Vishweshkumar V Salampuria , 62 , E4

NLP - LAB 6

Date : 01 - 03- 24

AIM : Perform the experiment to calculate emission and transition matrix for tagging parts of speech using hidden markov model and viterbi decoding .

RESOURCES : Google Colab / Jupyter notebook.

OBJECTIVES : 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 and to find POS tags of words in a sentence using Viterbi decoding.

THEORY :

Emission Matrix:

- The emission matrix, also known as the observation matrix, represents the probabilities of emitting each observation (word) from each hidden state .
- In the context of part-of-speech tagging, the emission matrix indicates the likelihood of observing a specific word given a particular part-of-speech tag.
- Each row of the emission matrix corresponds to a hidden state ,and each column corresponds to an observation .

Transition Matrix:

- The transition matrix represents the probabilities of transitioning between different hidden states .
- In the context of part-of-speech tagging, the transition matrix indicates the likelihood of transitioning from one part-of-speech tag to another in a sequence of words.
- Each row of the transition matrix corresponds to a current hidden state , and each column corresponds to a next hidden state.

A hidden markov model is a statistical markov model in which the system being modeled is assumed to be a markov process with unobserved (hidden) states. It is a stochastic model and future states depend only on the current state, not on the events that occurred before it. It's called "hidden" because the states are not directly observable, but the sequence of observable events depends on these hidden states. HMMs are widely used in various fields, including natural language processing, speech recognition, bioinformatics, and finance.

Viterbi Decoding is based on dynamic programming , used to find the most likely sequence of hidden states in a hidden markov model given a sequence of observed events or symbols. This is particularly useful in tasks such as part-of-speech tagging, speech recognition, and bioinformatics.

PROCEDURE :

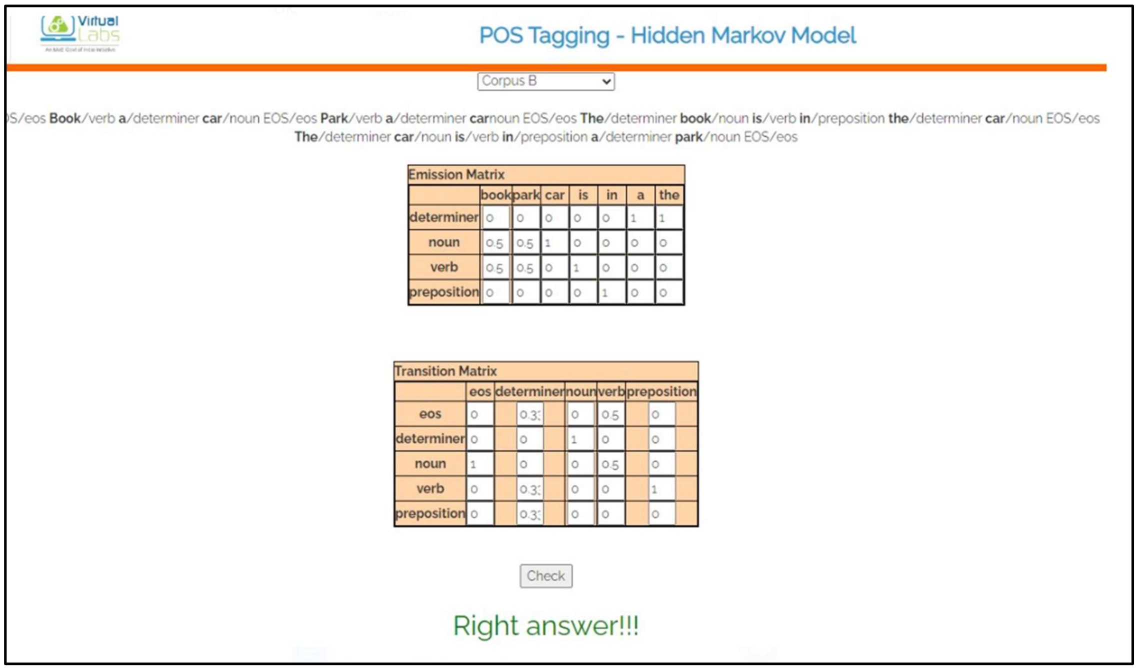
(Based on Hidden Markov Model)

- Select the corpus .
- For the selected corpus ,according to the sentences fill the emission and transition matrix.
- Press Check to check your answer.

(Based on Viterbi decoding)

- Select the corpus .
- Fill the column with the probability of possible POS tags given the word.
- Check the column , and if answers are right go to next step.
- Repeat the above steps until all words are covered.
- At last check the POS tag for each word obtained from backtracking.

SIMULATION :



)S/eos **Book**/verb **a**/determiner **car**/noun EOS/eos **Park**/verb **a**/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

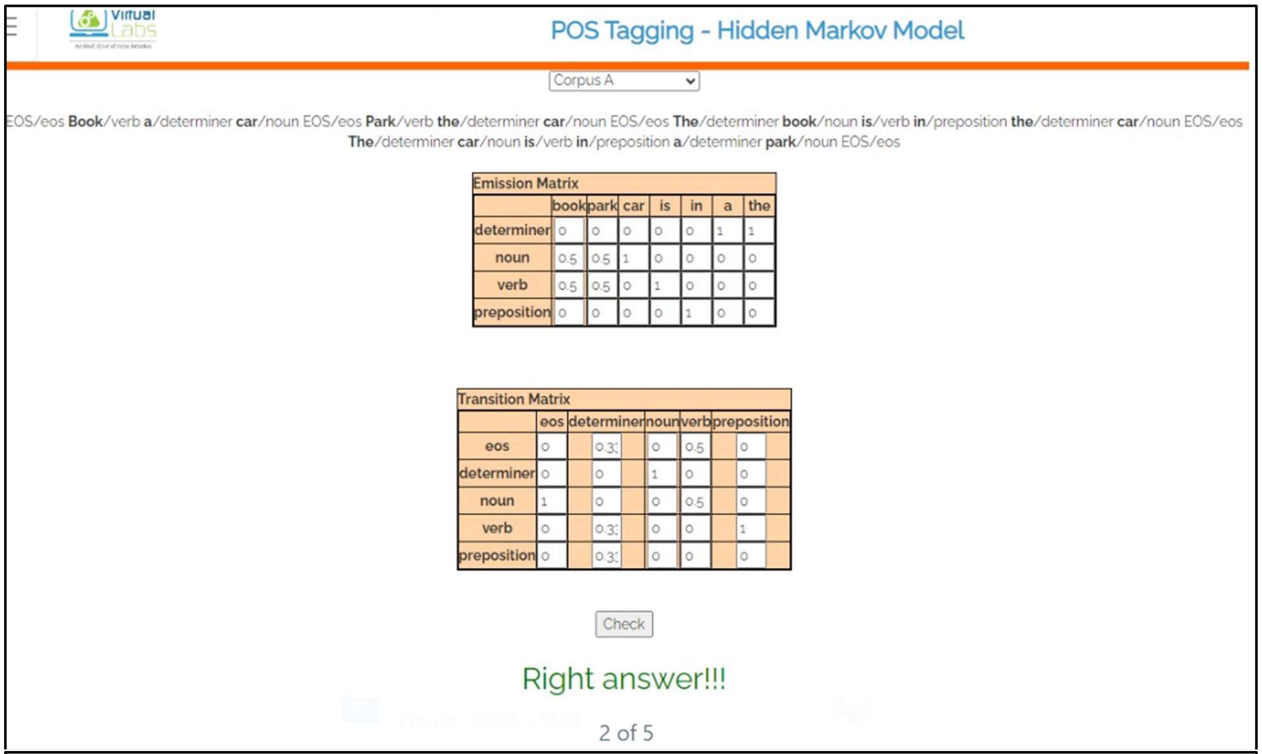
	book	park	car	is	in	a	the
determiner	0	0	0	0	0	1	1
noun	0.5	0.5	1	0	0	0	0
verb	0.5	0.5	0	1	0	0	0
preposition	0	0	0	0	1	0	0

Transition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0.3:	0	0.5	0
determiner	0	0	1	0	0
noun	1	0	0	0.5	0
verb	0	0.3:	0	0	1
preposition	0	0.3:	0	0	0

Right answer!!!

Vishweshkumar V Salampuria , 62 , E4



POS Tagging - Hidden Markov Model

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	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

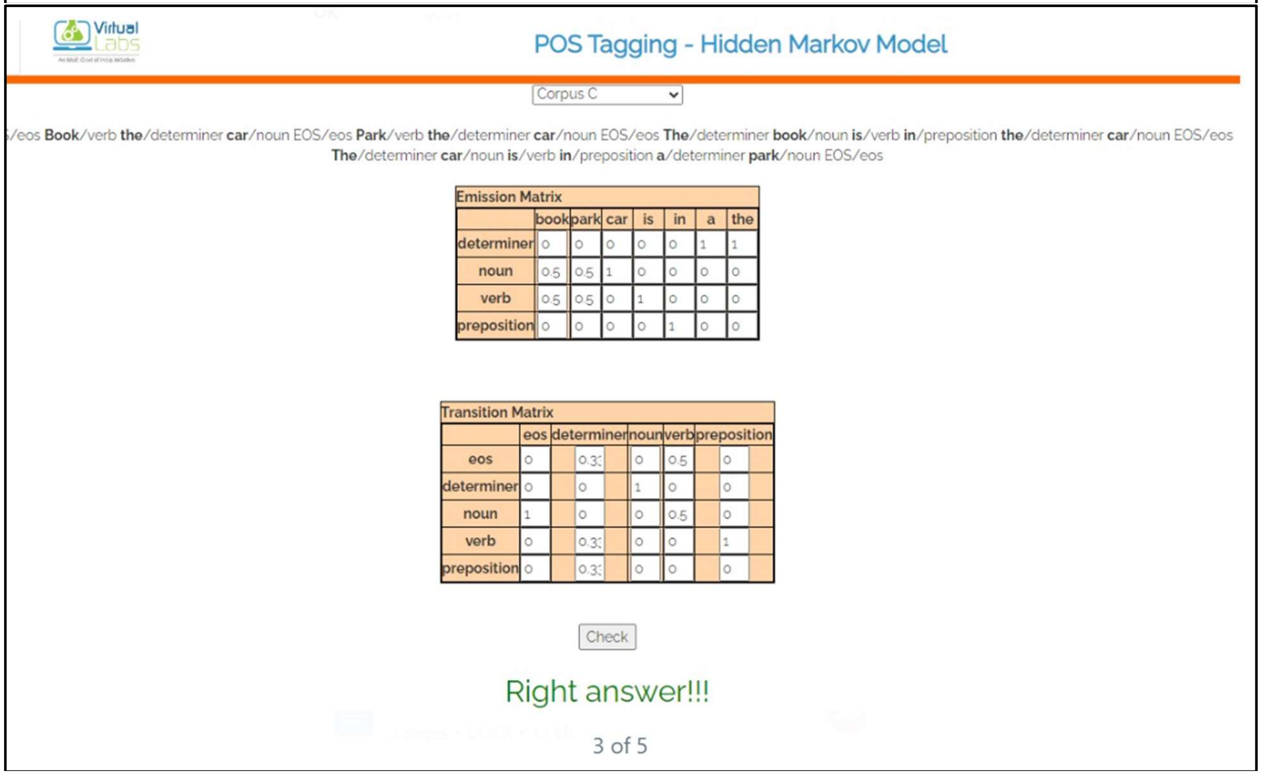
Transition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0.3	0	0.5	0
determiner	0	0	1	0	0
noun	1	0	0	0.5	0
verb	0	0.3	0	0	1
preposition	0	0.3	0	0	0

Check

Right answer!!!

2 of 5



POS Tagging - Hidden Markov Model

Corpus C

/eos **Book**/verb **the**/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	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

Transition Matrix

	eos	determiner	noun	verb	preposition
eos	0	0.3	0	0.5	0
determiner	0	0	1	0	0
noun	1	0	0	0.5	0
verb	0	0.3	0	0	1
preposition	0	0.3	0	0	0

Check

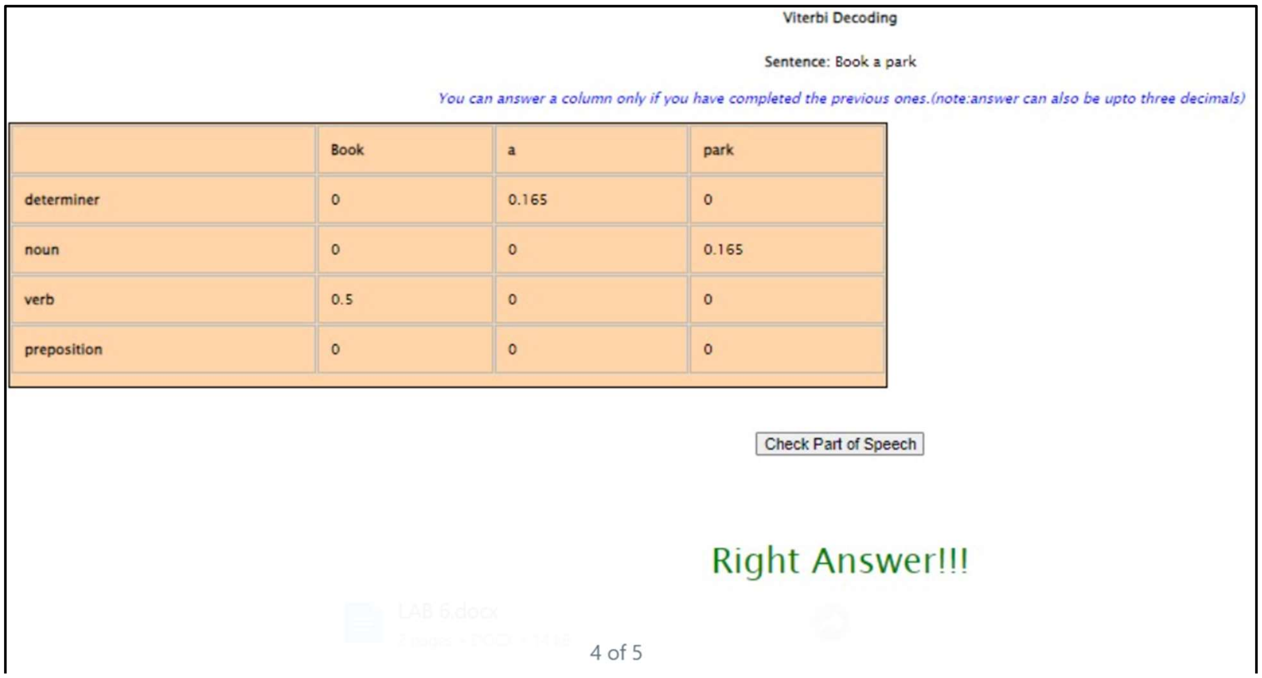
Right answer!!!

3 of 5

Count the occurrences of each word-tag pair in the training data to populate the emission matrix Count the transitions between POS tags to populate the transition matrix.

Calculating Emission Probabilities : Divide the count of each word-tag pair by the total count of words with the same POS tag to obtain emission probabilities.

Calculating Transition Probabilities : vDivide the count of each transition between POS tags by the total count of transitions from the same POS tag to obtain transition probabilities.



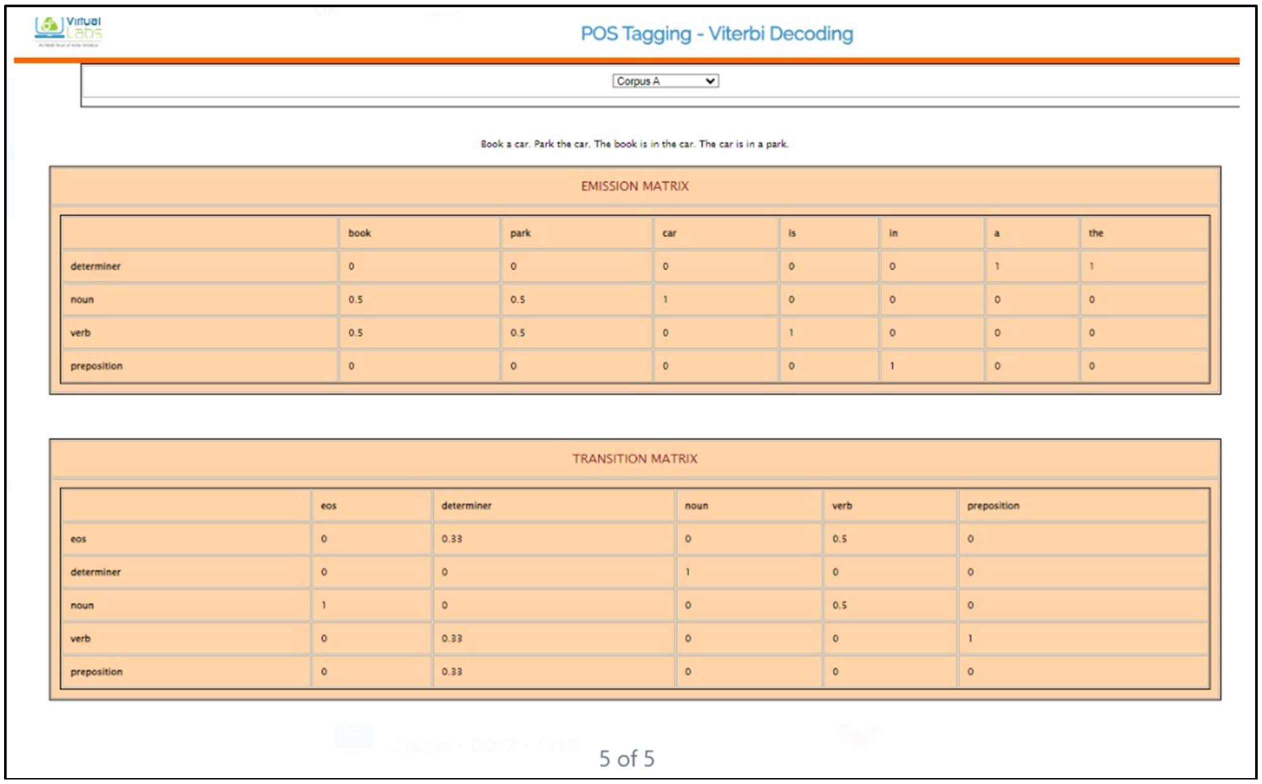
Viterbi Decoding

Sentence: Book a park

You can answer a column only if you have completed the previous ones.(note:answer can also be upto three decimals)

	Book	a	park
determiner	0	0.165	0
noun	0	0	0.165
verb	0.5	0	0
preposition	0	0	0

Right Answer!!!



POS Tagging - Viterbi Decoding

Book a car. Park the car. The book is in the car. The car is in a park.

EMISSION 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

TRANSITION 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

Count the occurrences of each word-tag pair in the training data to populate the emission matrix. Count the transitions between POS tags to populate the transition matrix.

Calculating Emission Probabilities : Divide the count of each word-tag pair by the total count of words with the same POS tag to obtain emission probabilities.

Calculating Transition Probabilities : Divide the count of each transition between POS tags by the total count of transitions from the same POS tag to obtain transition probabilities.

Using the Viterbi algorithm to find the most likely sequence of POS tags given the observed sequence of words . This involves recursively calculating the probabilities of each possible state sequence, considering both emission and transition probabilities.

ASSIGNMENTS :

POS Tagging - Hidden Markov Model

Q1) How is Hidden Markov Model different from Markov Model?

Ans:

1. Markov Models have directly observable states with transitions based solely on the current state.

2. Hidden Markov Models introduce hidden states not directly observable, influencing observable outputs.

3. In HMMs, transitions between states depend on both the current state and hidden variables.

4. HMMs are employed when the underlying system has unobservable states affecting observed outcomes.

Q2) What is the basic design for HMM for finding out POS?

Ans:

1. Define states representing different POS tags and transitions between these states.

2. Emission probabilities signify the likelihood of observing a word given its associated POS tag.

3. Initial initialization of transition and emission probabilities followed by iterative refinement.

4. Utilize Expectation-Maximization (EM) algorithm for parameter estimation.

Q3) What are the basic assumptions for the above model?

Ans:

1. Words in a sentence are generated sequentially, each associated with a hidden POS tag.
2. POS tags form a Markov chain, where the probability of a tag depends only on the preceding tag(s).
3. Emission probabilities depend solely on the associated POS tag and not on other words or tags.
4. The model assumes a finite set of POS tags and a finite vocabulary of words.

Q4) How does the corpus size effect the transition and emission matrix?

Ans:

1. Larger corpora lead to more accurate estimation of transition probabilities in the matrix.
2. Rare transitions become less sparse with more data, resulting in more reliable transition probabilities.
3. Extremely large corpora may introduce computational challenges for accurately estimating probabilities.
4. Larger corpora provide more instances of word-tag pairs, improving the reliability of emission probabilities.

POS Tagging - Viterbi Decoding

Q1) How is viterbi decoding different from forward algorithm?

Ans:

Viterbi Decoding : Viterbi decoding is used to find the most likely sequence of hidden states in a Hidden Markov Model (HMM) given a sequence of observations. It determines the most probable state sequence based on both the transition probabilities and the emission probabilities of the HMM.

Forward Algorithm : The forward algorithm, on the other hand, calculates the probability of observing a sequence of observations given an HMM. It computes the probability of a sequence of observations by summing over all possible state sequences.

Q2) How is viterbi decoding efficient than forward algorithm in terms of computation?

Ans:

- Computation Efficiency : Viterbi decoding is more computationally efficient compared to the forward algorithm. This efficiency arises because Viterbi decoding employs a dynamic programming approach that avoids redundant calculations by keeping track of the most likely path to each state at each step.

- Reduction in Complexity : The Viterbi algorithm reduces the computational complexity from exponential to polynomial time, making it more suitable for practical applications, especially with larger HMMs and longer observation sequences.

- Elimination of Redundancy : Viterbi decoding eliminates the need to consider all possible state sequences, which is necessary in the forward algorithm. Instead, it focuses only on the most probable path, significantly reducing computation time.

- Single Path Determination : Viterbi decoding directly determines the most likely state sequence, whereas the forward algorithm computes the likelihood of all possible state sequences, which may not be necessary for tasks like sequence decoding or recognition.

---XXX---