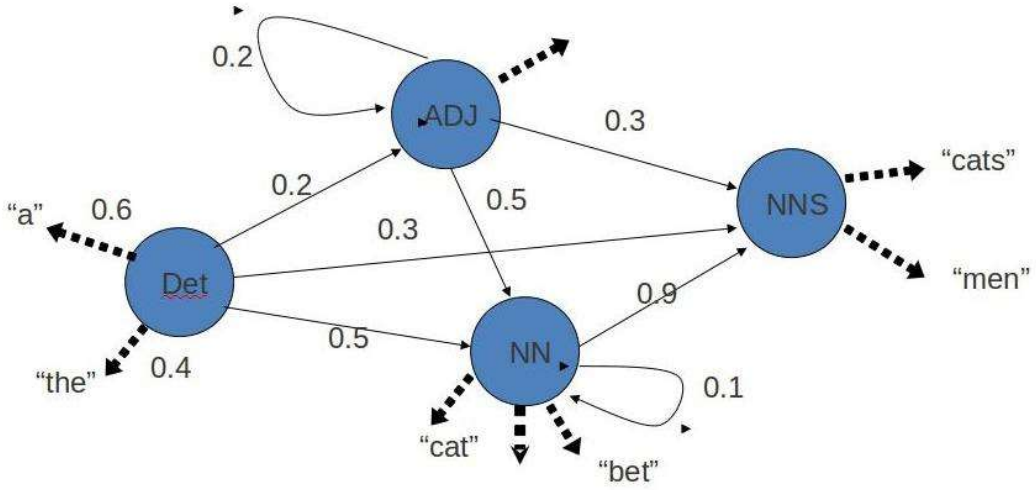
POS Tagging-HMM

Aim:

POS tagging or part-of-speech tagging is the procedure of assigning a grammatical category like noun, verb, adjective etc. to a word. In this process both the lexical information and the context play an important role as the same lexical form can behave differently in a different context.

For example the word "Park" can have two different lexical categories based on the context.

- The boy is playing in the park. ('Park' is Noun)
- Park the car. ('Park' is Verb)



Assigning part of speech to words by hand is a common exercise one can find in an elementary grammar class. But here we wish to build an automated tool which can assign the appropriate part-of-speech tag to the words of a given sentence. One can think of creating hand crafted rules by observing patterns in the language, but this would limit the system's performance to the quality and number of patterns identified by the rule crafter. Thus, this approach is not practically adopted for building POS Tagger. Instead, a large corpus annotated with correct POS tags for each word is given to the computer and algorithms then learn the patterns automatically from the data and store them in form of a trained model. Later this model can be used to POS tag new sentences.

In this experiment we will explore how such a model can be learned from the data.

Objective:

The objective of the experiment is to calculate emission and transition matrix which will be helpful for tagging Parts of Speech using Hidden Markov Model.

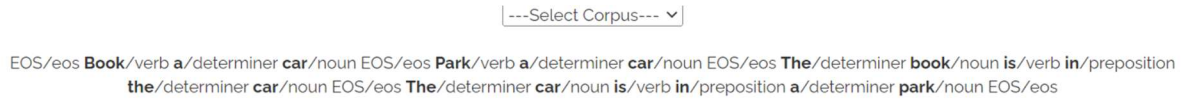
Procedure:

STEP1: Select the corpus.

STEP2: For the given corpus fill the emission and transition matrix. Answers are rounded to 2 decimal digits.

STEP3: Press `Check` to check your answer.
Wrong answers are indicated by the red cell.

Output



Corpus A

EOS/eos **Book**/verb **a**/determiner **car**/noun EOS/eos **Park**/verb **the**/determiner **car**/noun EOS/eos **The**/determiner **book**/noun **is**/verb **in**/preposition **the**/determiner **car**/noun EOS/eos **The**/determiner **car**/noun **is**/verb **in**/preposition **a**/determiner **park**/noun EOS/eos

Emission Matrix

	book	park	car	is	in	a	the
determiner	0	0	0	0	0	0	0
noun	0	0	0	0	0	0	0
verb	0	0	0	0	0	0	0
preposition	0	0	0	0	0	0	0

Transition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0	0	0	0
determiner	0	0	0	0	0
noun	0	0	0	0	0
verb	0	0	0	0	0
preposition	0	0	0	0	0

Transition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0	0	0	0
determiner	0	0	0	0	0
noun	0	0	0	0	0
verb	0	0	0	0	0
preposition	0	0	0	0	0

Check

Wrong Emission and Transition Matrix!!!

Get Answers

Hide Answers

mission Matrix

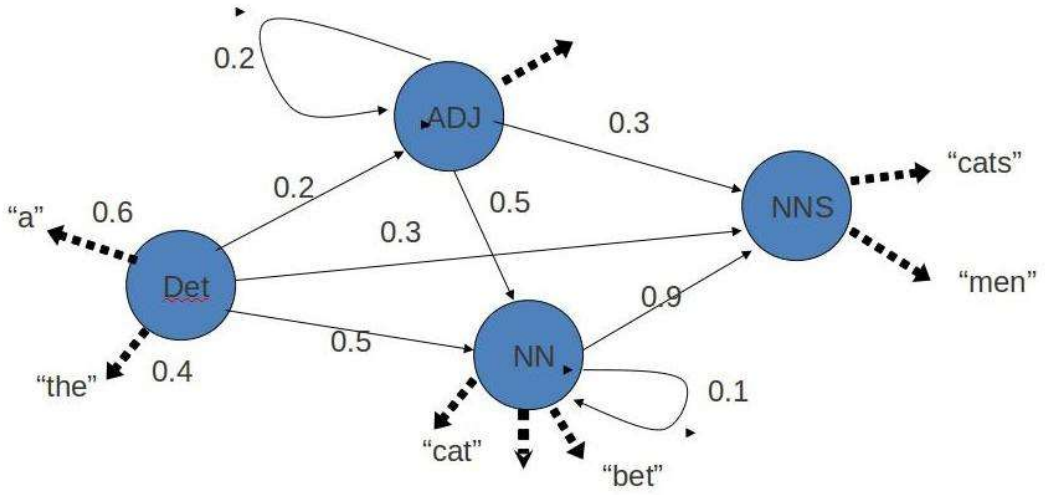
	book	park	car	is	in	a	the
determiner	0	0	0	0	0	1	1
noun	0.5	0.5	1	0	0	0	0
verb	0.5	0.5	0	1	0	0	0
preposition	0	0	0	0	1	0	0

ransition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0.33	0	0.5	0
determiner	0	0	1	0	0
noun	1	0	0	0.5	0
verb	0	0.33	0	0	1
preposition	0	0.33	0	0	0

Theory:

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.In a regular Markov model (Markov Model (Ref: http://en.wikipedia.org/wiki/Markov_model)), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible.



Hidden Markov Model has two important components-

1)Transition Probabilities: The one-step transition probability is the probability of transitioning from one state to another in a single step.

2)Emission Probabilties: : The output probabilities for an observation from state. Emission probabilities B = { $b_{i,k} = b_i(o_k) = P(o_k \mid q_i)$ }, where o_k is an Observation. Informally, B is the probability that the output is o_k given that the current state is q_i

For POS tagging, it is assumed that POS are generated as random process, and each process randomly generates a word. Hence, transition matrix denotes the transition probability from one POS to another and emission matrix denotes the probability that a given word can have a particular POS. Word acts as the observations. Some of the basic assumptions are:

```
1. First-order (bigram) Markov assumptions:
            a. Limited Horizon: Tag depends only on previous tag
               P(t<sub>i+1</sub>       =       t<sub>k</sub>       |
t<sub>1</sub>=t<sub>j1</sub>,.....,t<sub>i</sub>=t<sub>ji</sub>)    =
P(t<sub>i+1</sub> = t<sub>k</sub> | t<sub>i</sub> = t<sub>j</sub>)
```

```
            b. Time invariance: No change over time
               P(t<sub>i+1</sub> = t<sub>k</sub> | t<sub>i</sub> =
t<sub>j</sub>) = P(t<sub>2</sub> = t<sub>k</sub> | t<sub>1</sub> =
t<sub>j</sub>) = P(t<sub>j</sub> -> t<sub>k</sub>)
          2. Output probabilities:
            - Probability of getting word wk for tag tj: P(wk | tj)
is independent of other tags or words!
```

Calculating the Probabilities

Consider the given toy corpus

```
EOS/eos
                                        They/pronoun
                                        cut/verb
                                        the/determiner
                                        paper/noun
                                        EOS/eos He/pronoun
                                        asked/verb
                                        for/preposition
                                        his/pronoun
                                        cut/noun.
                                        EOS/eos
                                        Put/verb
                                        the/determiner

                                        paper/noun
                                        in/preposition
                                        the/determiner
                                        cut/noun
                                        EOS/eos
```

Calculating Emission Probability Matrix

Count the no. of times a specific word occus with a specific POS tag in the corpus.
Here, say for **"cut"**

```
count(cut,verb)=1
          count(cut,noun)=2
          count(cut,determiner)=0
```

and so on zero for other tags too.

```
count(cut) = total count of cut = 3
```

Now, calculating the probability
Probability to be filled in the matrix cell at the intersection of cut and verb

```
P(cut/verb)=count(cut,verb)/count(cut)=1/3=0.33
```

Similarly,
Probability to be filled in the cell at he intersection of cut and determiner

```
P(cut/determiner)=count(cut,determiner)/count(cut)=0/3=0
```

Repeat the same for all the word-tag combination and fill the

Calculating Transition Probability Matrix

Count the no. of times a specific tag comes after other POS tags in the corpus.
Here, say for **"determiner"**

```
count(verb,determiner)=2
          count(preposition,determiner)=1
          count(determiner,determiner)=0
          count(eos,determiner)=0
          count(noun,determiner)=0
```

and so on zero for other tags too.

```
count(determiner) = total count of tag 'determiner' = 3
```

Now, calculating the probability Probability to be filled in the cell at he intersection of determiner(in the column) and verb(in the row)

```
P(determiner/verb)=count(verb,determiner)/count(determiner)=2/3=0.66
```

Similarly,
Probability to be filled in the cell at he intersection of determiner(in the column) and noun(in the row)

```
P(determiner/noun)=count(noun,determiner)/count(determiner)=0/3=0
```

Repeat the same for all the tags

Note: **EOS**/eos is a special marker which represents ***End Of Sentence***.

Assignment

1. How is Hidden Markov Model different from Markov Model?
2. What is the basic design for HMM for finding out POS?
3. What are the basic assumptions for the above model?
4. How does the corpus size effect the transition and emission matrix?