Sequences

CS4248 Natural Language Processing

Week 07

Rahul BAID, Reza QORIB and Min-Yen KAN

7

Recap of Week 06

Move from term–document matrix to a term–context matrix

- Solves semantically relatedness

Embed the resultant term–context vectors into a denser space

- The side effect is the objective!
- Solves sparsity problem

Vectorial differences yields semantics relationships

Many extensions, we'll see some later

Week 07 Agenda

Sequences

Parts of Speech (POS)

- POS Tagging

Hidden Markov Model (HMM)

Forward Computation

Viterbi Algorithm

Sequences

Matters Ordering

Bag of words loses semantic information

Bob kills mosquitos using the book of Hamlet

vs

Hamlet kills Bob using the book of mosquitos

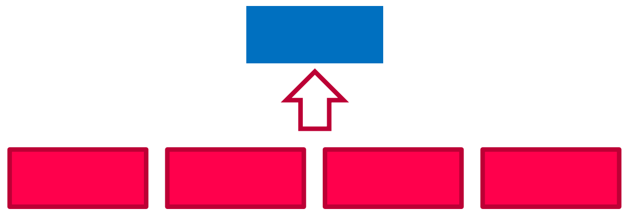
The food tastes good and does not look bad

vs

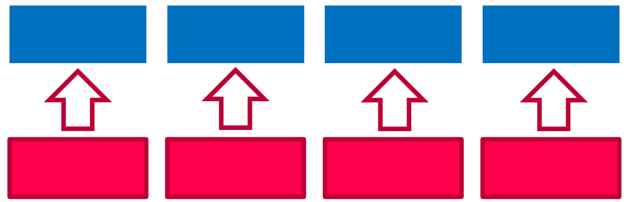
The food tastes bad and does not look good

Type of Sequence Tasks

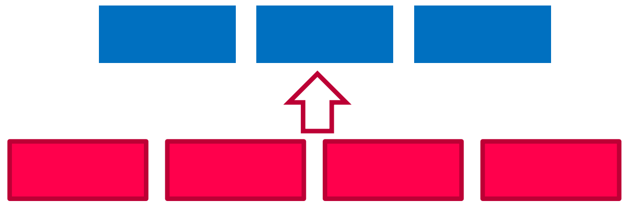
Sequence classification ($N \rightarrow 1$)



Sequence labelling ($N \rightarrow N$)



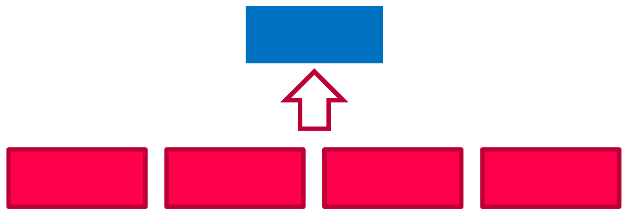
Sequence to sequence ($N \rightarrow M$)



Type of Sequence Tasks

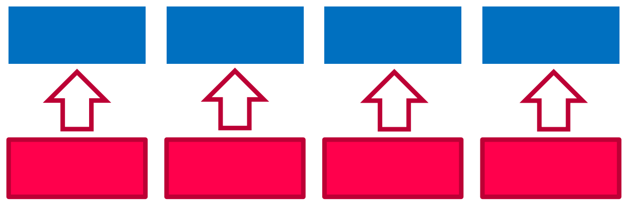
Sequence classification (N → 1)

- Sentiment analysis
- Sentence factual checking



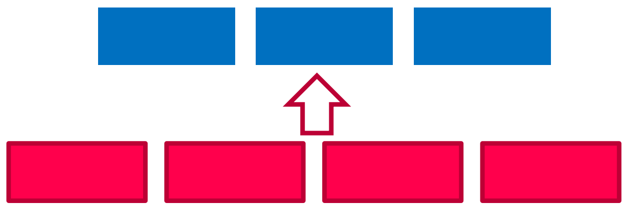
Sequence labelling (N → N)

- Named-entity recognition
- Part-of-speech tagging



Sequence to sequence (N → M)

- Machine translation



Parts of Speech

A case study of sequence tagging

National University of Singapore | School of Computing

Parts of Speech

Also called word classes or syntactic categories

- Every word in the vocabulary belongs to one or more of these classes
- ~ 8 of them (in English)

Diyi Yang (GaTech)

POS – Two broad categories

Closed class

- Small fixed membership
- Usually function words
  (words which play a grammatical role)

- Example:
  - Prepositions: on, in, of
  - Pronoun: she, him, them
  - Particles: up, down, on, off
  - Determiners: a, an, the
  - Conjunctions: and, but, or
  - Numerals: one, two, three
  - Auxiliary verbs: was, should

Open class

- New vocabulary items can be created
- Most languages have four:
  - Noun: Singapore, boy/boys
  - Verb: eat/eats/eaten
  - Adjective: good, bad, worse
  - Adverb: quickly, extremely

POS Tagging

The process of assigning a part of speech to words in a text

Input: a sequence of tokenized words and a tagset
Output: a sequence of tags, one per token

- A word can belong to more than POS.
  For example, the word *back*
  - The *back* door (adjective)
  - On my *back* (noun)
  - Win the voters *back* (adverb)
  - Promised to *back* the bill (verb)

POS Tagging

Why is it useful?

First step for a vast number of practical tasks

- Named Entity Recognition (Trump = proper or common noun?)
- Information Extraction (finding relations, names, etc.)
- Parsing (need to know if word is noun or verb before we create dependency tree)
- Speech synthesis/recognition (DIScount-disCOUNT, how do we pronounce *lead*)

POS Tagging

Words are ambiguous, tagging is a disambiguation task

How common is this ambiguity?

		WSJ		Brown	
Types:					
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)
Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

Goal: find the correct tag for each particular token.

Diyi Yang (GaTech)

POS Tagging

A simple way to deal with ambiguity

Most frequent class baseline: Always compare a classifier against a baseline at least as good as the most frequent class baseline:

- assign each word to the class it occurred in most often in the training set
- assign unknown words as nouns

POS Tagging

How good is this baseline: Accuracy for POS taggers is measured as the percent of tags that are correctly labeled as compared to human labels on a test set.

Model	Accuracy (on a news corpus: *Wall Street Journal*)
Most frequent class baseline	92.34%
Flair (SOTA)	97.85%

Baseline already has >90% accuracy

- many words are unambiguous
- you get points for them (*the*, *a*) and for punctuation marks

Penn Treebank Tagset

45 possible tags

Used to label many corpora

Example:

Preliminary/JJ findings/NNS were/VBD reported/VBN in/IN today/NN ’s/POS New/NNP England/NNP Journal/NNP of/IN Medicine/NNP ./.

Penn Treebank Tagset

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	$
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	#
JJS	superlative adj	*wildest*	SYM	symbol	+,%, &	“	left quote	‘ or “
LS	list item marker	*1, 2, One*	TO	“to”	*to*	”	right quote	’ or ”
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Penn Treebank Tagset – More examples

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS .

Mrs/NNP Shaefer/NNP never/RB got/VBD around/RP
to/TO joining/VBG

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

Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

POS Tagging via Supervised Learning

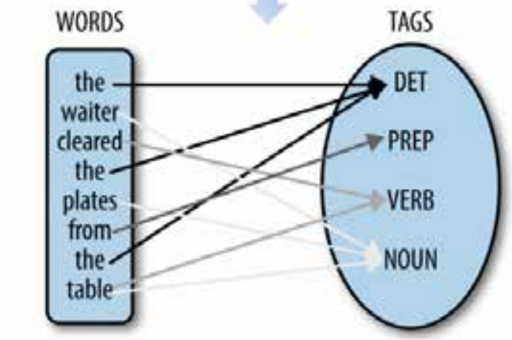
Linguists annotate input sentences with their POS.

We use this as input training data to build a supervised model.

Tagset:
DT/JJ/NN/VBD

POS Tagger

The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.



Diyi Yang (GaTech)

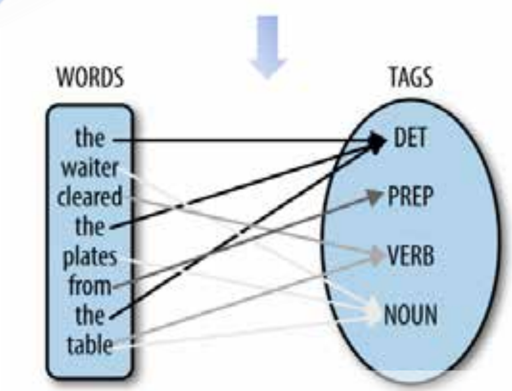
POS Induction via Unsupervised Learning

We assume we only have unannotated corpora:

Tagset:
DT/JJ/NN/VBD

The grand jury commented on a number of other topics.

POS Tagger

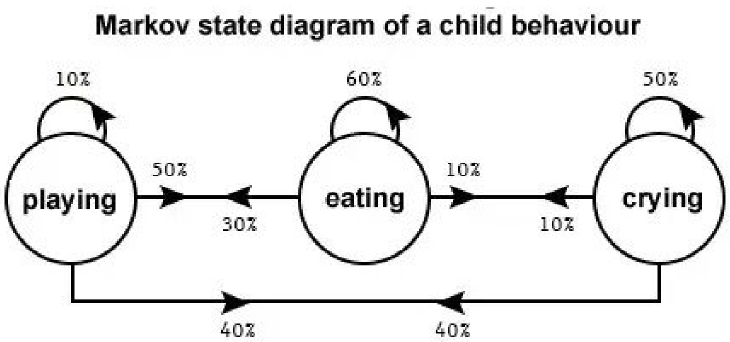


Diyi Yang (GaTech)

Hidden Markov Model (HMM)

What is an HMM

Based on the Markov Chain



Embodies the Markov assumption

“when predicting the future, the past doesn’t matter, only the present” – Markov

“hidden” $\longrightarrow$ contains unobserved event

Image Credit: https://www.quora.com/What-is-a-state-of-Markov-chain

Component of HMM

- Set of states (Q)
- Transition probability (a_{ij})

  the probability of moving from state i to state j
- Sequence of observations (o_t)

  observation drawn for each time-step t from vocabulary V
- Emission probability ($b_i(o_t)$)

  the probability of observation o generated from a state i
- Initial probability distribution (π)

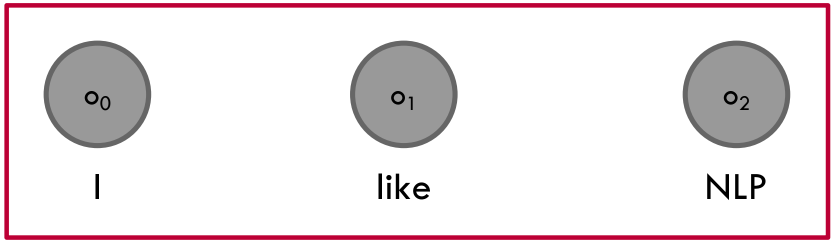
  the probability for the Markov chain to start at state i

https://web.stanford.edu/~jurafsky/slp3/A.pdf

HMM Components: Observations

I	like	NLP

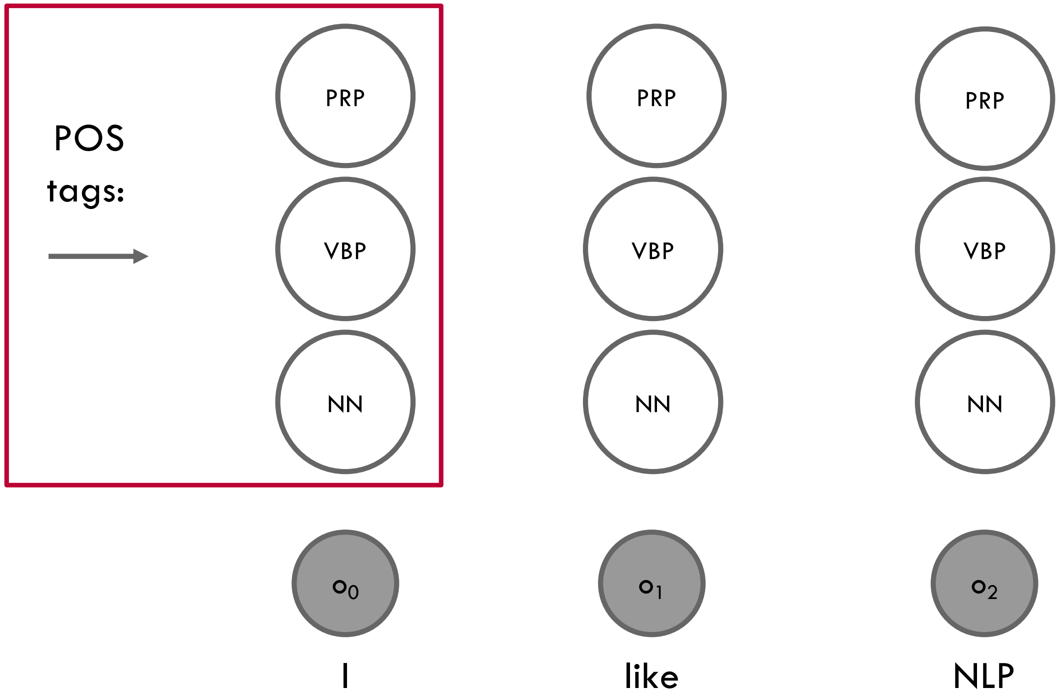
HMM Components: Emission Probabilities



Emission prob.:
$b_i(o_t) = p(o_t \mid q_i)$

Given a state q_i, what's the probability that it emits o_t?

HMM Components: States



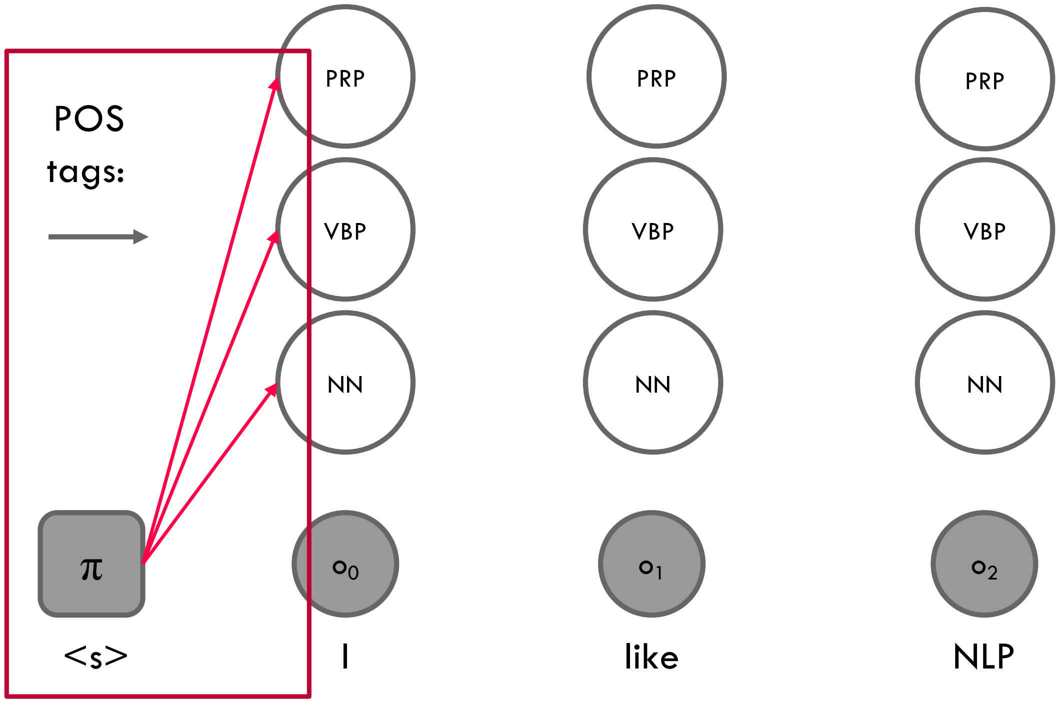
States:

$q_1, q_2, \dots, q_N$

N possible labels:

PRP, VBP, NN

HMM Component:
Initial Start Probability



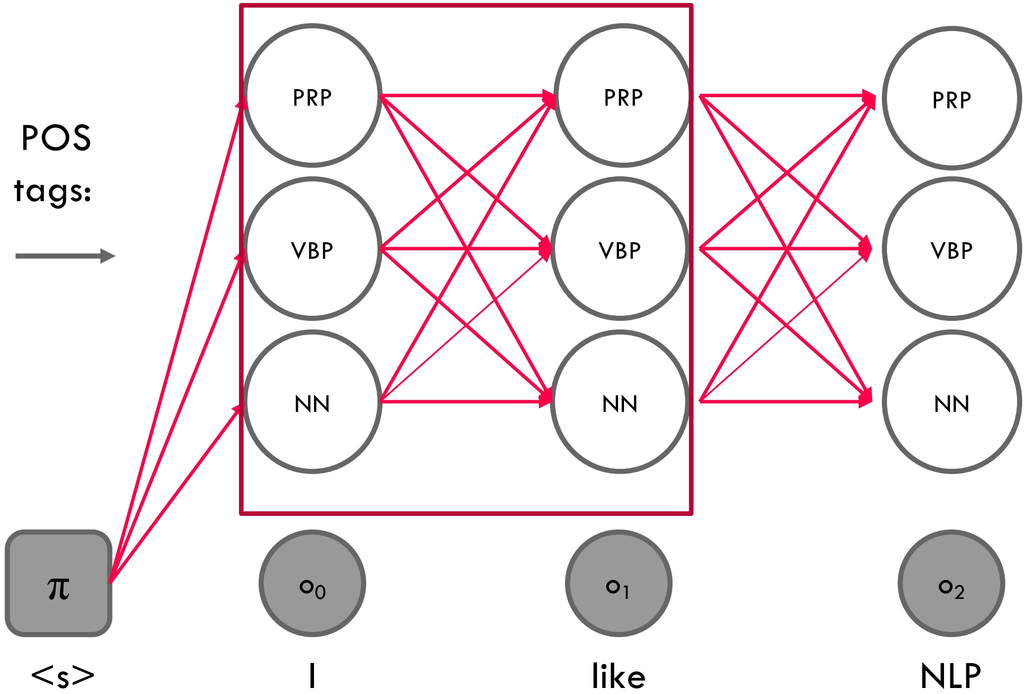
Start probability:

$\pi_i = p(q_i \mid <s>)$

What's the probability that a sequence starts with label q_i?

Component of HMM:

Initial Transition Distribution



Transition prob.:

$$a_{ij} = p(q_i \mid q_j)$$

Given a label q_j, what's the probability that it is followed by label q_i?

Getting the probabilities

Maximum Likelihood Estimates:

$$\pi_i = p(q_i|\langle s\rangle) = \frac{c(\langle s\rangle q_i)}{c(\langle s\rangle)}$$

$$a_{ij} = p\left(q_i|q_j\right) = \frac{c(q_j q_i)}{c(q_j)}$$

$$b_i(o_t) = p(o_t|q_i) = \frac{c(o_t, q_i)}{c(q_i)}$$

Getting the probabilities

Maximum Likelihood Estimates!

$$\pi_i = p(q_i|\langle s \rangle) = \frac{c(\langle s \rangle q_i)}{c(\langle s \rangle)}$$

$$\frac{\text{Number of sequences that starts with label } q_i}{\text{Number of sequences}}$$

$$a_{ij} = p(q_j|q_i) = \frac{c(q_i q_j)}{c(q_i)}$$

$$\frac{\text{Occurrences of label } q_i \text{ followed by label } q_j}{\text{Occurrences of label } q_i}$$

$$b_i(o_t) = p(o_t|q_i) = \frac{c(o_t, q_i)}{c(q_i)}$$

$$\frac{\text{Occurrences of word } o_t \text{ labeled as } q_i}{\text{Occurrences of label } q_i}$$

Forward Computation

What's the likelihood of a sequence?

School of Computing

Forward computation

Calculate the probabilities, given the HMM & sentence length T

$O = o_1 o_2 o_3 \cdots o_T \rightarrow$ observations, words from 1 to T

$Q = q_1 q_2 q_3 \cdots q_T \rightarrow$ order of the labels for each word, q_0=<s>

$$P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(o_i|q_i) \times P(q_i|q_{i-1})$$

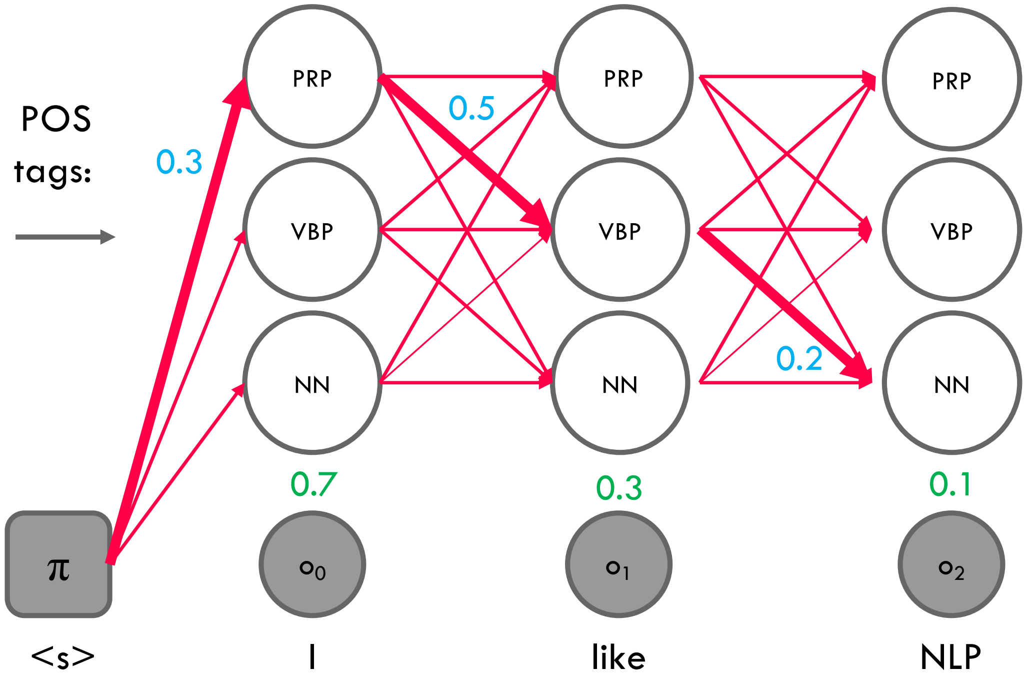
$$P(O,Q) = P(o_1|q_1) \times P(q_1|q_0) \times \cdots \times P(o_T|q_T) \times P(q_T|q_{T-1})$$

Forward computation

$$P(O,Q) = P(O|Q)\times P(Q) = \prod_{i=1}^{T} P(o_i|q_i)\times P(q_i|q_{i-1})$$

$$P(\text{"I like NLP"}, PRP - VBP - NN) =$$
$$P(I|PRP)\times P(PRP|\langle s\rangle)\times$$
$$P(Like|VBP)\times P(VBP|PRP)\times$$
$$P(NLP|NN)\times P(NN|VBP)$$

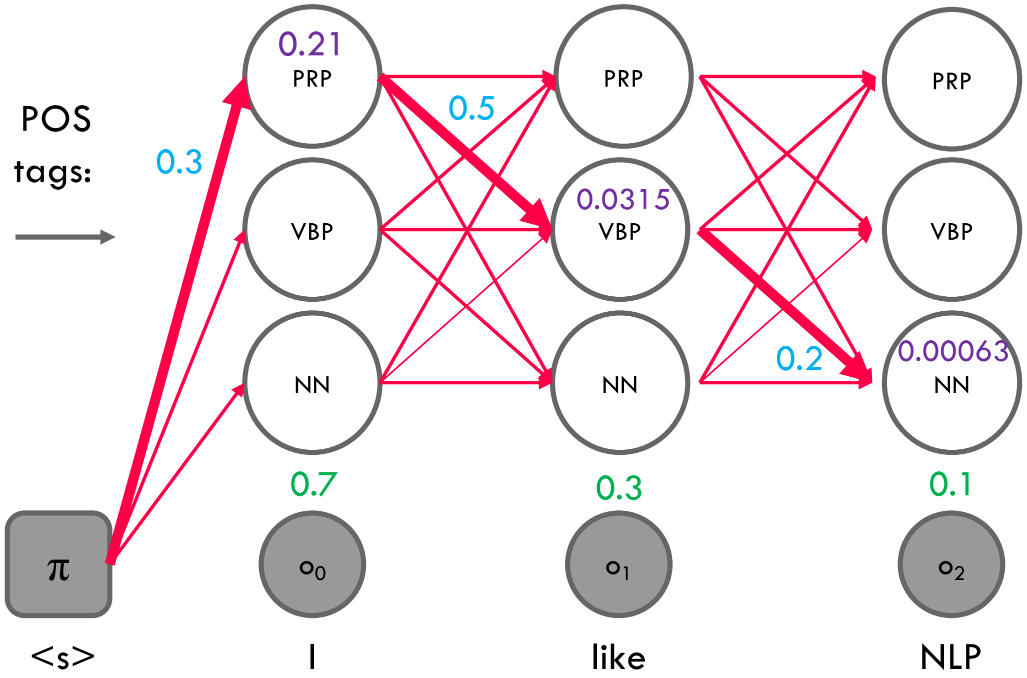
Forward Computation



$$P\left(\begin{matrix}\text{"I like NLP"},\\ PRP-VBP-NN\end{matrix}\right)$$

$$= 0.3 \times 0.7 \times 0.5 \times 0.3 \times 0.2 \times 0.1$$

Forward Computation



$$P\binom{\text{"I like NLP"},}{PRP - VBP - NN}$$

= 0.3 x 0.7 x
0.5 x 0.3 x
0.2 x 0.1

= 0.21 x
0.5 x 0.3 x
0.2 x 0.1

= 0.0315 x
0.2 x 0.1

= 0.00063

Viterbi Decoding

What's the most likely (hidden state) path?

NUS National University of Singapore | School of Computing

Decoding task

Determine the underlying sequence of variables that generate the observation

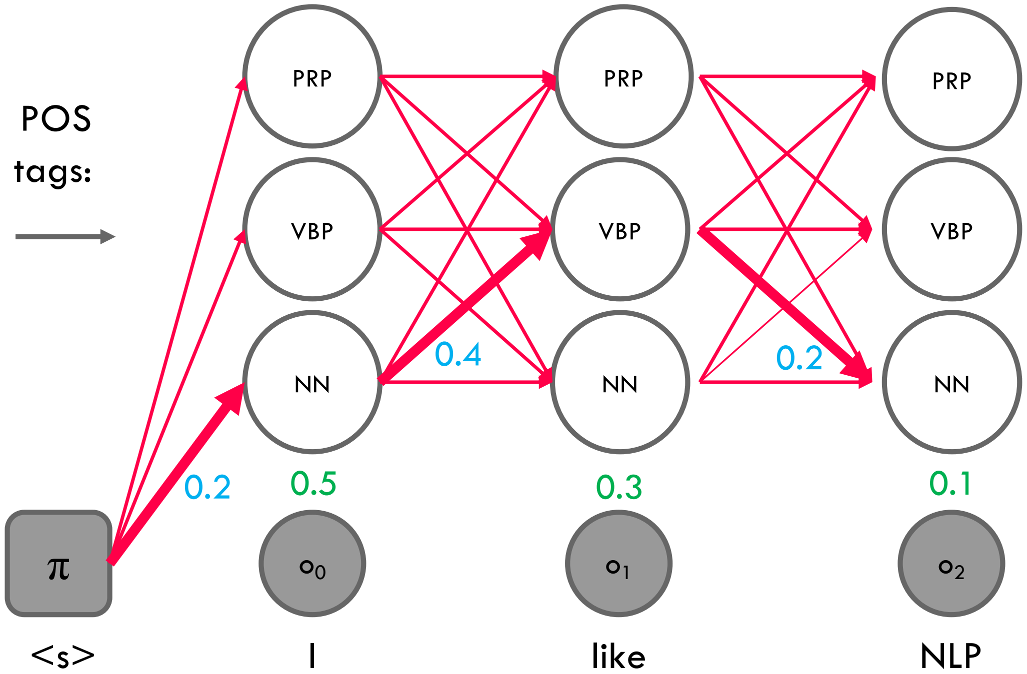
- What is the most probable labels for the sentence "I like NLP"?.

$$Q = \underset{q_1 \dots q_T}{\operatorname{argmax}} \prod_{i=1}^{T} P(o_i|q_i) \times P(q_i|q_{i-1})$$

Try all combinations?

We can save the states and use dynamic programming

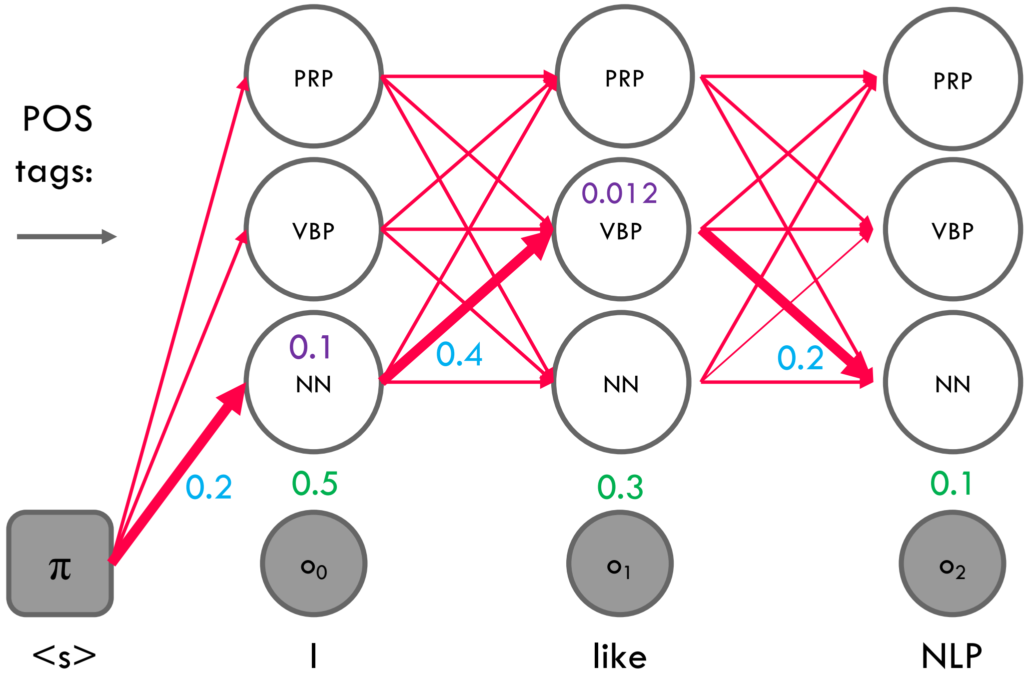
Forward Computation



$$P\left(\begin{matrix}\text{"I like NLP"},\\ NN - VBP - NN\end{matrix}\right)$$

$$= 0.2 \times 0.5 \times 0.4 \times 0.3 \times 0.2 \times 0.1$$

Forward Computation



$$P\left(\begin{matrix}\text{"I like NLP"},\\ NN - VBP - NN\end{matrix}\right)$$

$= 0.2 \times 0.5 \times$
$\quad 0.4 \times 0.3 \times$
$\quad 0.2 \times 0.1$

$= 0.1 \times$
$\quad 0.4 \times 0.3 \times$
$\quad 0.2 \times 0.1$

$= 0.012 \times$
$\quad 0.2 \times 0.1$

$= 0.00024$

Save the state

$$P(\text{"I like NLP"}, PRP - VBP - NN) =$$
$$P(I|PRP) \times P(PRP|\langle s \rangle) \times$$
$$P(Like|VBP) \times P(VBP|PRP) \times$$
$$P(NLP|NN) \times P(NN|VBP)$$

$$= v_1(PRP) \times$$
$$P(Like|VBP) \times P(VBP|PRP) \times$$
$$P(NLP|NN) \times P(NN|VBP)$$

$$= v_2(VBP) \times$$
$$P(NLP|NN) \times P(NN|VBP)$$

$$P\begin{pmatrix}\text{"I like NLP"},\\ PRP - VBP - NN\end{pmatrix}$$

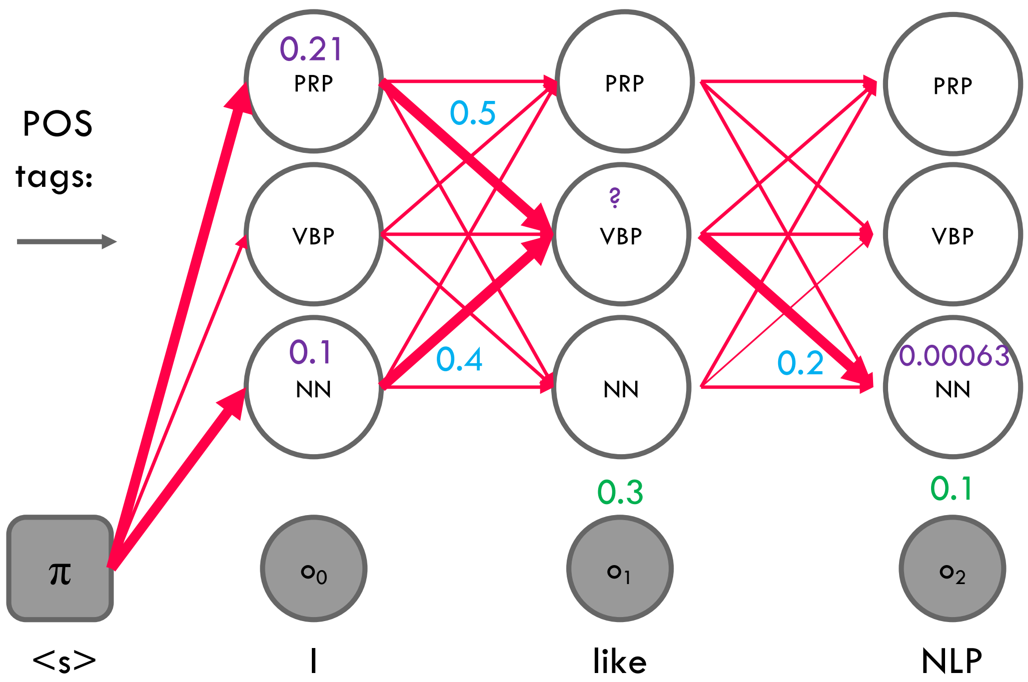
= 0.3 x 0.7 x
0.5 x 0.3 x
0.2 x 0.1

= 0.21 x
0.5 x 0.3 x
0.2 x 0.1

= 0.0315 x
0.2 x 0.1

= 0.00063

Which path to choose?



$P\left(\begin{matrix}\text{"I like NLP"}, \\ PRP - VBP - NN\end{matrix}\right)$

$= 0.0315 \times 0.2 \times 0.1$

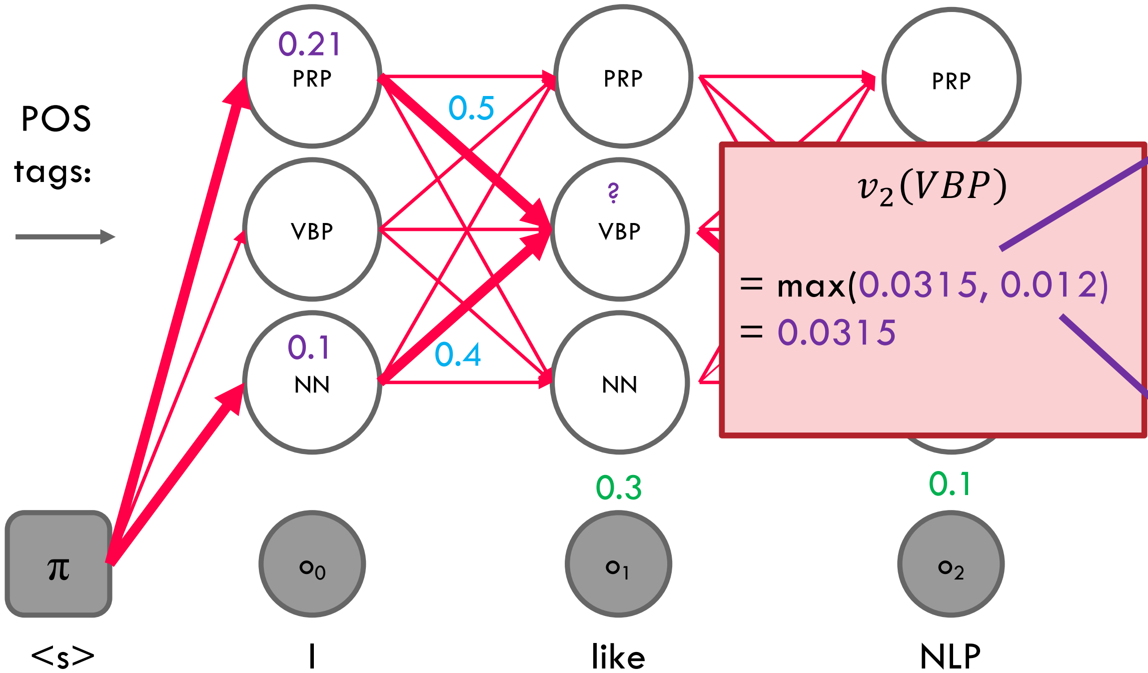
$= 0.00063$

$P\left(\begin{matrix}\text{"I like NLP"}, \\ NN - VBP - NN\end{matrix}\right)$

$= 0.012 \times 0.2 \times 0.1$

$= 0.00024$

Which path to choose?



$$P\binom{\text{"I like NLP"},}{PRP - VBP - NN}$$

= 0.0315 x
0.2 x 0.1

= 0.00063

$$P\binom{\text{"I like NLP"},}{NN - VBP - NN}$$

= 0.012 x
0.2 x 0.1

= 0.00024

Formal definition

Save the probability of best path to state j in step $t \longrightarrow v_t(j)$

$$v_t(j) = \max_{q_1 \dots q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j|\lambda)$$

$$v_t(j) = \max_{i=1 \dots N} v_{t-1}(i) \times p(q_j|q_i) \times p(o_t|q_j)$$

Highest probability of getting to state i in step $t-1$

Transition from state i to state j

Emision of o_t from state j

Formal definition

Save the probability of best path to state j in step $t \longrightarrow v_t(j)$

$$v_t(j) = \max_{q_1 \dots q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j|\lambda)$$

$$v_t(j) = \max_{i=1\dots N} v_{t-1}(i) \times p(q_j|q_i) \times p(o_t|q_j)$$

$$v_t(j) = \max_{i=1\dots N} v_{t-1}(i) a_{ij} b_j(o_t)$$

Remember,

$$a_{ij} = p(q_j|q_i)$$
$$b_i(o_t) = p(o_t|q_i)$$

1. **Initialization:**

$$
\begin{aligned}
v_1(j) &= \pi_j b_j(o_1) \qquad 1 \le j \le N \\
bt_1(j) &= 0 \qquad\qquad\quad 1 \le j \le N
\end{aligned}
$$

2. **Recursion**

$$
\begin{aligned}
v_t(j) &= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \\
bt_t(j) &= \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T
\end{aligned}
$$

3. **Termination:**

$$
\begin{aligned}
\text{The best score:} \quad P* &= \max_{i=1}^{N} v_T(i) \\
\text{The start of backtrace:} \quad q_T* &= \operatorname*{argmax}_{i=1}^{N} v_T(i)
\end{aligned}
$$

https://web.stanford.edu/~jurafsky/slp3/A.pdf

function VITERBI(*observations* of len *T*,*state-graph* of len *N*) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi[N,T]*
for each state *s* **from** 1 **to** *N* **do** ; initialization step
    $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
    $backpointer[\mathrm{s},1] \leftarrow 0$
for each time step *t* **from** 2 **to** *T* **do** ; recursion step
  for each state *s* **from** 1 **to** *N* **do**
    $viterbi[\mathrm{s,t}] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[\mathrm{s,t}] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
bestpath ← the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*

https://web.stanford.edu/~jurafsky/slp3/A.pdf

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	1Fruit	2Flies	3Like	4Bananas	</s>
NN					
VBZ					
IN					

Example probabilities

$$p(NN|\langle s\rangle) = 0.7$$
$$p(VBZ|\langle s\rangle) = 0.2$$
$$p(IN|\langle s\rangle) = 0.1$$
$$p(\langle /s\rangle|\langle s\rangle) = 0.0$$

$$p(NN|NN) = 0.4$$
$$p(VBZ|NN) = 0.3$$
$$p(IN|NN) = 0.1$$
$$p(\langle /s\rangle|NN) = 0.2$$

$$p(NN|VBZ) = 0.5$$
$$p(VBZ|VBZ) = 0.1$$
$$p(IN|VBZ) = 0.2$$
$$p(\langle /s\rangle|VBZ) = 0.2$$

$$p(NN|IN) = 0.7$$
$$p(VBZ|IN) = 0.1$$
$$p(IN|IN) = 0.1$$
$$p(\langle /s\rangle|IN) = 0.0$$

$$p(fruit|NN) = 0.4$$
$$p(flies|NN) = 0.2$$
$$p(like|NN) = 0.1$$
$$p(bananas|NN) = 0.1$$

$$p(fruit|VBZ) = 0.1$$
$$p(flies|VBZ) = 0.4$$
$$p(like|VBZ) = 0.4$$
$$p(bananas|VBZ) = 0.1$$

$$p(fruit|IN) = 0.0$$
$$p(flies|IN) = 0.0$$
$$p(like|IN) = 0.3$$
$$p(bananas|IN) = 0.7$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	$p(NN \mid \langle s \rangle) \times p(fruit \mid NN)$				
VBZ	$p(VBZ \mid \langle s \rangle) \times p(fruit \mid VBZ)$				
IN	$p(IN \mid \langle s \rangle) \times p(fruit \mid IN)$				

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	= 0.7 x 0.4 = 0.28				
VBZ	= 0.2 x 0.1 = 0.02				
IN	= 0.1 x 0.0 = 0.0				

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	0.28	0.0224			
VBZ	0.02				
IN	0.0				

$$v_2(NN) = \max_{i=1\ldots N} v_1(i)\times p(NN|q_i)\times p(Flies|NN)$$

$$= \max\{0.28\times p(NN|NN), 0.02\times p(NN|VBZ), 0.0\}\times p(Flies|NN)$$

$$= \max\{0.28\times 0.4, 0.02\times 0.5, 0.0\}\times 0.2$$

$$= \max\{\mathbf{0.112}, 0.01, 0.0\}\times 0.2 = 0.0224$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	0.28	0.0224			
VBZ	0.02	0.0336			
IN	0.0				

$$v_2(VBZ) = \max_{i=1\ldots N} v_1(i) \times p(VBZ|q_i) \times p(Flies|VBZ)$$

$$= \max\{0.28 \times p(VBZ|NN), 0.02 \times p(VBZ|VBZ), 0.0\} \times p(Flies|VBZ)$$

$$= \max\{0.28 \times 0.3, 0.02 \times 0.1, 0.0\} \times 0.4$$

$$= \max\{\mathbf{0.084}, 0.02, 0.0\} \times 0.4 = 0.0336$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	0.28	0.0224			
VBZ	0.02	0.0336			
IN	0.0	0.0			

$$v_2(IN) = \max_{i=1\ldots N} v_1(i) \times p(IN|q_i) \times p(Flies|IN)$$

$$= \max\{0.28 \times p(IN|NN), 0.02 \times p(IN|VBZ), 0.0\} \times p(Flies|IN)$$

$$= \max\{0.28 \times 0.1, 0.02 \times 0.2, 0.0\} \times 0.0$$

$$= \max\{\mathbf{0.028}, 0.004, 0.0\} \times 0.0 = 0.0$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	0.28	0.0224	0.00168		
VBZ	0.02	0.0336			
IN	0.0	0.0			

$$v_3(NN) = \max_{i=1\ldots N} v_2(i) \times p(NN|q_i) \times p(Like|NN)$$

$$= \max\{0.0224 \times p(NN|NN), 0.0336 \times p(NN|VBZ), 0.0\} \times p(Like|NN)$$

$$= \max\{0.0224 \times 0.4, 0.0336 \times 0.5, 0.0\} \times 0.1$$

$$= \max\{0.00896, \mathbf{0.0168}, 0.0\} \times 0.1 = 0.0168$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	[1]Fruit	[2]Flies	[3]Like	[4]Bananas	</s>
NN	0.28	0.0224	0.00168	0.00014112	?
VBZ	0.02	0.0336	0.002688	0.0000504	
IN	0.0	0.0	0.002016	0.00037632	

$$v_5(\langle/s\rangle) = \max_{i=1\ldots N} v_4(i)\times p(\langle/s\rangle|q_i)$$

$$= \max\{0.00014112\times p(\langle/s\rangle|NN), 0.0000504\times p(\langle/s\rangle|VBZ), 0.00037632\times p(\langle/s\rangle|IN)\}$$

$$= \max\{0.00014112\times 0.2, 0.0000504\times 0.2, 0.00037632\times 0.1\}$$

$$= \max\{0.000028224, 0.00001008, 0.000037632\} = 0.000037632$$

Smith UW (CSE447)

Example

Fruit flies like bananas

Tags: NN, VBZ, IN

	1Fruit	2Flies	3Like	4Bananas	</s>
NN	0.28	0.0224	0.00168	0.00014112	0.000037632
VBZ	0.02	0.0336	0.002688	0.0000504	
IN	0.0	0.0	0.002016	0.00037632	

The optimal path: Fruit flies like bananas

NN NN VBZ IN

Smith UW (CSE447)

Sequences

A primary form of natural language data with many applications.

The classic model of the Hidden Markov Model, where a latent (unobserved) variable is key aspect of the inference.

- What's the likelihood? Solved by Forward
- What's the best path? Solved by Viterbi Decoding