Lecture 4: Viterbi, NER

Claire Cardie, Tanya Goyal

CS 4740 (and crosslists): Introduction to Natural Language Processing

Announcements

- HW1 released today! Due Fri Feb 21 11:59pm.
  - START IT NOW!!!!!

Today

- HMMs as a tagging technology: Viterbi
  - You will implement for HW1!!!
- HMMs as a generative model
- Where do the probabilities come from?
- Named entity tagging: the task for HW1!!!

Recall: HMM POS Tagger

?   ?   ?
Cornell beat Harvard

Goal: Find the tag sequence that maximizes $P(t_1 \ldots t_N \mid w_1 \ldots w_N)$

Need to Bayes flip:

$$= \frac{P(w_1 \ldots w_N | t_1 \ldots t_N) \cdot P(t_1 \ldots t_N)}{\cancel{P(w_1 \ldots w_N)}}$$

Make Independence and Markov Assumptions

?   ?   ?

Cornell beat Harvard

$$P(w_1 ... w_N | t_1 ... t_N) \cdot P(t_1 ... t_N)$$

$P(t_1, \ldots, t_n)$: approximate using **n-gram model**

bigram $\prod_{i=1,n} P(t_i \mid t_{i-1})$

trigram $\prod_{i=1,n} P(t_i \mid t_{i-2} t_{i-1})$

Make Independence and Markov Assumptions

?　　?　　?
Cornell beat Harvard

$$\frac{P(w_1 \ldots w_N | t_1 \ldots t_N) P(t_1 \ldots t_N)}{}$$

Assume each word appears with a particular tag independent of its neighbors

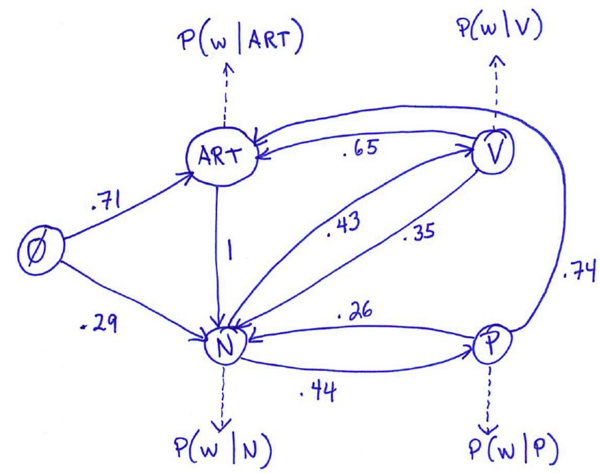
$$P(w_1 \ldots w_n | t_1 \ldots t_n) \cong \prod_{i=1,n} P(w_i | t_i)$$

? ? ?

Cornell beat Harvard

$$P(t_1 \ldots t_N \mid w_1 \ldots w_N) \cong \prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$

- Equation is modeled by an HMM
  (probabilistic finite-state machine)
  - **States:** represent the possible POS
  - **Transition probabilities:** bigram probabilities for tags
  - **Emission (observation) probabilities:** indicate, for each word, how likely that word is to be selected if we randomly select a POS



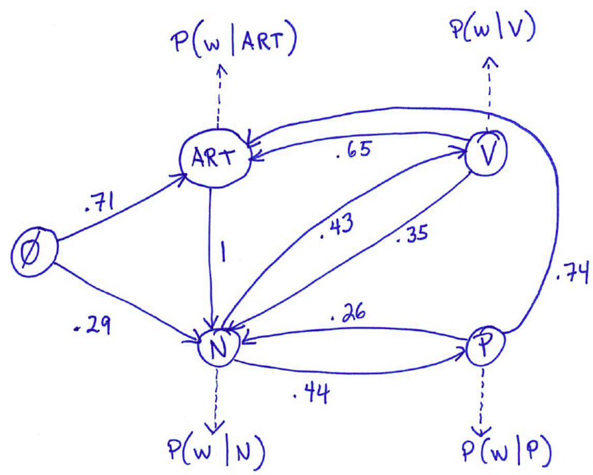
Tagging algorithm

N V N
Cornell beat Harvard

$$\prod_{i=1,n} P(t_i \mid t_{i-1}) \cdot P(w_i \mid t_i)$$

Given a new sentence to tag

- For every possible tag sequence,
  - Apply equation to calculate the score
- Select the highest-scoring tag sequence



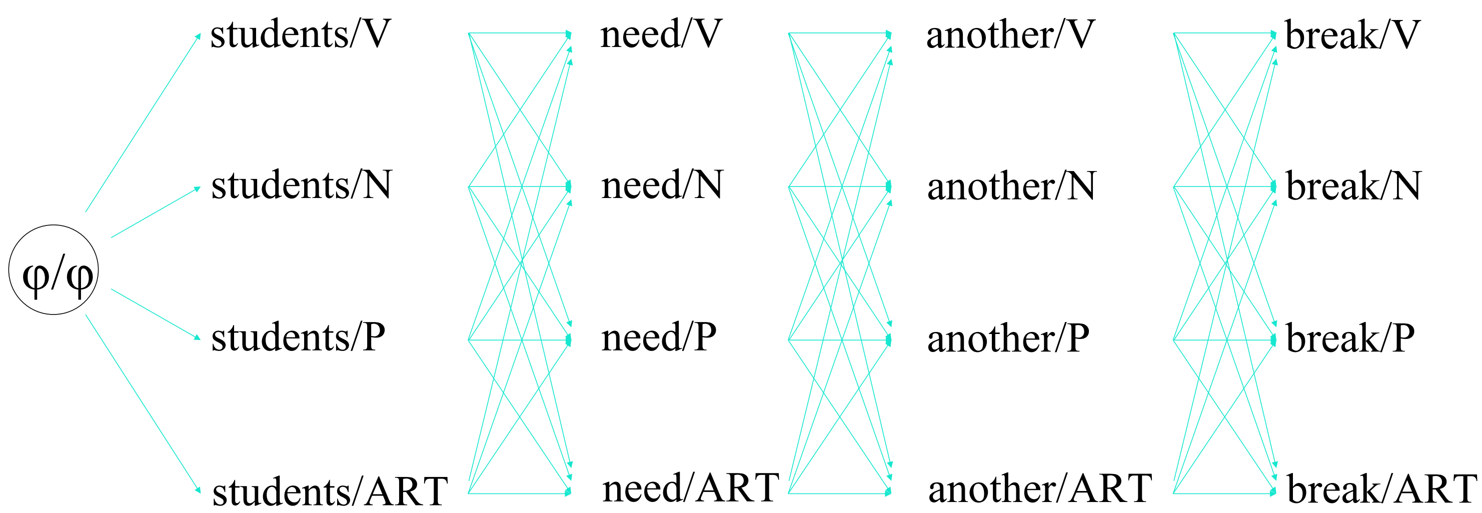
Uh-oh…Too many possible tag sequences to do this!!!

Sentence length m=20

Tagset of size T = 15

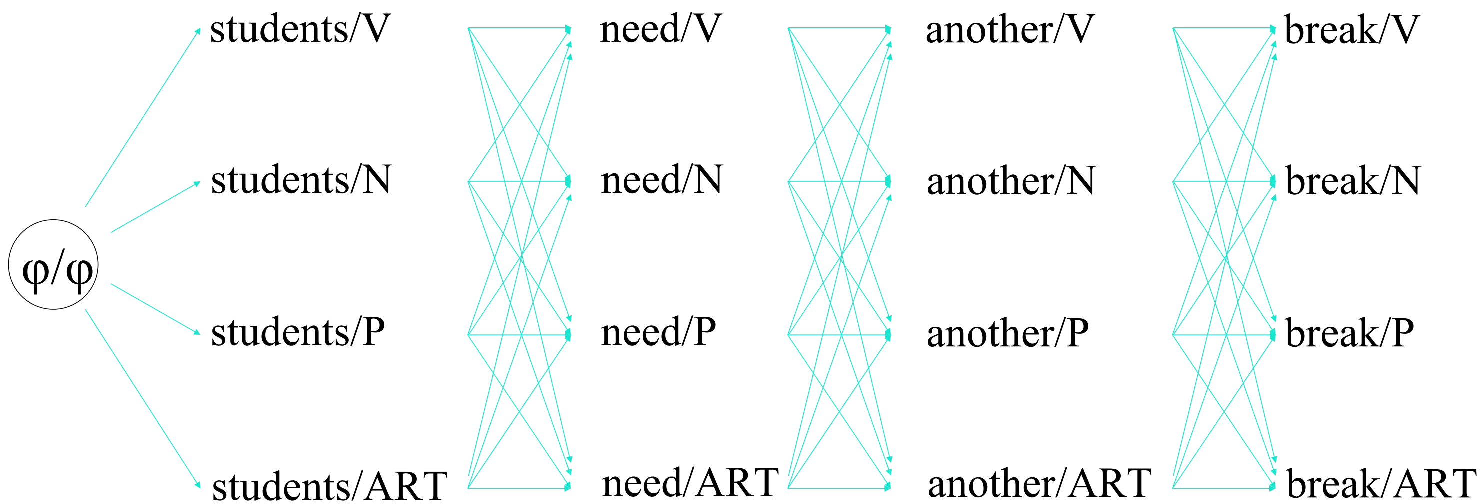
$T^m = 15^{20}$ tag sequences!!!

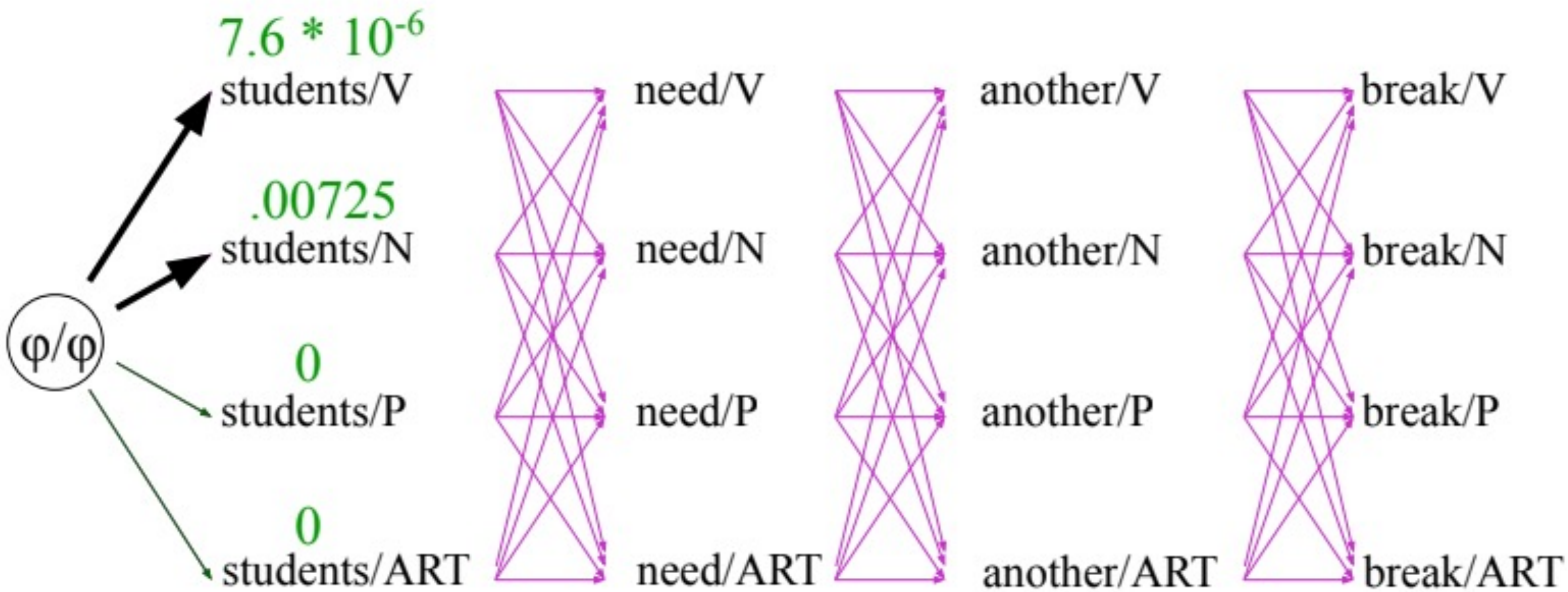
How do we avoid computing the probabilities for all possible paths?



Viterbi Algorithm Allows Efficient Search for the Most Likely Sequence

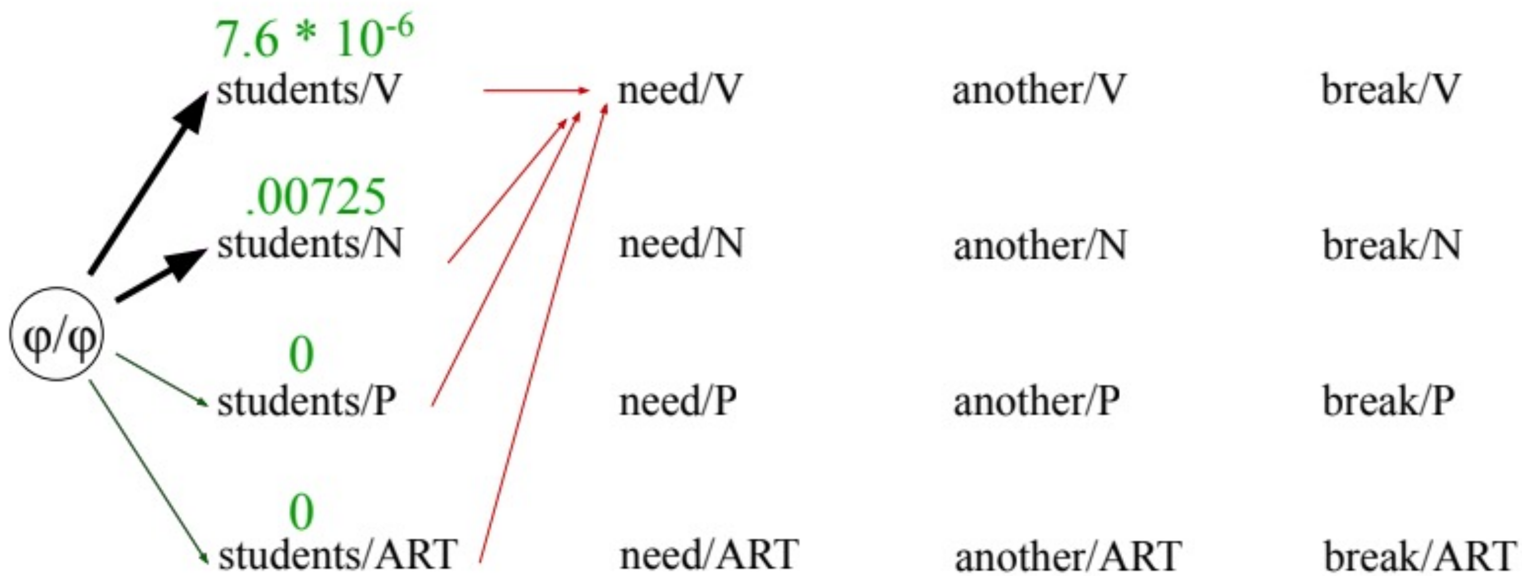
- Key idea: Markov assumptions mean that we do not need to enumerate all possible sequences
- Viterbi algorithm
  - Sweep forward, one word at a time, finding the most likely (highest-scoring) tag sequence ending with each possible tag
  - With the right bookkeeping, we can then “read off” the most likely tag sequence once we reach the end of the sentence


φ/φ
students/V
students/N
students/P
students/ART
need/V
need/N
need/P
need/ART
another/V
another/N
another/P
another/ART
break/V
break/N
break/P
break/ART



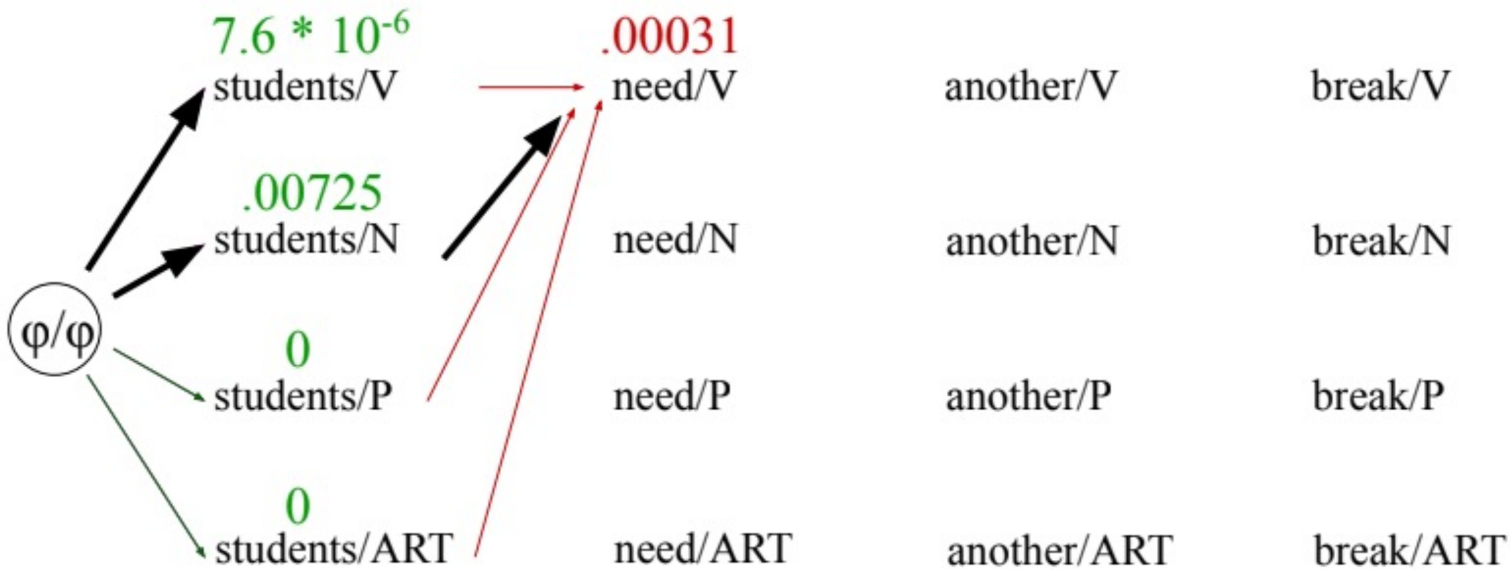
$v_0(j) = P(tag_0=j \mid \langle s \rangle) * P(students \mid tag_0=j)$

$vb_0(j) =$ <s>



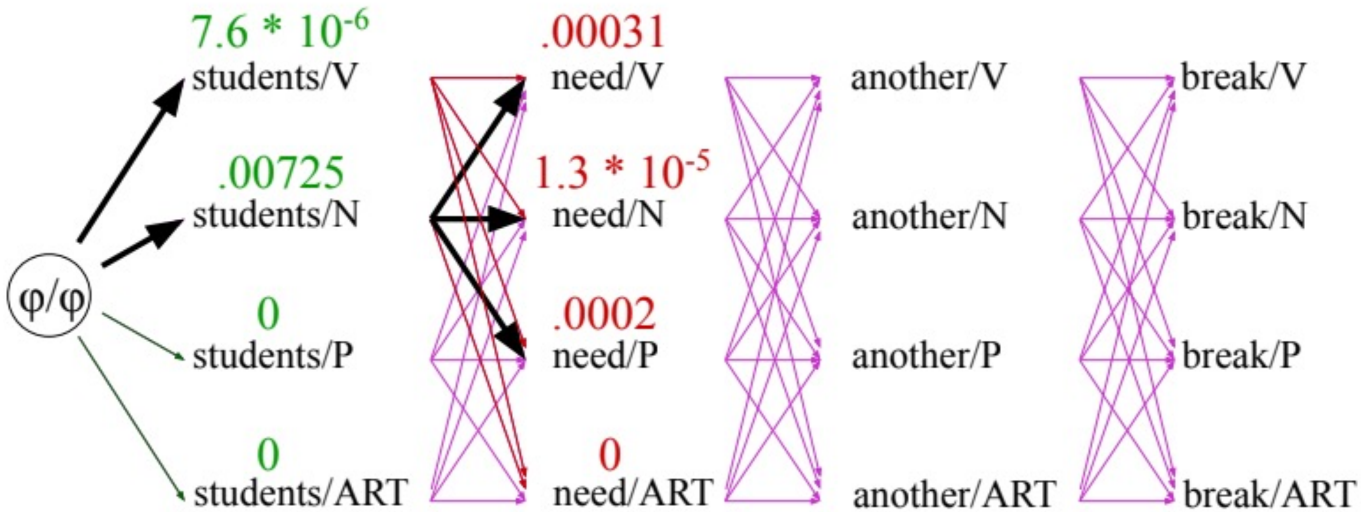
$v_1(j) = \max^{J}_{i=1} v_0(i) * P(tag_1{=}j \mid tag_0{=}i) * P(need \mid tag_1{=}j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$



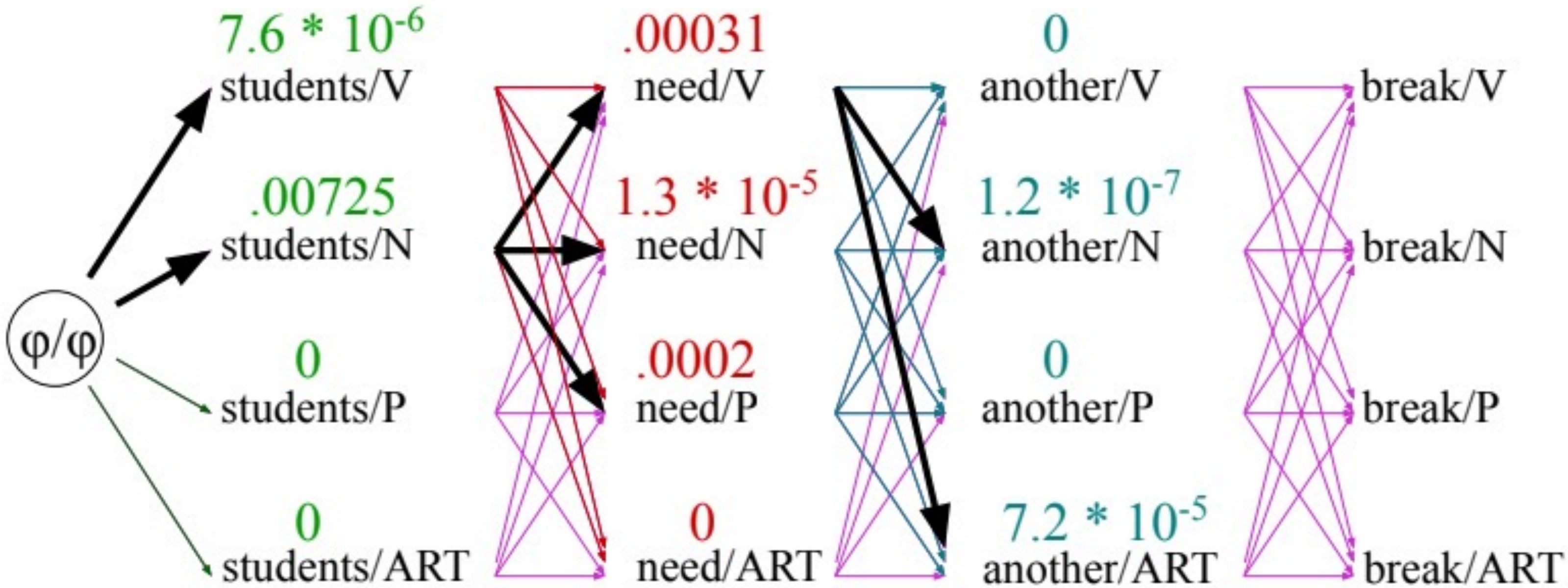
$v_1(j) = \max^{J}_{i=1} v_0(i) * P(tag_1=j \mid tag_0=i) * P(need \mid tag_1=j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$



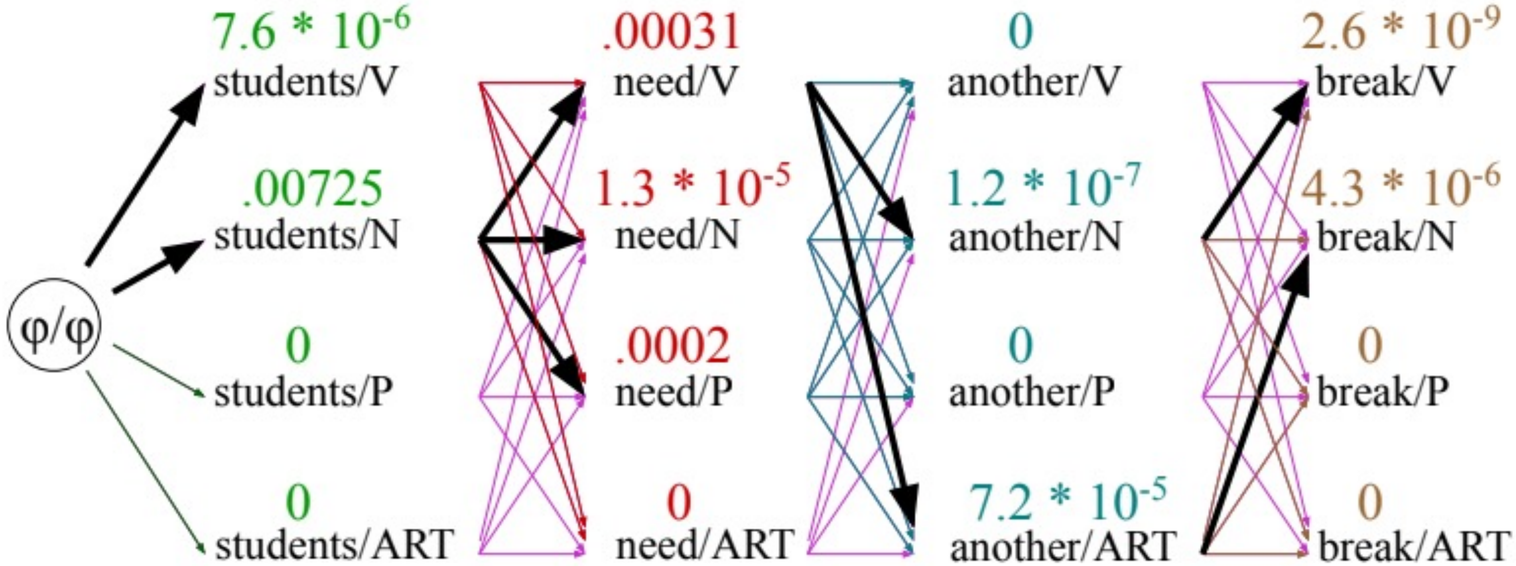
$v_1(j) = \max^{J}_{i=1} v_0(i) * P(tag_1{=}j \mid tag_0{=}i) * P(need \mid tag_1{=}j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$



$v_2(j) = \max^{J}_{i=1} v_1(i) * P(tag_2=j \mid tag_1=i) * P(another \mid tag_2=j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$

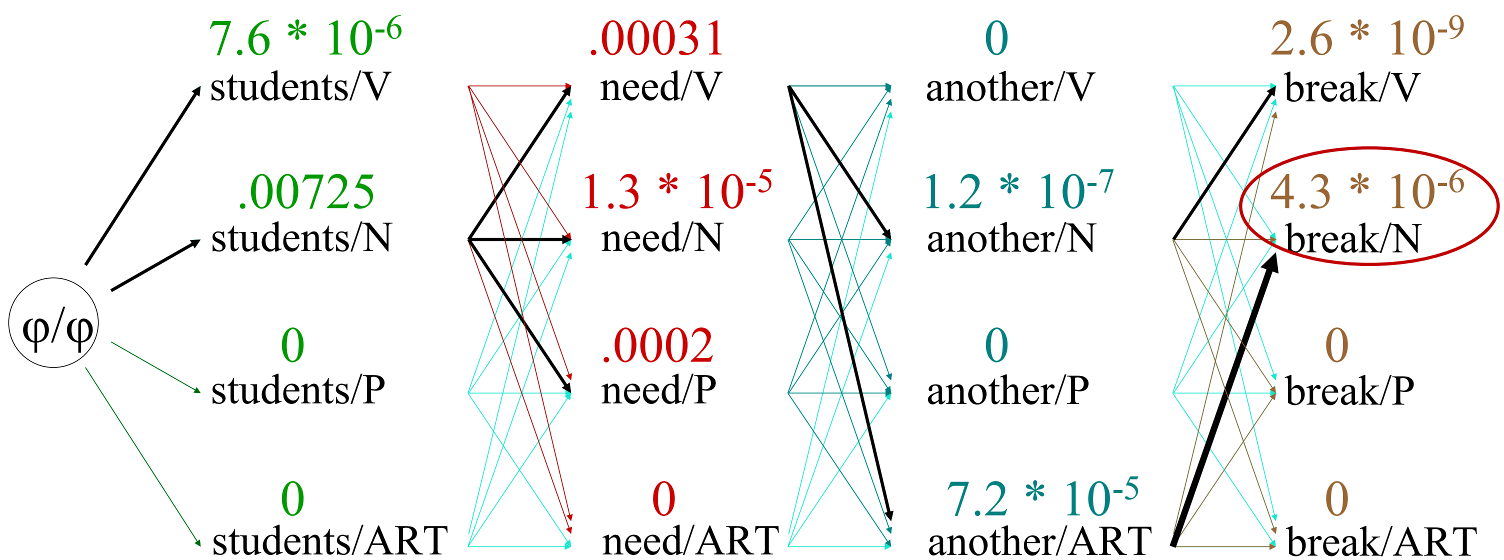


$v_3(j) = \max^{J}_{i=1} v_2(i) * P(tag_3=j \mid tag_2=i) * P(break \mid tag_3=j)$

$vb_t(j)$ = prev tag that maximizes $v_t(j)$

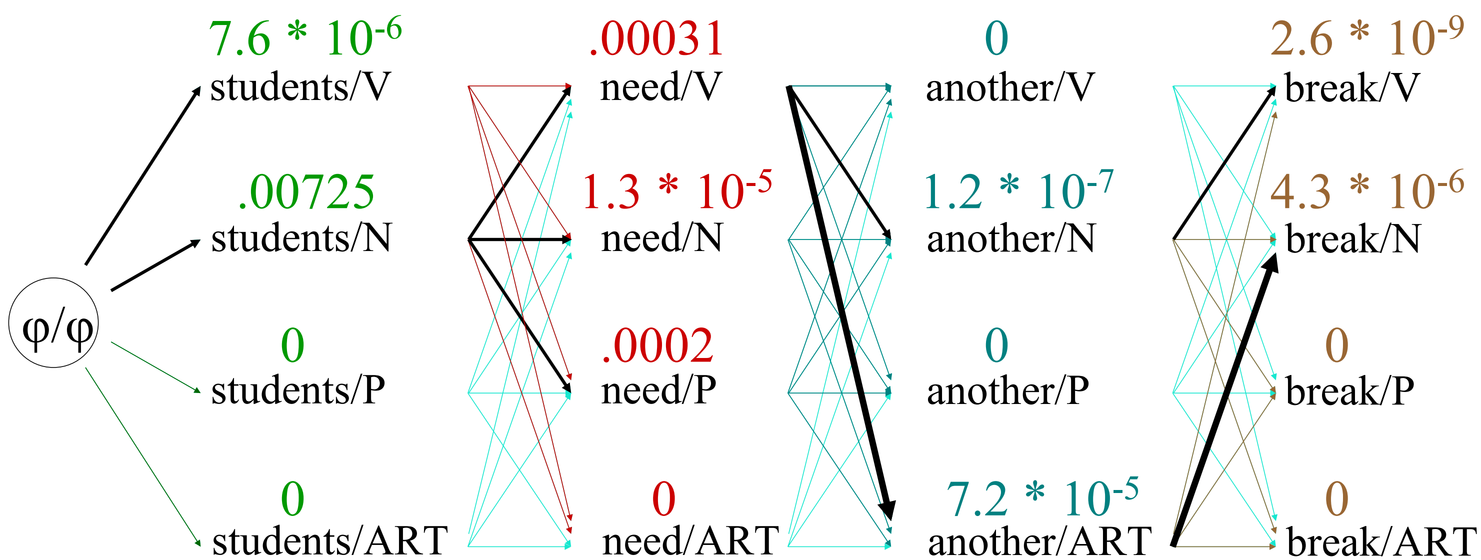
- To assign the maximum probability tag sequence, follow the backpointers that led to the largest product at v_3!

Viterbi Algorithm



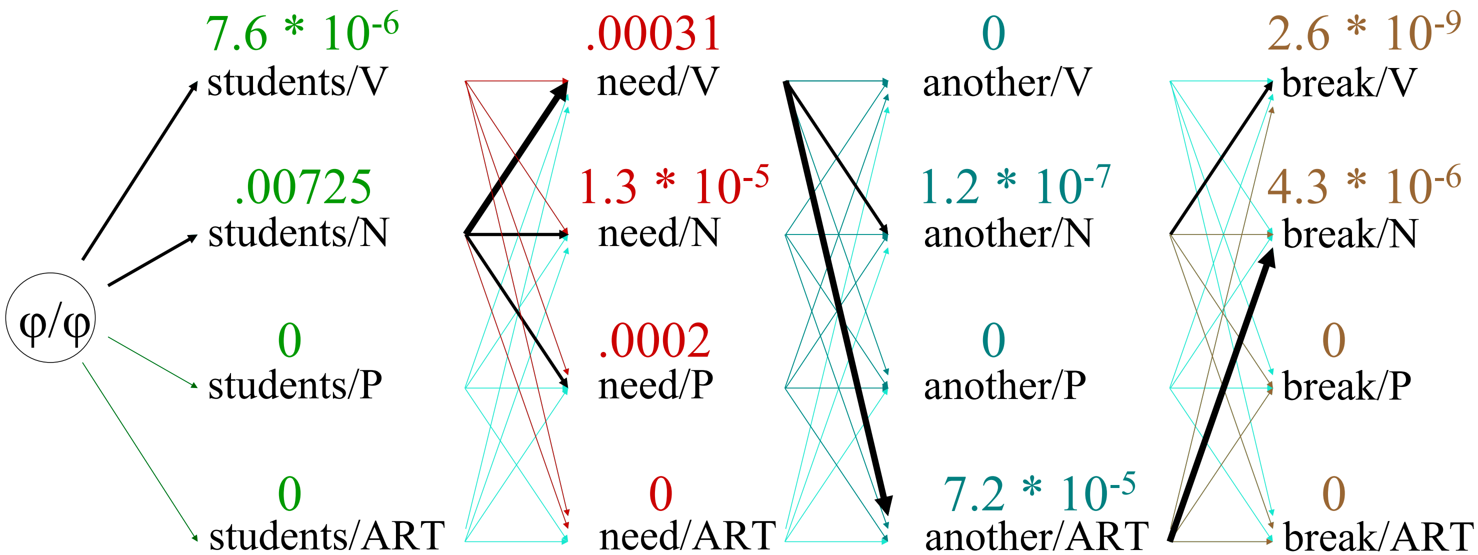
t_3= N

Viterbi Algorithm



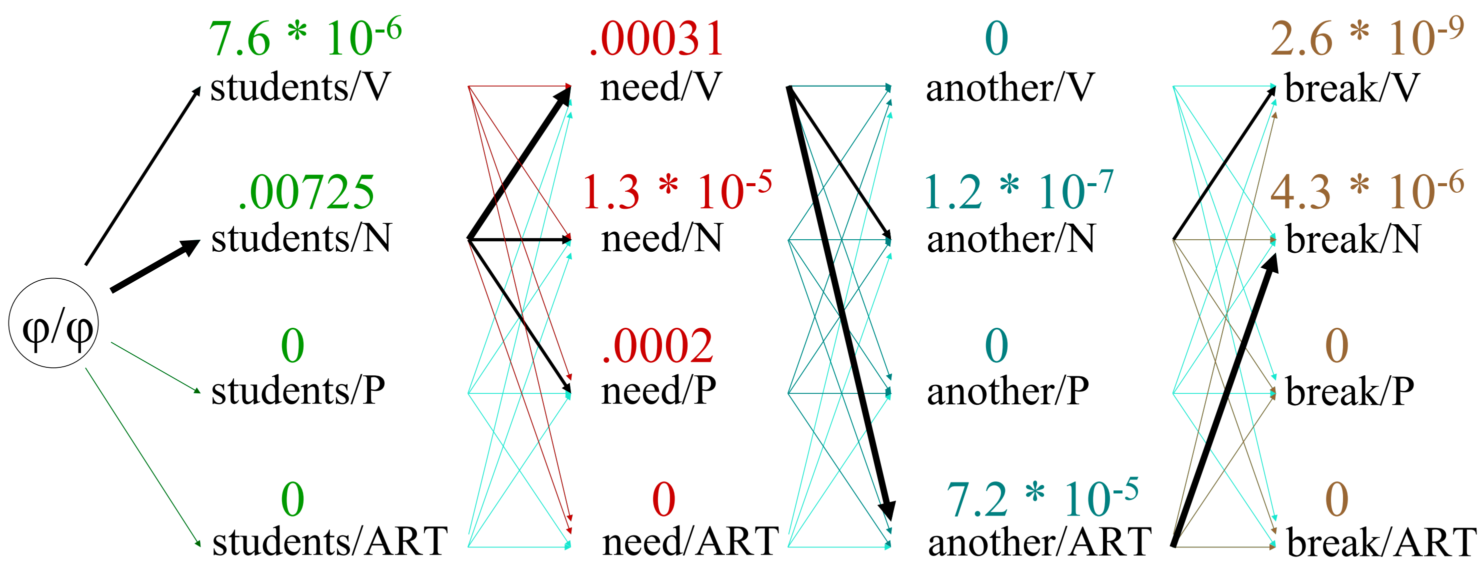
t_3= N, t_4= ART

Viterbi Algorithm



t_3= N, t_2= ART, t_1= V

Viterbi Algorithm



t_3= N, t_2= ART, t_1= V, t_0= N

Time/space complexity

- Space
  - Two c x n matrices
  - (and data structure for transition and lexical generation probabilities)
- Time
  - $O(c^2n)$ for forward pass
  - $O(n)$ for backward pass
  - Much better than the $O(c^n)$ brute force option

# of POS categories

length of sentence

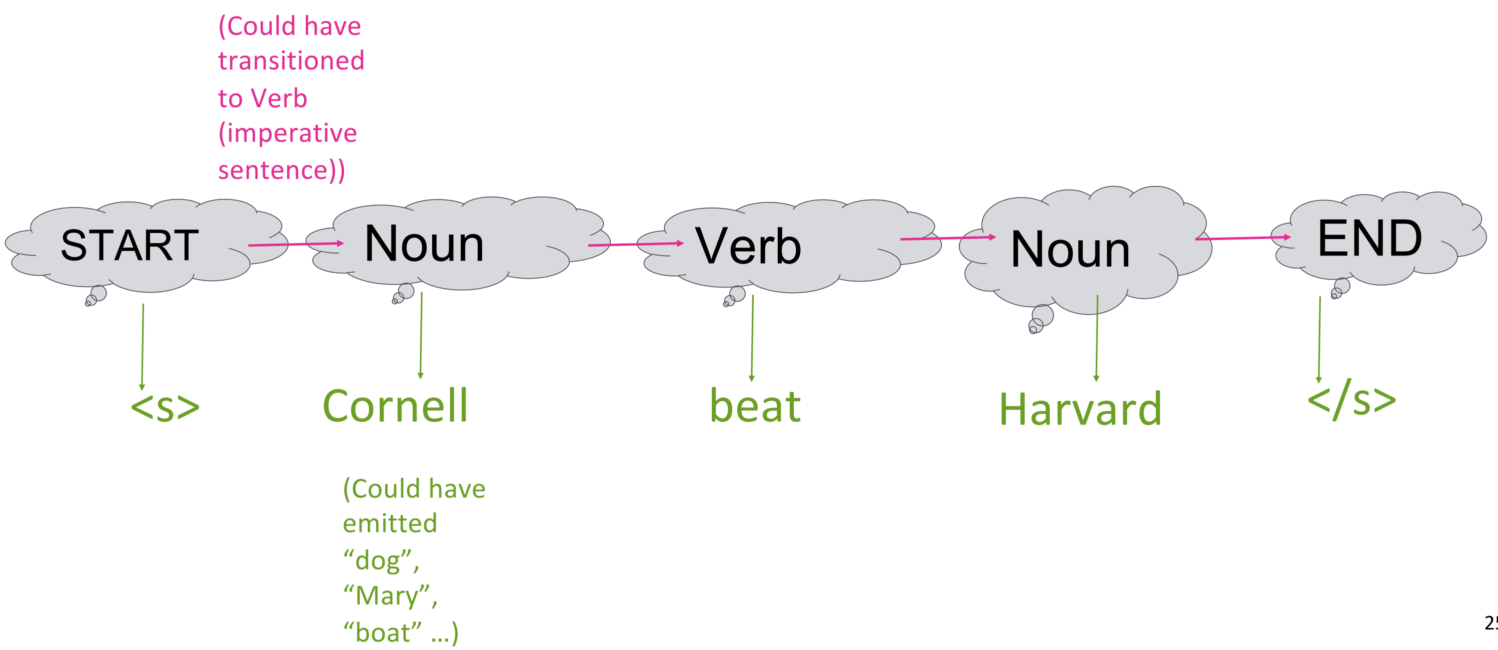
Today

- HMMs as a tagging technology: Viterbi
  - You will implement for HW1!!!
- **HMMs as a generative model**
- Where do the probabilities come from?
- Named entity tagging: the task for HW1!!!

HMMs as sentence generators

When in an underlying state (POS), generate a token.

Then, choose a next underlying state.



Today

- HMMs as a tagging technology: Viterbi
  - You will implement for HW1!!!
- HMMs as a generative model
- **Where do the probabilities come from?**
- Named entity tagging: the task for HW1!!!

Where do HMM transitions/emission probs come from?

Assume that we have *labelled data*:
For every observed token x_i , the (usually hidden) true tag c_i is given.

<s>/**<s>** I/**PP** am/**VBP** sitting/**VBG** in/**IN** Mindy/**NNP** ’s/**POS** restaurant/**NN** eating/**VBG** the/**DT** gefilte/**NN** fish/**NN** ./**.** </s>**/</s>**

- Looks like VBG generates things like “sitting” and “eating”; and a period (.) can be followed by </s>.

Warning: training data might omit <s>, <s>, </s>, </s>. You’ll want to insert them (implicitly or explicitly).

“Raw count” method for setting transition and emission probs

$$P_{HMM}(w_j \mid c) := \frac{\text{count } (\text{word } w_j \text{ in training with tag } c)}{\text{count } (\text{word tokens in training with tag } c)}$$

$$P_{HMM}(c' \mid c) := \frac{\text{count } (c \text{ followed by } c')}{\text{count } (c)}$$

Smoothing: "lack of evidence is not evidence of lack"

An unseen event isn't necessarily impossible! Safer to have all probs be non-zero.

One common smoothing technique: add-k.

P(b | a) := [Count(a b) + k] …

…divided by the normalization term:

sum over all possible b' of [C(a b') + k]

Today

- HMMs as a tagging technology: Viterbi
  - You will implement for HW1!!!
- HMMs as a generative model
- Where do the probabilities come from?
- **Named entity tagging: the task for HW1!!!**

Named Entity Recognition

Identify all:

- Named locations, named persons, named organizations, dates, times, monetary amounts…
- Fixed set of NE types

Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	**Turing** is a giant of computer science.
Organization	ORG	companies, sports teams	The **IPCC** warned about the cyclone.
Location	LOC	regions, mountains, seas	The **Mt. Sanitas** loop is in **Sunshine Canyon**.
Geo-Political Entity	GPE	countries, states, provinces	**Palo Alto** is raising the fees for parking.
Facility	FAC	bridges, buildings, airports	Consider the **Golden Gate Bridge**.
Vehicles	VEH	planes, trains, automobiles	It was a classic **Ford Falcon**.

Figure 17.1 A list of generic named entity types with the kinds of entities they refer to.

NER output

Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

Named Entity Recognition (NER): note the multi-word named entities, like “North America”

In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the ‘future AI PERSON platforms’. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Lots of errors!!!

Ambiguity in NER

Name	Possible Categories
Washington	Person, Location, Political Entity, Organization, Vehicle
Downing St.	Location, Organization
IRA	Person, Organization, Monetary Instrument
Louis Vuitton	Person, Organization, Commercial Product

Figure 17.2 Common categorical ambiguities associated with various proper names.

[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary seatbelt law.
The [$_{\text{VEH}}$ Washington] had proved to be a leaky ship, every passage I made...

Figure 17.3 Examples of type ambiguities in the use of the name *Washington*.

NE Recognition

- Identify the text **spans** that correspond to the proper names (or dates, times, money expressions)
  - How do we describe a **chunk** of text using individual-word tags?
- Assign the correct named entity (NE) type

BIO tag set for NER

- Allows distinguishing adjacent NEs
  - We'll fly to **New Orleans** **Friday**
- B_{XXX}: First (ie. Beginning) token in an NE of type XXX
- I_{XXX}: Inside of an entity type XXX
- O: Outside of all NEs

BIO Tagging

B: token that *begins* a span

I: tokens *inside* a span

O: tokens outside of any span

# of tags (where n is #entity types):

1 O tag,

n B tags,

n I tags

total of *2n+1*

Words	**BIO Label**
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

HMMs for NE detection

Just like in POS tagging

- States ***Q***
  - BIO tags
- Observations ***O***
  - Word tokens
- Transition Probabilities ***A***
  - P $(BIOtag_i \mid BIOtag_{i-1})$
- Emission (lexical generation) Probabilities ***B***
  - P $(w_i \mid BIOtag_i)$

Find most likely BIO tag sequence using Viterbi

Reconstruct the NEs from the BIO tags

Take-aways

- HMMs as a tagging technology: the Viterbi algorithm for efficiently assigning the highest probability tag sequence
- HMMs as a generative model (just 1 slide)
- Where do the probabilities come from? Labeled data
- Named entity tagging: the task for HW1!!!