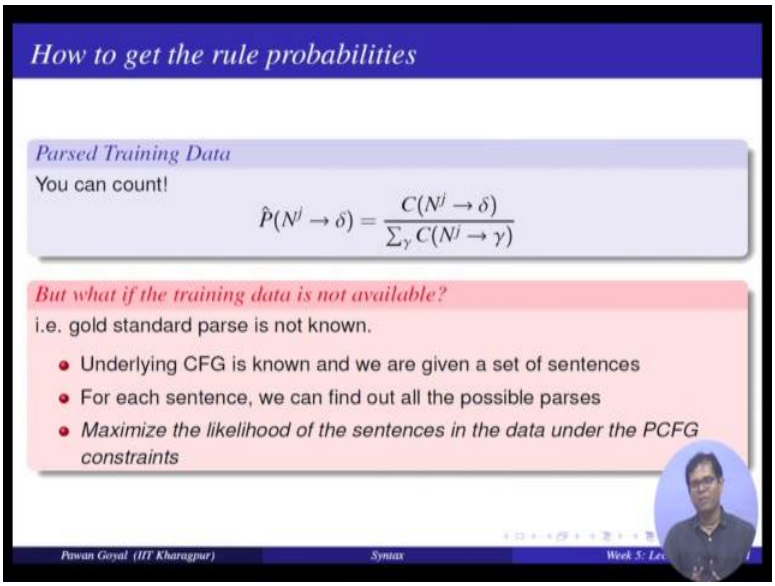
Natural Language Processing
Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 25
Inside-Outside Probabilities

Welcome back for the final lecture of this week. So, in the last lecture, we had started with the concept of inside outside probabilities and how do you use them for answering certain questions like what is the probability of a sentence as per my grammar and what is the most likely parse. So, I have use the inside algorithm specifically to find out what is the probabilities of this sentence as per my grammar and then in the end I was saying that we will also use this concept to find out the rule probabilities of my grammar and how do we do that again exactly what we will discuss in this lecture.

(Refer Slide Time: 01:00)



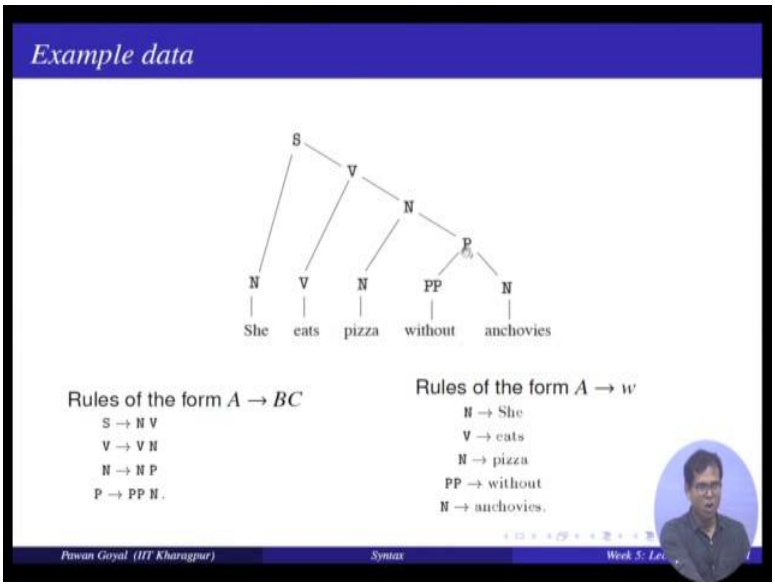
Now in general, how do you obtain the rule probabilities and remember this is very similar to what we were also talking about in the case of HMMs, when I have to learn the parameters of my HMM, I can do that in 2 manner, one where I am given already the labeled data set; that means, I am given what are the sentences and what is their past reach. If I am given all that so, then from there I can compute how many times a particular non terminal derives a particular sequence divide by number of times a particular non terminal has been used that will give me the probability of this particular

rule. If there is only one rule possible from N j this is always you want, but if there multiple rules are possible I can find out what fraction of times this particular instance has been used in my labeled data sets and by that I can compute all the rule probabilities.

Now, so this is easy, but what about the case where the training data is not given to us that is I have no way to find out in which sentence what is the rule that has been used. So, what is in fact, given to us? So, we are; assume that we are given the underlying context free grammar. So, I am given all the possible rules, but what I am not given what is the probabilities for each individual rules so; that means, the particular PCGFs part the rule probabilities is not given to us and I am given a lot of sentences and I am the given the grammar that will generate these sentences, but not the rule probabilities and my task is how do I find out these rules probabilities.

So, in other words how do I find out the parameters of my PCGFs and we will use the same sort of idea that we did earlier that is you are given the observation of sentences find out the parameters of a model that maximize the likelihood of observing the sentences. So, this is what we are going to do maximize the likelihood of the sentences in the data under the PCGF constraints and for that we will use some sort of expectation expression algorithm.

(Refer Slide Time: 03:25)

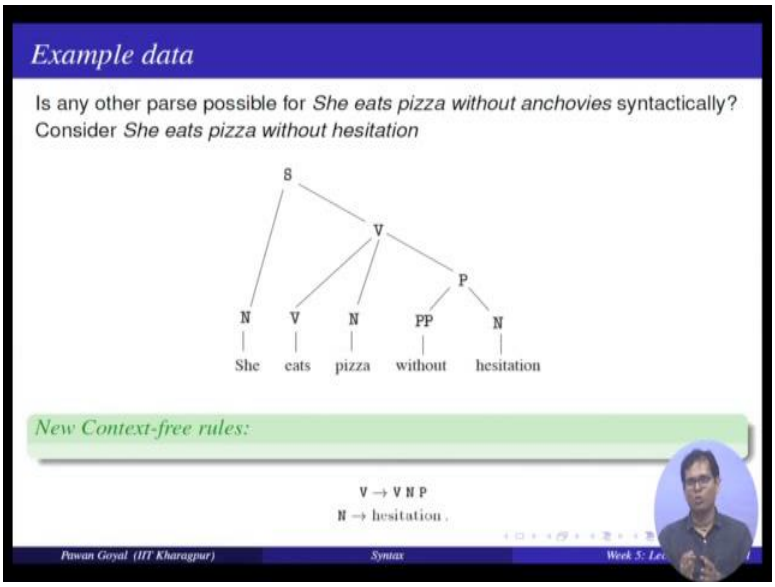


Let us take a simple example and what is the intuition for using this. So, we start with this simple sentence like she eats pizza without anchovies and a particular parse; tree is

given to us. So, what kind of rules do you see here? So, you have rules of the form a single non terminal derives to 2 different non terminals like S derives N and V, V derives V and N and so on and there are rules of the form a non terminal derives a terminal like here and derives anchovies PP derives without N derives pizza V derives eats and so on. So, these are different sort of non terminals that are given the rules that are given to me I do not know the probabilities for each individual rule.

Now, can you think of any other parse for this sentence, she eats pizza without anchovies, so one parse that we saw was she eats pizza and pizza without anchovies modifying pizza what is the other possibilities. So, other could be the without anchovies does not modify pizza it is a separate phrase altogether starting from V, so something like she eats pizza with fog.

(Refer Slide Time: 05:00)

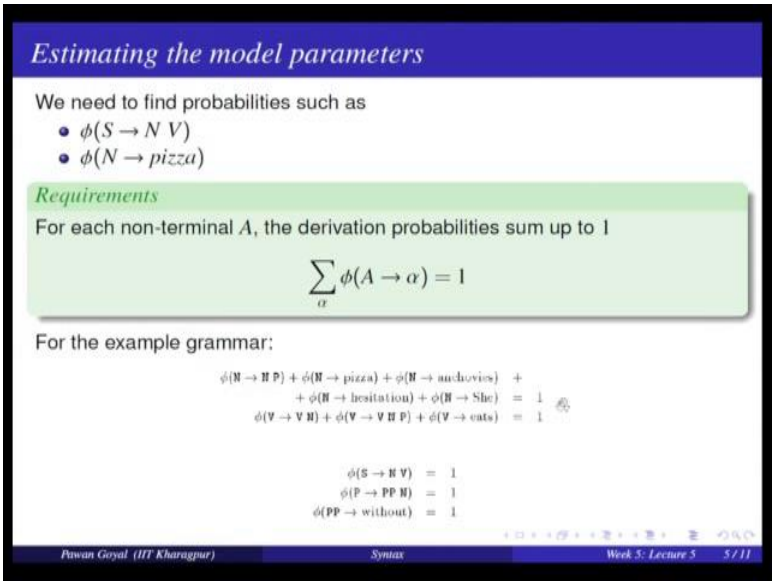


In that sort of meaning so yeah so, she eats pizza without hesitation that is another sort of possibility, now a single V here is giving me V and N P. So, this is the different parse tree for the same sentence

Now, I can what are the new rules that I have added V gives me V N and P and N gives me hesitation these are 2 new rules I have added. So, now, what do you see I have 2 different sentences; she eats pizza without anchovies and she eats pizza without hesitation and both have 2 possible parse edge and I know what are the all possible rules

now my task is given these sentences how do I find out what should be the ideal rule probabilities.

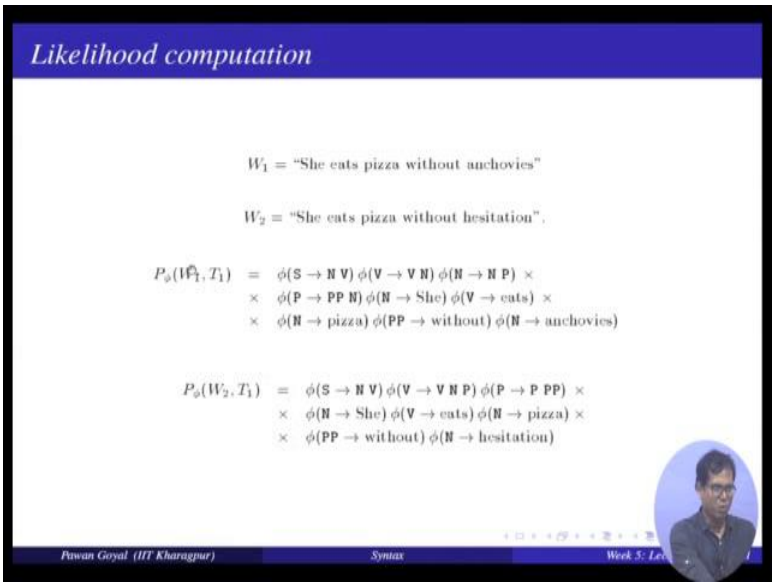
(Refer Slide Time: 05:49)



Now for that let me first define so we need to find out all these probabilities. So, probability of this rule S gives me N and V probability of rule N gives me pizza and so on, now what is the PCGF requirement? The PCGF requirement is that it starting from a single non terminal all the possible sets of right hand side that I can generate (Refer Time: 06:11). So, that is so, all the possible rules starting from a to any alpha the probability should also added to 1 so now, if I look at my grammar so, I have 5 rules pair and occurs in the left hand side N gives me NP, N gives me pizza anchovies and so on. So, all these should also add to 1.

Similarly, 3 rules where V occurs in the left hand side so, all these should also added to 1 and then there are some other rules like S gives me N V, N V nothing else, this would be 1 and so on. So, this constraints I can obtain from my from my grammar, I know what are the rules and what is the constraint starting from the left hand side all possible rule should have a probability adding up to 1.

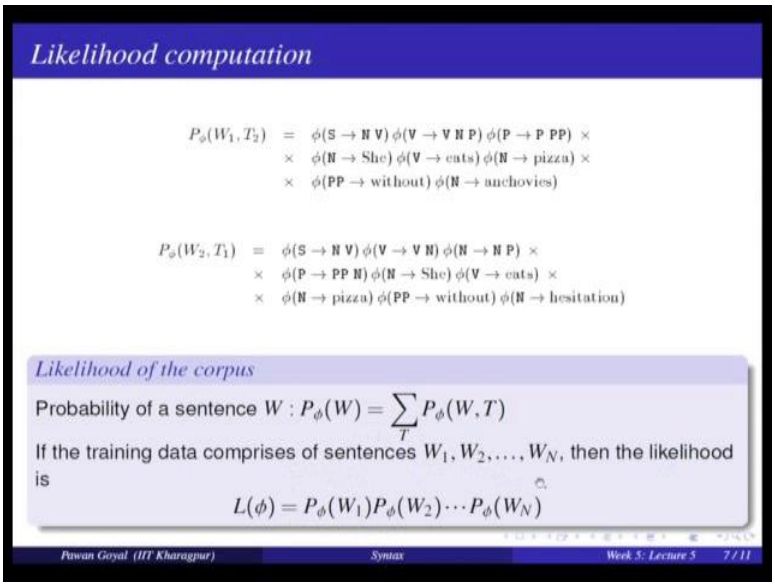
(Refer Slide Time: 06:58)



Now, I have 2 sentences, can I compute the likelihood of the sentences W 1 and W 2. So, what do I mean by likelihood what is the probability of generating the sentence as of my grammar? Right now I am not giving the rule probabilities, but I can write down in terms of the variable rule probabilities. So, what will the likelihood of W 1? So, for that I have to take the 2 possible parse trees. So, here T 1 is the first parse tree.

So, here I am giving the probabilities of both the sentences as per the first parse tree. So, probability of sentence W 1 as per the first parse tree T 1 and probability of sentence W 2 as per the first parse tree T 1, how do I compute that it is very similar to what we saw in the case of PCGF? How do we compute the probability of a parse tree if the rule probabilities are given that was very easy here the possibilities are not given, but you can parameterize. So, you will say what is the probability of S giving me N V and so on up to you go to the leaves and I do not know these whole probabilities similarly I can write down this likelihood of the sentence W 2 as per my first parse tree.

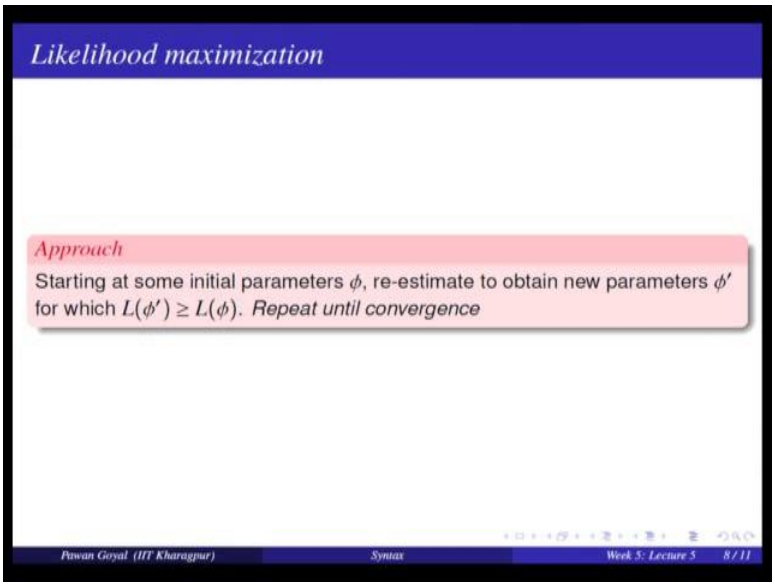
(Refer Slide Time: 08:11)



Similarly, I can do further second parse tree also for both the sentences. So, this tells me. So, if I know all the possible parse trees because my CFGS given to me, I will know all the possible parse tree, I can put my all the rule probabilities as variables and define what is the likelihood of various parse tree now what is the likelihood of the sentence, it will be summation of the likelihood as per different possible parse trees. So, probability sentence such summation over all the possible trees that can generate this P phi W T, in this case for both sentences W 1 W 2, I had 2 different parse trees.

So, I will just add the 2; 2 probabilities to get the likelihood of the sentence and how do I get the likelihood of the whole corpus that has multiple sentences for that I will come to the likelihood of each sentence and multiply those. So, if I have a sentences W 1 to W N, I compute the likelihood of each and keep on multiplying now we know that how do I express the likelihood of my corpus in terms of the rule probabilities right the only variable here are all the rule probabilities now I can further define my problem.

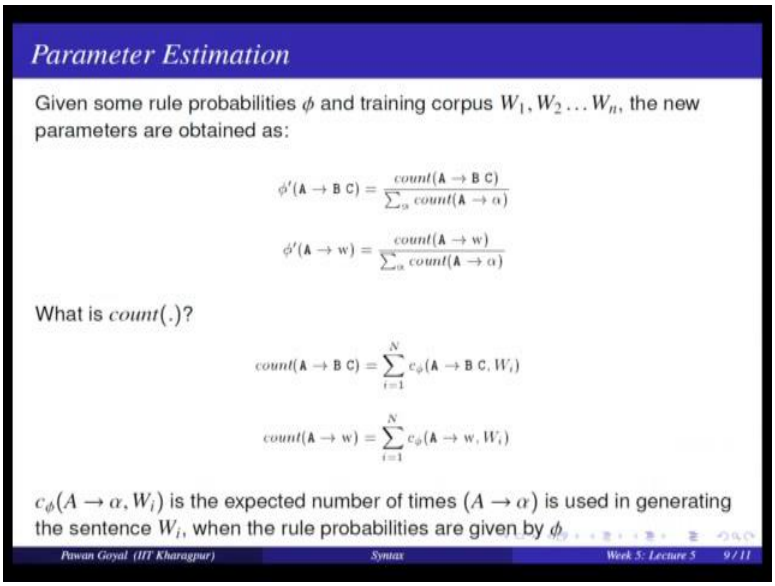
(Refer Slide Time: 09:33)



My problem would be, so, this is some sort of E M approach, I will start with some initial parameters phi, phi means the rule probabilities I want to re estimate so that I obtain some new parameters phi prime such that the likelihood of my corpus increases now. So, L phi prime will be greater than equal to L phi and I keep on doing that until I converge. So, now, here we have to apply (Refer Time: 10:04) algorithm so that we can keep on updating our rule probabilities, this is the parameter of my system and how do we do that if you remember that like what we did in the case of learning parameters for G S analogous to that what we will do here we will start with some arbitrary rule probabilities phi and use that to compute something intermediate.

In this case what we will compute? What is the expected number of times a particular rule has been used if the rule probabilities are as per the current parameters? I will compute the expected value; again use the expected value to compute the probabilities. So, I will compute my phi prime again use the phi prime to compute the expected number of times each rule has been used and again compute 5 dwell prime and keep on doing that until you converge and that is why we will be using the inside outside probabilities. So, let us see.

(Refer Slide Time: 11:02)

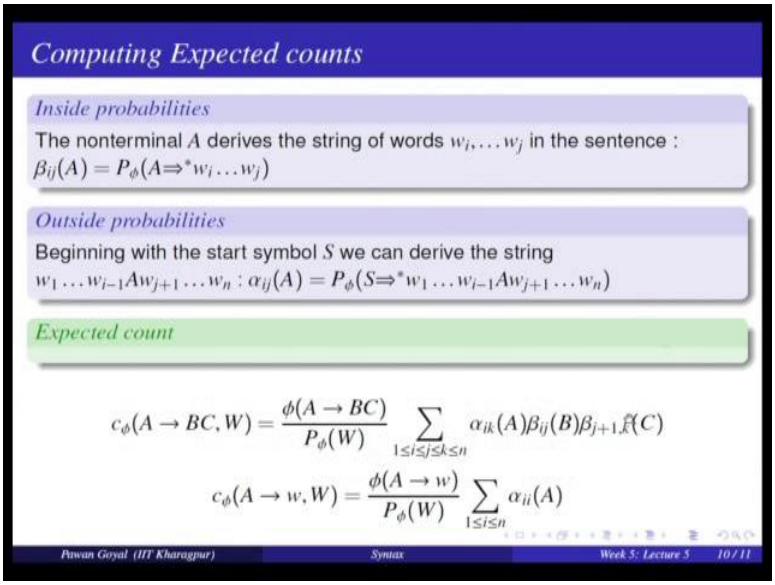


Idea is that I start with some rules probabilities phi and I am given a corpus that that what are sentences that I observing W i W j and I will obtain the new parameters phi prime using the simple idea. So, this is something that we were saying if we are given the labeled data that is why I will compute the rule probabilities. So, I am saying I can always define probability of the rule a given B C as the number of times the rule A given B C A gives goes to B C or a derives to B C has been applied in my corpus divide by all the possible all the different times where A derives alpha for all possible alpha and this gives me the probability for a deriving B C.

Similarly, I can compute probability a deriving W by saying how many times, this rule has been used divide by number of times a gives me alpha has been used in my corpus in my actual corpus, but we do not have any labeled corpus, we only know what parse are possible and for each parse, we can compute the probabilities using the previous parameter phi. So, how do I write down this count A derives B C number of times, this rule has been used for that I use the idea that I have multiple sentences any sentence I can find the expected number of times this rule a deriving B C has been used. So, that is count A deriving B C is nothing but summation over all, the sentences number of times a deriving B C has been used for the particular sentence same 1 for the count of a deriving W each sentence find out the expected number of times a particular rule has been used, now how do I actually come up with this formulation expected number of times, a

particular rule has been used in a sentence and for that we use the inside probabilities and outside probabilities.

(Refer Slide Time: 13:17)

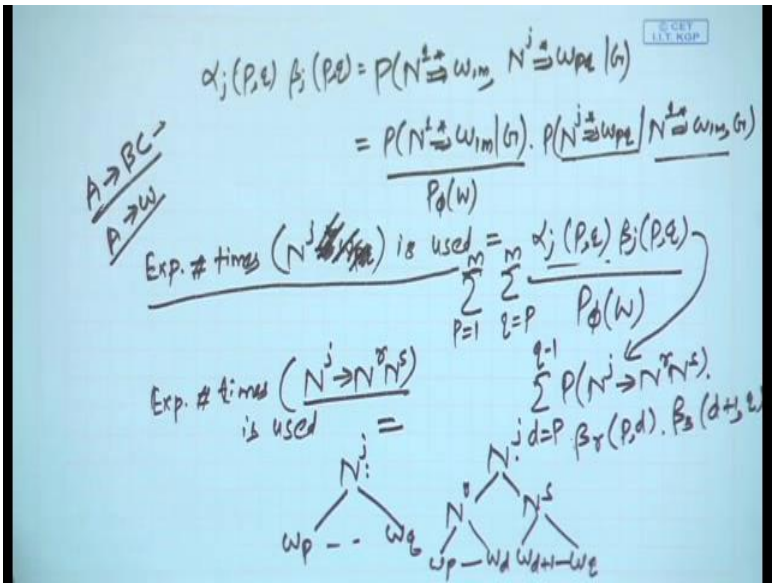


Now coming back to this inside and outside probabilities and how do we use that to compute the expected number of times a rule has been used in a sentence now this must be clear by the previous slide that if I can compute the expected number of times, a rule has been used in a sentence, I can keep on updating my parameters, this is the only bottleneck, in the previous computation and we will see how to do that using the inside and outside probabilities. So, let me give the definitions again. So, inside probability is starting from non terminal A, I derive the words W i to W j in the sentence so that is probability that A derives W i to W j as per my grammar.

And the outside probabilities starting from the symbol S, I can derive the string W 1 to W i minus 1 A and W j plus 1 to W N. So, it starting from S, I can derive W 1 to W i minus 1 A W j plus 1 to W N, now once we are given the inside and outside probabilities, we can actually compute the expected number of times, the rule has been used and the expression comes out to be this one expected number of times a rule A has been A derives B C has been used in my sentence W each the rule probability a given B C divide by the probability of the sentence and this very peculiar term that you see that you are seeing here. So, you are seeing here alpha i k A beta i j B and beta j plus 1 k C, now how do I actually come up with a term like that and how do I come up with this

expression that is expected count is given by this. So, for that let us go back to what we were discussing in the last lecture that I can multiply inside and outside probabilities to know something about the probability for the sentence.

(Refer Slide Time: 15:16)



Let us go back to that. So, what we were saying if I multiply alpha j p q and beta j p q what does it give me? It gives me the probability that it starting from N 1, I can derive W 1 m and it starting from N j, I can derive W p q as per my grammar. So, now, I can use the chain rule here to write it like that. So, its probability N 1 derives W 1 m. So, this means it derives in any number of steps given by grammar times probability N j derives W p q given N 1 derives W 1 m and my grammar. So, now, what is this probability that N one derives W 1 m given by grammar second write as the P phi W probability of the sentence and what is this say what is the probability that this rule N j has been used to derive W Pp q given that the sentences there and my grammar is there.

And how many times this has been used in this particular context? Only 1 time, so, can I write down expected number of times N j derives W p q is used that would be alpha j p q beta j p q given divided by P phi W and suppose because I do not want to fix this p q, I just want to say expected number of times the rule N j has been used. So, each time it has been used only once for deriving W to W q. So, here I will have to sum over all the possible p and q. So, I will say p can go from one to m suppose there are W 1 to W m.

So, these are number of words and q can go from p up to m. So, this is the expected number of times my rule and j has been used.

Now, what is something that I have to express I want to find out for example, expected number of times a rule like A goes to B C has been used or a goes to W has been used what is the expected number of times these has been used now for that suppose let us take the easy case expected number of times the rule a goes to W or a derives W has been used. So, in this case what I am saying N j derives a particular terminal here that is some p p. So, I can write the beta j p p for that case and beta j p p is simply the rule probability that is what is the probability that in non terminal derives this word W p, so we will see the expression for that so, this one is easy.

But what about this case when the rule A derives B C so, in the particular notation that I have written suppose you want to say expected number of times N j derives N r, N S is used . So, now, what would happen? So, this beta j p q is when the terminal N j derives the whole sequence p q, W p to W q and can use any possible rules yes N r N S or N j N z whatever it can use any non terminals now what I am my limiting I am saying this rule should only use this. So, this non terminal should only derive N r N s. So, then I am saying so; that means, my N j will derive N r N s and this N r N s will again derive say W p to some W d and this will derive W d plus 1 to W q.

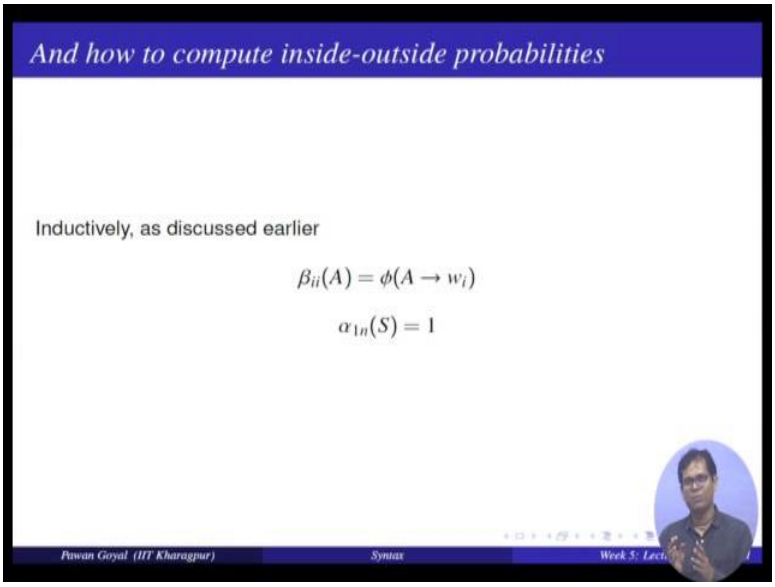
Now how do I modify this equation? So, alpha j p q is the outside probability that will remain the same nothing has changed for outside probability, but inside probability because I am saying this should be the situation. So, I further express it like with a particular path. So, I will write in place of beta j p q, I will write probability of the rule N j derives N r N s times this beta probability that is beta r p d times beta s d plus 1 to q, but now the d can vary, I have already been given that N j gives me N r N s, but they can take different possible d s. So, this will be summation over d and d can vary from p to q minus 1. So, these between p and q now if you put that can you see that you can you can actually obtain the same expression that was given in the slide.

If you go back to the slide that is what we have been doing here. So, you see the expression we have 3 parameters beta I j b. So, 3 parameters i j and k that corresponds to p d and q and this was the outside probability and this is the inside probability for the two children; i j and j plus 1 k and then you have the rule probability here. So, this

expected number of count has been derived in this particular form and same thing you can try with the next formula the expected number of times the rule a deriving the W has been used in my graph and you will obtain this particular expression.

What we are seeing here suppose I start with some initial rule probabilities. So, I can use the inside outside probability. So, all the recursive formulation to compute all the inside outside probabilities for my various rules and stages once I do that I can compute what is the expected number of times each and each individual rule has been used in my corpus as per the current parameters. Once I have the expected number of times rule has been used I can further estimate my parameters by number of times the rule used divide by number of times any particular rule starting with that non terminal has been used and that will give me the new parameters, again I will compute inside outside probabilities expected count, re-estimate the parameters and this I will continue until this converges.

(Refer Slide Time: 23:47)



And yeah, so computing inside outside probabilities is as we discussed earlier by this inductive manner. So, what we discussed in this module was that what is parsing in terms of a constituency structure and how do you use the formulation of context free grammar to do parsing, how do we incorporate the rule probabilities there? How do we learn the rule probabilities using this interesting concept of inside outside probabilities? So, I hope by the example that we did in the class, you will be able to understand how it is actual exactly works in practice.

In the next week, we will be starting with this different notion of parsing. So right now, we have done a constituency parsing. So, we will see there is a different notion of parsing called dependency parsing. So, what is the formulation that that dependency parsing follows? How it is different from this constituency parsing and what are different methods we can use for that that will be a topic for the next week.

Thank you.