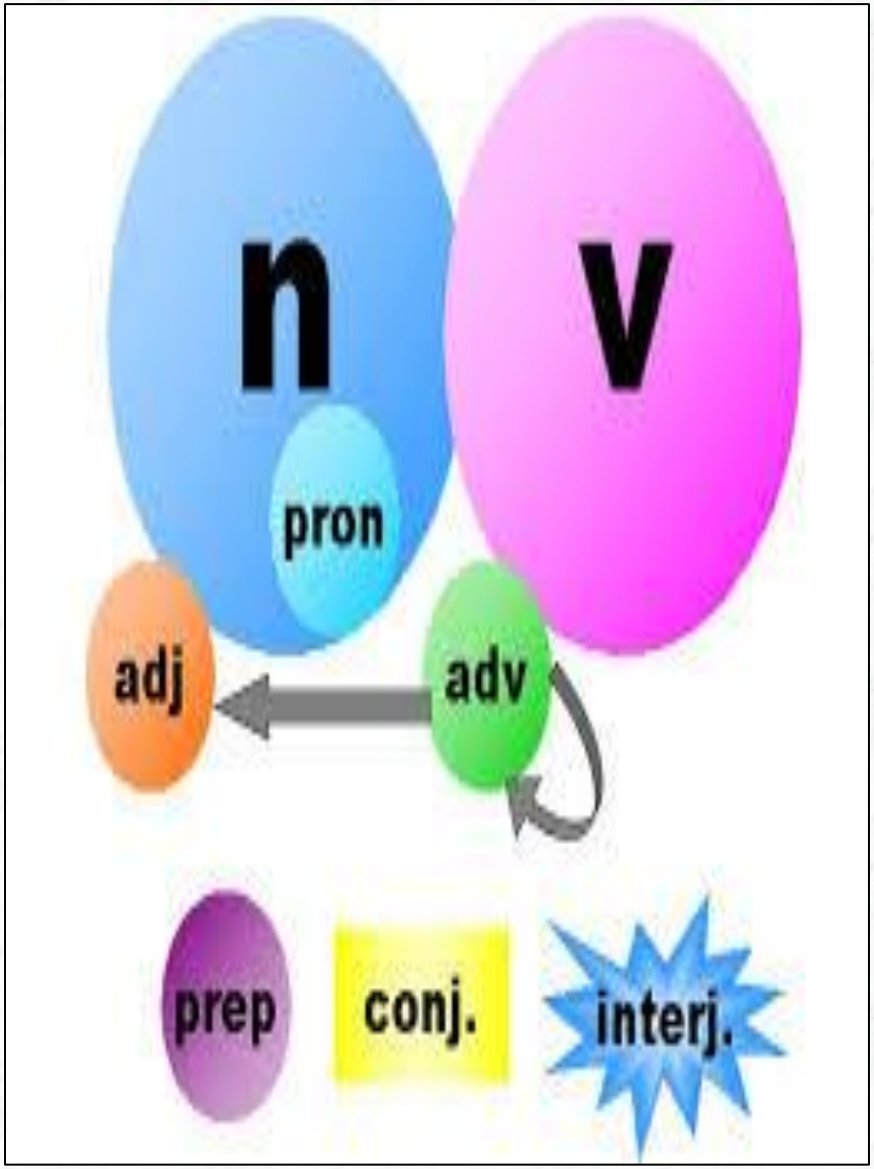
Part of speech tagging



Word Classes
and
Part Of Speech Tagging

Instructor : Dr. Hanaa Bayomi Ali
Mail : h.mobarz @ fci-cu.edu.eg

Stage 4: Syntax

- Up until now we have been dealing with individual words and simple-minded (though useful) notions of what sequence of words are likely.
- Now we turn to the study of how words
  - -Are clustered into classes
  - -Group with their neighbors to form phrases and sentences
  - -Depend on other words
- Interesting notions:
  - -Word order (Subject + Verb +object)
  - -Constituency parser
  - -Grammatical relations

What is a word class?

- Words that somehow ‘behave’ alike:

  - Appear in similar contexts

    “earth” and “soil”

  - Perform similar functions in sentences

    “verb class” ⇨ Actions (walk, ate) and states (be, exude)

  - Undergo similar transformations

parts-of-speech

- Traditional parts of speech

  - Noun, verb, adjective, preposition, adverb, article, interjection (!), pronoun, conjunction, etc

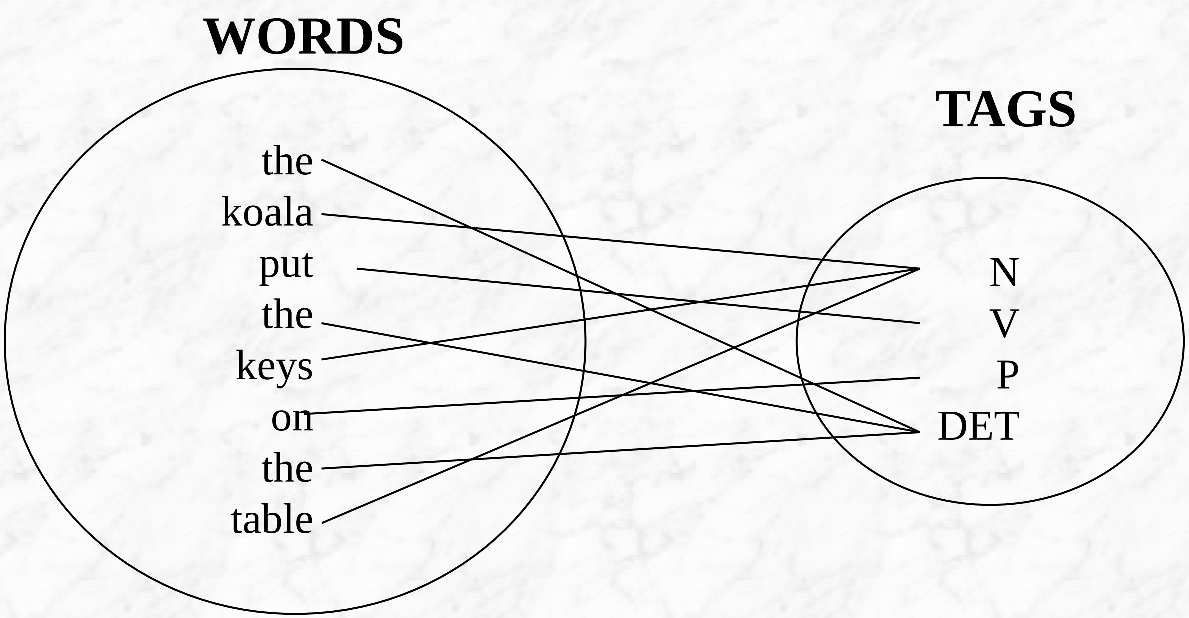
  - Called: parts-of-speech, lexical category, word classes, morphological classes, lexical tags, POS

POS examples

Tags	***meaning***	***Examples***
N	noun	chair, bed, apple
V	verb	study, debate, eat
ADJ	adjective	purple, tall, smart, beautiful
ADV	adverb	unfortunately, slowly,
P	preposition	of, by, to
PRO	Pronoun	I, me, mine
DET	determiner	the, a, that, those

POS Tagging: Definition

The process of assigning a part-of-speech or lexical class marker to each word in a corpus:



POS Tagging example

WORD	*tags*
the	DET
Koala	N
Put	V
The	DET
Keys	N
On	P
The	DET
Table	N

What is POS tagging good for?

- Understanding how words can and should be **joined together to make sentences that are both grammatically correct and readable**.

- Used in Stemming for Machine translation, since knowing the word's POS can help tell us which logical affixes it can take.
  - Book(s)   help(s)

- Word prediction in speech recognition

  Possessive pronouns (my, your, her) followed by nouns

  Personal pronouns (I, you, he) likely to be followed by verbs.

What is POS tagging good for?

- Help in building **automatic word sense disambiguation algorithm**
  - watch “verb”   and watch “noun”
- Corpora that have been marked for part-of-speech are very useful for linguistic research, for example to help find instances or frequencies of particular constructions in large corpora.

Open and closed class words

Parts of speech are divided into two broad categories:

1- **Open class (or content) words accept the addition of new** words through morphological processes such as compounding, derivation, etc.

(emailed , faxable, skype)

2- **Closed class (or function) words do not normally accept** addition of new items

- Prepositions: of, in, by, …
- Auxiliaries: may, can, will had, been, …
- Pronouns: I, you, she, mine, his, them, …
- Usually function words (short common words which play a role in grammar)

Tagset

- What set of parts of speech do we use?

- Most tagsets implicitly encode fine-grained specializations of 8 basic parts of speech (POS, word classes, morphological classes, lexical tags):

    Noun, verb, pronoun, preposition, adjective, conjunction, article, adverb

- Vary in number of tags: a dozen to over 200

- **Size of tag sets** depends on *language, objectives and purpose*

Tagset

•These categories are based on *morphological and distributional similarities* and not, as you might think, semantics.

•In some cases, tagging is fairly straightforward (at least in a given language), in other cases it is not.

Distribution of Tags

- Parts of speech follow the usual frequency-based distributional behavior
  - **Most word** types have **only one part** of speech
  - Of the rest, **most have two**

    (“Like” can be a verb or a preposition)
  - A **small number** of word types **have lots of parts** of speech

    (“Around” can be a preposition, particle, or adverb)

- Unfortunately, **the word types with lots of parts of speech** occur **with high frequency** (and words that occur most frequently tend to have multiple tags)

Distribution of Tags – Brown

•**The Brown Corpus** of Standard American English was the first of the modern, computer readable general corpora. (Compiled at Brown University)

•Corpus consists of 1 million words of American English text printed in 1961.

<u>To see the problem:</u>

Unambiguous (1 tag): 35,340

Ambiguous (2-7 tags): 4,100

11.5% ambiguous

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

POS tagging: Choosing a tagset

- There are so many parts of speech, potential distinctions we can draw
- To do POS tagging, need to choose a standard set of tags to work with

1- Could pick very coarse tagets

N, V, Adj, Adv.

2- Brown Corpus (Francis & Kucera ‘82), ***1M words,87 tags***

3- Penn Treebank: hand-annotated corpus of *Wall Street Journal*, ***1M words, 45-46 tags***

- Commonly used

Penn TreeBank POS Tag set

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 ”)
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	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*			

PART OF SPEECH TAGGING METHODS

POS Tagging

The Tagging Task

Input : `the lead paint is unsafe`

Output: `the/Det lead/N paint/N is/V unsafe/Adj`

POS Tagging Methods:

1. Manual Tagging
2. Machine Tagging
3. A Combination of Both

Manual Tagging

Methods:

1. Agree on a Tagset after much discussion.
2. Chose a corpus, annotate it manually by two or more people.
3. Check on inter-annotator agreement.
4. Fix any problems with the Tagset (if still possible).

Machine Tagging

1. Rule based tagging.
2. Stochastic tagging.

1- Rule-based tagging

A Two-stage architecture

1- Use lexicon FST (dictionary) to tag each word
with all possible POS
- Apply hand-written rules to eliminate tags.

2- The rules eliminate tags that are inconsistent with the context and should reduce the list of POS tags to a single POS per word.

Start with a dictionary

- she: PRP
- promised: VBN,VBD
- To: TO
- back: VB, JJ, RB, NN
- the: DT
- bill: NN, VB

Personal pronoun

Past-participle , past tens

Verb,Adjective,adverb,Noun

Determine

Verb,Noun

- Etc… for the ~100,000 words of English

Use the dictionary to assign every possible tag

			NN		
			RB		
	VBN		JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

Write rules to eliminate tags

Eliminate VBN if VBD is an option when VBN|VBD follows “<start> PRP”

			NN		
			RB		
			JJ		VB
PRP	VBD	TO	VB	DT	NN
She	**promised**	**to**	**back**	**the**	**bill**

The ENGTWOL tagger

- Morphology for lemmatization.
- 56 000 entries for English word stems (first pass)
- 1100 handwritten constraints to eliminate tags (second pass)

Sample ENGTWOL Lexicon

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NOINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	IMPERATIVE VFIN
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

2- Stochastic Tagging

- Based on probability of certain tag occurring given various possibilities

- Requires a training corpus

- Simple Method: Choose most frequent tag in training text for each word!

- HMM is an example

The Most Frequent Tag algorithm

- For each word
  - Create dictionary with each possible tag for a word
  - Take a tagged corpus
  - Count the number of times each tag occurs for that word
- Given a new sentence
  - For each word, pick the most frequent tag for that word from the corpus.

Hidden Markov Map (HMM)

Making some simplifying Markov assumptions, the basic HMM equation for a single tag is:

$t_i = \text{argmax}_i\ P(t_i \mid t_{i-1}) * P(w_i \mid t_i)$

-The function $\text{argmax}_x F(x)$ means "the x such that F(x) is maximized"

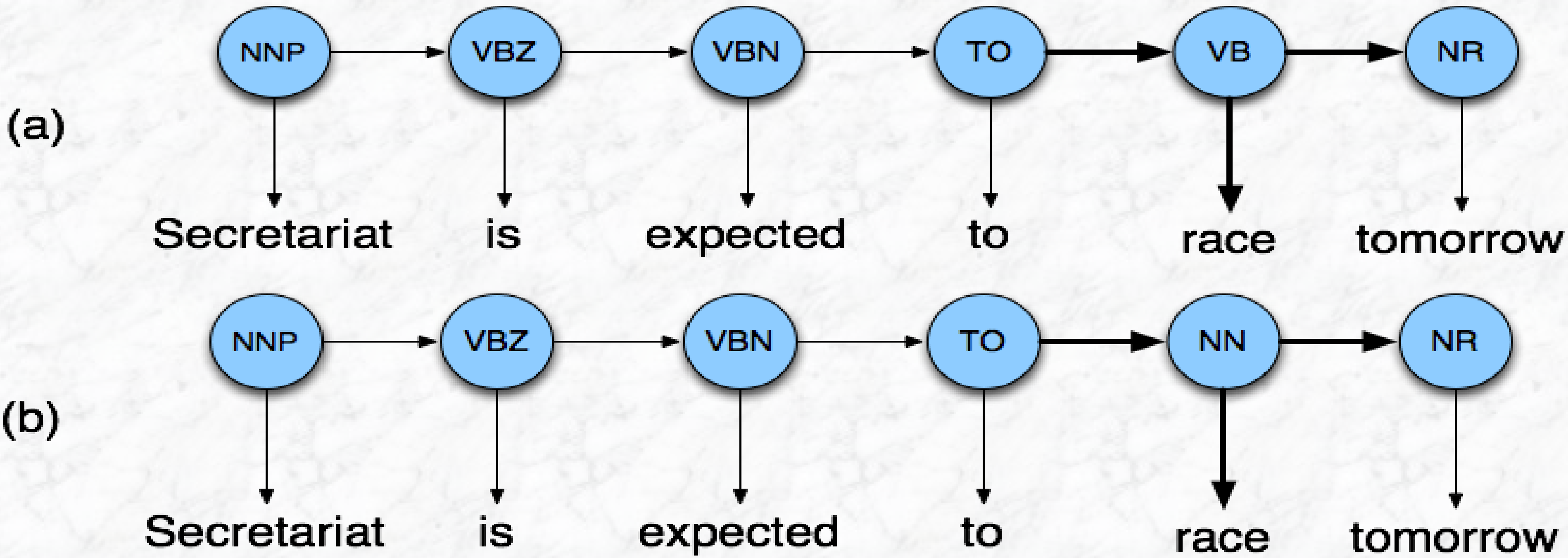
-The first P is the tag sequence probability, the second is the word likelihood given the tag.

Most of the **better statistical models report around 95% accuracy on standard datasets**

But, note you get **91% accuracy just by picking the most likely tag!**

A Simple Example

From the Brown Corpus



Assume previous words have been tagged, and we want to tag the word *race*.

Bigram tagger

- to/TO *race/?*
- the/DT *race/?*

A Simple Example

Goal: choose between **NN** and **VB** for the sequence *to race*

Plug these into our bigram HMM tagging equation:

P(race | VB) * P(VB | TO)

P(race | NN) * P(NN | TO)

How do we compute the tag sequence probabilities and the word likelihoods?

Word Likelihood

We must compute the likelihood of the word race given each tag. I.e., **P(race | VB)** and **P(race | NN)**

Note: we are **NOT** asking which is the most likely tag for the word.

Instead, we are asking, **if we were expecting a verb, how likely is it that this verb would be *race*?**

From the Brown and Switchboard Corpora:

P(*race* | VB) = .00003

P(*race* | NN) = .00041

Tag Sequence Probabilities

Computed from the corpus by counting and normalizing.

We expect VB more likely to follow TO because infinitives (*to race, to eat*) are common in English, but it is possible for NN to follow TO (*walk to school, related to fishing*).

From the Brown and Switchboard corpora:

P(VB | TO) = .340

P(NN | TO) = .021

And the Winner is…

Multiplying tag sequence probabilities by word likelihoods gives

P(*race* | VB) = .00003
P(*race* | NN) = .00041
P(VB | TO) = .340
P(NN | TO) = .021

P(*race* | VB) * P(VB | TO) = .000010

P(*race* | NN) * P(NN | TO) = .000007

So, even a simple bigram version correctly tags race as a VB, despite the fact that it is the less likely sense.

And the Winner is…

Multiplying tag sequence probabilities by word likelihoods gives

P(*race* | VB) = .00003
P(*race* | NN) = .00041
P(VB | TO) = .340
P(NN | TO) = .021

P(*race* | VB) * P(VB | TO) = .000010

P(*race* | NN) * P(NN | TO) = .000007

So, even a simple bigram version correctly tags race as a VB, despite the fact that it is the less likely sense.

Statistical POS Tagging (whole sequence)

Challenges

1- Multivariable output

- make multiple prediction simultaneously

2- Variable length output

- Sentence length not fixed

Statistical POS Tagging (whole sequence)

Goal: choose the best sequence of tags T for a sequence of words W in a sentence

$$T' = \underset{T \in \tau}{\arg\max}\, P(T|W)$$

By Bayes Rule (giving us something easier to calculate)

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

Since we can ignore P(W), we have

$$T' = \underset{T \in \tau}{\arg\max}\, P(T)P(W|T)$$

Statistical POS Tagging (whole sequence)

Goal: choose the best sequence of tags T for a sequence of words W in a sentence

$$T' = \operatorname*{argmax}_{T \in \tau} P(T \mid W)$$

By Bayes Rule (giving us something easier to calculate)

$$P(T \mid W) = \frac{P(T)P(W \mid T)}{P(W)}$$

Since we can ignore P(W), we have

$$\hat{t}_1^n = \operatorname*{argmax}_{t_1^n} \overbrace{P(w_1^n \mid t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

Statistical POS Tagging: the Prior

$P(T) = P(t_1, t_2, \ldots, t_{n-1}, t_n)$

By the Chain Rule:

$= P(t_n \mid t_1, \ldots, t_{n-1})\, P(t_1, \ldots, t_{n-1})$

$$= \prod_{i=1}^{n} P(t_i \mid t_1^{i-1})$$

Making the Markov assumption:

$\approx P(t_i \mid t_{i-N+1}^{i-1})$ e.g., for bigrams, $\prod_{i=1}^{n} P(t_i \mid t_{i-1})$

Statistical POS Tagging: the (Lexical) Likelihood

$P(W|T) = P(w_1, w_2, \ldots, w_n \mid t_1, t_2, \ldots, t_n)$

From the Chain Rule:

$$= \prod_{i=1}^{n} P(w_i | w_1 t_1 \ldots w_{i-1} t_{i-1} t_i)$$

Simplifying assumption: probability of a word depends only on its own tag $P(w_i|t_i)$

$$\approx \prod_{i=1}^{n} P(w_i | t_i)$$

So...

$$T' = \underset{T \in \tau}{\arg\max} \prod_{i=1}^{n} P(t_i | t_{i-1}) \prod_{i=1}^{n} P(w_i | t_i)$$

Estimate the Tag Priors and the Lexical Likelihoods from Corpus

Maximum-Likelihood Estimation

For bigrams:

$P(t_i | t_{i-1}) = c(t_{i-1}, t_i) / c(t_{i-1})$

$$P(w_i | t_i) = \frac{c(w_i, t_i)}{c(t_i)}$$

Statistical POS Tagging (whole sequence)

- Want to compute
  - $P(T)\,P(W|T) \approx P(t_1)\,P(t_2|t_1) \ldots P(t_n|t_{n-1})\;\; P(w_1|t_1)\,P(w_2|t_2) \ldots P(w_n|t_n)$
- Let
  - $c(t_i)$ = frequency of t_i in the corpus
  - $c(w_i,t_i)$ = frequency of w_i/t_i in the corpus
  - $c(t_{i-1},t_i)$ = frequency of $t_{i-1}\,t_i$ in the corpus
- Then we can use
  - $P(t_i|t_{i-1}) = c(t_{i-1},t_i)/c(t_{i-1})$,
  - $P(w_i|t_i) = c(w_i,t_i)/c(t_i)$

Question

What is the tagging of the following sentence?

Computers process programs accurately

with the following HMM tagger: (part of) lexicon:

computers	N	0.123
process	N	0.1
process	V	0.2
programs	N	0.11
programs	V	0.15
accurately	Adv	0.789

(part of) transitions:

$P(N\|V)=0.5$	$P(N\|Adv)=0.12$	$P(V\|Adv)=0.05$	$P(V\|N)=0.4$
$P(Adv\|N)=0.01$	$P(Adv\|V)=0.13$	$P(N\|N)=0.6$	$P(V\|V)=0.05$

Answer

Solutions

4 choices (it's a lattice):

```
computers process programs accurately
    N        N        N          Adv
             V        V
```

Differences are (skept the common factors):

```
P(N|N) P(process|N) P(N|N) P(programs|N) P(Adv|N)
P(N|N) P(process|N) P(V|N) P(programs|V) P(Adv|V)
P(V|N) P(process|V) P(N|V) P(programs|N) P(Adv|N)
P(V|N) P(process|V) P(V|V) P(programs|V) P(Adv|V)
```

i.e.:

```
    0.6    0.1    0.6    0.11    0.01
--> 0.6    0.1    0.4    0.15    0.13 <--MAX
    0.4    0.2    0.5    0.11    0.01
    0.4    0.2    0.05   0.15    0.13
```

Tagging obtained (not corresponding to the one expected by an average English reader ;-)):

```
computers process programs accurately
   N        N        V        Adv
```

What is the tagging of the following sentence?

Computers process programs accurately

with the following HMM tagger: (part of) lexicon:

computers	N	0.123
process	N	0.1
process	V	0.2
programs	N	0.11
programs	V	0.15
accurately	Adv	0.789

(part of) transitions:

P(N|V)=0.5	P(N|Adv)=0.12	P(V|Adv)=0.05	P(V|N)=0.4
P(Adv|N)=0.01	P(Adv|V)=0.13	P(N|N)=0.6	P(V|V)=0.05