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

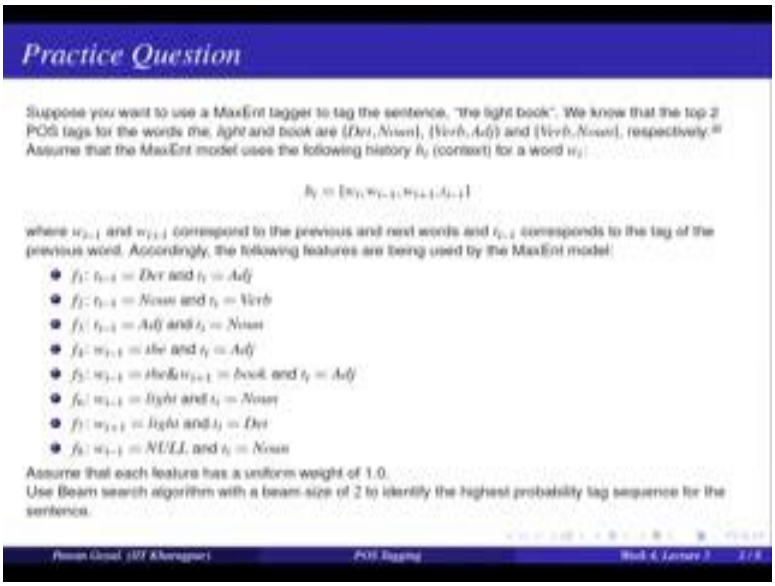
Lecture – 20
Conditional Random Fields

Welcome back for the final lecture of this week. So, we are talking about the problem of part of speech tagging and the methods that we are discussing our generic methods for any sequence labeling task. So, input is sequence like here sequence of the words; and output is again a sequence where for each word I going to predict what is a corresponding part of speech tags. So, we talked about hidden Markov models and also Maxent model. So, in general in Maxent model, we talked about how do you we use that simple classifier for a sequence labeling task.

So, we will call that the maximum entropy Markov model. And the formulation was very easy that is if predict the tag for each word, and then multiply the probabilities for the whole sequence. There was one problem though that because we need the tag of the previous word in certain features to assign the tag at this particular word, we will need to use beam search algorithm and we also discuss; what is the beam search algorithm.

So, what I will do in this lecture I will take a take an example there we will see how to use beam search algorithm and then I briefly discuss what are condition random fields and how are the different from maximum entropy models. So, condition random fields again are a very vast topic, we will not cover fully. We will only give you the hint that once you know Maxent or MEMA models what are condition random fields how are they different from those.

(Refer Slide Time: 01:56)



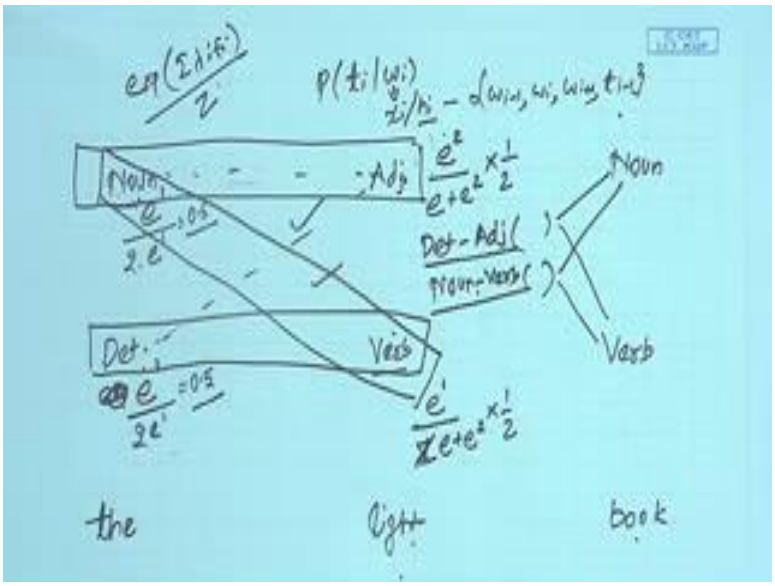
So, we are starting with a practice question. So, here so we are having the same sentence the light book; and you are given that for all the three words the, light and book, the top two excel determine and noun, verb, adjective and verb noun. Now, you want to use your MEMA model. So, for the given tag or given word w i, you use particular context what is a context here the previous word next word and the tag given to the previous word this is the context so that means, all your features we will be defined over this context. So, here is a example, we have showing some sample like you have given a different features. And now we have to use the beam search algorithm to find out what will be the appropriate tag sequence for the sentence.

So, what are the features giving? So, features are simple like the previous tag, the tag given to the previous word is determiner, and current tag is adjective; previous tag is noun, current tag is verb; previous tag is adjective, current tag is noun; previous word is the current tag is adjective and so on. Then you also have feature like the next word is light, current tag is determiner; previous word is null and current tag is noun. What it means is that this ith word will be starting a sentence that is why the previous word will be null.

So, now, you are given these features and for simplicity you are should given that each feature has a uniform weight of 1. Now, your task is to use beam search algorithm with the beam size of 2. What do you mean by a beam size of 2, at any given point, you will

keep only the top two highest probability tag sequence, and everything else you will forget. So, at any point you will know; what are the top two tag sequences till this point. And overall you have to find the highest probability tag sequence for the sentence that is the light book.

(Refer Slide Time: 03:56)



So, let us see how do we solve this. So, we are having three words in the sentence the, light and book. The word has two text two possible texts it can be either a determiner or a noun. The word light can be a verb or an adjective; and the word book can be a verb or a noun. So, now how do we start you to find out probability tag i given w i or instead of w i, let me write the context x i or h i give different notations for that. So, here context is w i minus 1 w i w i plus 1 and t is tag. And what is the formulation of this tag i given the current word current here, it will be exponent summation lambda i f i and feature we know is a function of input and the tag divide by Z. And Z is nothing but a normalization constant, so that all the text probability adds up to 1.

So, let us try to do that. When the tag is determiner word is the. So, let we just write down exponent of summation lambda i f i and lambda i is lambda h is 1 here in this problem. So, it will be simply exponent summation over f i. So, what features are 1 and what features are 0. So, for this word, so everywhere where we need the previous tag or the previous word should be 0, because this is the start of the sentence. So, I do not have any previous word tag or any previous word. So, all these features value will become 0.

Now, what is the feature that will become 1 what should be this one, this needs the next word i plus 1 word is light and the current tag is determiner is that will that 1. So, if you see here current tag is determiner on the next word is light. So, it could be 1 that is for feature f I. So, it is 1, so I will say it is exponent or let me write e to the power 1, this lambda is 1 divided by z, let me find out that Z. Let me find out the value for noun. So, again similar to this one all of features from f 1 to f 6 will become 0, because you do not know what is a previous word or previous tag this feature could have become one, but the current tag is not determiner, but noun. So, this will also be 0. This feature previous word is null yes that is true is a start of the sentence, and the current tag is noun, this will also become 1.

So, this is the only features that become the 1 for noun. So, this is e to the power 1. And what is the normalization this plus this. So, 2 times e to the power divided by 2 times e to the power 1. So, both will become 0.5. So, at this point all the two tags have we both the tags has probability 0.5. And any of because I am using a beam size of 2, I will have to keep both these tags. So, now I keep both this tags with probability 0.5, 05.

Now let us go to the next word light. Now again so I have to use probability of verb given this history; now this history is previous word, current word, next word and the previous tag. So, now, when I going about talking about verb, if I have to compute these features, I need to know what is the previous tag, here it can be a either determiner or noun. So, that is why I need talk in terms of the sequences. So, here I have one sequence, determiner-noun, determiner-verb, and second sequence with noun-verb. So, we have to take both the sequences separately in compute the probability. Similarly, I will have to do the same for adjective.

So, let us try do that for one sequence. So, noun-verb sequence. So, what will be the probability of tag i given the word. So, this we can write here. So, let me write only this summation lambda i f i part. So, this will be summation lambda i f i. So, let us go to the features here, so noun-verb. So, let us go to the features first features previous tag is determiner no previous tag is noun here this is 0; previous tag is noun, yes and the current tag is verb this is this could be 1, so f 2 is 1. F 3 -0, because previous tag is not a adjective. F 4 previous tag is the noun, yes previous word is the current tag is adjective, no it is verb then this will be also not correct, because the current tag is not a adjective, but verb; previous word is light noun, next word is light noun, previous word is null

noun. So, only f 2 is 1. So, this will be for this sequence it will be e to the power 1 divided by the normalization Z.

Now what will this normalization depend on this will depend on from all everywhere where this context is taken what are the probabilities. So, I have to compute the probability for this one also to find out this Z. So, I know this probability, I know this function this function and I will normalize them to add to 1. So, what will be for the function for noun and adjective I just try that from the features again. Previous tag is determiner gone, previous tag is noun, but current tag is verb noun, previous tag is adjective noun, previous word is the yes and tag is which adjective yes. So, this will become one f four will become 1. Previous word is the yes, next word is book yes, current tag is adjective this will also become 1 all three will be 0. So, now, this will become e to the power 2 divide by Z; and Z, I can write as e plus e square; same here e plus e square.

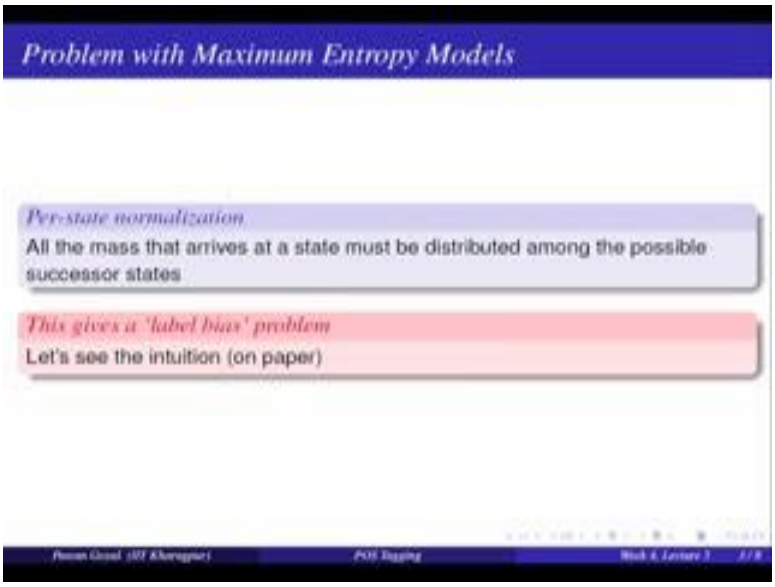
So, now I know the probability of getting verb at this position given the previous tag is noun; and adjective at this position given the previous tag is noun. But what is the probability of this whole sequence it will be multiplied by the probability of getting noun that is half, similarly here half. So, this is the probability of selecting this sequence similarly I will compute the probability of selecting this and this normalized them multiplied by 0.5. So, now, I will get the probability for four sequences, so that is noun-verb, determiner-verb, determiner-adjective, and noun adjective, I have the probability of four sequence of this system. And then I will select only top two from there.

So, suppose the top two could be say noun-verb and determiner-adjective. So, what will happen now, for the next step, I will consider only say determiner-adjective and noun-verb, and I will know also their probabilities. Then I will take each individual as a history and see determiner-adjective then noun, determiner-adjective then verb normalize the probability. Similarly noun-verb - noun, noun-verb - verb normalize the probability accordingly multiply this. So, again you will have four sequences here, you will have probability for all four sequences and take the one that is having the highest probability and that will be your final tag sequence in this example.

So, I hope the idea is clear I am not solving this fully, but I am I will encourage you that you do it on your own, and see that you can find out what is the appropriate sequence

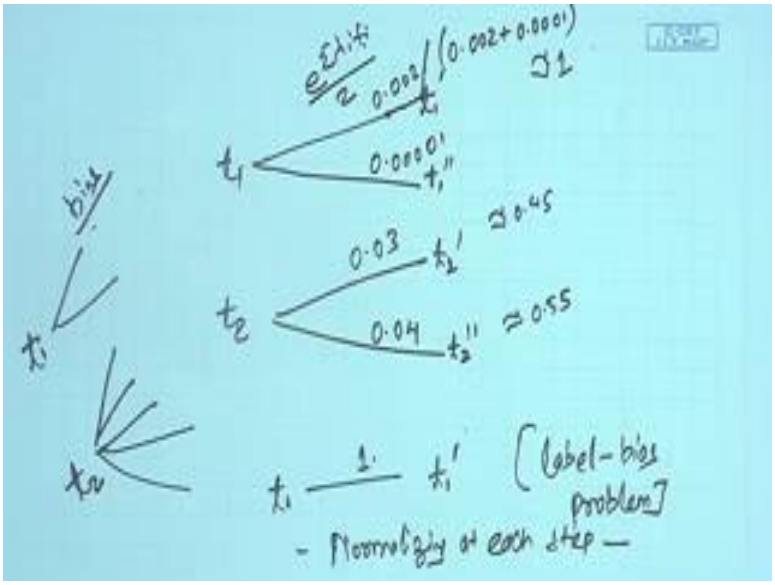
using the MMA model. So, this was how to use beam search algorithm for MMA model. Now I will just talk briefly with what is the problem with this, what is the single problem with this maximum entropy model then? So, let we have to think about condition random fields.

(Refer Slide Time: 13:30)



So, in maximum entropy model, we do a per state normalization that is all the mass that arrives at a state must be distributed among the possible successor states, and this is giving rise to a label bias problem. So, let us see; what is the intuition. So, what do I mean by this. So, let me take the same example. So, first let me take the same example to explain what do I mean by normalization at each state. So, take this one. So, you are computing noun-verb and noun-adjective. And you had the features like e square and e 1, but you where normalizing them by so you are normalizing then such that these two add up to 1. Same thing, you will do with determiner; you will make sure that these two add up to 1. So, you are normalizing at each state, so why will that a problem.

(Refer Slide Time: 14:40)



So, let me just taken hypothetical example. Suppose, in your maximum entropy model you were having a tag t 1 and tag t 2 at any given point. So, next point, suppose from t 1, you can go to go two different tags t 1 prime and t 1 double prime. And from t 2 again you can go to t 2 prime t 2 double prime; they may be same, they may not be same. Now, how do you compute this probability, it will be e to the power summation lambda i f i divide by z; z is nothing but the addition of these two. Same here now this is what is the importance for how many features you were having and so on.

Suppose, for a particular choice of these tags, it happens that in one of the branch, they are having this value has 0.002; and this value has 0.0001 or let us say even much smaller value. And this branch is having value of 0.03, 0.04. So, what will happen now this is not normalized values. So, it tells that this summation lambda f i is getting a higher score in these two cases; and lower score in these two cases. But because you are doing for normalization, you will divided by 0.002 plus 0.0001 and that will be closed to say very closed to 1, 0.98 or something; on the other hand this will be closed to 0.45 or 0.55.

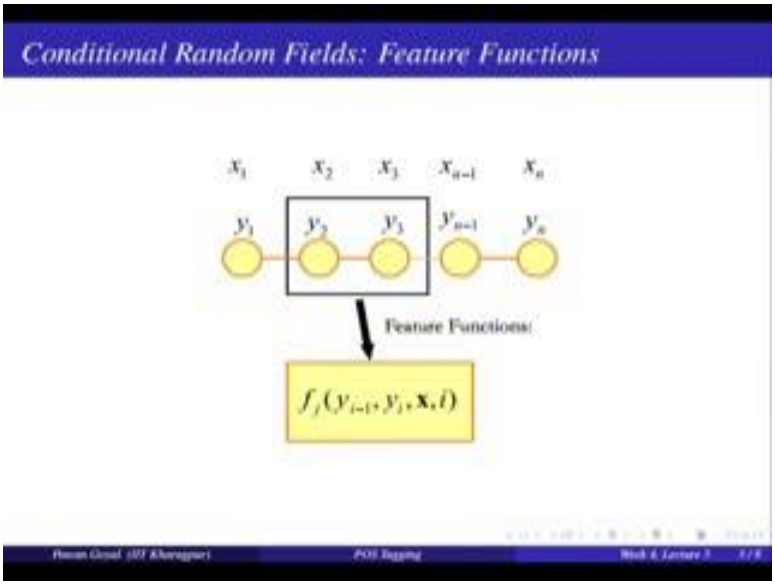
So, what is happening here even though this probability was low when I normalized this became very this became very high that is one particular problem. On the other hand, suppose that from t 1, there was no possibility of going to t 1 double prime. So, there was only one tag possible. So, you will in independent of the context, it will always get a

probability of 1, because you have to normalize at each state that is if from t 1, I can only go to one tag t 1 prime and everything else has a probability of 0, because if normalization this will become 1. So, we multiply this probability t 1 independent of the context and this is called as the label bias problem. And this comes because you are normalizing at each step that is one problem with the maximum entropy Markov models.

So, let us see how we avoid this problem in condition random fields. So, I hope this problem is clear that you are doing normalization at easy step at each state and that is giving some bias towards those states that are having few transitions then the (Refer Time: 18:10) having more number of transitions. So, let me just telling one thing. So, suppose I have two different tags from t 1, you have two possible transitions; from t 2, you have five possible transitions. What will happen these will get there will be a bias towards choosing this state because this will be normalized and one of this will get a higher value and this may not happen here because there are five possible transitions. So, this gets a bias and that is not what is idle.

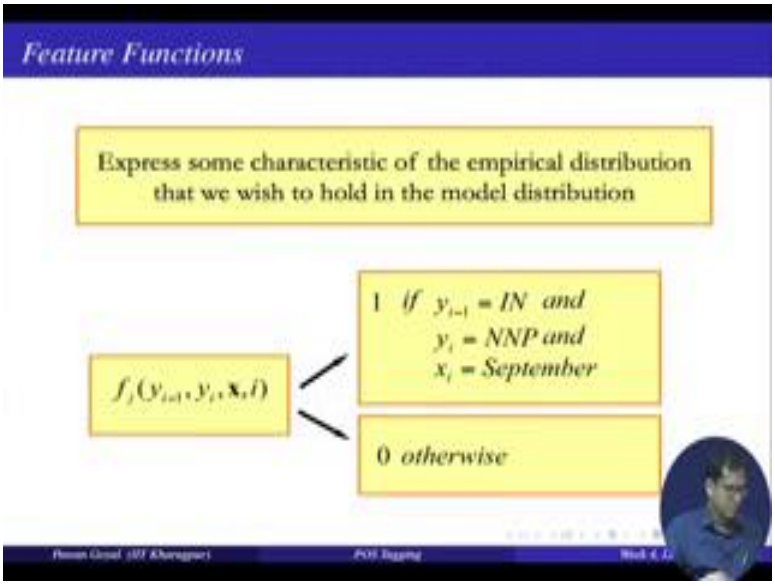
So, how do we avoid this problem and conditional random fields? So, conditional random fields are undirected graphical models, and while there are many variations of conditional random fields, so there is a generic structure, we will look at the linear generic structure.

(Refer Slide Time: 19:08)



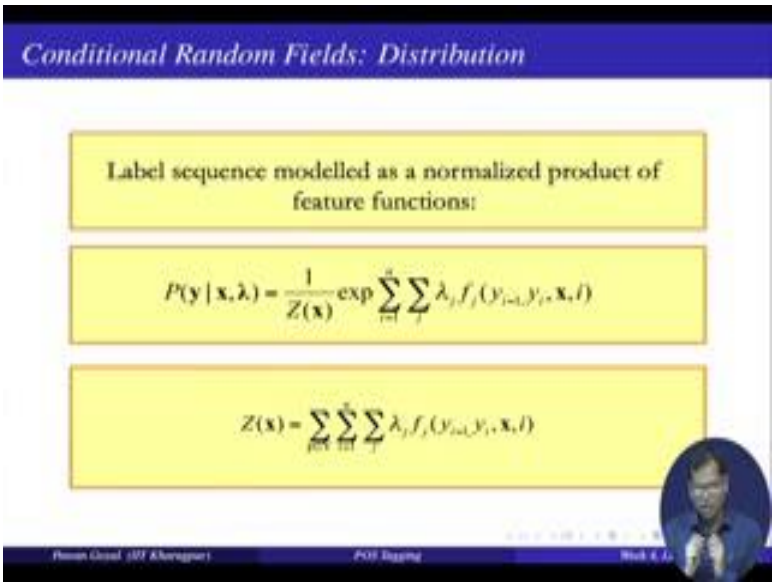
So, here we are seen the linear generic structure of conditional random fields. So, again like in the previous case, you are having a sequence x 1 to x n and these are the tags y 1 to y n assign to these tag. Now how they are so in what way they are very similar to maximum entropy model they are similar in the sense that we they use the same for the feature functions. So, what you are seeing here, for the ith point the feature function, so suppose i is equal to 3 would be a function over the previous tag y 2, current tag y 3, the whole the input you can take any number of words before and after x 3 and this is ith index. So, feature functions are again function of the input current tag and previous tag, this is in the linear generic structure.

(Refer Slide Time: 20:17)



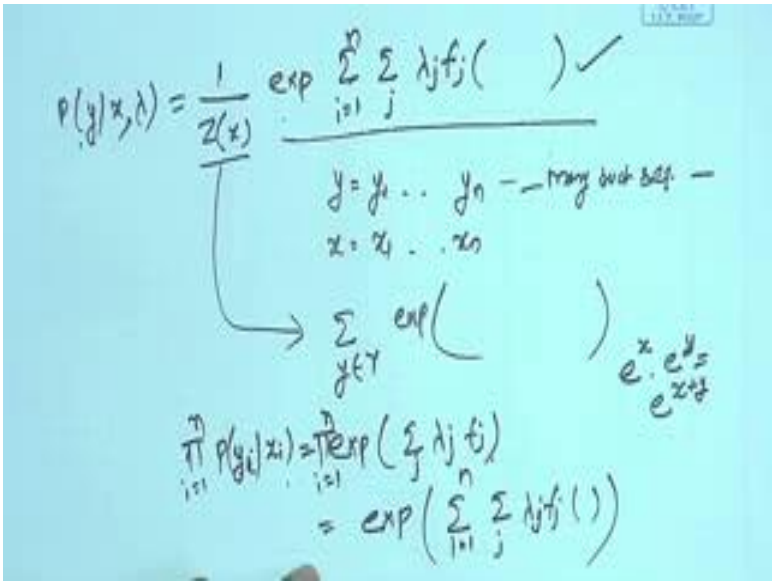
So, conditional random fields are like factor graphs. So, what happens the probability of each node will depending on only its neighbor. So, and you can use the same sort of features. So, like this is what we had discuss in maximum entropy model also. So, I have a feature that is 1, if previous tag is I N current tag is NNP and the current word is September and 0 otherwise. So, we will see there are very similar sort functions that we are using in Maxent. So, so they are same in as Maxent in that sense, but how they are different.

(Refer Slide Time: 20:45)



So, a difference comes in how the normalization is done. In Maxent model or maximum entropy Markov model, we are doing normalization for each state, so that is at easy state if I have multiple transitions, I will make sure that the probability for them at adds up to 1. This does not happen in conditional random fields. So, we will compute the features expectations of feature values for each possible transition, whole sequence and then I will do normalization, so that we can see from the probability function here. So, y is a whole sequence y 1 to y n given a current the current input sequence x 1 to x n and lambda is the feature weights that you will learn. And this is 1 by Z- x say a single normalization parameter exponent summation over i is equal to 1 to n summation j lambda j f j. So, let us try to understand quickly what this function means.

(Refer Slide Time: 21:52)

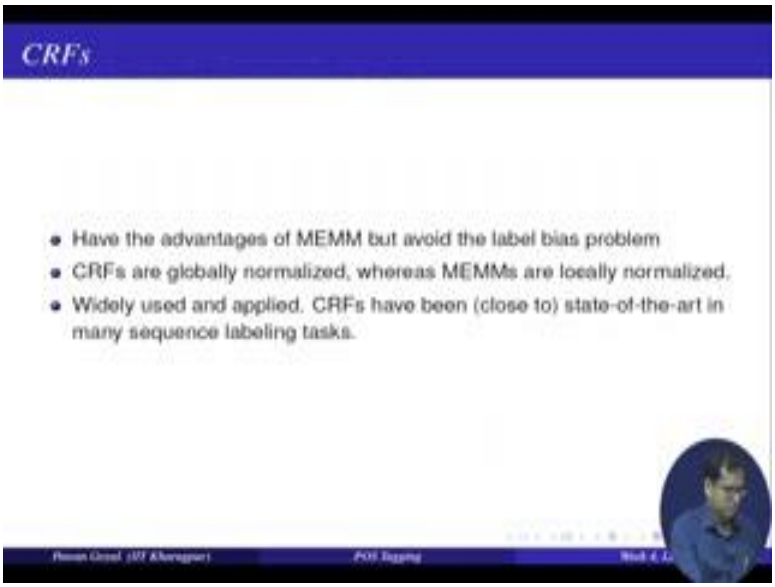


So, here having 1 up on Z x exponent summation i is equal to 1 to n summation over j lambda j f j and say f j is a jth feature. And this is probability y given x lambda, lambda are the feature of x. Now, let us try to understand this. What do you mean by y, y is a sequence y 1 to y n; and x is the input sequence, and there are many such sequences possible, so this is a probability for a given sequences. And this Z is a normalization that is done over all, so that is a summation over all the sequences. So, we can write it as, this will be summation over all the sequence. I have this value for only sequence y current sequence y, I will add it over all the sequence that will give me Z, so summation over all y possible y exp and whatever was in set.

So, now how do we get this equation? So, remember this equation summation j lambda j f j that is for particular tag given at the ith position. So, I have exponent summation j lambda j f j that is probability of a tag y i given x i; x i can be all history at the given point divided by Z was there, but forget about the Z term. Now, here we are computing probability for the whole sequence y i, i is equal to 1 to n. So, so this will be multiplication. So, multiply i is equal to 1 to n. So, multiply i is equal to 1 to n. Now, if I multiplying multiple exponent this is like, so for example, e to the power x times e to the power y becomes e to the power x plus y. So, that is multiplication of all the exponent is nothing but like summation of what is inside exponent summation i is equal to 1 to n summation j lambda j f j and that is the function a. And Z x is normalization over all such transitions all such sequences.

So, you are not doing normalization here. So, what happens in Maxent you are doing normalization here? So, for a given i, you make sure that everything adds up to one and that is not being done here. You are not making sure that at each I all the tags will the probability for the text will add up to 1. Now, you are doing a normalization only in the end, I know the probability or something that is proportion to probability for each tag sequence and then I will normalize everything by this z x and that is way this avoids the label bias problem. So, this is the particular function that is using conditional random fields.

(Refer Slide Time: 25:24)



So, what we can see that. So, conditional random fields have the advantage of maximum entropy Markov model, they use a same sort of features, the kind of model is very, very similar to maximum entropy Markov model. But they avoid the label bias problem. So CRFs are globally normalized, whereas MEMMs are locally normalized, so that we had discussed. And they are very widely used and applied for many, many sequence labeling task. So, they were very closed to the state of the art models for many of these sequence labeling task. So, whatever sequence labeling task that comes your mind, so starting from part of speech tagging to name and recognition.

So, there you can apply a conditional random field model and lots of libraries are also available. So, (Refer Time: 26:13) one particular app that is very popular, and then there are CRA plus plus and many other libraries that you can use. What is important is that

you understand, what is the sort of features that that you need to use and then the model will help you to train you are own CRF.

So, this ends our discussion on part of speech tagging we did discuss a lot about what will be the models for sequence tagging. So, from the next week, we will start discussion on syntax that how do we find out what are the word, what are the word arrangement in a sentence and how do we group them in various sort of phrases. So, I will see you next week.

Thank you.