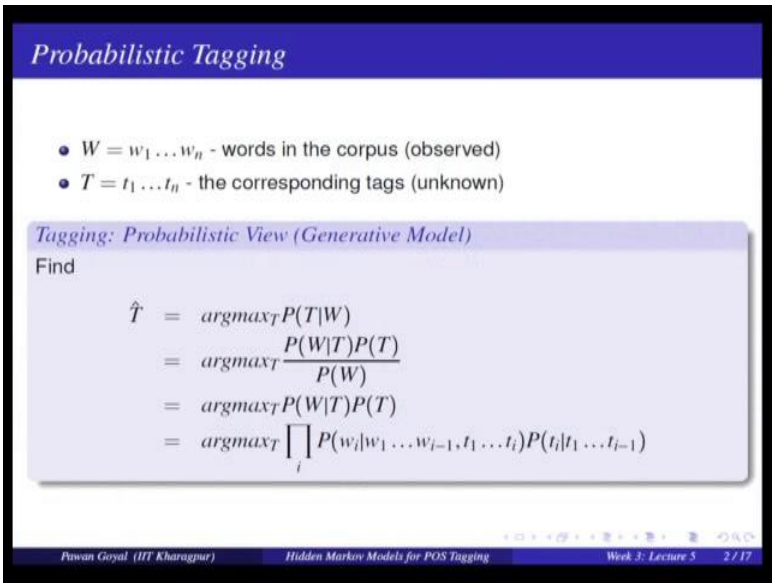
Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute Technology, Kharagpur

Lecture - 15
Hidden Markov Models for POS Tagging

Hello everyone, welcome back friend for this lecture 5 of week 3. So in the last lecture we started with the problem on part of speech tagging. We defined the problem image given a text that can be a sentence. So you need to find out what is the actual part of speech category for each of the individual word. So there may be many various ambiguities, but you need to resolve the ambiguities and find out the unique part of each tag for each of the words.

And we said that you can solve it using rule based methods or some probabilistic methods. We also discuss that the methods can be generative or discriminative. And then they differ in the philosophy of the model. So in general generative model the class comes first and it is assumed that the words are generated the data is generated from the class. And in the discriminative model you directly find out the probability of the class given the data. So in this lecture we will start with the generative model that is hidden Markov model and see how it can be used for solving the task of part of speech tag. So this is a probabilistic model.

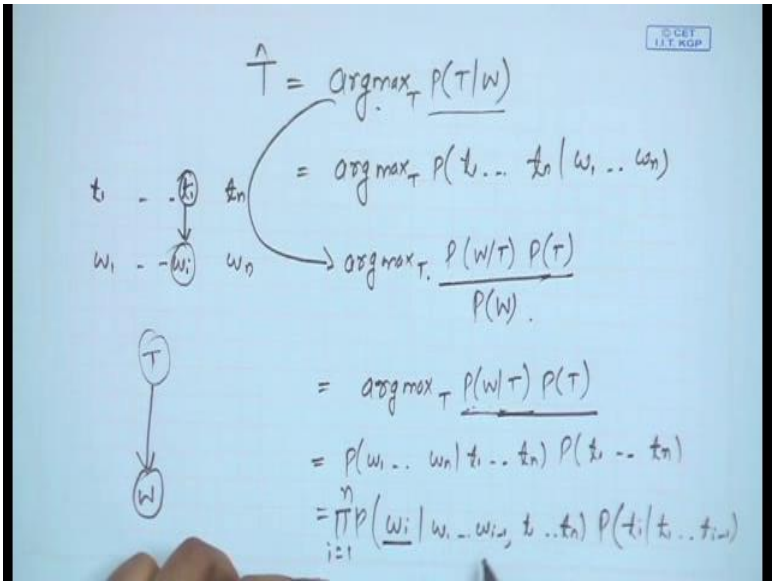
(Refer Slide Time: 01:40)



So starting with what is my problem. So I have some n words w 1 to w n in my corpus that I observed in, and I need to find out the participation text for each of these words. So suppose that you have to find out the sequence capital T, to that assigns t 1 to t n for all of these n words. Now each of these t i's that a participant text can belong to my actual say suppose I am using the university of when we tag set. So it can take any of those 45 value age. So there are many different values that these this sequence can take. I need to find out the actual and ambiguous tag sequence given this word sequence as my input.

So now how do I start solving this problem as per this probabilistic model? So the idea is that among all the possible sequences of part of speech tags that this word sequence can take I need to find the one that has the maximum probability. So I can write it like that.

(Refer Slide Time: 02:51)



So this is I have to find out that that gives you the maximum probability argmax overall possibility probability T given W. So I need to find out the purpose sequence that gives the highest probability.

Now, so as we have already said T is nothing, but t 1 to t n and W is nothing, but w 1 to w n. So I can write it as argmax t 1 to t n given w 1 to w n this is what I will be doing. So in generative model remember what is the idea, idea is that the class or the tags come first and then my words are generated from there. So I cannot find out the probability of tagging the word, but from the model I can find out the probability of word given the tag.

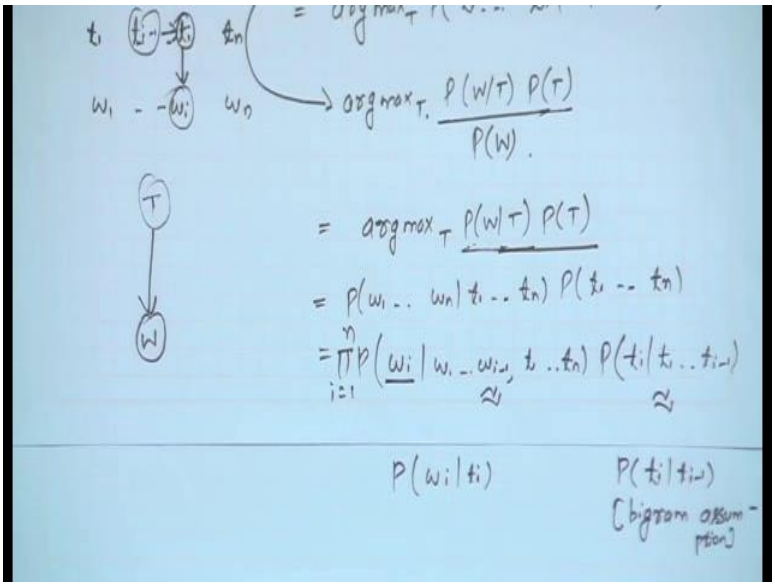
So I need to invert this, the direction of the probabilities here. So instead of finding t given w i need to find I need to use w going.

So what is the theorem popular theorem that I can use I can apply Bayes theorem. So instead of directly opening it here I can try same argmax T P W given T P T given divided by P W and P W it is common for all these sequences the probability of the sequence. So this is again it does not matter. So that will give me argmax over T P W given T and P T. So this because a using generating model. So first my class that is my T comes and then my sequence is generated W that is why I have to take it in this format probability W T P T.

Now I can try and open this. So this will be. So let me just take this particular thing and this will be the probability w 1 to w n given t 1 to t n probability t 1 to t n. So now, so I can again use chain rule to write that, so that we can see in the slide. So I can write it as using the chain rule. So this will be nothing, but probability. So multiplication over all i is equal to 1 to n, w i given w 1 to w i minus 1 t 1 to t n and probability t i given t 1 to t i minus 1 this is simply by using the chain rule I can write it like that.

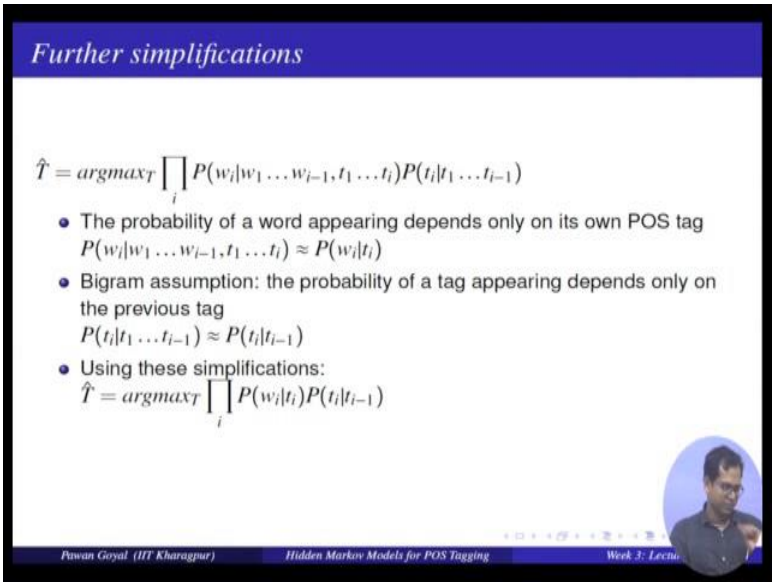
So now, it is very difficult to get the estimates for all these probabilities that we are thinking it. That means, I need to do certain simplifications over this formula. So what are the simplifications that we do? So one simplification that we do here is that, so in this formula we are saying that the probability of the word the current word. So you see we have a sequence of words w 1 to w n and correspondingly we have a sequence of x t 1 to t n whatever w i what I am saying the probability of this word depends on all the previous words and all the tags. So instead of that because the generative model I might say that this probably depends only on the current deck, this is simplification that I can make.

(Refer Slide Time: 07:18)



So this I can simplify as probability w i given t i. So this is one simplification, then here we are saying that the probability of t i the tag t i depends on all the previous tags. So again I might simplify it by using some Markov assumption that it depends on either the only the previous tag or previous to previous tag. So if I take only the bigram assumption. So I will set t i depends on t i minus 1. So in terms of this model I will say that t i only depends on t i minus 1 and here I will simplify it using probability t i k 1 t i minus 1 and this is my bigram assumption. So now if I make the simplification what is the model that we actually see?

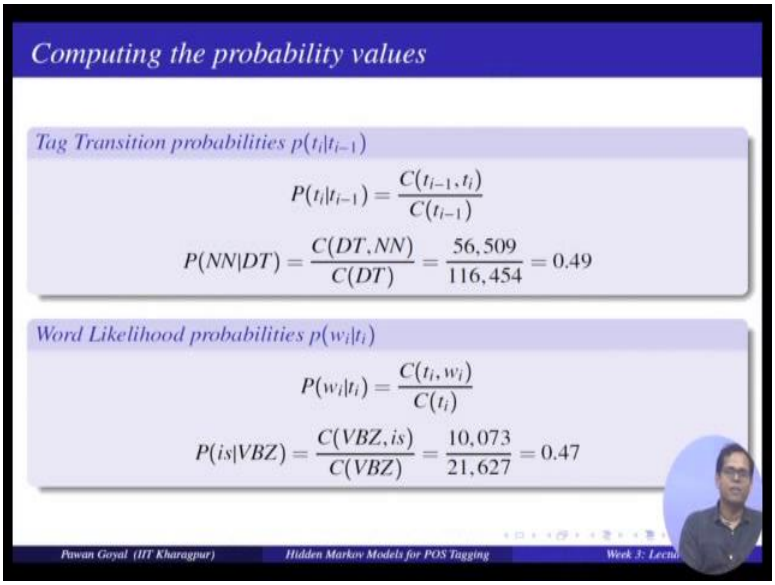
(Refer Slide Time: 08:20)



So this is the formula that we came up with. So I want a tag sequence that has that gives the highest probability for this particular form and we make some simplification that is the probability of a word appearing depends only on it is part of speech tag remember this is genetic model. So the word is generated from the part of speech tag. So first the part of speech tags is generated and then the tags are giving each an individual word. So we are making this assumption that the word is generated only from it is own part of speech tag then. So this gives me the first simplification second one I will say that the probability of attack depends only on it is forward previous tags.

So if I make the bigram assumption, it will depend only on the previous tag. So this will give me this function. So together the same the simplification will give me this formula. So I want to find it tag sequence that gives me the maximum probability for this particular formula. So now, once we have come to this formula. So what is this model actually?

(Refer Slide Time: 09:29)



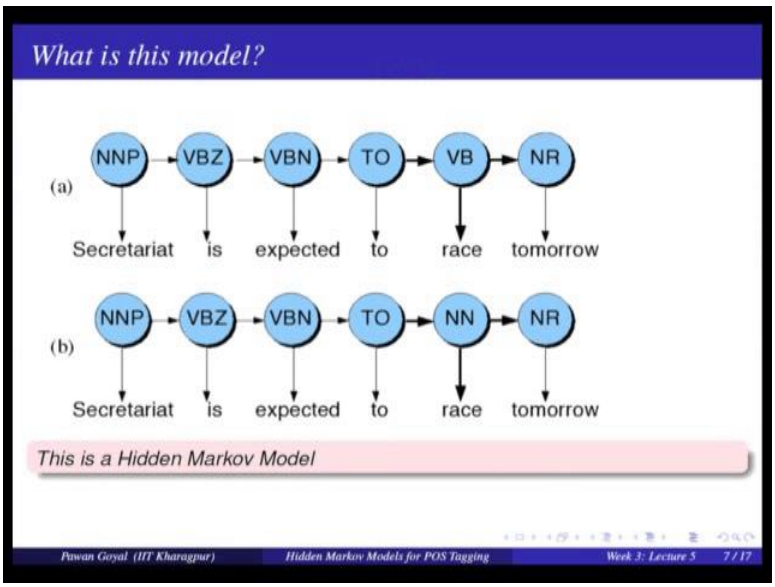
So first let us see that it can be easily compute these probabilities. Probability w i given t i and probability t i given t i minus 1, so how will you actually compute these probabilities? So one way is that you are given a corpus where you know all the words you also know what are their part of speech tags somebody has manually annotated each this, this data file, now given this data can you compete these probabilities. So for example, computing probability t i given t i minus 1. If you want to find out how many

times this tag t i comes after the tag t i minus 1. So you will compute it by using the maximum likelihood estimate that is the number of times t i minus 1 and t i come together in the corpus divided by the number of times the word t i minus 1 the tag t i minus 1 comes. So if you see here in the slide. So p t i given t i minus 1 can be found by this by using these counts, count of the 2 texts together divided by the count of the previous tag only.

So if I want to compute probability N, N given DT I will say number of times DT occurs followed by CN and N divided by number of times DT occurs. And this if I have the numbers I can compute this probability. What is the other probability of the compute? I have to compute probability of word given the tag. So the complete probability where I give the tag I will find out again how many times this word occurs with this tag divided by how many times that tag occurs, so this again from my corpus.

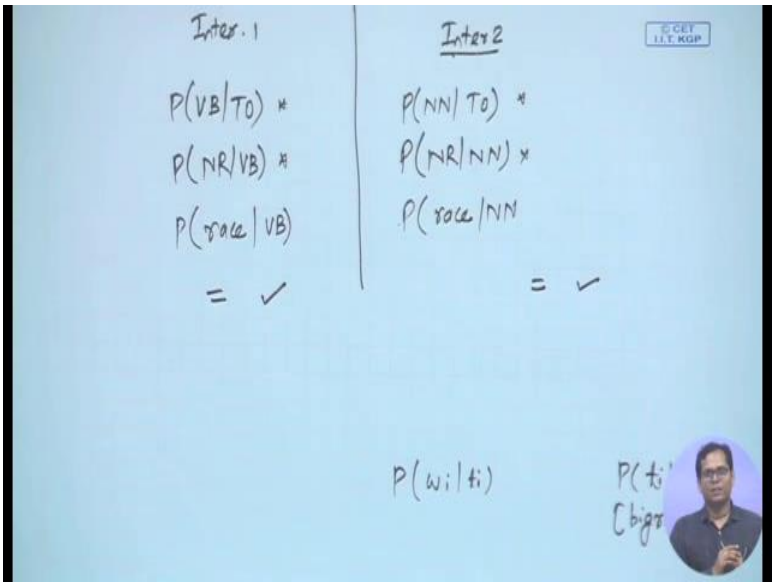
So if I have the numbers I want to compute probability is given V w Z I will find out how many times the word each occurs with the direct V w Z divided by the number of times the tag V w Z actually occurs in my corpus. So here if I have the numbers I can compute these probabilities. So all the probabilities that are required for this model can easily be computed if I have a data where I know the words and the part of speech tags for each and individual words.

(Refer Slide Time: 11:39)



So now let us see how we can use that for some disambiguation. It is not the complete model just to give you some idea. So once we have this how we can use that for some disambiguation. So I have a sentence a part of a sentence here. Secretariat is expected to race tomorrow. And the ambiguity here is in the word race. So whether the other word races in noun NN or VB you see everything as the same here. Now what are the probabilities as from my model that differ in the 2 interpretations.

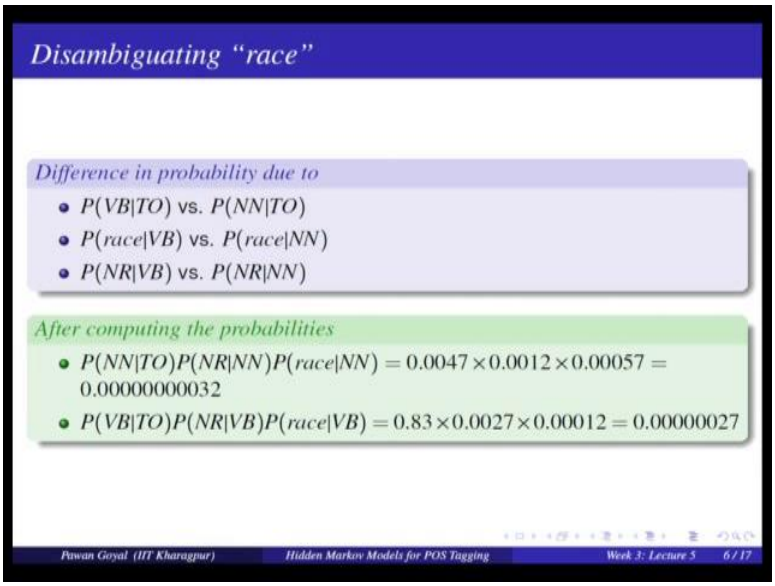
(Refer Slide Time: 12:15)



So if you see the first interpretation, what is the second interpretation? What are the probabilities that are defined, in the first one you find if the probability of the tag VB given 'To' versus in the second you find probability of NN given 'To' then in the first when you find probability of NR given VB and second you find probability of NR given NN. In the first when you find probability of race given VB and second you find probability of race given NN and because in your model you are multiplying all the probabilities; that means, you will multiply all the prob all these 3 probabilities in the integration 1 and all 3 in the interpretation 2.

So whichever multiplication gives you the highest value will decide what is the actual part of speech tag of race that should be used here. It should be VB or NN. So if you have the corpus you know all the probabilities you will multiply you will find the number and this will tell me whether I should prefer interpretation 1 or interpretation 2.

(Refer Slide Time: 13:33)

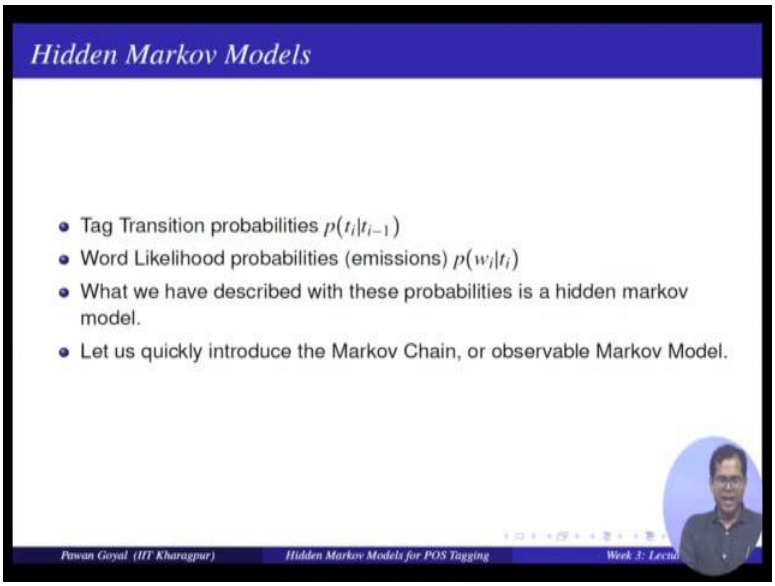


Suppose I take some numbers. So here so difference is because of these probabilities suppose from my corpus I find some numbers. So I will see here is that the possibility here is that the word race should be a verb not as a noun. And if you see the sentence to race tomorrow the race should be race, the word race should be used as a verb not as a noun, although noun is a more common category count part of speech tag for race then verb.

But because if the context it is more likely to have race edge verb in this case. So that is how we can disambiguate in one part in this simplistic case, but in general for the whole sentence even if you do not know any of the tags we can try to use this model to find out what is the actual sequence of tag that should be used.

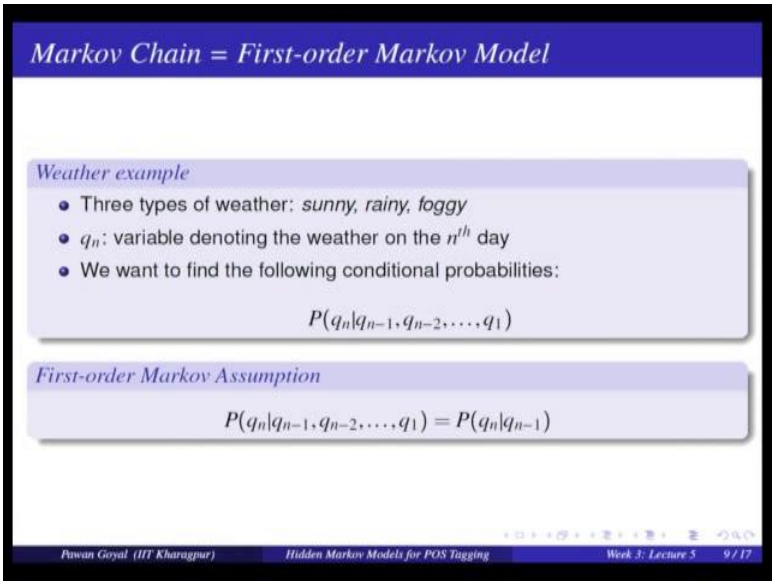
So now so coming back to this question what is this model. So what are you seeing here? So you are seen, so you have a sentence. So that is the words that is what you are observing and then there are certain tags that assigned to each of the individual words. So in this model you have a probability of going from one tag to another tag. And then from a tag you get a word. So can you find can you think of what is the model it corresponds to what is the actual model. So if you have come across hidden Markov model where you have the states and from one state you transit to another state and so on and in each state that is hidden you can emit the observation. So the word here is observation. So this is nothing, but the hidden Markov model.

(Refer Slide Time: 15:23)



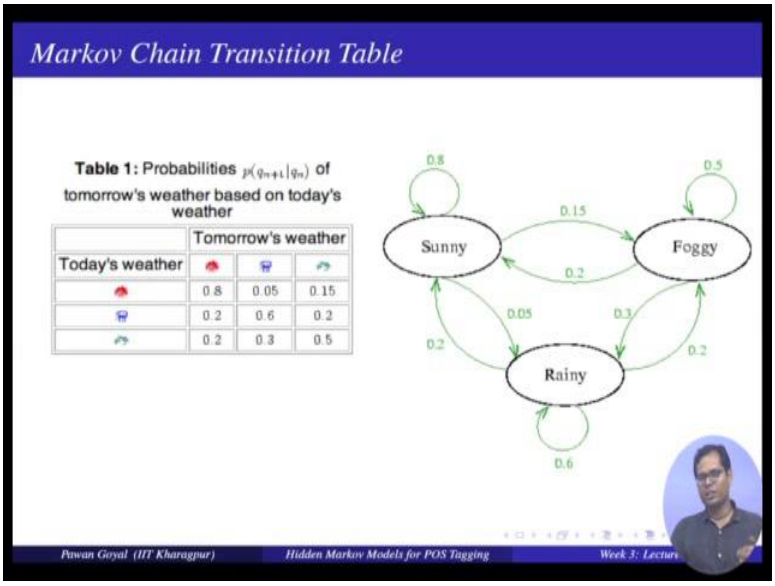
So what are hidden Markov models? So just to give the idea in brief, here you have the tag transition probabilities, t i given t i minus 1 you have the emission probabilities. So that is word observation probabilities probability w i given t i. So using this, whatever we are describing is a hidden Markov model. Now to tell you what is hidden Markov model? So let me just quickly tell you what is a Markov model and how a (Refer Time: 15:53) Markov model different from a Markov model.

(Refer Slide Time: 15:58)



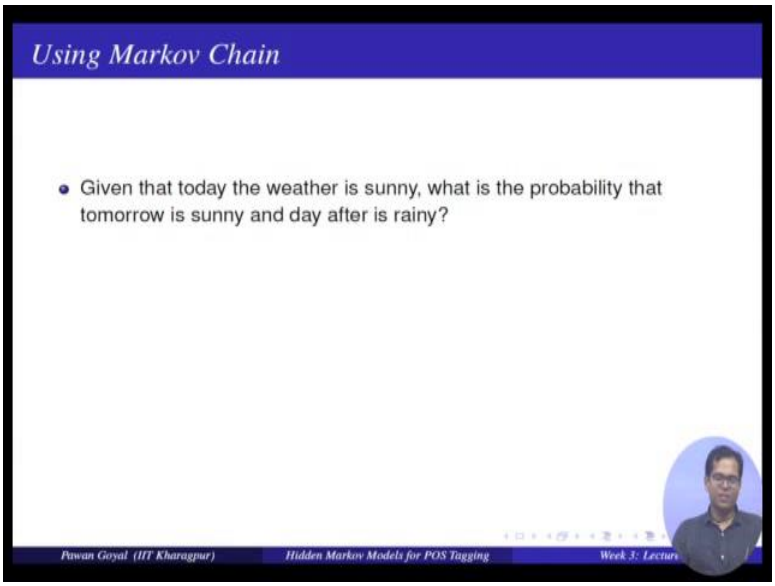
So what is a Markov model? So this Markov model is best explained using this simple example. So in Markov model what happens you again have a states, but the states are also your observations. Suppose you are studies the weather the weather on the current day and the weather can be sunny, rainy or foggy. These are the 3 different kinds of weathers that can happen on a given day. Now what you will get you will have these 3 states you will know, given that today is sunny what is the probability that tomorrow will be sunny or foggy or rainy. So this is state transition probabilities you will obtain. So suppose qn is a variable that denotes the variable on the nth day and using this model we can find out the probability of qn the weather on the nth day given all the previous days back and here if you use the first row mark Markov model. So we say this will depend only on the previous days' back qn will depend on your n qn minus 1.

(Refer Slide Time: 17:03)



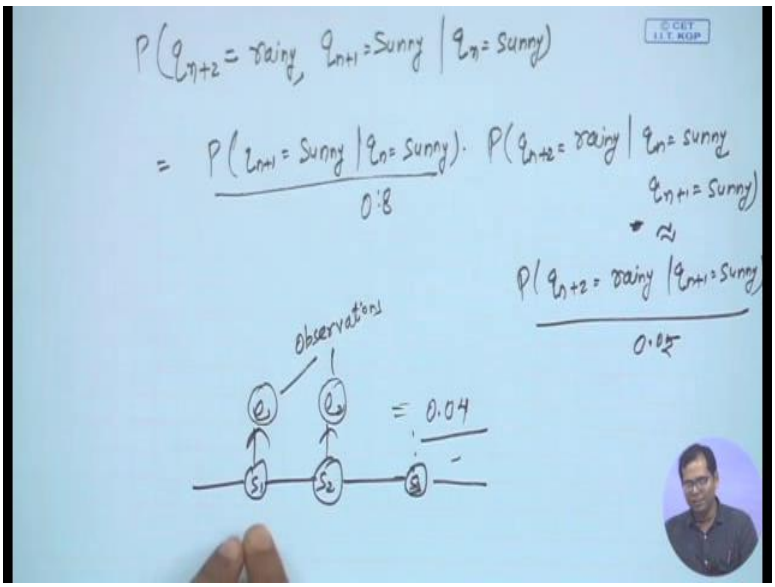
So let us take one simple example. So here you are seeing the state transition. So that says that if today's weather is sunny then tomorrow will be sunny with the probability of 0.8 and if today is sunny tomorrow will be foggy with a probability 0.15. So you can see that from the edges that go from one state to another state, also from the table that is shown in this. So now, once you are given these probabilities you can do certain you can do certain computations.

(Refer Slide Time: 17:35)



For example, suppose you have to find out given that today the weather is sunny what is the probability that tomorrow is sunny and day after is rainy.

(Refer Slide Time: 17:48)



So let us use the variable and I want to find out probability qn plus 2, days after is rainy qn plus 2 is to rainy and tomorrow is sunny qn plus 1 it is sunny given qn sunny.

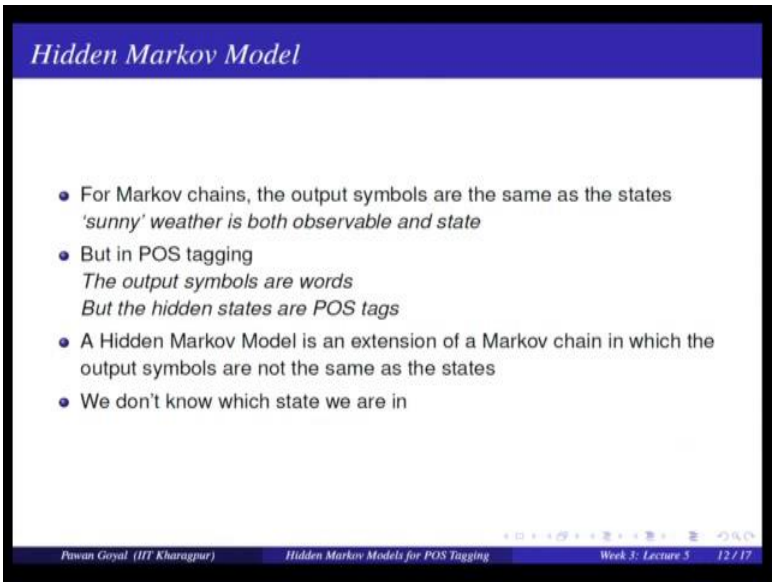
So how would you compute that? So if you simply are the channel, you will say that is nothing, but probability qn plus 1 is equal to sunny given qn is equal to sunny times probability qn plus 2 is equal to rainy given qn is equal to sunny qn plus 1 is equal to

sunny and now because you are using a first row Markov assumption. So this will be equivalent to. So this you will compute using probability qn plus 2 is equal to rainy given qn plus 1 is equal to sunny. So, now you have to compute this probability and this probability and that you can obtain from the state transition graph. So if you go to the previous slide you will find out probability sunny given sunny is 0.8 m probability rainy given sunny is 0.05. So once you multiply this you will get the answer is 0.04. So here so you will get the answer is 0.04. So that gives you the probability that tomorrow will be rainy and sorry; tomorrow will be sunny day after tomorrow will be rainy given that today is sunny. So that is how you use the Markov model.

Now, this is the Markov model. So what you see here? You have the states, so in this case the weather. So on a day is a solidus state and transitions are happening among the states, but what is your observation that is also state. So you also observing the weather that is shear state you are observing that now remember the example of participate text. What are we observing? We observing the words, then when I give you a text I you only observe the words, but what is hidden that is the text. The text is hidden. So that is where the hidden Markov models are different from Markov model.

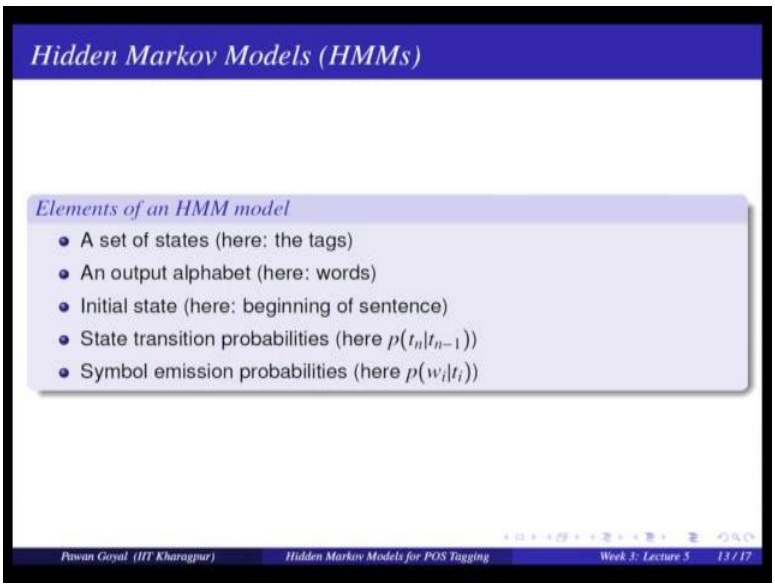
The states there are not observed, they are hidden variables and what is observed is different. So you have the words are being objective and from the state you can emit the words. So this is the idea. So the generative model would be. So you are starting from some state 1 you keep on transiting to other states S3 and so on and v is a state you also emit a word emission 1, emission 2 and so on and these emissions are nothing, but your observations, these are your observations. So you need to find out the underline sequence of a states given the sequence of observations.

(Refer Slide Time: 20:54)



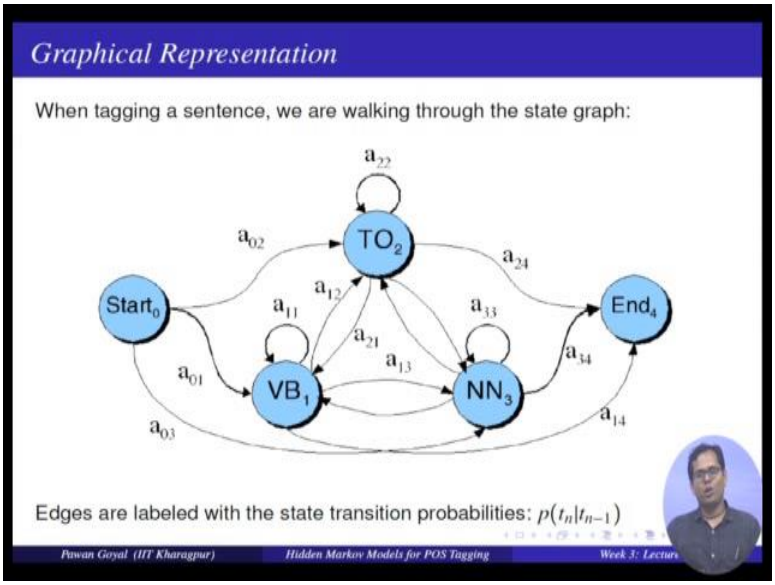
So now this is the difference we have seen for Markov mode chains the output symbols are the same as the states. So the word sunny is both a state and the observable. So what happens in part of speech tagging, words are the output symbols, but the hidden states are part of speech tag. So hidden Markov model is nothing, but an extension of Markov of chain, so in which what happens that the output symbols that you are having are not the same as the states. So states are different from the output. And we actually do not know what should we are in until we try to use our model.

(Refer Slide Time: 21:33)



So what are the elements of a hidden Markov model? So what do you need from hidden Markov model? So you need a set of states, yes. You need the probability of transiting from one state to another state. In the probability of emission given a state which words will be emitted with what probability. And you might also need what is the beginning of a state and so on. So if you try to correspond a hidden Markov model with part of speech is tagging. So we need set of states in our part of speech taking case. So the tags are the states. We need an output alphabet that is then what are emissions. So the words are the emissions with the initial state. So in our case that is the beginning of the sentence. We need the transition probabilities that is given a previous tag what will be the next tag. And we need the emission probabilities that is given this tag what will be the word.

(Refer Slide Time: 22:40)

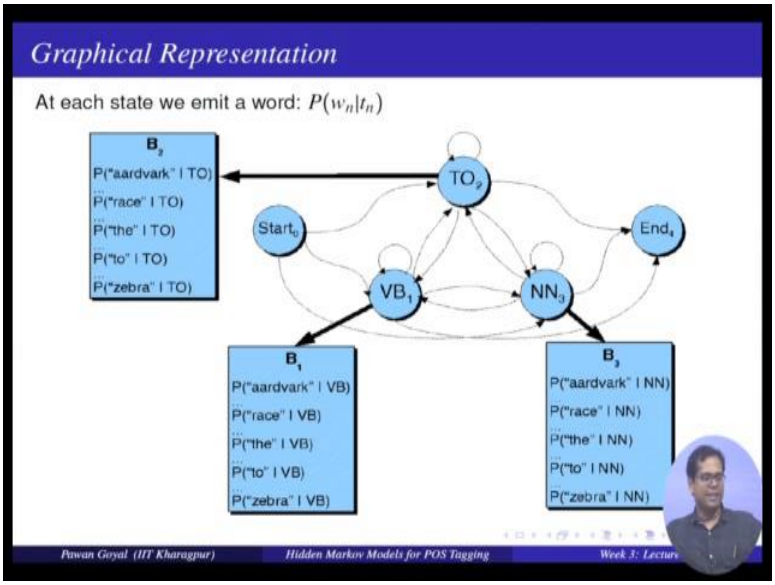


So once we have this we can also give a graphical representation to our hidden Markov model for part of speech tagging. So what is happening? See here we start state. So here by start steady shown. From the start state you have a probability of transiting to any of the other tags. So what does that mean? You can start the sentence with some particular tags. So this probably should depend on what is the tag that is more likely to start the sentence. And once you have this tag what is the next likely tag and so on. So this is your state transition graph. So this is what is shown in the slide. So when we are tagging the sentence you are actually walking on this a state graph from one state we are going to another state and so on and they are very transition probabilities that you can have over

this state graph. And this you can model by using probability of a state t n given t n minus 1.

Now, what else is missing here? With each now within the hidden Markov model with each a state you also have the word emission. So from history you also want to know what is the probability that you will output or emit a particular word.

(Refer Slide Time: 23:52)

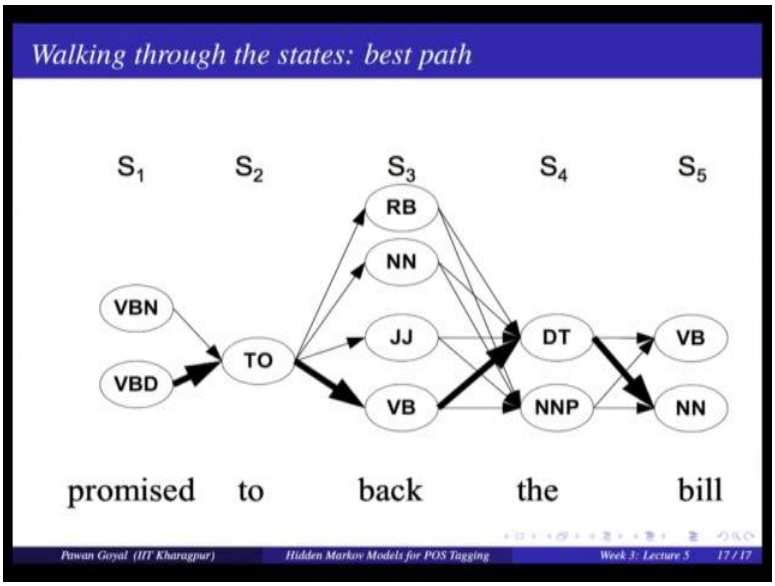


So this is what is additionally needed here. So in the graphical representation, now with these states like to I should also the probability of different words emitting from that state. So here all the words are in the vocabulary are written and given the particular state what is the probability of out putting or emitting that particular word in the vocabulary, so that you need to define for all the very possible states in your graph.

So now once we have both these what is my problem. So I can define, so if you remember I can define all these probabilities once you give me a corpus that contains a set of words and their corresponding part of speech tags. If you give me that I can find out all these probabilities. So I can define these graph once here once I have a corpus.

What is my problem? At runtime I am only given an observation that is, I am only given a sentence that is sequence of words. I have to find out what is the corresponding part of speech tag sequence that should be used for this word sequence.

(Refer Slide Time: 25:09)



So that is suppose I am given a sentence like this, is the part of the sentence promised to back the bill, there are 5 words here. So my problem is to find out what is the sequence of x that would be used for this sentence. So how would I approach this problem given the state transition graph that I already have? Now for each word one thing to simplify this I will do is that, for each for I will find out what are all the possible part of speech text a word can take.

So we will be talked about this said that some words can take multiple part of speech tags, but none of the words went beyond 7 part of speech tags. And mostly the words were having if they were ambiguous they were having 2 or 3 part of space tags. So for a word if I can identify what are the part of speech tags. So I can only use that to have my set of possibilities. So here for example, I will say the word promised can take only VBD or VBN past tense of participial. It can be only one of those. I have to disambiguate among the 2 - the word to can have only one part of speech tag. Back can have 4 tags, the can have 2 tags, and bill can have 2 tags. These are all the possibilities. Now each word can take any of these possibilities. So now, can you just quickly see how many possibilities are there? So if you see here I have 2 times 4, 8 times 2, 16 times 2, 32 possibilities of the part of speech tag sequences.

Now, I have to find out among these which is the most likely sequence of participation tags that is being used. So how will I solve this problem? One neither might be, I

enumerate all the possibilities and for each possibility I compute the probability separately. So I have 32 possibilities and I compute 32 different. So for all the possibility I compute the full probability. That is one possibility, but firstly, this is not very efficient, and think of some sentences which might have say 15-20 volts. This will just go exponentially with the number of words. So this is not a good solution.

So I need to have a solution where it does not grow exponentially with the size of the sentence. So what will be a good algorithm for doing that? So ideally I want to come up with the actual part of speech tag sequence like here it will be VBD TO VB DT and NN from all the possibilities. And how will I do that and that is what we will be discussing in the next lecture. So this is the Viterbi algorithm.

How do I apply in my HMM model this will be algorithm to come up with this part of speech tags sequence in efficient manner? Instead of doing something naive implementation that is exponential and that is actually not feasible for doing it for a large corpus, so that will be the focus in the next lecture.

Thank you.