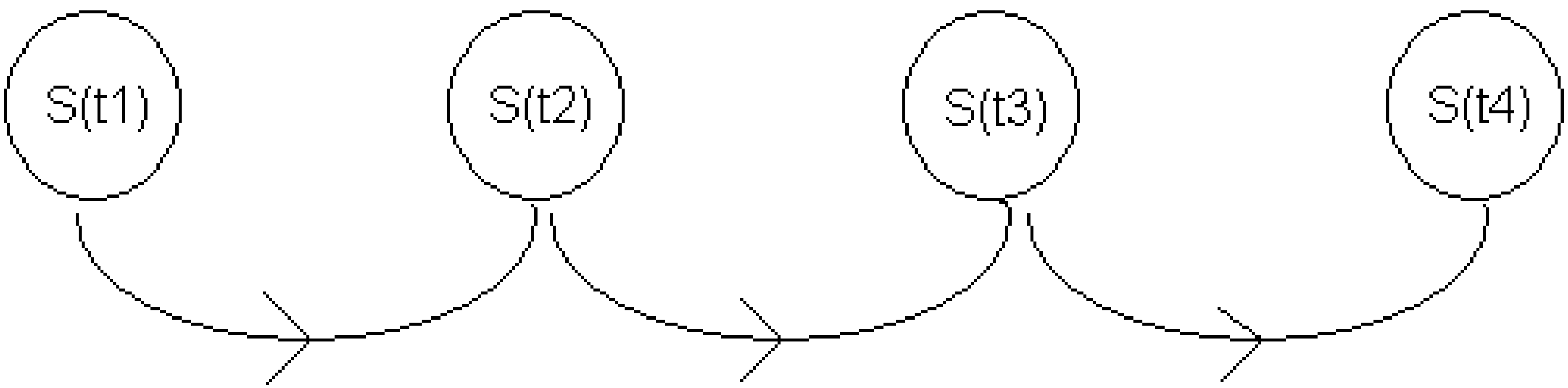
Hidden Markov Models: Algorithms and Applications

Introduction

- Often we are interested in finding patterns in signals which change over a space or time.
- For example:
  - commands used in instructing a computer
  - sequences of words in sentences
  - sequence of phonemes in spoken words
- *i.e. areas where a sequence of events occurs could produce useful patterns.*

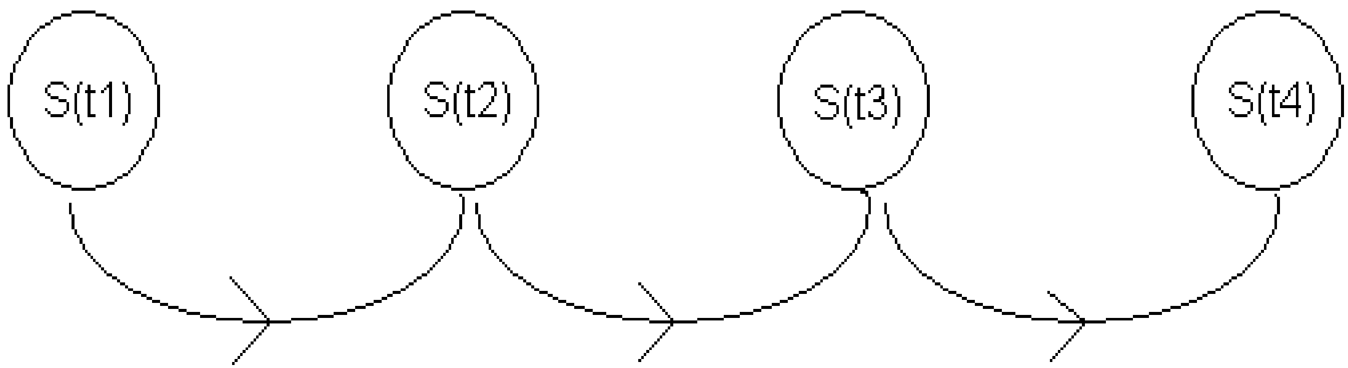
Markov models

- In Markov Model we model a system as a finite set of *states*
- The system makes transitions from one state to another with some transition probability



Markov models contd.

- In Markov Models **we assume** that a state of a system depends only on the previous ***k*** states.
- e.g. if ***k*** = 2 then S(t4) depends only on S(t3) and S(t2) and not on S(t1).



An example

- Consider the current value of shares of a particular company.
- A stock-broker would be interested in knowing whether the value of the share is going to increase, decrease or remain unchanged.
- Thus, there are three possible states:

  I (increase)

  D (decrease)

  U (unchanged)

The example contd.

- Say we make observe the share value for several days and note whether it increased, decreased or remained same.
- We get a sequence like

U U I I I U U I I D D D D I I U U D U D

- What conclusions would we like to draw from the above sequence?
- Obviously we would like to know whether the share values are going to increase / decrease / remain unchanged in the near future.
- In other words, given the state today and of the immediate past, we would like to ***predict*** tomorrow's state.

The example contd.

- Recall, the sequence was:

U U I I I U U I I D D D D I I U U D U D

We can get the probabilities of

- observing a particular state (7/20 for **U***, 7/20 for* **I***, 6/20 for* **D***)*

- this observation is not very informative. It does not tell us that if the state is **D** *today then what state it is going to be in tomorrow since all states have nearly equal probabilities.*

The example contd.

- We can also calculate the *transition probabilities*
- For example:

In the sequence

U U I I I U U I I D D D D I I U U D U D

\- *transitions from one state to another*

(3/7 for **U** → **U**, 2/7 for **U** → **I**, 2/7 for **U** → **D** etc.)

The example contd.

- We can create a *transition probability matrix:*
- For the sequence

U U I I I U U I I D D D D I I U U D U D

$$A = \begin{array}{c|ccc|c} & U & I & D & \\ & 3/7 & 2/7 & 2/7 & U \\ & 2/7 & 4/7 & 1/7 & I \\ & 1/5 & 1/5 & 3/5 & D \end{array}$$

Note: the discrepancy in the last row.

The last **D** *is not considered since there is no transition corresponding to it.*

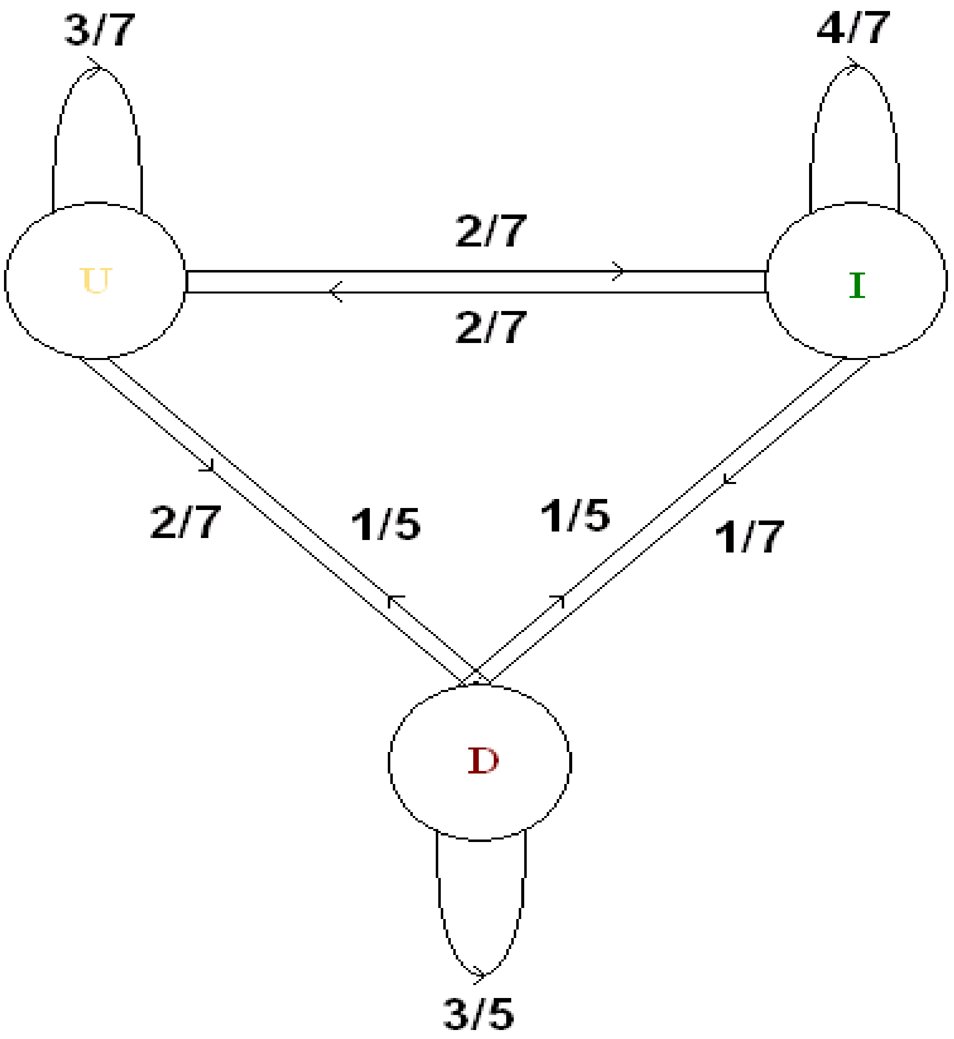
For sufficiently large data the discrepancy will be small.

The example contd.

- We have built a Markov model with **k** = 1 from the given data.
- We can now use it to predict.
- The probability of getting a **D** on the next day is 3/5 while the probabilities of getting **U** or **I** are 1/5 each.
- **Advice: *Don’t buy the share today.***

State transition diagrams

- We commonly represent the system by a state transition diagram.
- The numbers on the directed edges indicate the transition probabilities.



Role of initial state

- Let us start with the system in the state **U**
- Given the transition matrix as

$$
A = \begin{array}{ccc|c} U & I & D & \\ 3/7 & 2/7 & 2/7 & U \\ 2/7 & 4/7 & 1/7 & I \\ 1/5 & 1/5 & 3/5 & D \end{array}
$$

- We can represent the initial state as the vector

  $S(t_0) = \pi = (1, 0, 0)$

- To find the state of the system at the next time slot we have

  $S(t_1) = \pi * A$

- In general:

  $S(t_{j+1}) = S(t_j) * A$

Hidden Markov Models

- We have assumed that we know the system i.e. we know the possible states of the system.

Note: It is possible that we are not be able to observe the system directly. Instead we may be able to observe some effect of the system.

- Assumptions:

- there is an underlying system
- the system follows the Markov assumption
- we can not observe the system directly
- we can observe some effect of the system
- the underlying state of the system is responsible for the observation.

Some applications of HMM

- Speech recognition

  (observed: acoustic signal, hidden: words)
- Hidden states – phonemes
- Observations – words as heard
- Transitions – probability of one phoneme following another to make a word

- Handwriting recognition

  (observed: image, hidden: words)
- Part-of-speech tagging

  (observed: words, hidden: part-of-speech tags)
- Machine translation

  (observed: words in source language, hidden: words in target language)

A possible scenario

- Assume that we can not observe the value of the share directly.
- Instead we can observe what a stock-broker does with those shares. He either buys more shares, sells the shares bought earlier or does nothing.

Possible observables are:

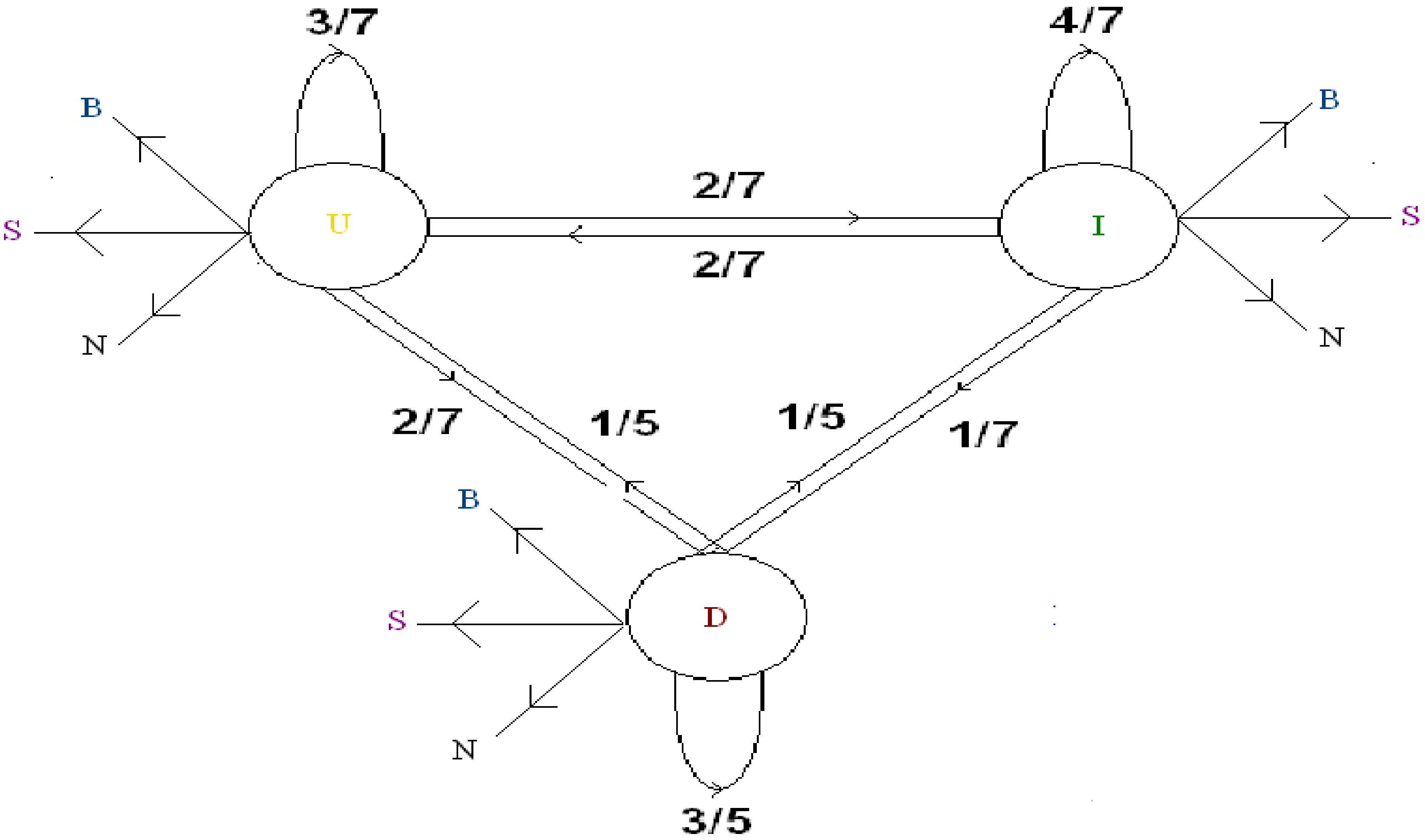
B (buy), **S** (sell) and **N** (do nothing)

- We can get sequences like

B B S N B S B B N ….

- Each possible state of the system can generate any of the given observations with a given probability
- i.e. The states, **U**, **I** or **D** can generate all the observables, **B**, **S** or **N** with some probabilities (*emission probabilities*).

State diagram for HMM



Inputs for an HMM

- A system that can be in some states, x_i
- Transition probabilities between states, a_{ij}
- A set of observables, y_k
- Emission probabilities of observables from a state, b_{jk}
- A start state
- An HMM is characterized by the triplet

  $\Lambda = (\{a_{ij}\}, \{b_{ij}\}, \pi)$
- Where

- $a_{ij} = P(x_i(t+1) \mid x_j(t))$; $a_{ij} \geq 0$; $\Sigma_{j=1}^{N} a_{ij} = 1$ for all i

- $b_{jk} = P(y_k(t) \mid x_j(t))$; $b_{jk} \geq 0$; $\Sigma_{k=1}^{M} b_{jk} = 1$ for all j

Three Basic HMM problems

- Problem 1 (Evaluation):

  Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model, how do we compute the probability of O given the model?

- Problem 2 (Decoding):

  Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model, how do we find the state sequence that best explains the observations?

- Problem 3 (Learning):

  How do we adjust the model parameters $\Lambda = (\{a_{ij}\}, \{b_{ij}\}, \pi)$, to maximize $P(O|\Lambda)$?

Decoding (Viterbi) Algorithm

- Given an HMM:

  $\Lambda = (\{a_{ij}\}, \{b_{jk}\}, \pi)$

  and a sequence of observations

  $O = \{o(1), o(2), o(3), \ldots, o(T)\}$

  what is the most likely sequence of hidden states that produced the given set of observations?

Decoding (Viterbi) Algorithm contd…

- We want to maximize $\delta_t(i)$ i.e.

$$\delta_t(i) = \max_{s_1, s_2, s_3, \ldots, s_{t-1}} P(s_1, s_2, s_3, \ldots, s_{t-1}, s_t = i; o_1, o_2, o_3, \ldots, o_{t-1} | \Lambda)$$

- We get the recursion

$$\delta_{t+1}(j) = b_{j\,o(t+1)} \{ \max_{1 \le i \le N} \delta_t(i)\, a_{ij} \}$$

With initial condition $\delta_1(j) = \pi_j \; b_{j\,o(1)}$

Word classes &

Part of speech tagging

Word Classes

- Basic word classes: Noun, Verb, Adjective, Adverb, Preposition, …
- Open vs. Closed classes
  - Open:
    - Nouns, Verbs, Adjectives, Adverbs.
  - Closed:
    - determiners: a, an, the
    - pronouns: she, he, I
    - prepositions: on, under, over, near, by, …

Open Class Words

Every known human language has nouns and verbs

Nouns: people, places, things

- Classes of nouns
  - proper vs. common
  - count vs. mass

Verbs: actions and processes

Adjectives: properties, qualities

Adverbs: hodgepodge!

- *Unfortunately*, John walked *home extremely slowly yesterday*

Closed Class Words

- Differ more from language to language
- Examples:
  - prepositions: on, under, over, …
  - particles: up, down, on, off, …
  - determiners: a, an, the, …
  - pronouns: she, who, I, ..
  - conjunctions: and, but, or, …
  - auxiliary verbs: can, may, should, …
  - Numerals: one, two, three, third, …

Prepositions (and particles) of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

Coordinating and subordinating conjunctions of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	why	1,333	directly	26	insomuch as	0
because	23,626	now	1,290	ere	12	insomuch that	0
so	12,933	neither	1,120	notwithstanding	3	like	0
before	10,720	whenever	913	according as	0	neither nor	0
though	10,329	whereas	867	as if	0	now that	0
than	9,511	except	864	as long as	0	only	0
while	8,144	till	686	as though	0	provided that	0
after	7,042	provided	594	both and	0	providing that	0
whether	5,978	whilst	351	but that	0	seeing as	0
for	5,935	suppose	281	but then	0	seeing as how	0
although	5,424	cos	188	but then again	0	seeing that	0
until	5,072	supposing	185	either or	0	without	0

Pronouns of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

it	199,920	how	13,137	yourself	2,437	no one	106
I	198,139	another	12,551	why	2,220	wherein	58
he	158,366	where	11,857	little	2,089	double	39
you	128,688	same	11,841	none	1,992	thine	30
his	99,820	something	11,754	nobody	1,684	summat	22
they	88,416	each	11,320	further	1,666	suchlike	18
this	84,927	both	10,930	everybody	1,474	fewest	15
that	82,603	last	10,816	ourselves	1,428	thyself	14
she	73,966	every	9,788	mine	1,426	whomever	11
her	69,004	himself	9,113	somebody	1,322	whosoever	10
we	64,846	nothing	9,026	former	1,177	whomsoever	8
all	61,767	when	8,336	past	984	wherefore	6
which	61,399	one	7,423	plenty	940	whereat	5
their	51,922	much	7,237	either	848	whatsoever	4
what	50,116	anything	6,937	yours	826	whereon	2
my	46,791	next	6,047	neither	618	whoso	2
him	45,024	themselves	5,990	fewer	536	aught	1
me	43,071	most	5,115	hers	482	howsoever	1
who	42,881	itself	5,032	ours	458	thrice	1
them	42,099	myself	4,819	whoever	391	wheresoever	1
no	33,458	everything	4,662	least	386	you-all	1
some	32,863	several	4,306	twice	382	additional	0
other	29,391	less	4,278	theirs	303	anybody	0
your	28,923	herself	4,016	wherever	289	each other	0
its	27,783	whose	4,005	oneself	239	once	0
our	23,029	someone	3,755	thou	229	one another	0
these	22,697	certain	3,345	'un	227	overmuch	0
any	22,666	anyone	3,318	ye	192	such and such	0
more	21,873	whom	3,229	thy	191	whate'er	0
many	17,343	enough	3,197	whereby	176	whenever	0
such	16,880	half	3,065	thee	166	whereof	0
those	15,819	few	2,933	yourselves	148	whereto	0
own	15,741	everyone	2,812	latter	142	whereunto	0
us	15,724	whatever	2,571	whichever	121	whichsoever	0

Word Classes: Tag Sets

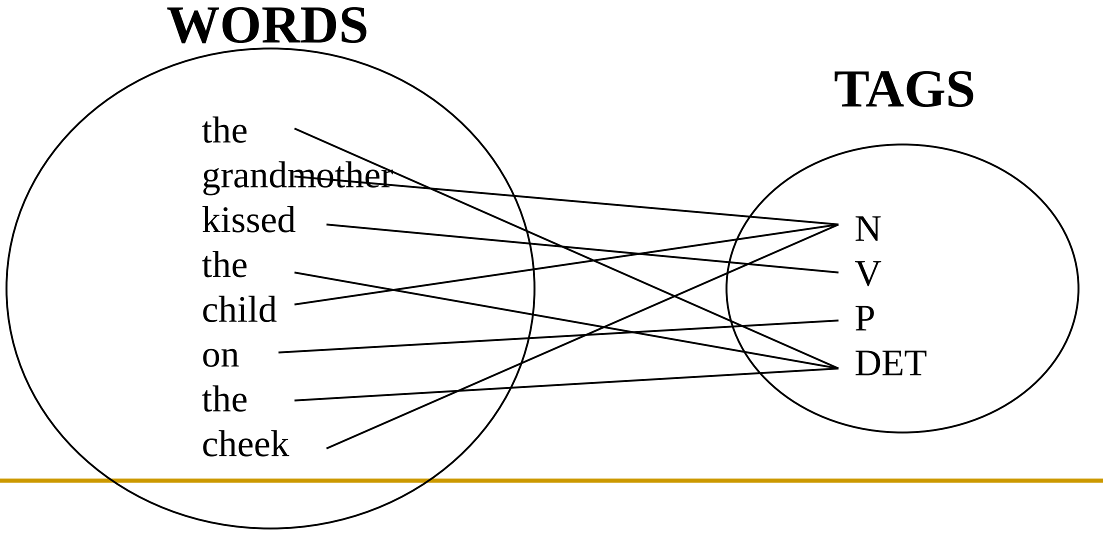
- Vary in number of tags: a dozen to over 200
- Size of tag sets depends on language, objectives and purpose

Penn Treebank part-of-speech tags (including punctuation).

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*'s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Definition

"The process of assigning a part-of-speech or other lexical class marker to each word in a corpus" (Jurafsky and Martin)



An Example

WORD	LEMMA	TAG
the	the	+DET
grandmother	grandmother	+NOUN
kissed	kiss	+VPAST
the	the	+DET
child	child	+NOUN
on	on	+PREP
the	the	+DET
cheek	cheek	+NOUN

From: http://www.xrce.xerox.com/competencies/content-analysis/fsnlp/tagger.en.html

Example of Penn Treebank Tagging of Brown Corpus Sentence

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

VB	DT	NN	.
Book	that	flight	.

VBZ	DT	NN	VB	NN	?
Does	that	flight	serve	dinner	?

The Problem

Words often have more than one word class: *this*

- *This* is a nice day = PRP
- *This* day is nice = DT
- You can go *this* far = RB

Word Class Ambiguity (in the Brown Corpus)

Unambiguous (1 tag): 35,340

Ambiguous (2-7 tags): 4,100

2 tags	3,760
3 tags	264
4 tags	61
5 tags	12
6 tags	2
7 tags	1

(Derose, 1988)

Part-of-Speech Tagging

- Rule-Based Tagger: ENGTWOL (ENGlish TWO Level analysis)
- Stochastic Tagger: HMM-based
- Transformation-Based Tagger (Brill)

Stochastic Tagging

- Based on probability of certain tag occurring given various possibilities
- Requires a training corpus
- No probabilities for words not in corpus.
- Training corpus may be different from test corpus.

HMM Tagger

- Intuition: Pick the most likely tag for this word.
- HMM Taggers choose tag sequence that maximizes this formula:
  - P(word|tag) × P(tag|previous n tags)
- Let $T = t_1,t_2,\ldots,t_n$
  Let $W = w_1,w_2,\ldots,w_n$
- Find POS tags that generate a sequence of words, i.e., look for most probable sequence of tags T underlying the observed words W.

Start with Bigram-HMM Tagger

$\text{argmax}_T\ P(T|W)$

$\text{argmax}_T P(T)P(W|T)$

$\text{argmax}_t P(t_1 \ldots t_n)P(w_1 \ldots w_n|t_1 \ldots t_n)$

$\text{argmax}_t [P(t_1)P(t_2|t_1) \ldots P(t_n|t_{n-1})][P(w_1|t_1)P(w_2|t_2) \ldots P(w_n|t_n)]$

To tag a single word: $t_i = \text{argmax}_j\ P(t_j|t_{i-1})P(w_i|t_j)$

How do we compute $P(t_i|t_{i-1})$?

- $c(t_{i-1}t_i)/c(t_{i-1})$

How do we compute $P(w_i|t_i)$?

- $c(w_i,t_i)/c(t_i)$

How do we compute the most probable tag sequence?

- Viterbi

Markov Model Taggers

- Bigram tagger
  - Make predictions based on the preceding tag
  - The basic unit is the preceding tag and the current tag
- Trigram tagger
  - We would expect more accurate predictions if more context is taken into account
  - RB(adverb) VBD(past tense) Vs RB VBN(past participle) ?
    - Ex: “clearly marked”
    - Is clearly marked : *P*(BEZ RB VBN) > *P*(BEZ RB VBD)
    - He clearly marked : *P*(PN RB VBD) > *P*(PN RB VBN)

An Example

Secretariat/NNP is/VBZ expected/VBN to/TO **race**/VB tomorrow/NN

People/NNS continue/VBP to/TO inquire/VB the DT reason/NN for/IN the/DT **race**/NN for/IN outer/JJ space/NN

to/TO race/???

the/DT race/???

$t_i = \text{argmax}_j\ P(t_j|t_{i-1})P(w_i|t_j)$

max[P(VB|TO)P(race|VB) , P(NN|TO)P(race|NN)]

For example: *race* has the following probabilities in the Brown corpus:

P(NN|race) = .98

P(VB|race)= .02

An Early Approach to Statistical POS Tagging

- PARTS tagger (Church, 1988): Stores probability of tag given word instead of word given tag.
- P(tag|word) × P(tag|previous n tags)
- Compare to:
  - P(word|tag) × P(tag|previous n tags)