Natural Language Processing
CSE 325/425

Sihong Xie

<u>Lecture 6:</u>

- Part-of-Speech (POS)
- POS tagging
- Hidden Markov Models (HMM)

Part-of-Speech

English word classes: cover a few common classes that will be used in tagging.

1. Nouns: pronouns (she, he, I, who, others), proper nouns (Russia), countable nouns (desk), mass noun (air)
2. Verbs: participles (paced), gerund (pacing), auxiliaries (be, do, have, can, may, should)
3. Adjectives: comparative, superlative.
4. Adverbs: I went *Church yesterday*
5. Prepositions: in, on, over, ...
6. Particles: phrasal verb like "go *over*".
   - Easy to be confused with prepositions.
   - Combination of verb and particle does not have their meanings combined simply.
7. Determiners: a, the, an
8. Conjunctions: and, but, that, when
9. Other smaller classes.

Part-of-Speech

Syntatic information: how words are ordered in a sentence.

- noun-verb
- determiner-noun
- adjective-noun
- verb-adverb
- preposition-noun

Useful for grammar checking: go to (a?the?) hospital

Semantic information: meaning of a word in a context. Useful for:

- machine translation (building a building): building -> (建 vs. 楼)
- question-answering, need to understand the semantics of what a person is asking.
- relation extraction (Bill Gates founded MS): gates (verb vs. noun)
- event extraction (They went to a concert): concert (verb vs. noun)
- entity extraction (I will visit DC): DC?

Syntactic info $\Leftrightarrow$ Semantic info

POS tagging

No free

large-scale training corpus for Language models, POS tagger, Syntactic Parsing

The Penn Tree Bank tagsets.

- JJ (adjective), JJR (comparative), JJS (superlative)
- NN, NNS (plural), NNP (pronoun), NNPS (pronoun plural)
- RB (adverb), RBR (comparative), RBS (superlative)
- VB, VBD (past tense), VBG (present participle, gerund), going
  VBN (past participle), VBZ (third-person single) goes
- RP: particle (as in *go on*)
- IN: preposition (as in *go to church*)

go
gone

See SLP3 for a more comprehensive List & examples.

Example: Determiner

- 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

Def of POS tagging task:

Given a sentence = $[w_1, w_2, \ldots, w_n]$

output a single POS-tag for each word in $[w_1 \ldots w_n]$

corpus with sentences POS-tagged → NLP model (POS-tagger) → make predictions on future sentences.

POS tagging

Why POS-tagging is difficult

- VBZ *Does that flight serve dinner* (that is used as a determiner)
  NN VB NN
- *I thought that your flight was earlier* (that is used as a complementizer)
  main sentence q_1 q_2 q_3 clause

VB get RP around

Difficult even for humans

- Mrs./NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG
- All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN
- Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

adverb

Naïve solution?

- always predict the most frequent tag for each word.
- contexts are useful

good performance (90%) on easy words, such as "difficult"/JJ

Most **common** words are ambiguous; most words are not ambiguous.

Hidden Markov Model (HMM)

Want to find the best sequence of POS tags for a given sentence.

- Vocabulary V
- Set of N POS tags $S = \{s_1, \ldots, s_N\}$

T = length of the sentence

T=9

- Observed sentence $O = [o_1, \ldots, o_T], o_t \in V$
- Hidden states: $Q = [q_1, \ldots, q_T], q_t \in S$

All	we	gotta	do	is	go	around	the	corner
DT	PRP	VBN	VB	VBZ	VB	IN	DT	NN

- MAP (maximum a posterior) prediction

$$Q^* = \arg\max_Q \boxed{\Pr(Q|O)}$$

posterior probability of seq Q given seq O.

- Using the Bayes theorem:

Pr(O) = Prob. of seeing the sentence O

$$\Pr(Q|O) = \frac{\Pr(O|Q)\Pr(Q)}{\Pr(O)}$$

$\sum_Q P(O,Q) =$

Pr(O|Q): likelihood of observing seq O, conditioned on seq Q.

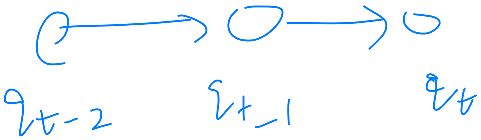
Pr(Q): prob. of observing seq Q

- Difficulty: there are exponentially many possible sequences.

How many possible Q sequences?

$O(N^T)$: exponential time complex.

D-seperation :



Conditional Independence

$Pr(q_t, q_{t-2} | q_{t-1})$

$= Pr(q_t | q_{t-1}) \times Pr(q_{t-2} | q_{t-1})$

The Markov assumption

(Strong ?) ⇒ pushed further $Pr(q_i, q_j)$

$= Pr(q_i) Pr(q_j)$

⇒ Naive Solution : tagging each word with the most frequent POS-tag.

Motivations:

- Linguistically, any two POS tags do not occur independently.
  - P(NN|DT) and P(JJ | DT) should be higher than P(DT | JJ)
  - e.g., *a yellow cab* vs. *yellow a cab* vs. *yellow cab a*
- Computationally, reduce the complexity of computing the prior $\Pr(Q)$

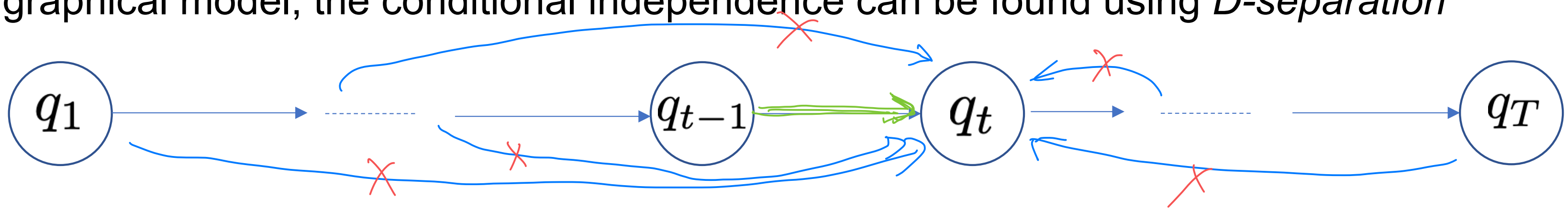
Conditioned on the **previous** POS tag, the probability of the **current** tag is

- Transition probability

$$\Pr(q_t|q_1,\ldots,q_{t-1}) = \Pr(q_t|q_{t-1})$$

Using a graphical model, the conditional independence can be found using *D-separation*

No long-range dependency



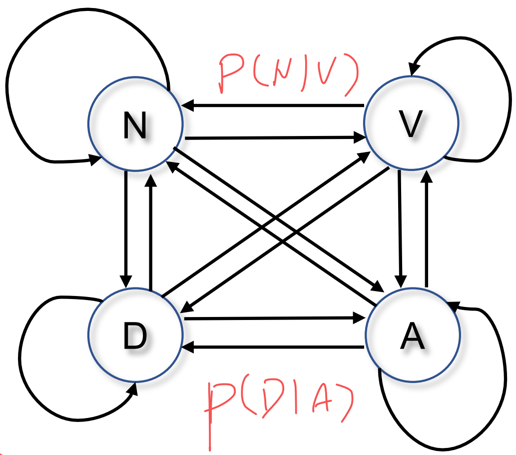
The Markov assumption

$O(N^2)$ = $N \cdot N \cdots N$

$\boxed{N \text{ Pos tags}}$

exponential ?!

fully connected directed graph



N = Noun
V = Verb
D = Determiner
A = Adjective

Each row of A is a prob dist.

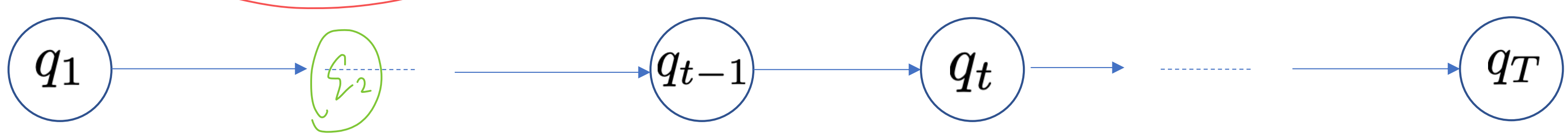
Transition probability matrix A

A_{ij} = Probability of going from hidden state i to hidden state j,

i-th row

j-th col

$$\begin{array}{c|cccc} & N & V & D & A \\ \hline N & & & & \\ V & & & & \\ D & & & & \\ A & & & & \end{array}$$



$q_1 \in$ {N, V, D, A}

$q_t \in$ {N, V, D, A}

$q_T \in$ {N, V, D, A}

$P(q_2 \mid \underline{q_1 = V})$

Sample q_2 using the probabilities $P(N|V)$, $P(V|V)$, $P(D|V)$, $P(A|V)$

Emission probability

Motivations:

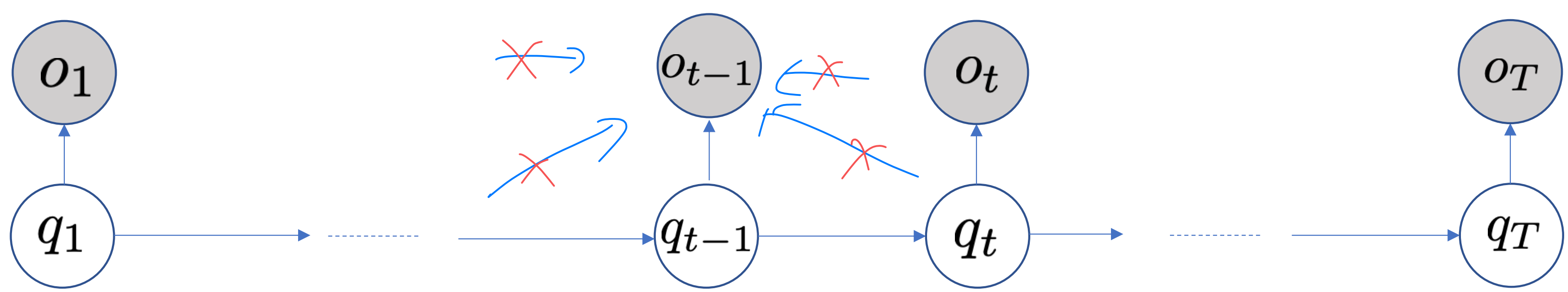
- Linguistically, a word is selected (partially) based on the current POS tag.
- Computationally, reduce the complexity of computing the likelihood $\Pr(O|Q)$

Conditioned on the **current** POS tag, the probability of the **current** word is

- Emission probability

$$\Pr(o_t|q_1,\ldots,q_T,o_1,\ldots,o_{t-1},o_{t+1},\ldots,o_T) = \Pr(o_t|q_t)$$

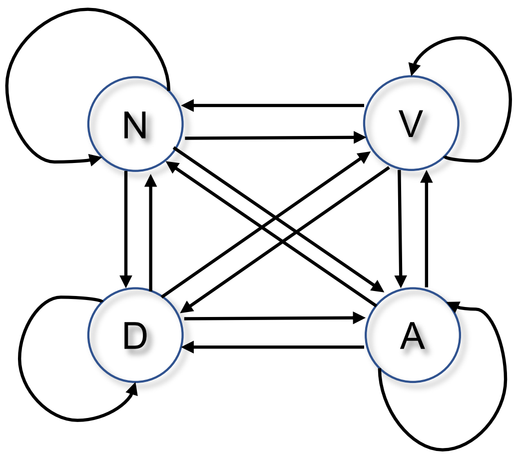
Using a graphical model



Each row of B is a prob. dist.

$B = |S| \Big\{ \underbrace{\big[[\cdot \cdot \cdot \cdot - - -] \big]}_{|V|}$

Generating a sentence using HMM



N = Noun
V = Verb
D = Determiner
A = Adjective

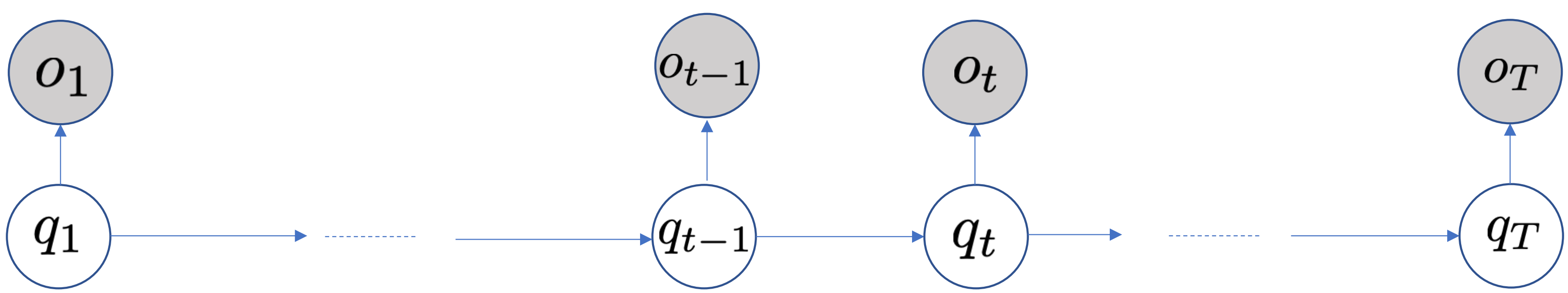
Emission probability matrix B

B_{io} = emission probability of word o from hidden state i.

i-th row

o-th col

$o_t \in V$



$q_t \in$ {N, V, D, A} $= S$

$q_1 \in \{N, V, D, A\}$

$\pi = [Pr(N), Pr(V), Pr(D), Pr(A)]$

High low High higher

$q_1 \sim \pi$ over

$S = \{s_1, \dots, s_N\}$

$$q_1 \to q_2 \to q_3 \to \dots \to q_T$$

$\Pr(q_2|q_1)$ $\Pr(q_3|q_2)$

HMM

Now we can simplify the following probabilities

- Prior

$$\Pr(Q) = \Pr(q_1)\Pr(q_2|q_1)\dots\Pr(q_T|q_{T-1}) = \Pr(q_1)\prod_{t=2}^{T}\Pr(q_t|q_{t-1})$$

$= \Pr(q_1|\phi)$

with the starting probability $\Pr(q_1) = \pi_{q_1}$ = probability of using q_1 as the first POS tag.

$q_1 \in S$

- Likelihood

$$\Pr(O|Q) = \prod_{t=1}^{T}\underbrace{\Pr(o_t|q_t)}_{\text{emission prob.}} = \prod_{t=1}^{T} B_{q_t, o_t}$$

$\pi = \underbrace{[\Pr(s_1) \dots \Pr(s_N)]}_{N}$

- Posterior

$$\Pr(Q|O) = \frac{\Pr(O|Q)\Pr(Q)}{\Pr(O)}$$

MAP:

$Q^* = \arg\max_Q \Pr(Q|O)$

$$= \frac{\Pr(O|Q)\Pr(Q)}{\sum_Q \Pr(O,Q)} = \frac{\Pr(O|Q)\Pr(Q)}{\sum_Q \Pr(O|Q)\Pr(Q)}$$

HMM

Similar to n-gram model

- The transitions between tags is a bi-gram model over pos-tags

Bi-gram

$Pr(w_t \mid w_{t-1})$

Different from n-gram model

- The transitions happen on the hidden state (POS-tag) level;
- Words are independent given the POS sequence.

vs. $Pr(q_t \mid q_{t-1})$

Three tasks with HMM:

- find the likelihood of a POS-tag sequence (forward algorithm); $P(Q \mid O)$
- find the most likely POS-tag sequence given an observed sentence (Viterbi algorithm); $\max_Q P(Q \mid O)$
- find the start, transition, and emission probabilities (MLE with backward algorithm).

train HMM from training corpus (pos-tagged sentences)

$\triangleq$ Estimating

$$\boxed{\begin{array}{l} A: \text{transition prob} \\ B: \text{emission prob.} \\ \pi: \text{start prob} \end{array}}$$

EM-alg w. forward backward