

# COMP9414: Artificial Intelligence

## Lecture 5b: Language Models

Wayne Wobcke

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

## This Lecture

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

## 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 ...

## Penn Treebank Tagset

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

## 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

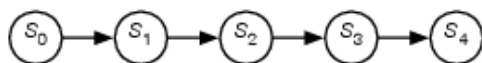
- Events: Occurrence of word  $w$ , occurrence of a word with tag  $t$
- Given sequence of words  $w_1, \dots, w_n$ , choose  $t_1, \dots, t_n$  so that –  $P(t_1, \dots, t_n | w_1, \dots, w_n)$  is maximized
- Apply Bayes' Rule
  - ▶  $P(t_1, \dots, t_n | w_1, \dots, w_n) = \frac{P(w_1, \dots, w_n |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

## Unigram Model

Maximize  $P(w_1, \dots, w_n | t_1, \dots, t_n) \cdot P(t_1, \dots, t_n)$

- Apply independence assumptions
  - $P(w_1, \dots, w_n | t_1, \dots, t_n) = P(w_1 | t_1) \cdot \dots \cdot P(w_n | t_n)$
  - Probability of word  $w$  generated by  $t$  independent of context
  - $P(t_1, \dots, t_n) = P(t_1) \cdot \dots \cdot P(t_n)$
  - Probability of tag sequence independent of order
- Estimate probabilities
  - $P(w | t) = \#(w \text{ occurs with tag } t) / \#(\text{words with tag } t)$
  - $P(t) = \#(\text{words with tag } t) / \#(\text{words})$
  - Choose tag sequence that maximizes  $\prod P(t_i | w_i)$
  - Chooses most common tag for each word
- Accuracy around 90% – but still  $\approx 1$  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

- ▶  $P(S_{i+1}|S_0, \dots, S_i) = P(S_{i+1}|S_i)$
- ▶ Transition probabilities dependent **only** on current state  $S_i$  – **independent** of history to reach that state  $S_0, \dots, S_{i-1}$
- ▶ The future is independent of the past, given the present

## Bigram Model

Maximize  $P(w_1, \dots, w_n | t_1, \dots, t_n) \cdot P(t_1, \dots, t_n)$

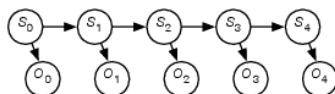
### Apply independence assumptions (Markov assumptions)

- ▶  $P(w_1, \dots, w_n | t_1, \dots, t_n) = \prod P(w_i | t_i)$
- ▶ Observations (words) depend **only** on states (tags)
- ▶  $P(t_1, \dots, t_n) = P(t_n | t_{n-1}) \cdot \dots \cdot P(t_0 | \phi)$ , where  $\phi$  = start
- ▶ Bigram model: state (tag) depends **only** on previous state (tag)

### Estimate probabilities

- ▶  $P(t_i | t_j) = \#((t_j, t_i \text{ occurs}) / \#(t_j \text{ starts a bigram}))$
- ▶ Choose tag sequence that maximizes  $\prod P(w_i | t_i) \cdot P(t_i | t_{i-1})$
- ▶ Parts of speech generated by finite state machine

## Hidden Markov Models



### Bayesian network

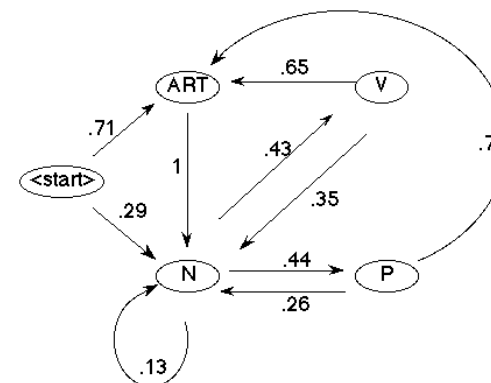
- ▶  $P(S_0)$  specifies initial conditions
- ▶  $P(S_{i+1}|S_i)$  specifies dynamics
- ▶  $P(O_i|S_i)$  specifies “observations”

### Independence Assumptions

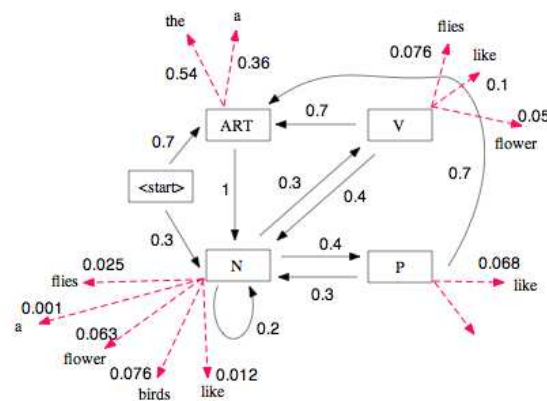
- ▶  $P(S_{i+1}|S_0, \dots, S_i) = P(S_{i+1}|S_i)$  (Markov Chain)
- ▶  $P(O_i|S_0, \dots, S_{i-1}, S_i, O_0, \dots, O_{i-1}) = P(O_i|S_i)$
- ▶ Observations (words) depend **only** on current state (tag)

## Markov Model for POS Tagging

Transition probabilities define **stationary** distribution



## Hidden Markov Model for POS Tagging



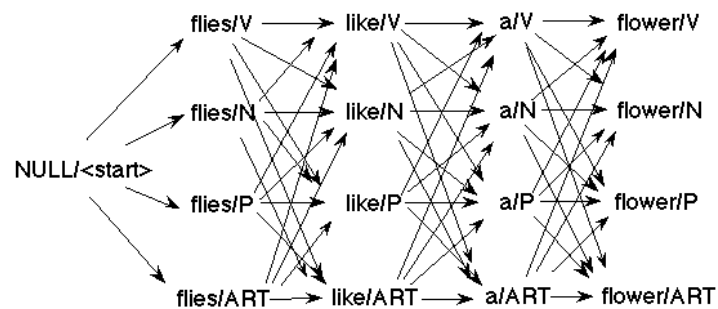
## Example

$w$	$P(w \text{ART})$	$P(w \text{N})$	$P(w \text{P})$	$P(w \text{V})$
a	0.36	0	0	0
flies	0	0.025	0	0.076
flower	0	0.063	0	0.05
like	0	0.012	0	0.1

$$\begin{aligned}
 P(\text{flies}/\text{N} \text{ like}/\text{V} \text{ a}/\text{ART} \text{ flower}/\text{N}) &= \\
 &= P(\text{N}|\text{start}) \cdot P(\text{flies}|\text{N}) \cdot P(\text{V}|\text{N}) \cdot P(\text{like}|\text{V}) \cdot P(\text{ART}|\text{V}) \cdot P(\text{a}|\text{ART}) \cdot P(\text{N}|\text{ART}) \cdot P(\text{flower}|\text{N}) \\
 &= 0.29 \times 0.025 \times 0.43 \times 0.1 \times 0.65 \times 0.36 \times 1 \times 0.063
 \end{aligned}$$

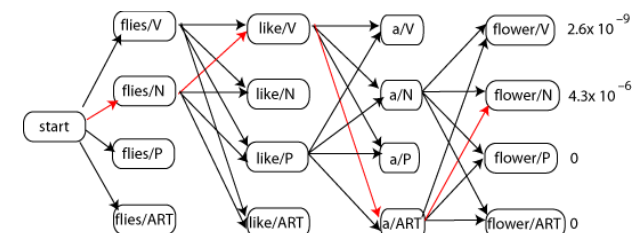
Most likely sequence, even though  $P(\text{flies}/\text{V}) > P(\text{flies}/\text{N})$

## 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 = ...

## Simple (Made Up) Example

Word	bridge/structure	bridge/dental	any window
teeth	1	10	300
suspension	200	1	2000
the	5500	180	500 000
dentist	2	35	900
TOTAL	5651	194	501 500

$$P(\text{bridge/structure}) = 5651/501\,500 = 0.0113$$

$$P(\text{bridge/dental}) = 194/501\,500 = 3.87 \times 10^{-4}$$

$$P(\text{teeth}|\text{bridge/structure}) = 1/5651 = 1.77 \times 10^{-4}$$

$$P(\text{teeth}|\text{bridge/dental}) = 10/194 = 0.052$$

bridge/dental preferred if window contains 'teeth'

## 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  $P(w \text{ as } s | w_1, \dots, w_n)$
- Apply Bayes' Rule
  - Maximize  $\frac{P(w_1, \dots, w_n | w \text{ as } s) \cdot P(w \text{ as } s)}{P(w_1, \dots, w_n)}$
- Apply independence assumptions
  - $P(w_1, \dots, w_n | w \text{ as } s) = \prod P(w_i | w \text{ as } s)$
- Estimate probabilities:  $P(w_i | w \text{ as } s)$ 
  - $\#(w_i \text{ in } n\text{-word window around } w \text{ as } s) / \#(\text{windows on } w \text{ as } s)$

## Mutual Information

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

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

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

## Class-Based Methods

---

- Use predefined “sense classes”, e.g. WordNet, Wikipedia
  - ▶ lock → *Mechanical Devices* ← tool, crank, cog, ...
  - ▶ lock → *Body Part* ← hair, eyes, hands, ...
- Calculate counts for word senses by adding those for words
- Advantages
  - ▶ Reduces space and time complexity
  - ▶ Reduces data **sparsity** 稀疏的
  - ▶ 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*)