5 Part-of-Speech tagging

Exercise V.1

What is the tagging of the following sentence

computers process programs accurately

with the following HMM tagger:

(part of) lexicon:

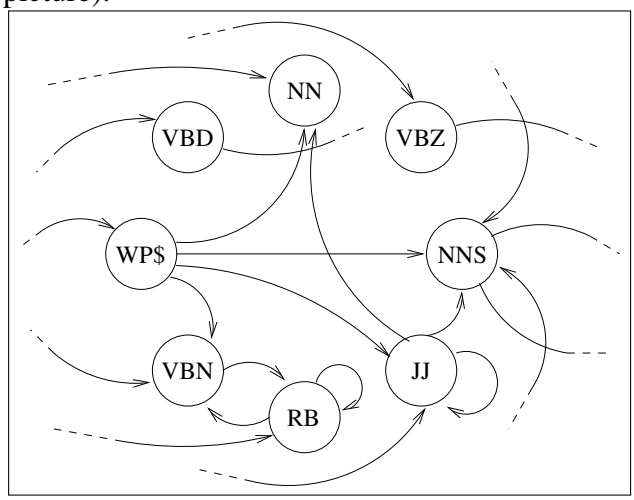
```
computers   N       0.123
process     N       0.1
process     V       0.2
programs    N       0.11
programs    V       0.15
accurately  Adv     0.789
```

(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
```

Exercise V.2

We aim at tagging English texts with "Part-of-Speech" (PoS) tags. For this, we consider using the following model (partial picture):



continues on back ☞

Explanation of (some) tags:

Tag	English expl.	Expl. française	Example(s)
JJ	Adjective	adjectif	*yellow*
NN	Noun, Singular	nom commun singulier	*cat*
NNS	Noun, Plural	nom commun pluriel	*cats*
PRP$	Possessive Pronoun	pronom possessif	*my*, *one's*
RB	Adverb	adverbe	*never*, *quickly*
VBD	Verb, Past Tense	verbe au passé	*ate*
VBN	Verb, Past Participle	participe passé	*eaten*
VBZ	Verb, Present 3P Sing	verbe au présent, 3e pers. sing.	*eats*
WP$	Possessive wh-	pronom relatif (poss.)	*whose*

① What kind of model (of PoS tagger) is it? What assumption(s) does it rely on?

② What are its parameters? Give examples and the appropriate name for each.

We use the following (part of) lexicon:

adult	JJ	has	VBZ
adult	NN	just	RB
daughter	NN	my	PRP$
developed	VBD	programs	NNS
developed	VBN	programs	VBZ
first	JJ	tooth	NN
first	RB	whose	WP$

and consider the following sentence:

my daughter whose first adult tooth has just developed programs

③ With this lexicon, how many different PoS taggings does this sentence have? Justify your answer.

④ What (formal) parameters make the difference in the choice of these different PoS taggings (for the above model)?

Give the explicit mathematical formulas of these parts that are different.

⑤ Assume that the following tagging is produced:

my/PRP$ daughter/NN whose/WP$ first/JJ adult/JJ tooth/NN has/VBZ just/RB developed/VBN programs/NNS

How is it possible? Give an explanation using the former formulas.

Exercise V.3

① What is the problem addressed by a Part-of-Speech (PoS) tagger?
Why isn't it trivial? What are the two main difficulties?

② Assume that you have to quickly search for the existence of given {word , part-of-speech} pairs within the set of all the English words associated with their part(s)-of-speech. Which data structure(s) would you use if memory is an issue?

③ Assume that the texts to be tagged contain unknown words, which are either capitalized words, or spelling errors, or simply general common words not seen during the learning. Almost all capitalized words correspond to proper nouns, and most of the spelling-errors correspond to words already in the lexicon (only a few of the spelling errors correspond to words not seen during the learning).

How would you handle such a situation in a concrete NLP application (that uses a PoS tagger)? Explicit your solution(s).

④ Assume that the texts to be tagged contain 1.5% of unknown words and that the performance of the tagger to be used is 98% on known words.

What will be its typical overall performance in the following two situations:

(a) all unknown words are systematically wrongly tagged?

(b) using the solution you proposed in ③ is used in a situation where 80% of the unknown words are capitalized among which 98% are proper nouns, 15% are general common words not seen during learning, and 5% are spelling-errors, among which 1% corresponds to correct words which were not in the learning set?

Provide both a calculation (a complete formula but not necessarily the final numerical result) and an explanation.

Exercise V.4

① Consider an HMM Part-of-Speech tagger, the tagset of which contains, among others:

$$\texttt{DET}, \texttt{N}, \texttt{V}, \texttt{ADV} \text{ and } \texttt{ADJ},$$

and some of the parameters of which are:

$$P_1(\text{a}|\texttt{DET}) = 0.1, \quad P_1(\text{accurately}|\texttt{ADV}) = 0.1, \quad P_1(\text{computer}|\texttt{N}) = 0.1,$$
$$P_1(\text{process}|\texttt{N}) = 0.095, \quad P_1(\text{process}|\texttt{V}) = 0.005,$$
$$P_1(\text{programs}|\texttt{N}) = 0.080, \quad P_1(\text{programs}|\texttt{V}) = 0.020,$$

continues on back ☞

$P_2(\texttt{Y}|\texttt{X})$: (for instance $P_2(\texttt{N}|\texttt{DET}) = 0.55$)

		Y →				
		DET	N	V	ADJ	ADV
X ↓	DET	0	0.55	0	0.02	0.03
	N	0.01	0.10	0.08	0.01	0.02
	V	0.16	0.11	0.06	0.08	0.08
	ADJ	0.01	0.65	0	0.05	0
	ADV	0.08	0.02	0.09	0.04	0.04

and:
$P_3(\texttt{DET}) = 0.20$, $P_3(\texttt{N}) = 0.06$, $P_3(\texttt{V}) = 0.08$, $P_3(\texttt{ADV}) = 0.07$, $P_3(\texttt{ADJ}) = 0.02$.

(a) How are the propabilities P_1, P_2 and P_3 usually called?

P_1:

P_2:

P_3:

(b) What are all the possible taggings of the sentence

`a computer process programs accurately`

(c) What would be the output of the HMM PoS tagger on the above sentence? Fully justify your answer.