



Experiment No. 5

Title: Virtual lab on Viterbi Algorithm



Batch: 1

Roll No.: 16010420008

Experiment No.:5

Aim: To explore the virtual lab on Viterbi Decoding Algorithm.

Resources needed:

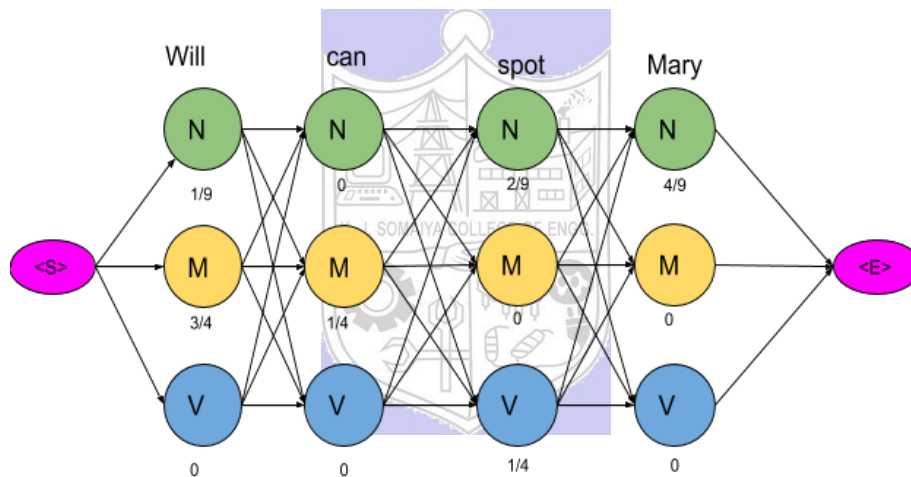
<https://nlp-iiith.vlabs.ac.in/exp/viterbi-decoding/>

Theory:

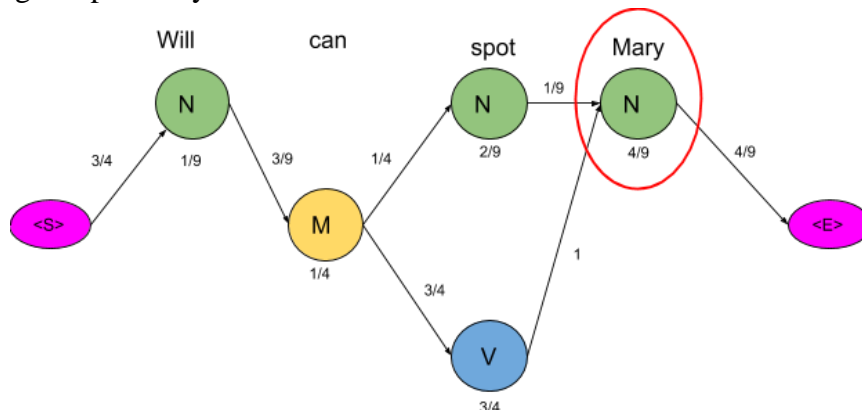
Viterbi Algorithm

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models (HMM). HMM is optimized using Viterbi Algorithm.

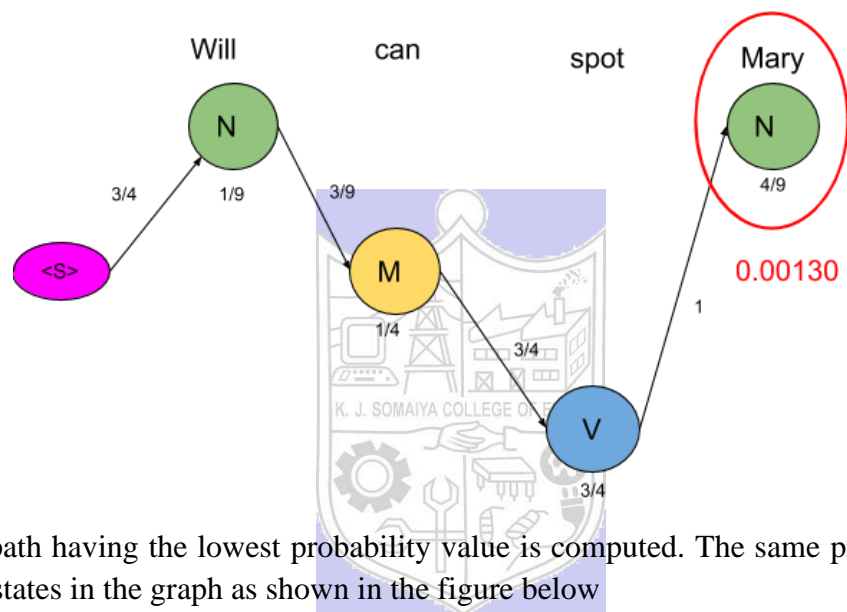
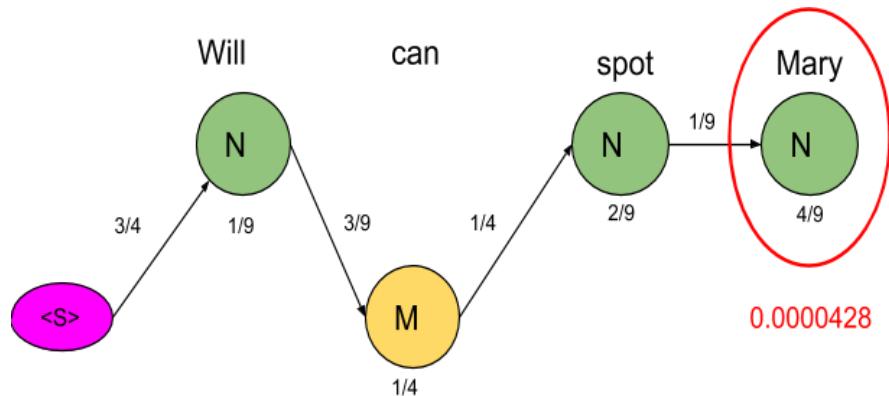
Consider the following example



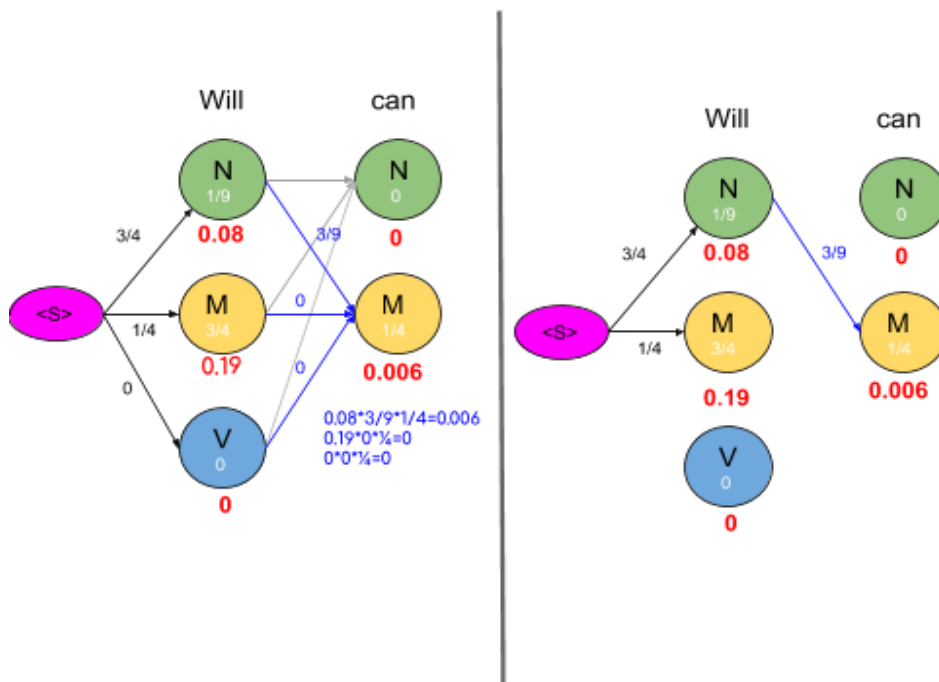
In this example, keeping into consideration just three POS tags, that is, Noun(N), Model(M) and Verb(V), 81 different combinations of tags can be formed. These number of combinations can be reduced and optimal path can be found using Viterbi Algorithm. Consider the given path with the transition and emission probability values marked at each vertex and edge respectively.



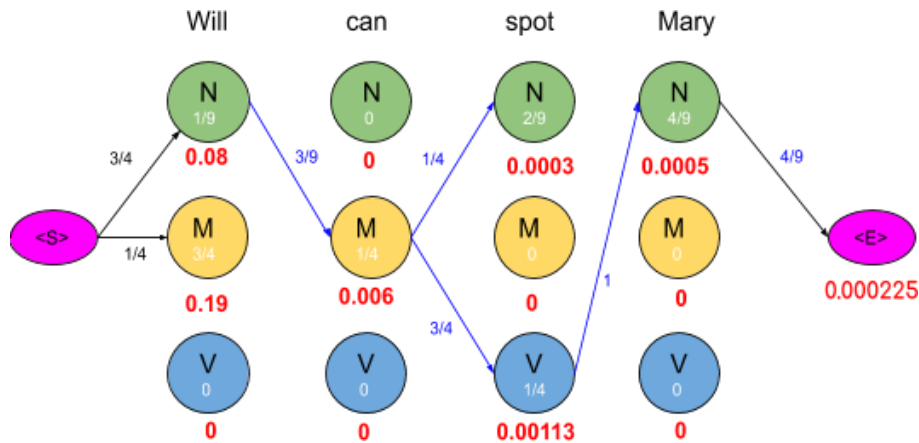
Consider the vertex encircled in the above example. There are two paths leading to this vertex as shown below along with the probabilities of the two mini-paths.



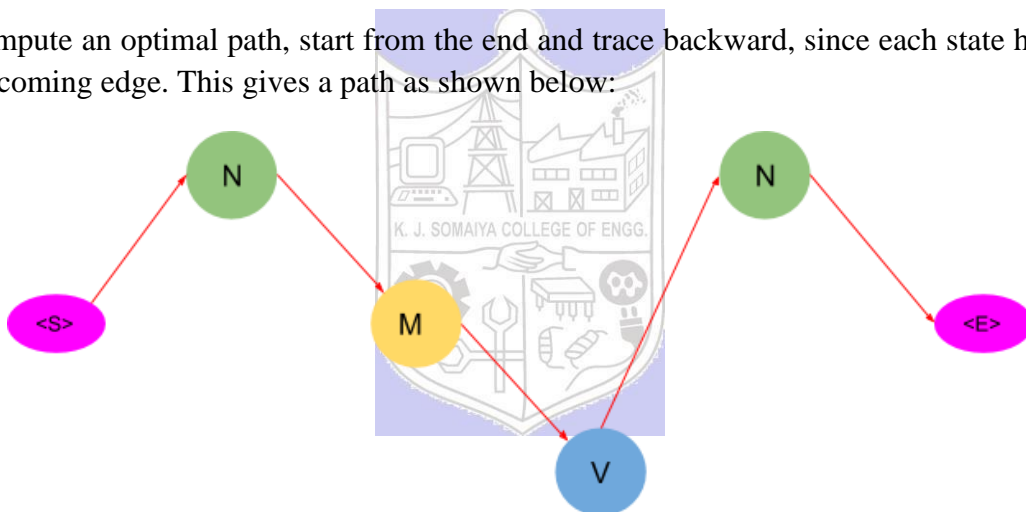
The mini path having the lowest probability value is computed. The same procedure is done for all the states in the graph as shown in the figure below



As seen in the figure above, the probabilities of all paths leading to a node are calculated and the edges or path which has lower probability cost are removed. Also, there are some nodes having the probability of zero and such nodes have no edges attached to them as all the paths are having zero probability. The graph obtained after computing probabilities of all paths leading to a node is shown below:



To compute an optimal path, start from the end and trace backward, since each state has only one incoming edge. This gives a path as shown below:



Thus after applying the Viterbi algorithm, the model tags the sentence as follows:

Will as a Noun
Can as a Model
Spot as a Verb
Mary as a Noun

Activity:

1. Perform the virtual lab experiment on Viterbi Decoding Algorithm

Results: (Snapshot of result of virtual lab experiment)

Virtual Labs

POS Tagging - Viterbi Decoding

Corpus A

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

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		
noun	0		
verb	0.5		
preposition	0		

Go to Next Step

Right Answer!!!

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	
noun	0	0	
verb	0.5	0	
preposition	0	0	

Go to Next Step

Right Answer!!!

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

Check Part of Speech

Right Answer!!!

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

Bookaparkdeterminer0 0.1650noun00 0.165verb 0.500preposition000POS tags
 for the words in the sentence are as following:
Book is verb
a is determiner
park is noun

Questions:

1. Explain bi-gram language model. Explain the use of bi-gram model in Viterbi algorithm.

A) Bi-gram Language Model:

A bi-gram language model, also known as a 2-gram language model, is a statistical language model used in natural language processing and computational linguistics. It is a simplification of language modeling where the probability of a word in a sequence is conditioned only on the preceding word. In other words, it assumes that each word in a text depends only on its immediately preceding word.

In a bi-gram language model, the probability of a word (W_i) is calculated based on the previous word (W_{i-1}), and this is expressed as $P(W_i | W_{i-1})$. This model is limited because it only considers one word of context, ignoring longer-range dependencies in language. Despite its simplicity, bi-gram models are often used due to their computational efficiency and usefulness in various natural language processing tasks.

Use of Bi-gram Model in Viterbi Algorithm:

The Viterbi algorithm is a dynamic programming algorithm commonly used in various applications, including part-of-speech tagging, speech recognition, and error correction.

In the context of part-of-speech tagging, the Viterbi algorithm is used to find the most likely sequence of part-of-speech tags for a given sequence of words.

The bi-gram model plays a crucial role in the Viterbi algorithm for part-of-speech tagging. Here's how it's used:

1) State Transition Probabilities:

In part-of-speech tagging, the states correspond to different possible part-of-speech tags for each word in a sentence. The bi-gram model provides state transition probabilities. Specifically, it calculates the probability of transitioning from one part-of-speech tag (state) to another based on the likelihood of word sequences in a training corpus. For example, it helps determine the likelihood of transitioning from a noun to a verb ($P(\text{verb} | \text{noun})$).

2) Initial State Probabilities:

The bi-gram model also helps estimate the initial state probabilities, which are the probabilities of starting a sentence with a particular part-of-speech tag. For example, it calculates the likelihood of a sentence starting with a noun or a verb.

3) Viterbi Algorithm Execution:

The Viterbi algorithm uses the state transition probabilities provided by the bi-gram model to calculate the most likely sequence of part-of-speech tags for a given sentence. It does so by efficiently considering all possible tag sequences and selecting the one with the highest overall probability, which is determined by multiplying the initial state probability, transition probabilities, and emission probabilities (the probability of observing a word given a particular tag) at each step.

Outcomes:

CO3: Establish concept of Structure and Semantics

Conclusion: (Conclusion to be based on the outcomes achieved)

We understood the concept of Viterbi Algorithm which helps in POS tagging of statements. We also performed simulation of Viterbi algorithm on corpus of data and found out the POS tag for it.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Allen.James, Natural Language Understanding, Benjamin Cumming, Second Edition, 1995
2. Jurafsky, Dan and Martin, James, Speech and Language Processing, Prentice Hall, 2008
3. Palash Goyal, Karan Jain, Sumit Pandey, Deep Learning for Natural Language

Processing: Creating Neural Networks with Python, Apress, 2018