary if same for each $i$)
- Independence Assumptions
  - $P(S_{i+1}|S_0,\cdots,S_i) = P(S_{i+1}|S_i)$
  - Transition probabilities dependent only on current state $S_i$ – independent of history to reach that state $S_0,\cdots,S_{i-1}$
  - The future is independent of the past, given the present





# Hidden Markov Models




- Bayesian network
  - $P(S_0)$ specifies initial conditions
  - $P(S_{i+1}|S_i)$ specifies dynamics
  - $P(O_i|S_i)$ specifies "observations"
- Independence Assumptions
  - $P(S_{i+1}|S_0,\cdots,S_i) = P(S_{i+1}|S_i)$ (Markov Chain)
  - $P(O_i|S_0,\cdots,S_{i-1},O_0,\cdots,O_{i-1}) = P(O_i|S_i)$
  - Observations (words) depend only on current state (tag)





# Bigram Model

Maximize $P(w_1,\cdots,w_n|t_1,\cdots,t_n).P(t_1,\cdots,t_n)$

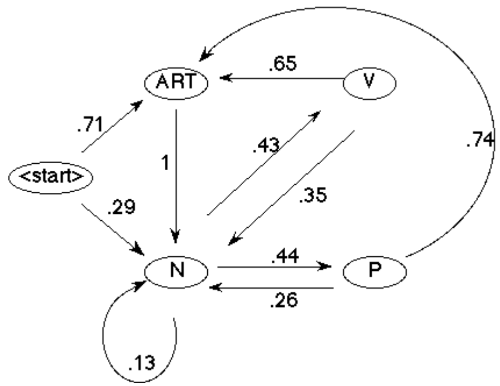
- Apply independence assumptions (Markov assumptions)
  - $P(w_1,\cdots,w_n|t_1,\cdots,t_n) = \Pi P(w_i|t_i)$
  - $P(t_1,\cdots,t_n) = P(t_n|t_{n-1}).\cdots.P(t_0|\phi)$, where $\phi$ = start
  - Bigram model: state (tag) depends only on previous state (tag)
  - Observations (words) depend only on states (tags)
- Estimate probabilities
  - $P(t_i|t_j)$ = #($(t_j,t_i$ occurs)/#($t_j$ starts a bigram)
  - Choose tag sequence that maximizes $\Pi P(w_i|t_i).P(t_i|t_{i-1})$
  - Parts of speech generated by finite state machine



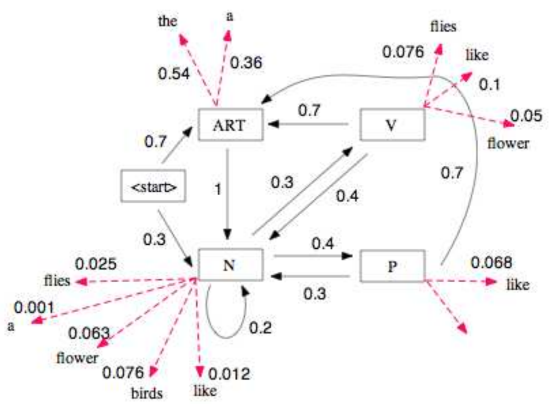


# Markov Model for POS Tagging

Transition probabilities define stationary distribution



# Hidden Markov Model for POS Tagging

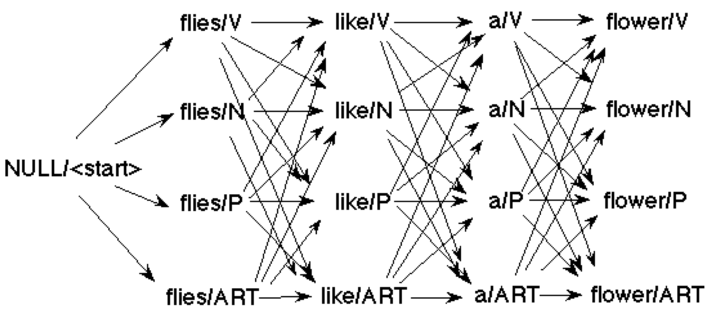








# Computing Probabilities



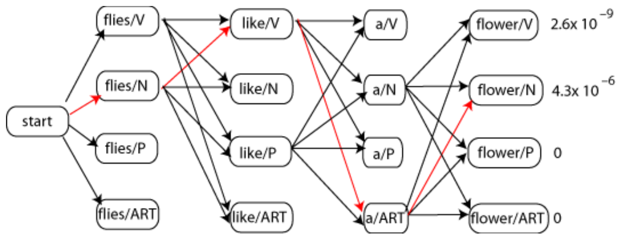






# Viterbi Algorithm

1. Sweep forward (one word at a time) saving only the most likely sequence (and its probability) for each tag $t_i$ of $w_i$
2. Select highest probability final state
3. Follow chain backwards to extract tag sequence









# Word Sense Disambiguation

Example

> I should have changed that stupid lock and made you leave your key, if I'd known for just one second you'd be back to bother me.

lock = ···

leave = ···

second = ···

back = ···

# Windows

- Consider co-occurrences in a window about $w$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $w_1$ | | | | | $w$ | | | | | $w_n$ |

- Senses of word should co-occur with classes of "related" words
- Choose sense $s$ of $w$ to maximize $P(w \text{ is } s | w_1, \cdots, w_n)$
- Apply Bayes' Rule
  - Maximize $\frac{P(w_1,\cdots,w_n | w \text{ is } s).P(w \text{ is } s)}{P(w_1,\cdots,w_n)}$
- Apply independence assumptions
  - $P(w_1, \cdots, w_n | w \text{ is } s) = \Pi P(w_i | w \text{ is } s)$
- Estimate probabilities: $P(w_i | w \text{ is } s)$
  - #($w_i$ in $n$-word window around $w$ as $s$)/#(windows on $w$ as $s$)





# Mutual Information

$$MI(x,y) = \log_2 \frac{P(x,y)}{P(x).P(y)}$$

$$MI(sense(w_1), w_2) = \log_2 \frac{N.\#(sense(w_1), w_2)}{\#(sense(w_1)).\#(w_2)}$$

- $MI < 0$: $sense(w_1)$ and $w_2$ occur together less than randomly
- $MI > 0$: $sense(w_1)$ and $w_2$ occur together more than randomly
- Adding mutual information is equivalent to assuming independence
- Choose sense $s$ for $w$ = $\arg\max_{s \in senses(w)} \Sigma_{w_i \in window(w)} MI(s, w_i)$





# Class-Based Methods

- Use pre-defined "sense classes", e.g. WordNet, Wikipedia
  - lock $\rightarrow$ *Mechanical Devices* $\leftarrow$ tool, crank, cog, $\cdots$
  - lock $\rightarrow$ *Body Part* $\leftarrow$ hair, eyes, hands, $\cdots$
- Calculate counts for word senses by adding those for words
- Advantages
  - Reduces space and time complexity
  - Reduces data sparsity
  - Allows unsupervised learning





# Conclusion

- Statistical (and neural network) models perform well on many tasks
  - Part-of-speech tagging
  - Word sense disambiguation
  - Control of traditional parser
  - Probabilistic parsing
- Problems
  - Unrealistic simplifying assumptions (that seem to work)
  - Requirement for very large amount of (labelled) text
  - Sparsity of word occurrences in (even large) text corpora
  - Changes in word usage over time (e.g. *Senator* Obama)