ECS763U/P Natural Language Processing

Julian Hough

Week 4: Sequence Classification

OUTLINE

1) Sequence Tagging Tasks: POS tagging and NER
2) Generative: Hidden Markov Models
3) Discriminative: Conditional Random Fields

OUTLINE

1) Sequence Tagging Tasks: POS tagging and NER
2) Generative: Hidden Markov Models
3) Discriminative: Conditional Random Fields

Sequence Modelling Tasks

- Many problems are about modelling (labelling, characterising, evaluating) **sequences**:
  - Part-of-speech tagging
  - Dialogue act tagging
  - Named entity recognition
  - Speech recognition
  - Spelling correction
  - Machine translation
  - ...

Sequence Likelihood Tasks

- Speech recognition

  `I saw a van`

  `eyes awe of an`
- Spelling correction

  `It's about fifteen minuets from my house`

  `It's about fifteen minutes from my house`
- Machine translation

  `vjetar će biti noćas jak:`

  `the wind tonight will be strong`

  `the wind tonight will be powerful`

  `the wind tonight will be a yak`

Sequence Tagging Tasks

- Part-of-Speech (POS) tagging:

mary	hires	a	detective
PN	VBZ	DET	CN

- Named Entity tagging/Named Entity Recognition (NER):

Today	President	Donald	J.	Trump	announced
O	B-PER	I-PER	I-PER	E-PER	O

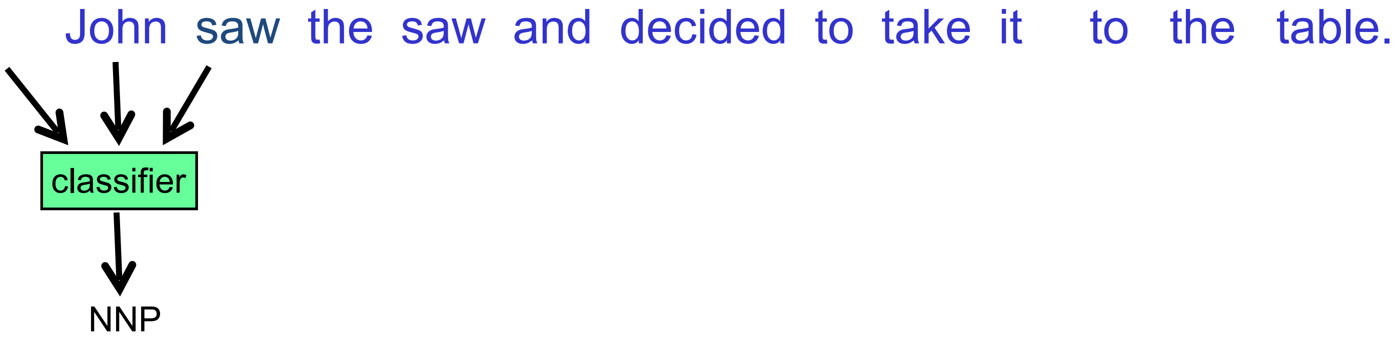
- Dialogue Act tagging:

A:	So do you go to college right now?	YN-QUESTION
B:	Yeah	YES-ANSWER
A:	Are yo-	ABANDONED
B:	it's my last year	STATEMENT
A:	What did you say?	CLARIFY
B:	my last year	NP-ANSWER
A:	Oh good for you	APPRECIATION
B:	uh-huh	BACKCHANNEL

- Why are these not just (word/sentence) classification tasks?

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

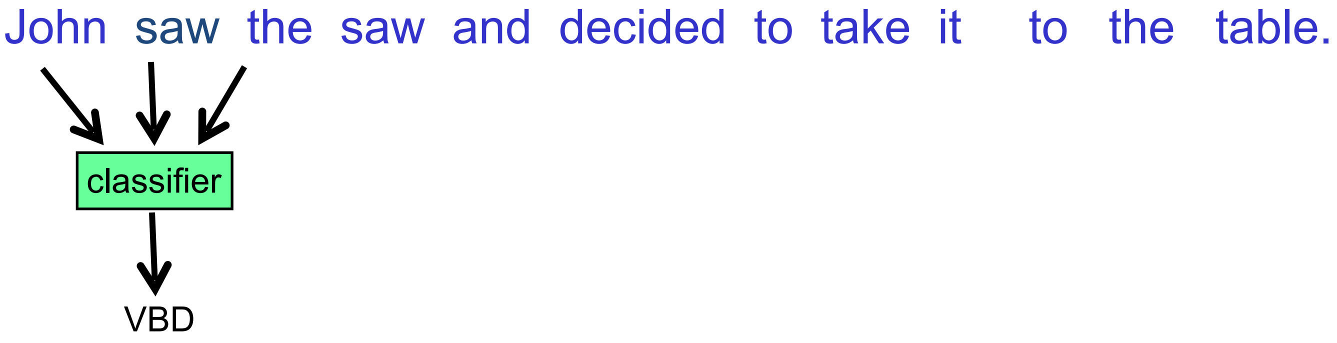
classifier

NNP

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

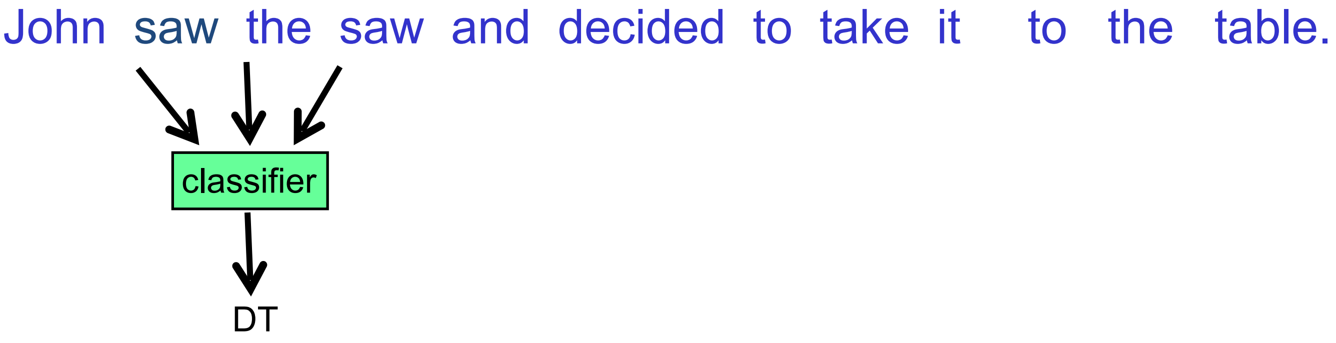
classifier

VBD

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

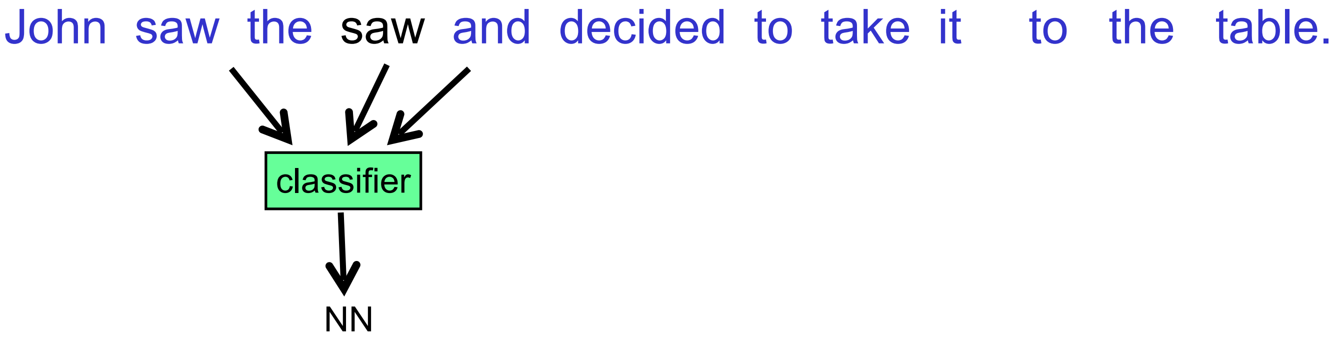
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Part-of-Speech (POS) tagging

Sequence Labeling as Classification

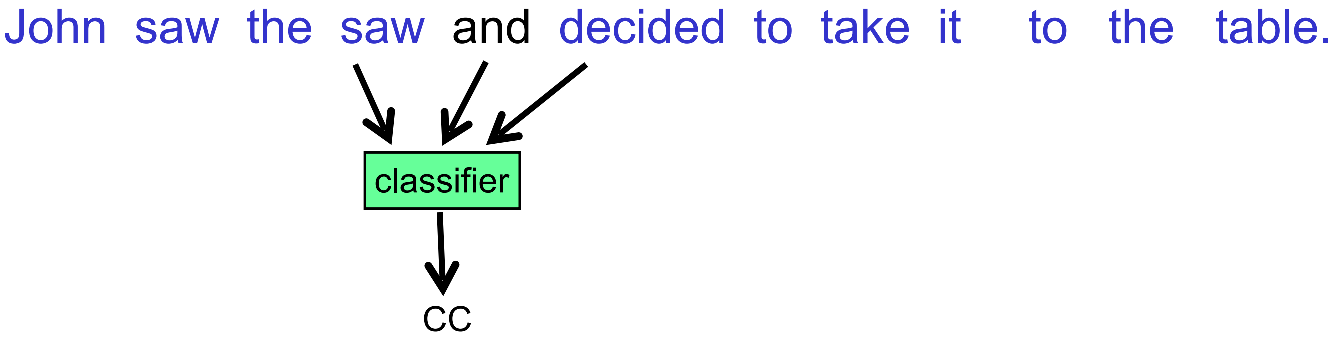
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

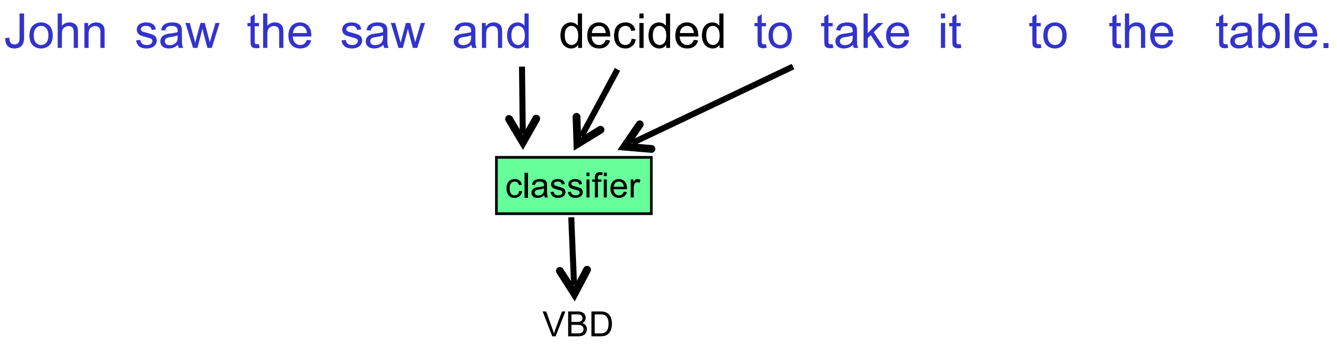
classifier

CC

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

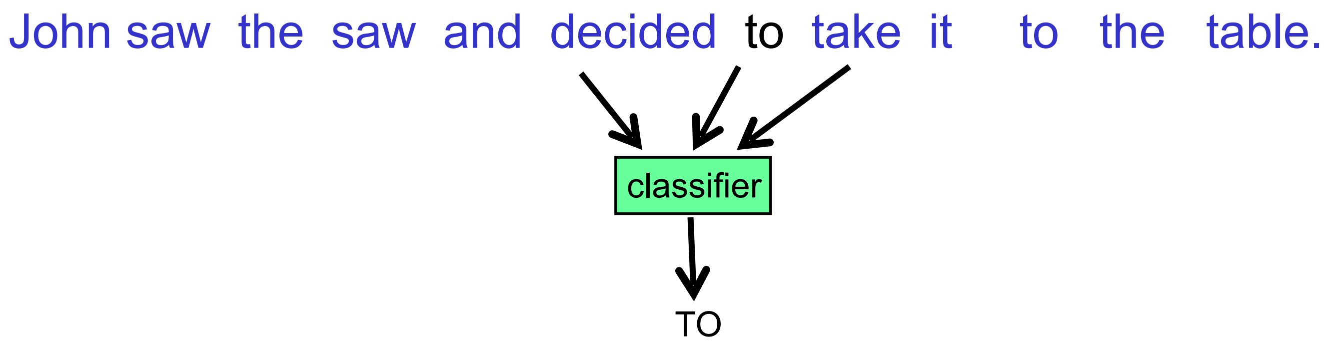
classifier

VBD

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

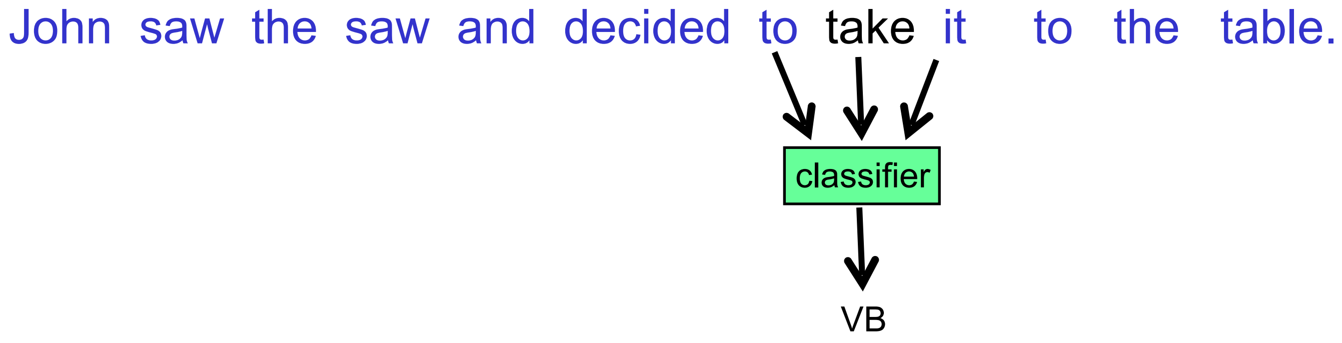
classifier

TO

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

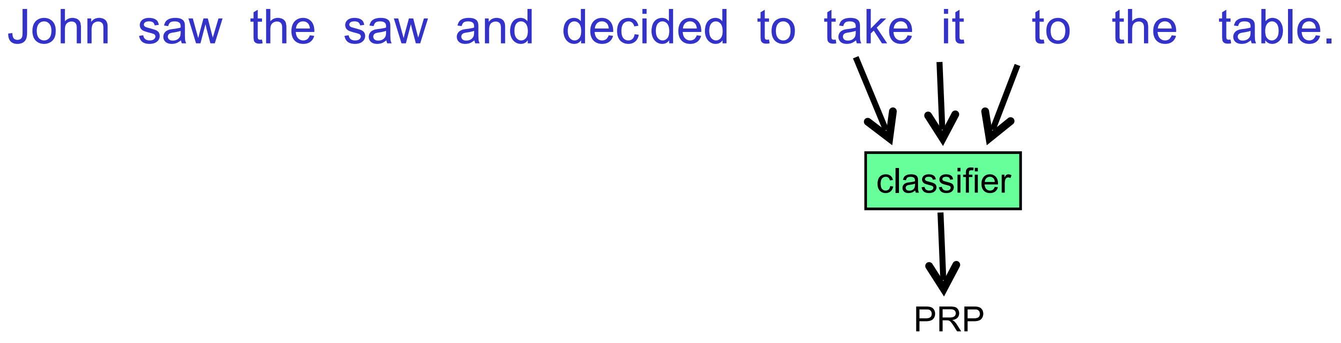
- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).


John saw the saw and decided to take it to the table.

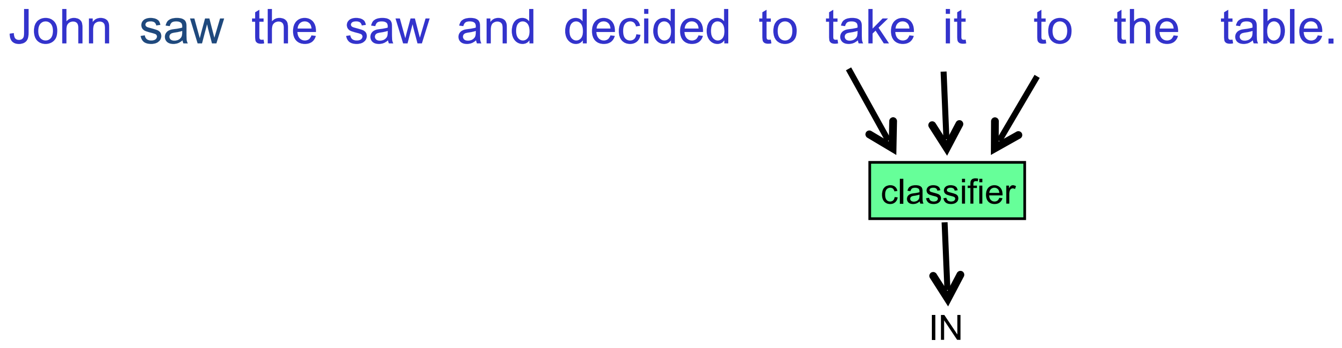
classifier

PRP

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).

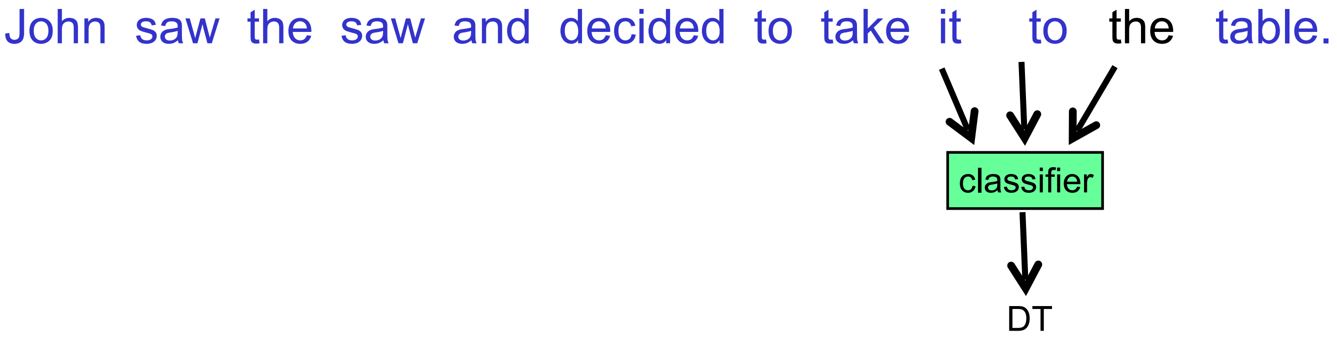


John saw the saw and decided to take it to the table.

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



John saw the saw and decided to take it to the table.

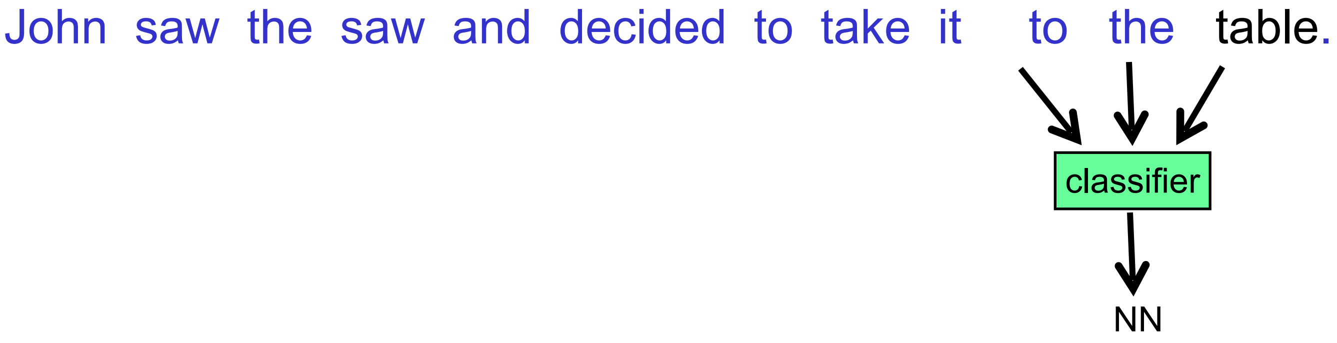
classifier

DT

Part-of-Speech (POS) tagging

Sequence Labeling as Classification

- Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



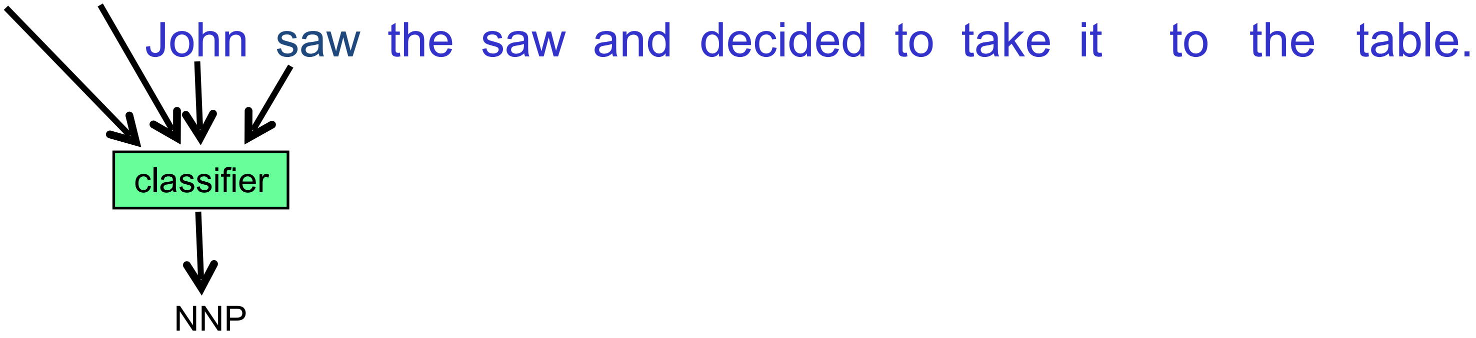
Part-of-Speech (POS) tagging

Using Outputs as Inputs

- Better input features are usually the **categories** of the surrounding tokens, but these are not available yet as they haven’t been classified.

- You can use category of either the preceding or succeeding tokens by going forward or back and using previous output from the classifier at test time.

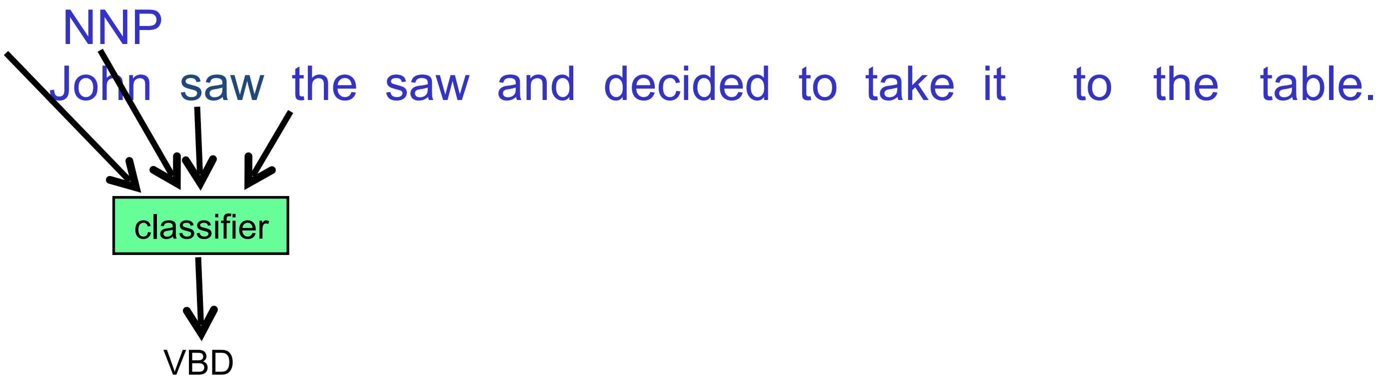
Forward Classification



John saw the saw and decided to take it to the table.

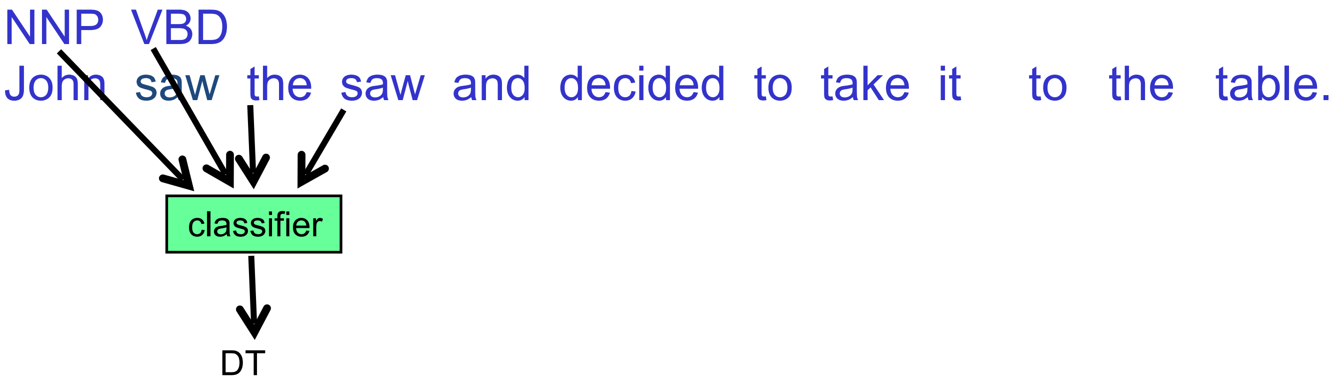
Part-of-Speech (POS) tagging

Forward Classification



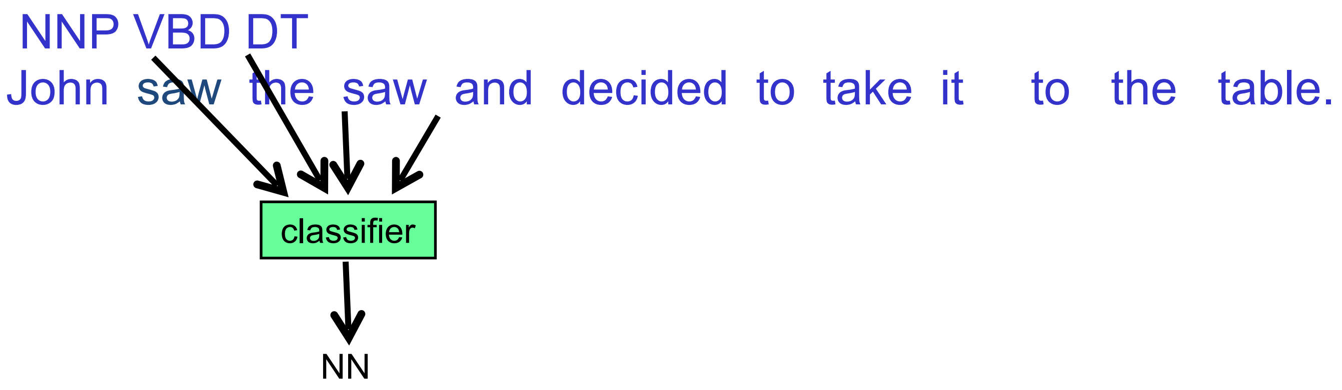
Part-of-Speech (POS) tagging

Forward Classification



Part-of-Speech (POS) tagging

Forward Classification



NNP VBD DT

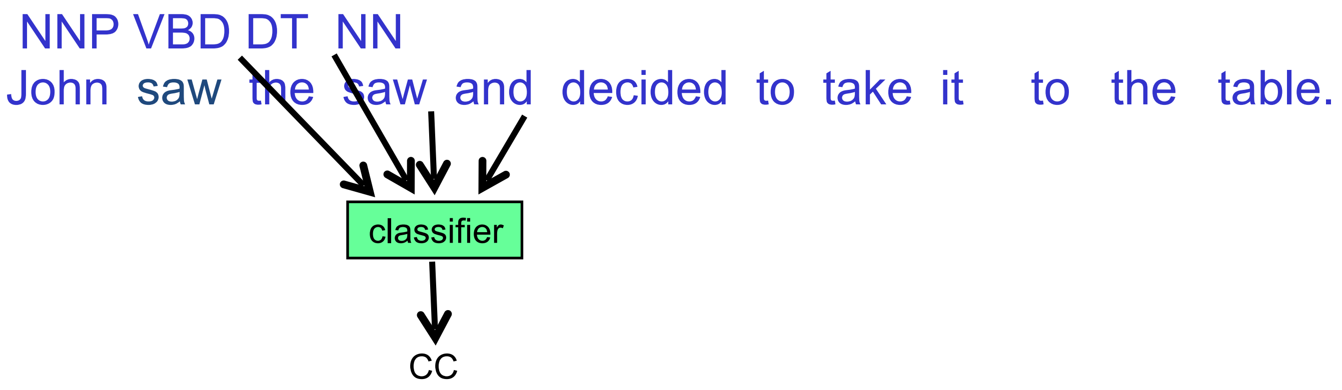
John saw the saw and decided to take it to the table.

classifier

NN

Part-of-Speech (POS) tagging

Forward Classification



NNP VBD DT NN

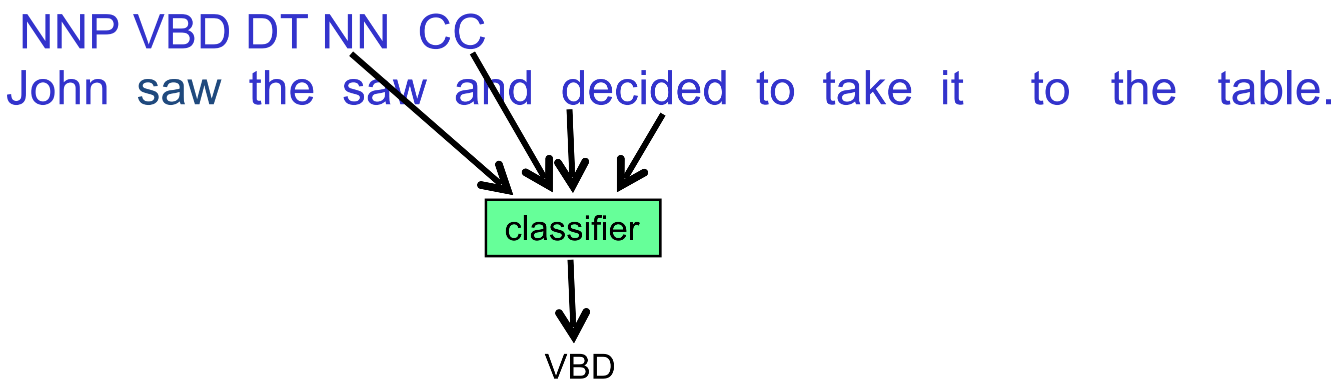
John saw the saw and decided to take it to the table.

classifier

CC

Part-of-Speech (POS) tagging

Forward Classification



NNP VBD DT NN CC

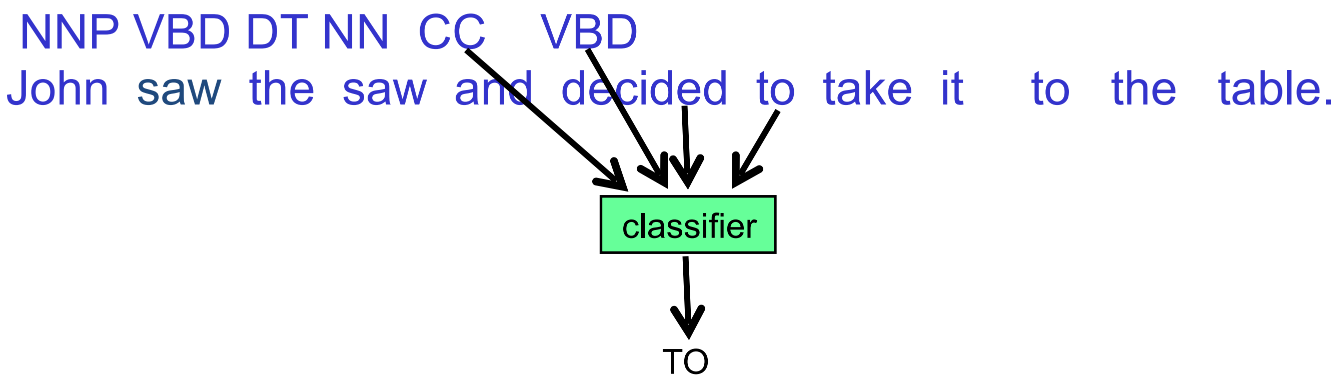
John saw the saw and decided to take it to the table.

classifier

VBD

Part-of-Speech (POS) tagging

Forward Classification



NNP VBD DT NN CC VBD

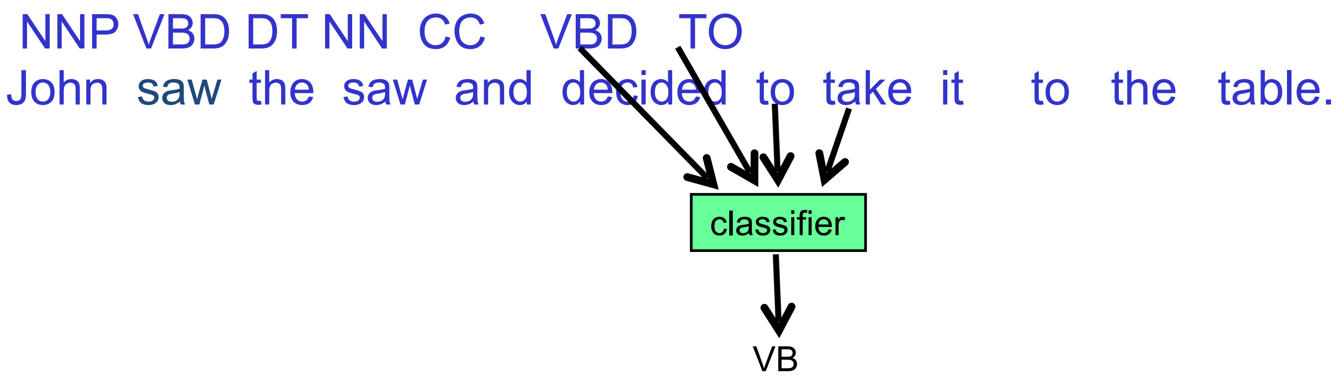
John saw the saw and decided to take it to the table.

classifier

TO

Part-of-Speech (POS) tagging

Forward Classification



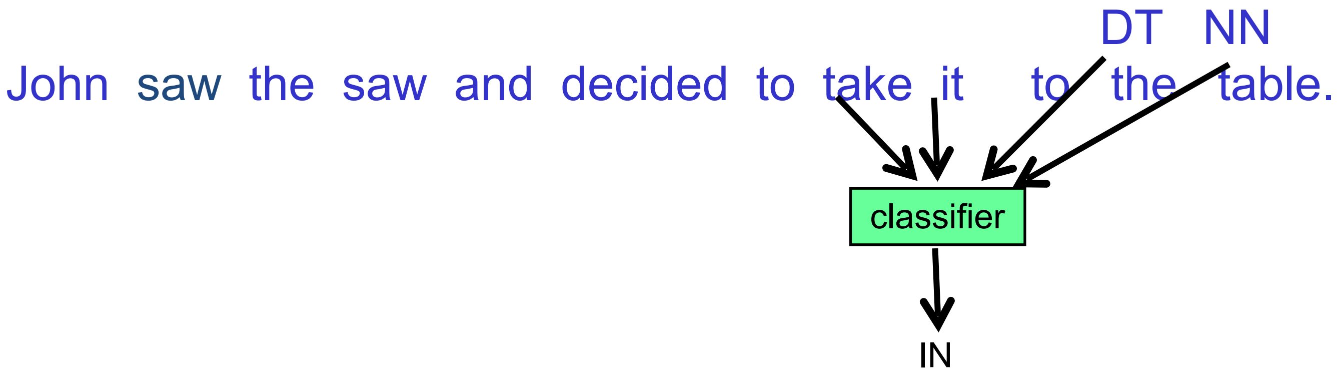
NNP VBD DT NN CC VBD TO

John saw the saw and decided to take it to the table.

Part-of-Speech (POS) tagging

Backward Classification

- Disambiguating “to” in this case would be even easier backward.



Named Entity Recognition (NER)

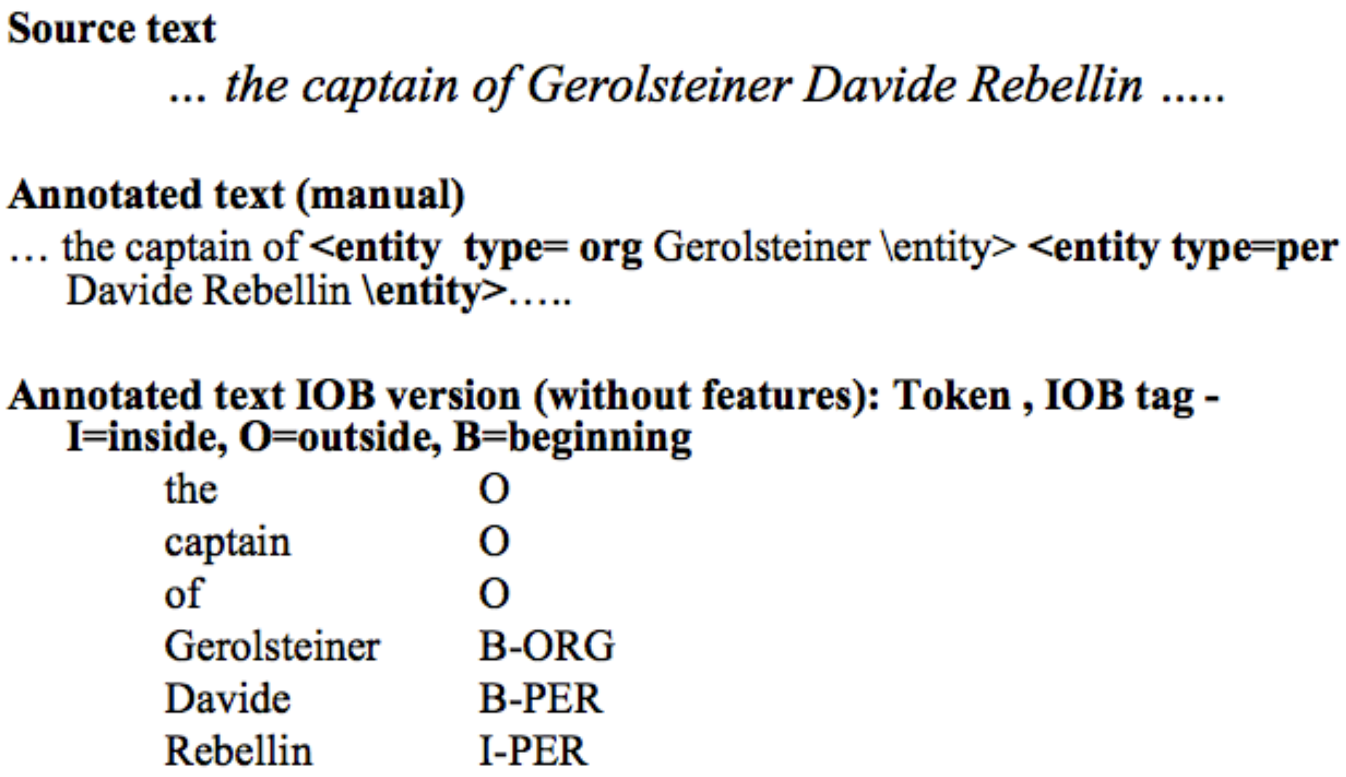
Input:

Apple Inc., formerly Apple Computer, Inc., is an American multinational corporation headquartered in Cupertino, California that designs, develops, and sells consumer electronics, computer software and personal computers. It was established on April 1, 1976, by Steve Jobs, Steve Wozniak and Ronald Wayne.

Output:

Apple Inc., formerly Apple Computer, Inc., is an American multinational corporation headquartered in Cupertino, California that designs, develops, and sells consumer electronics, computer software and personal computers. It was established on April 1, 1976, by Steve Jobs, Steve Wozniak and Ronald Wayne.

THE ML APPROACH TO NE: THE IOB REPRESENTATION



Source text

… the captain of Gerolsteiner Davide Rebellin …..

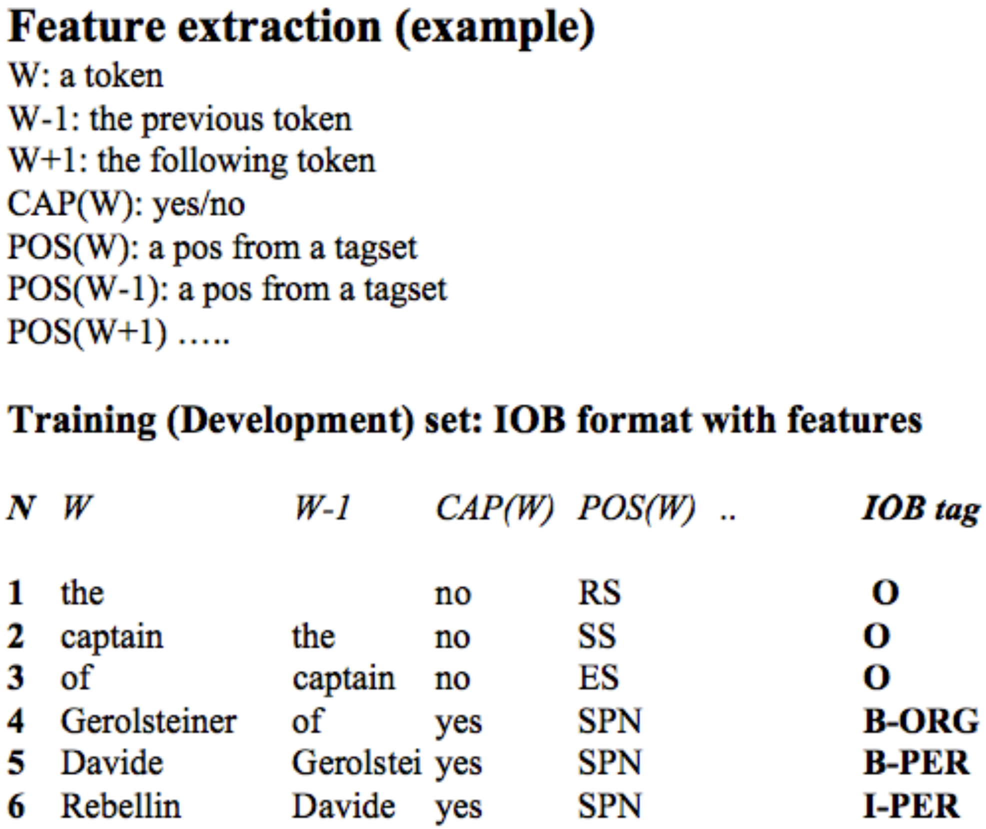
Annotated text (manual)

… the captain of **<entity type= org** Gerolsteiner \entity> **<entity type=per** Davide Rebellin **\entity>**…..

Annotated text IOB version (without features): Token , IOB tag - I=inside, O=outside, B=beginning

the	O
captain	O
of	O
Gerolsteiner	B-ORG
Davide	B-PER
Rebellin	I-PER

THE ML APPROACH TO NE: FEATURES



Feature extraction (example)

W: a token
W-1: the previous token
W+1: the following token
CAP(W): yes/no
POS(W): a pos from a tagset
POS(W-1): a pos from a tagset
POS(W+1) …..

Training (Development) set: IOB format with features

N	*W*	*W-1*	*CAP(W)*	*POS(W)*	..	***IOB tag***
1	the		no	RS		**O**
2	captain	the	no	SS		**O**
3	of	captain	no	ES		**O**
4	Gerolsteiner	of	yes	SPN		**B-ORG**
5	Davide	Gerolstei	yes	SPN		**B-PER**
6	Rebellin	Davide	yes	SPN		**I-PER**

FEATURES

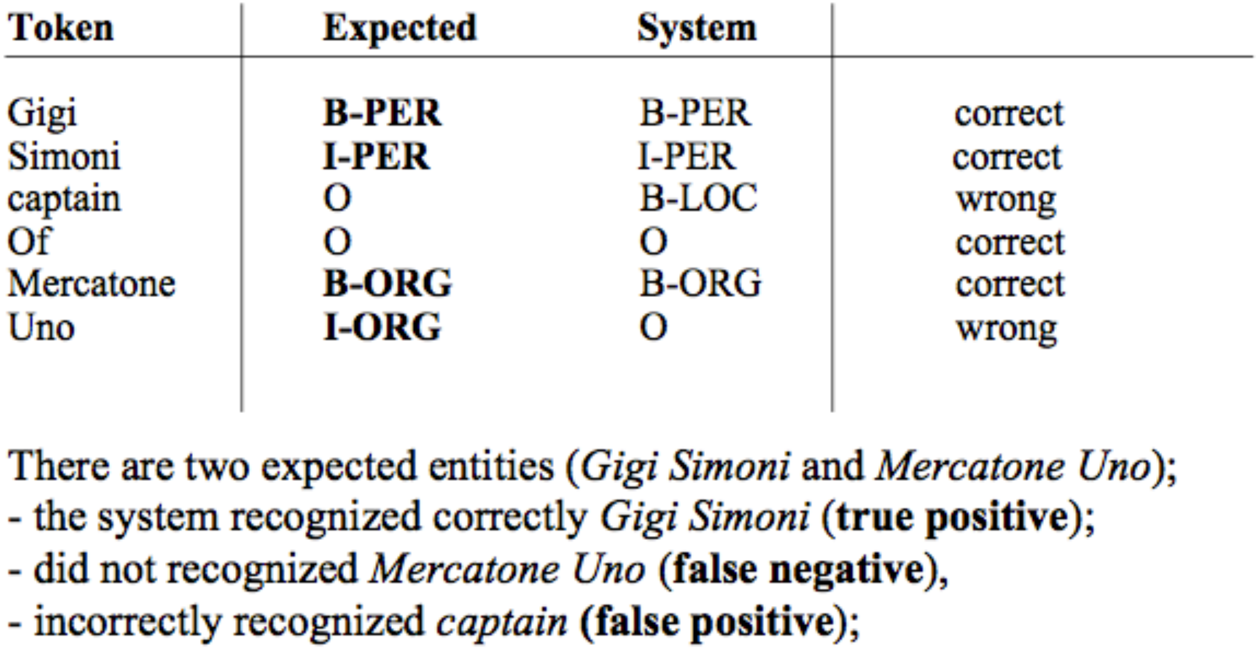
For each running word:

- **WORD**: the word itself (both unchanged and lower-cased)
  e.g. Casa casa
- **POS**: the part of speech of the word (as produced by TagPro)
  e.g. Oggi SS (singular noun)
- **AFFIX**: prefixes/suffixes (1, 2, 3 or 4 chars. at the start/end of the word)
  e.g. Oggi {o,og,ogg,oggi, – i,gi,ggi,oggi}
- **ORTHOgraphic** information (e.g. capitalization, hyphenation)
  e.g. Oggi C (capitalized)
  oggi L (lowercased)

FEATURES

- **COLLOCation** bigrams
  - 36.000, Italian newspapers ranked by MI values
- **Gazzetters**
  - **PERSONS**: Person proper names or titles (154.000, Italian phone-book, Wikipedia,)
  - **TOWNS**: World (main), Italian (comuni) and Trentino's (frazioni) towns (12.000, from various internet sites)
  - **STOCK-MARKET**: Italian and American stock market organizations (5.000, from stock market sites)
  - **WIKI-GEO**: Wikipedia geographical locations (3.200,)

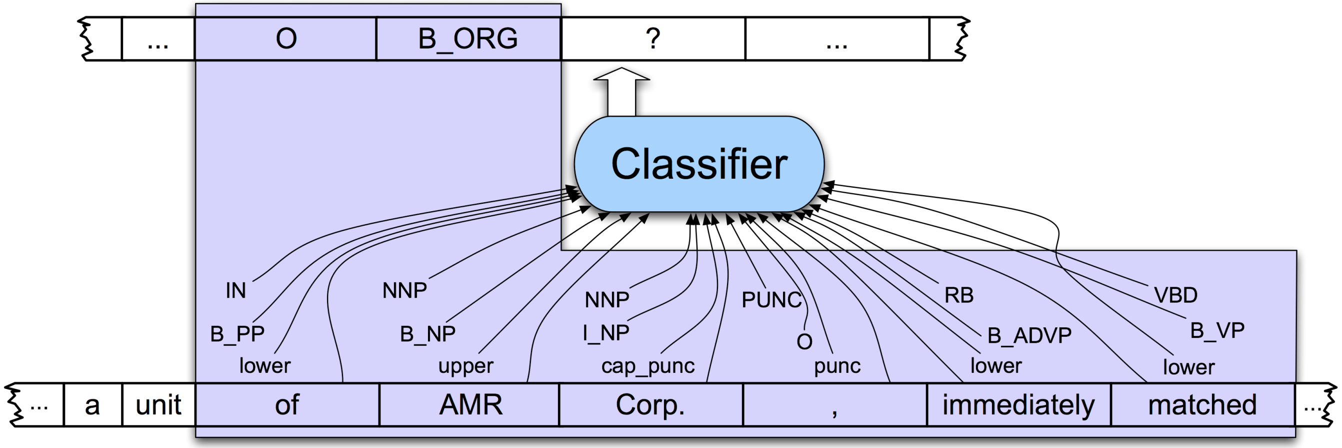
NER: EVALUATION



Token	Expected	System	
Gigi	**B-PER**	B-PER	correct
Simoni	**I-PER**	I-PER	correct
captain	O	B-LOC	wrong
Of	O	O	correct
Mercatone	**B-ORG**	B-ORG	correct
Uno	**I-ORG**	O	wrong

There are two expected entities (*Gigi Simoni* and *Mercatone Uno*);
- the system recognized correctly *Gigi Simoni* (**true positive**);
- did not recognized *Mercatone Uno* (**false negative**),
- incorrectly recognized *captain* (**false positive**);

NER as Sequence Labeling

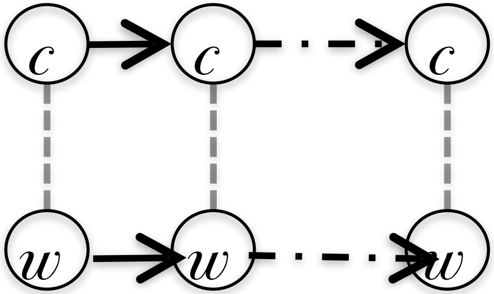


OUTLINE

1) Sequence Tagging Tasks: POS tagging and NER
2) Generative: Hidden Markov Models
3) Discriminative: Conditional Random Fields

Sequence Labelling

- Sequence labelling/tagging
  - A classification problem, but over sequences.
    - Often from words to a sequence of class labels. e.g.:
      - POS-tagging
      - Named Entity Recognition (NER)



- We could try:
  - Rule-based classifier:
    - E.g. transformation-based learning (old school)
  - **Generative sequence model:**
    - (remember Naïve Bayes?) – **Hidden Markov Models**
  - **Discriminative sequence model:**
    - (remember Logistic Regression?) – **Conditional Random Fields**

Generative models- look familiar?

- Unigram language model

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i)$$

- Bigram language model

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-1})$$

- N-gram language model

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

- Naïve Bayes

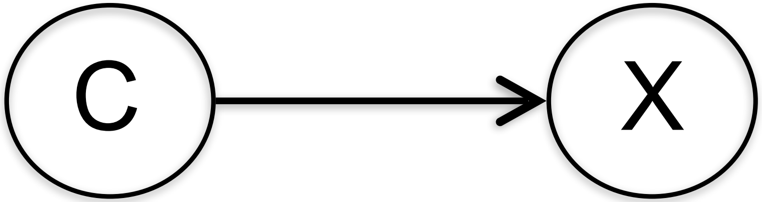
$$P(c_j \mid d) = P(c_j) \prod_i P(w_i \mid c_j)$$

Bayes Rule (Reminder)

- Generative models (non sequence):

$$P(C,X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$\boxed{P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}}$$



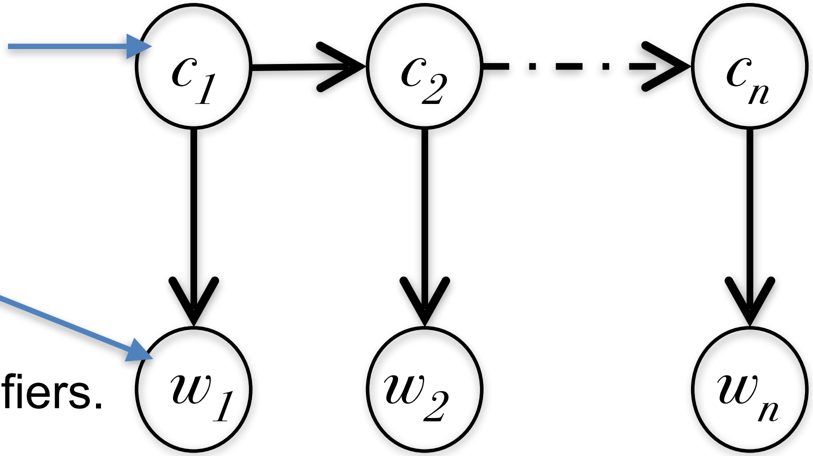
C = latent (hidden) variable/state/class
X = instance data (features)

Bayes Rule

- For lots of NLP sequence classification, observations are **words** and latent variables are **classes**:

$$P(C|W) = \frac{P(W|C)P(C)}{P(W)}$$

$$P(c_1 \ldots c_n | w_1 \ldots w_n) = \frac{P(w_1 \ldots w_n | c_1 \ldots c_n) P(c_1 \ldots c_n)}{P(w_1 \ldots w_n)}$$



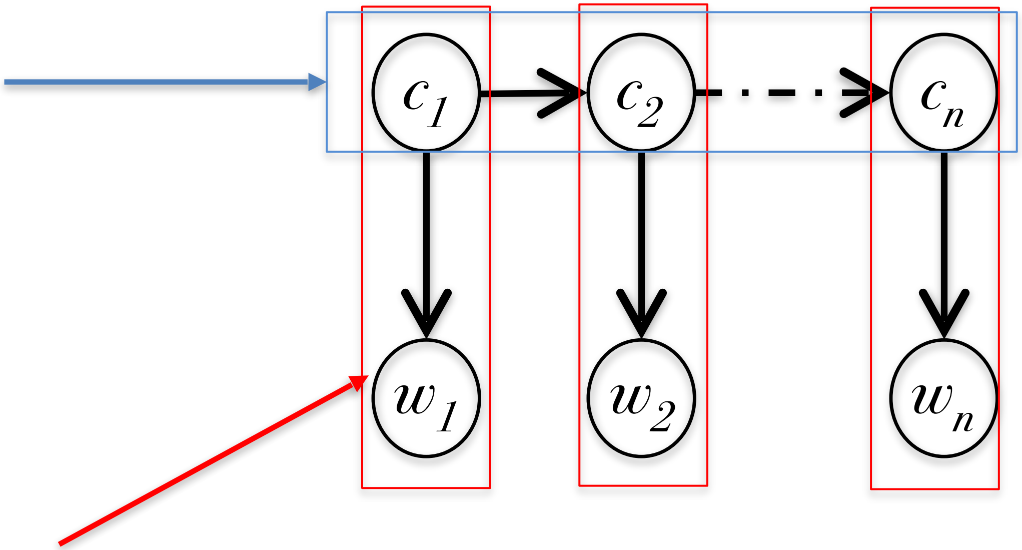
Hidden class/tag sequence (e.g. POS tag)

Observed word sequence

Model is like a sequence of Bayesian classifiers.

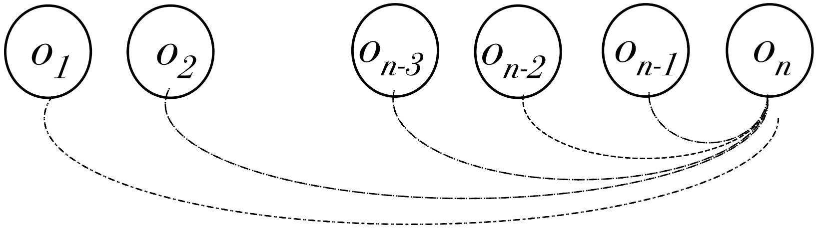
Hidden Markov Models

- HMMs use probability distributions from two models:
  - A class sequence model $p(c_i|c_1 \ldots c_{i-1})$ which is a Markov Model defined by **Transition probabilities** (like a language model)
  - A word/class association model $p(w_i|c_i)$ which are distributions of **Emission probabilities**



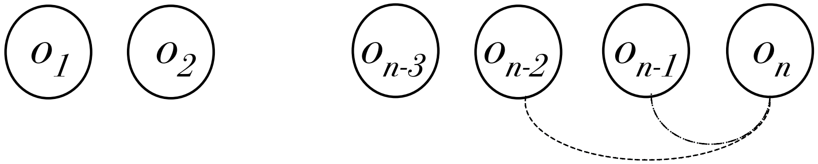
Markov Assumption

- Instead of:



- We approximate by:
  - "n-gram model of length k" (where k = n-1)

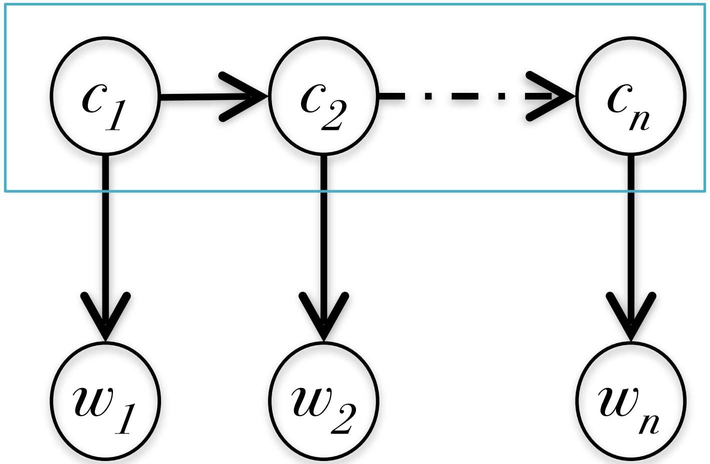
trigram model
(k=2):

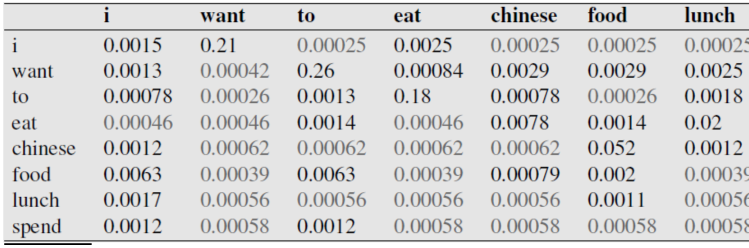


- In general not sufficient – but often good approximation for high *k*.
  - Ignores long-distance dependencies:

Hidden Markov Models

- **Remember Language Models?**
- We can define a **Markov Model** effectively have a Language Model (sequence likelihood model using the Markov assumption) which will give us the probability of a possible hidden sequence $C_1 \ldots C_n$
- Remember the probability matrix for bigrams? i.e. **Transition matrix** for **transition probabilities.**
- For 1st order Markov Models, we can do this for class/state sequences too.





	i	want	to	eat	chinese	food	lunch
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.0002
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.0003
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.0005
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.0005

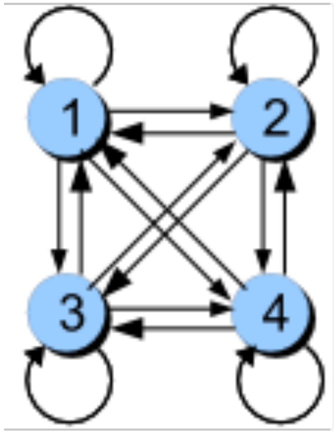
Hidden Markov Models

- **Transition matrix** constrains possible state paths:

C_i (state/class value at position i in sequence)

C_{i-1} (state/class value at position i-1 in sequence)

	c_1	**c_2**	**c_3**	**c_4**
c_1				
c_2				
c_3				
c_4				



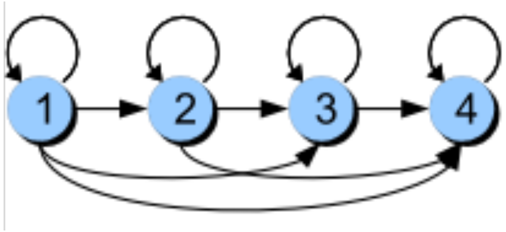
Hidden Markov Models

- **Transition matrix** constrains possible state paths:

C_i (state/class value at position i in sequence)

C_{i-1} (state/class value at position i-1 in sequence)

	c_1	**c_2**	**c_3**	**c_4**
c_1	▨	▨	▨	▨
c_2		▨	▨	▨
c_3			▨	▨
c_4				▨



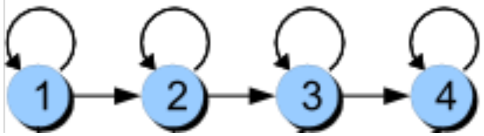
Hidden Markov Models

- **Transition matrix** constrains possible state paths:

C_i (state/class value at position i in sequence)

C_{i-1} (state/class value at position i-1 in sequence)

	c_1	c_2	c_3	c_4
c_1	▨	▨		
c_2		▨	▨	
c_3			▨	▨
c_4				▨



Hidden Markov Models

- **Transition probabilities** $P(c_i|c_{i-1})$ define a 1st order Markov model gives probability of a given sequence of states (classes/tags) having occurred at all using the chain rule.
- 1st order Markov models (bigram model) can be easily represented in a 2D transition matrix:

Transition probs $P(c_i|c_{i-1})$:

	NN	**NNS**	**VBZ**	**VB**	**end**
NN	0.3	0.3	0.3	0.0	0.1
NNS	0.0	0.2	0.6	0.2	0.0
VBZ	0.5	0.0	0.0	0.1	0.4
VB	0.3	0.5	0.0	0.0	0.2
start	0.3	0.3	0.0	0.4	0.0

C_{i-1} (state/class value at position i-1 in sequence)

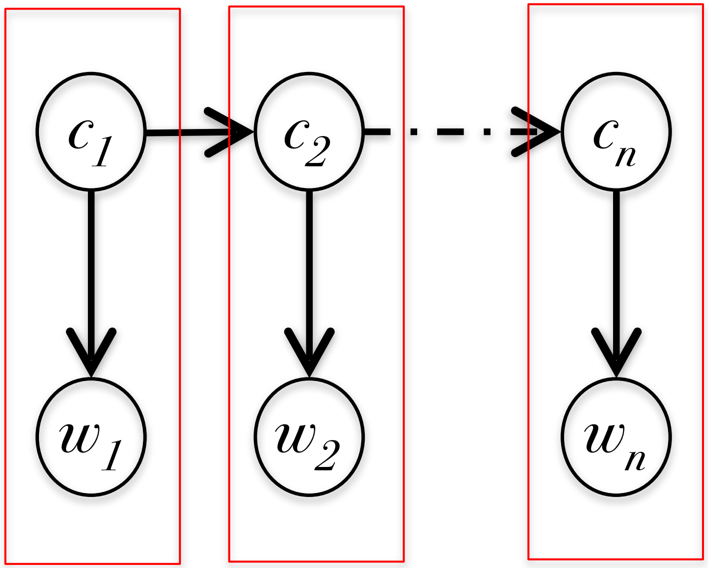
C_i (state/class value at position i in sequence)

`Rows are distributions. Probabilities sum to 1.`

- The class sequence is not directly observed, hence it is a **hidden** Markov model

Hidden Markov Models

- We can only estimate that a given sequence occurred based on what we **observe (observation sequence)**.
- **Emission probabilities** are needed for us to use Bayesian inference to answer: what is the likelihood that word *w* was generated (observed/emitted) by underlying class *c* ?



Hidden Markov Models

- **Emission probabilities** can be defined in a matrix $P(w_i|c_i)$:

Emission probs $P(w_i|c_i)$:

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

C_i (state/class value at position i in sequence)

W_i (observation/word value at position i in sequence)

`Rows are distributions over the vocab. Probabilities sum to 1.`

- As with Naive Bayes, we 'flip' the probability around- given 'time' was observed, what's the likelihood that 'NN' **generated** it, or that 'NNS' generated it? etc. i.e. what is the likelihood of different hidden sequences.

Hidden Markov Models

- Generative model:
  - Assume observations (e.g. words) generated from **states**
  - States depend on previous state sequence (Markov: just the most recent one, or fixed number in the past)
- Likelihood of observations given we know the classes for bigram underlying model:

$$P(W) = P(w_1, w_2, \ldots, w_n) = \prod_i p(w_i \mid c_i) p(c_i \mid c_{i-1})$$

- Bayes' Rule lets us use it to estimate likelihood of a class sequence given we know the word sequence:

$$P\left(C \mid W\right) = \frac{P(W \middle| C) P(C)}{P(W)}$$

And from this we have a classifier:

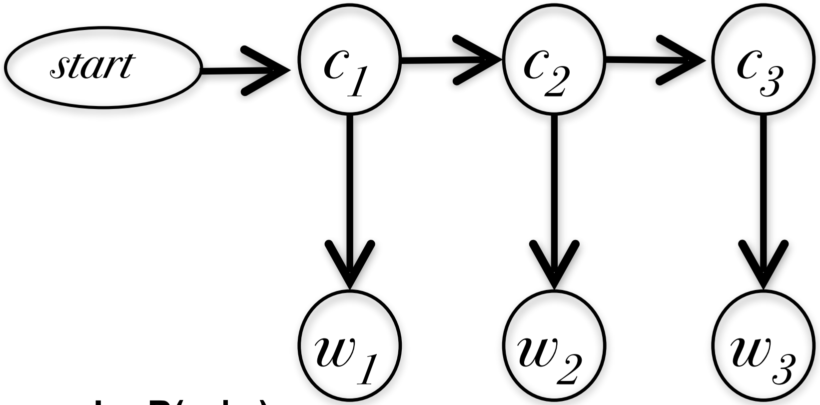
$$C_{MAP} = argmax_C\, p(C|W) = argmax_C\, p(W|C)p(C)$$

Likelihood

- Given HMM H, what kind of probabilities are available?

W = time flies like an arrow
C = NN VBZ PRP DT NN

W = fruit flies like a banana
C = NN NNS VB DT NN



Transition probs $P(c_i|c_{i-1})$:

c_i (columns), c_{i-1} (rows)

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i (columns), c_i (rows)

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN)
p(c2=NNS|c1=NN)
p(w1=fruit|c1=NN)
p(w1=flies|c1=VBZ)

More difficult, what are:

p(w1=fruit)
p(w1=time)
p(W=fruit flies)
p(c1=NN|w1=time)

Likelihood

- `(Solution)`

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN)

p(c2=NNS|c1=NN)

p(w1=fruit|c1=NN)

p(w1=flies|c1=VBZ)

Likelihood

- (Solution)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN) 0.4

p(c2=NNS|c1=NN)

p(w1=fruit|c1=NN)

p(w1=flies|c1=VBZ)

Likelihood

- (Solution)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN)	0.4
p(c2=NNS|c1=NN)	0.2
p(w1=fruit|c1=NN)	
p(w1=flies|c1=VBZ)	

Likelihood

- (Solution)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN) 0.4

p(c2=NNS|c1=NN) 0.2

p(w1=fruit|c1=NN) 0.3

p(w1=flies|c1=VBZ)

Likelihood

- (Solution)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

What are:

p(c2=VBZ|c1=NN)	**0.4**
p(c2=NNS|c1=NN)	**0.2**
p(w1=fruit|c1=NN)	**0.3**
p(w1=flies|c1=VBZ)	**1.0**

Likelihood

- `(Solution)`

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

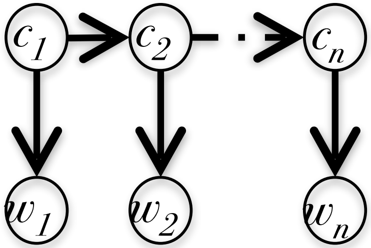
What are:

p(c2=VBZ|c1=NN)	0.4
p(c2=NNS|c1=NN)	0.2
p(w1=fruit|c1=NN)	0.3
p(w1=flies|c1=VBZ)	1.0

Only simple look-up required!

Likelihood of Observed Sequence (words)

- **Likelihood:** given observation W and HMM H, what is the likelihood p(W|H)?



- If we knew the class sequence, we could use:

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i \mid c_i) P(c_i \mid c_{i-1})$$

- But we don't …
  - HMM classes are hidden/unseen: **"latent variables"**

$$P(w_1 w_2 \dots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

Likelihood of Observed Sequence (words)

More difficult, what are:

p(w1=fruit)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_{i} P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

Transition probs $P(c_i|c_{i-1})$:

c_i (columns), c_{i-1} (rows)

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i (columns), c_i (rows)

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

More difficult, what are:

p(w1=fruit)

= p(w1=fruit|c1=NN) * p(c1=NN|c0=start) +
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start)
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$ →

More difficult, what are:

p(w1=fruit)
= p(w1=fruit|c1=NN) * p(c1=NN|c0=start) +
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start)
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=fruit)

= p(w1=fruit|c1=NN) * p(c1=NN|c0=start)
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- `(Solution)`

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j) \rightarrow$$

More difficult, what are:

p(w1=fruit)

= p(w1=fruit|c1=NN) * p(c1=NN|c0=start) +
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start)
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_{i-1} \ c_i	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

c_i \ w_i	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.0 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=fruit)

= p(w1=fruit|c1=NN) * p(c1=NN|c0=start)
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start)
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.0 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

= 0.06 +
0 +
0 +
0 +
0 +
0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \dots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=fruit)

= p(w1=fruit|c1=NN) * p(c1=NN|c0=start) +
p(w1=fruit|c1=NNS) * p(c1=NNS|c0=start)
p(w1=fruit|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=fruit|c1=VB) * p(c1=VB|c0=start)
p(w1=fruit|c1=PRP) * p(c1=PRP|c0=start)
p(w1=fruit|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i (columns), **c_{i-1}** (rows)

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i (columns), **c_i** (rows)

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.0 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

= 0.06 +
0 +
0 +
0 +
0 +
0

= 0.06

Likelihood of Observed Sequence (words)

More difficult, what are:

p(w1=time)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=time)
= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start)
p(w1=time|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i (columns), c_{i-1} (rows)

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i (columns), c_i (rows)

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j\in C}\prod_{i} P(w_i \mid c_i^j)\, P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=time)

= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start
p(w1=time|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	**NNS**	**VBZ**	**VB**	**PRP**	**DT**
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	**fruit**	**flies**	**arrow**	**like**	**an**
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- `(Solution)`

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=time)

= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start)
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start)
p(w1=time|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i (columns), c_{i-1} (rows)

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i (columns), c_i (rows)

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

More difficult, what are:

p(w1=time)

= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start
p(w1=time|c1=DT) * p(c1=DT|c0=start)

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.2 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood of Observed Sequence (words)

- `(Solution)`

$$P(w_1 w_2 \dots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j) \rightarrow$$

More difficult, what are:

p(w1=time)

= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start
p(w1=time|c1=DT) * p(c1=DT|c0=start)

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	**NNS**	**VBZ**	**VB**	**PRP**	**DT**
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	**fruit**	**flies**	**arrow**	**like**	**an**
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.2 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

= 0.06 +
0 +
0 +
0.02 +
0 +
0

Likelihood of Observed Sequence (words)

- (Solution)

$$P(w_1 w_2 \ldots w_n) = \sum_{j \in C} \prod_i P(w_i \mid c_i^j) P(c_i^j \mid c_{i-1}^j)$$

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	NNS	VBZ	VB	PRP	DT
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	fruit	flies	arrow	like	an
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

More difficult, what are:

p(w1=time)

= p(w1=time|c1=NN) * p(c1=NN|c0=start)
p(w1=time|c1=NNS) * p(c1=NNS|c0=start)
p(w1=time|c1=VBZ) * p(c1=VBZ|c0=start
p(w1=time|c1=VB) * p(c1=VB|c0=start)
p(w1=time|c1=PRP) * p(c1=PRP|c0=start
p(w1=time|c1=DT) * p(c1=DT|c0=start)

= (0.3 * 0.2) +
(0.0 * 0.2) +
(0.0 * 0.0) +
(0.2 * 0.1) +
(0.0 * 0.0) +
(0.0 * 0.5)

= 0.06 +
0 +
0 +
0.02 +
0 +
0

= **0.08**

Likelihood of Latent Variable (Class) sequence

- (Solution)

$$P(C|W) = \frac{P(W|C)P(C)}{P(W)}$$

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid c_i) P(c_i \mid c_{i-1})$$

More difficult, what are:

p(c1=NN|w1=time)

(where p(w1=time) = 0.08 from earlier!)

= (p(w1=time|c1=NN) * p(c1=NN|c0=start)) / 0.08

= (p(w1=time|c1=NN) * 0.2) / 0.08

= (0.3 * 0.2) / 0.08

= 0.75

Transition probs $P(c_i|c_{i-1})$:

c_i

c_{i-1}

	NN	**NNS**	**VBZ**	**VB**	**PRP**	**DT**
NN	0.2	0.2	0.4	0.2	0.0	0.0
NNS	0.0	0.1	0.5	0.4	0.0	0.0
VBZ	0.1	0.1	0.0	0.0	0.5	0.3
VB	0.2	0.2	0.0	0.0	0.1	0.5
PRP	0.2	0.2	0.0	0.0	0.0	0.6
DT	0.5	0.5	0.0	0.0	0.0	0.0
start	0.2	0.2	0.0	0.1	0.0	0.5

Emission probs $P(w_i|c_i)$:

w_i

c_i

	time	**fruit**	**flies**	**arrow**	**like**	**an**
NN	0.3	0.3	0.0	0.4	0.0	0.0
NNS	0.0	0.0	1.0	0.0	0.0	0.0
VBZ	0.0	0.0	1.0	0.0	0.0	0.0
VB	0.2	0.0	0.0	0.0	0.8	0.0
PRP	0.0	0.0	0.0	0.0	1.0	0.0
DT	0.0	0.0	0.0	0.0	0.0	1.0

Likelihood

- We can do these calculations in this way for short sequences for small numbers of states.

- However, summing all C is exponential, so use dynamic programming
  - we use the **Forward algorithm**
  - $\alpha_n(j)$ = probability of getting to word n and being in state j

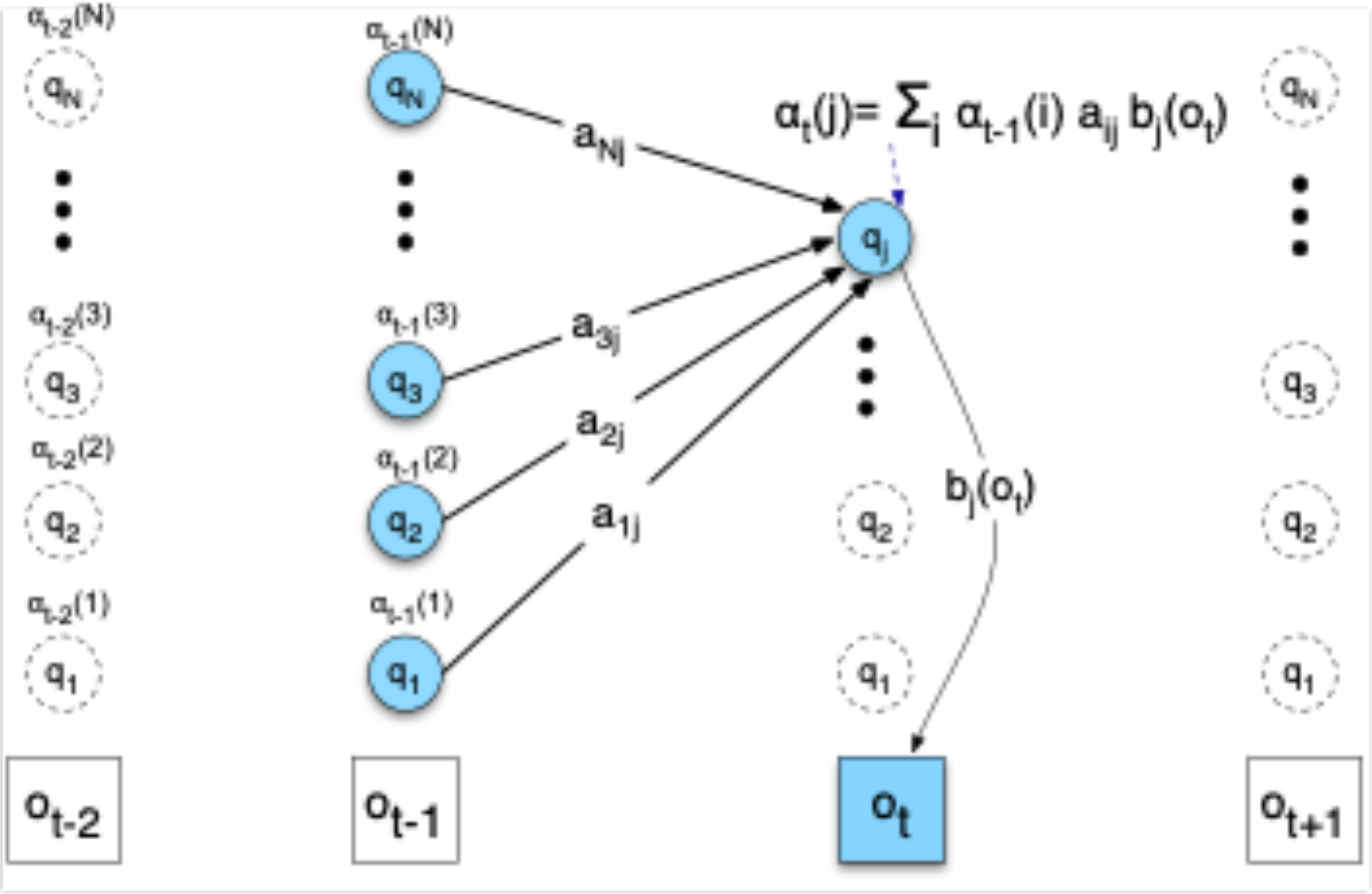
$$\alpha_1(j) = P(w_1 c_j) = P(w_1 \mid c_j) P(c_j)$$

$$\alpha_2(j) = P(w_1 w_2 c_j) = P(w_2 \mid c_j) \sum_i P(c_j \mid c_i) \alpha_1(i)$$

$$\alpha_n(j) = P(w_1 w_2 \ldots w_n c_j) = P(w_n \mid c_j) \sum_i P(c_j \mid c_i) \alpha_{n-1}(i)$$

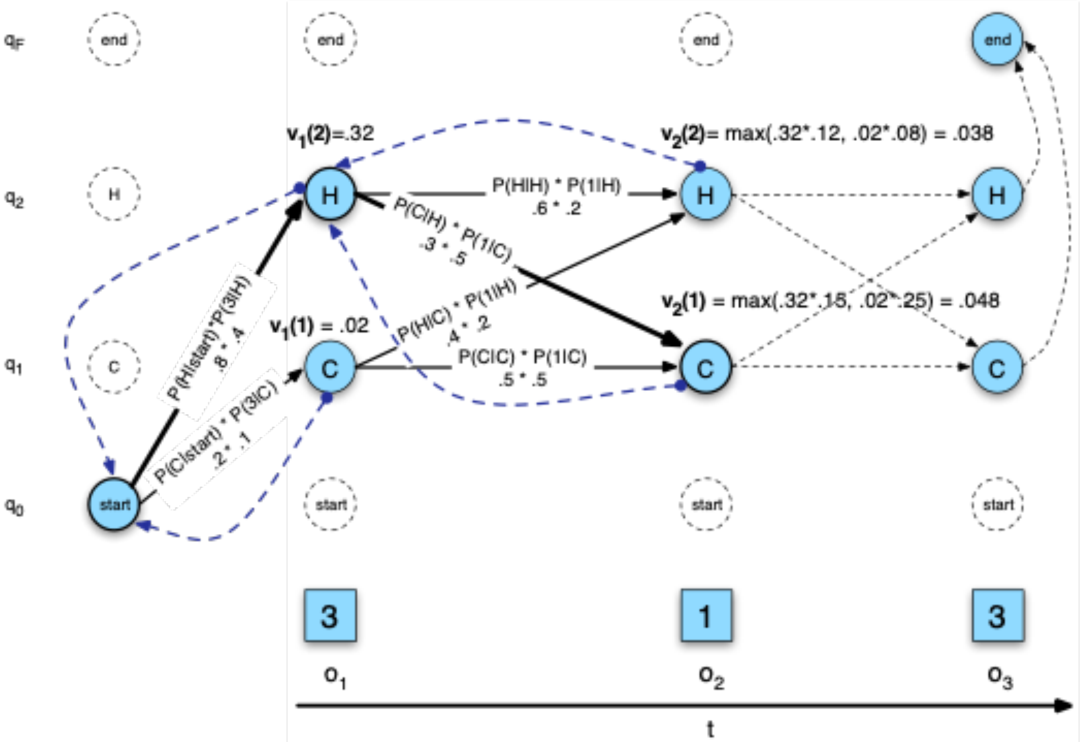
$\ldots$

Forward algorithm



Decoding- getting the most likely sequence

- **Decoding:** given observation O and HMM H, what is the most likely state sequence?
  - we use the **Viterbi algorithm**
    - Similar to Forward algorithm, but maintain **back-pointer** from each state to most likely previous state
    - Then **retrace** from most likely final state



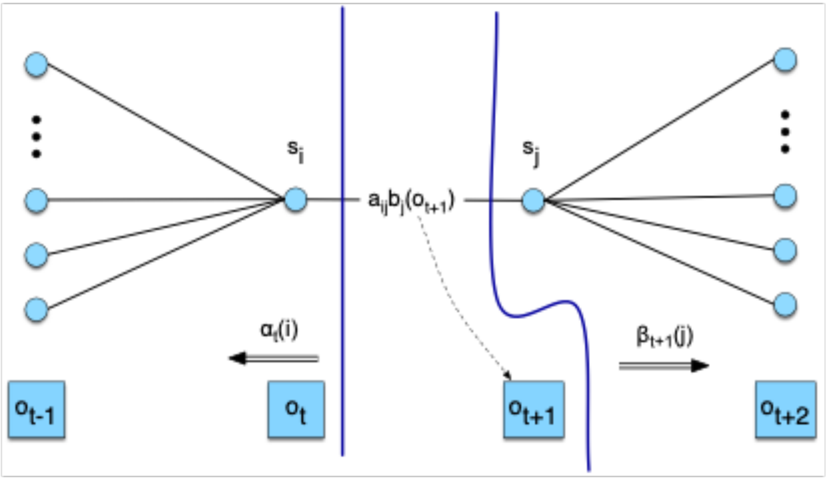
Learning

- **Learning/training:** given observation O, what is the optimum HMM model H? i.e. what are the optimal emission and transition probability models?
- If we have training data with fully labelled sequences, use Standard **Maximum likelihood estimation (MLE)** with counts:
  - Emission probabilities $p(w_i \mid c_j) = n(c_j \to w_i) \big/ n(c_j)$
  - Transition probabilities $p(c_j \mid c_k) = n(c_k \to c_j) \big/ n(c_k)$
- We can of course **smooth** these estimates to avoid 0s and not overfit the data.

(See Python notebook book for HMM POS tagging)

Learning

- What if we don’t have fully labelled data?
- We use the **Forward-Backward (Baum-Welch) algorithm**
  - Similar to Forward algorithm, but combine:
    - Forward probability of getting to this state from start
    - Backward probability of getting from this state to end
  - (wait for parsing lecture)



Generalising HMMs

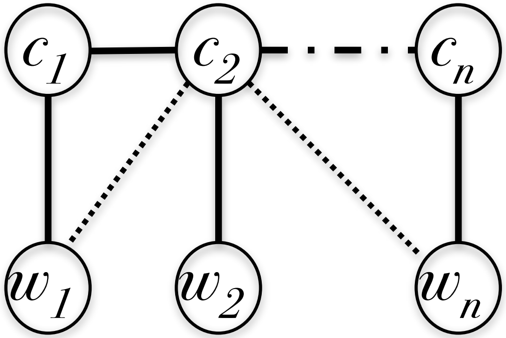
- So far we've assumed the emission probabilities only apply to single observations. What if there is a **class sequence** associated to each observed word?
  - Answer: The emission probabilities from a single underlying class can apply to a sequence of observations (can be an n-gram type structure).
- Also, we've only looked at 1st order (bigram) Markov models, largely because their transition probabilities are easy to show in a 2D matrix. What if it made sense for the underlying model to use other previous states (not just the last one)?
  - Answer: It is possible to generalize the Markov Model to an **arbitrary order** (see n-grams in language modelling lecture)

OUTLINE

1) Sequence Tagging Tasks: POS tagging and NER
2) Generative: Hidden Markov Models
3) Discriminative: Conditional Random Fields

Conditional Random Fields

- Can we use a **discriminative** approach instead? (usually better than generative models with enough data!)
  - Remember alternative text classification methods:
    - Naïve Bayes: generative – estimate p(d|c)p(c)
    - Logistic Regression: discriminative – p(c|d) directly



- Conditional Random Fields
  - "logistic regression for sequences"
    - (usually called "Maximum Entropy" in fact)
  - HMM (generative):

    C_{MAP} = argmax$_C$ p(C|W) = argmax$_C$ p(W|C)p(C)
  - CRF (discriminative):

    C_{MAP} = argmax$_C$ p(C|W)

$$p(C \mid W) = \frac{1}{Z}\prod_i \exp(\sum_j \lambda_j f_j(y_{i-1}, y_i, W, i))$$

  - Define features f, learn optimal weights λ
    - e.g. f_i = "w_i = flies, c_i = NNS", f_i' = "c_{i-1} = NN, c_i = NNS"
    - or even f_i'' = "w_{i-1} = fruit, w_i = flies, c_{i-1} = NN, c_i = NNS"

Conditional Random Fields

- A CRF model consists of
  - **F** = <f_1, …, f_k>, a vector of “feature functions”
  - **θ** = < θ_1, …, θ_k>, a vector of weights for each feature function.
- Let **O** = < o_1, …, o_T> be an observed sentence
- Let **A** = <a_1, …, a_T> be the latent variables.

$$P(\mathbf{A} = \mathbf{y} \mid \mathbf{O}) = \frac{\exp\left(\boldsymbol{\theta} \cdot \mathbf{F}\left(\mathbf{y}, \mathbf{O}\right)\right)}{\sum_{\mathbf{y}'} \exp\left(\boldsymbol{\theta} \cdot \mathbf{F}\left(\mathbf{y}', \mathbf{O}\right)\right)}$$

- This is the same as the Maximum Entropy equation.

Finding the Best Sequence

Best sequence is:

$$\begin{aligned}\arg\max_{\mathbf{y}} P(\mathbf{A} = \mathbf{y} \mid \mathbf{O}) &= \arg\max_{\mathbf{y}} \left[\frac{1}{Z(\mathbf{O})} \exp\left(\boldsymbol{\theta} \cdot \mathbf{F}\left(\mathbf{y}, \mathbf{O}\right)\right) \right] \\ &= \arg\max_{\mathbf{y}} \left[\boldsymbol{\theta} \cdot \mathbf{F}\left(\mathbf{y}, \mathbf{O}\right)\right]\end{aligned}$$

Recall from HMM discussion:

If there are:

K possible states for each y_i variable,

and N total y_i variables,

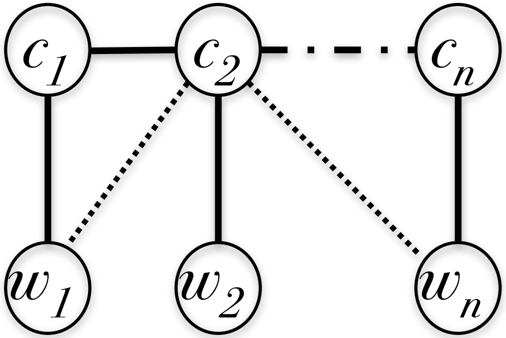
Then there are K^N possible settings for y

So brute force can't find the best sequence.

Instead, we resort to a Viterbi-like dynamic program.

Conditional Random Fields

- Advantages:
  - You can define (nearly) arbitrary features
  - Often outperform HMMs
  - Available implementations e.g. NLTK CRF tagger
- Disadvantages:
  - Complex inference (dynamic programming again)
  - Needs manual definition of features
  - Output is not a sequence probability
    - it's the confidence of sequence given the data
  - (i.e. it's not really a language model)



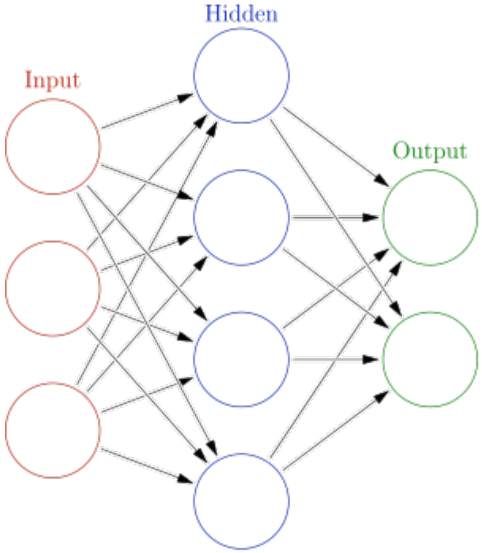
- In general, this is **structured prediction** rather than **classification**
  - Predicting structured objects not just classes/values

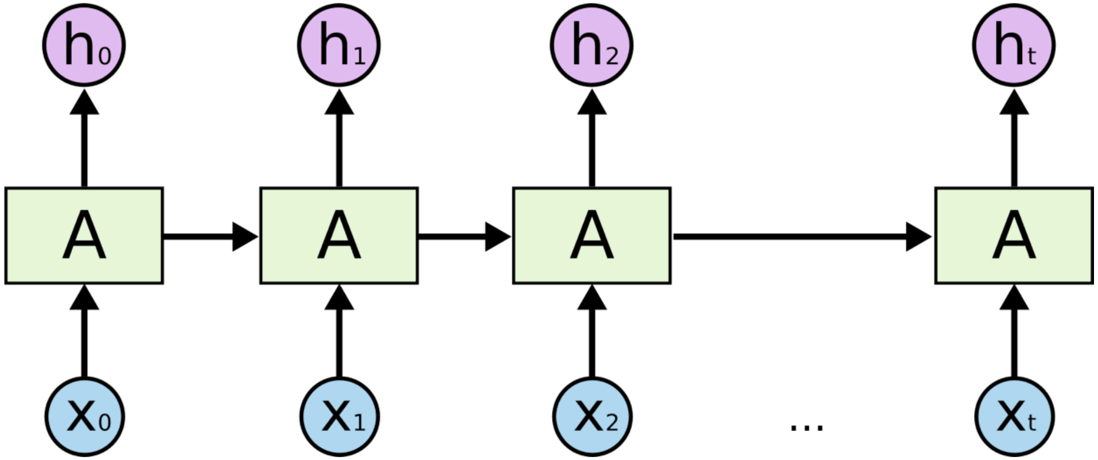
Conditional Random Fields

- **See Python Notebook CRF_POS_tagger.py**

- **In your own time you can train it on your own datasets.**

Extra: Recurrent Neural Networks





(see NLP and Deep Learning course next term!)

http://en.wikipedia.org
http://colah.github.io

Sequence Models

- Hidden Markov Models
  - Like Language Models, use Markov Models of a given order.
  - Though the Markov Model not directly observed.
  - ‘Flip’ the sequence likelihoods round in a Bayesian style.
  - Robust, good baseline for sequence tagging tasks
  - Learnable without much labelled data
    - But no exact solution – see next lecture
  - Be careful with smoothing!

- Conditional Random Fields / Recurrent Neural Nets
  - Discriminative: higher accuracy for many tasks
  - More complex learning; need more data
  - Can be more complex feature definition process
  - Be careful with regularisation, weighting, activation functions, …

Reading

- Jurafsky and Martin (3rd Ed.):
  - Chapter 8 (POS tagging and HMMs)
- Manning and Schuetze (1999):
  - Chapter 9 (Markov Models)
  - Chapter 10 (POS tagging & HMMs)