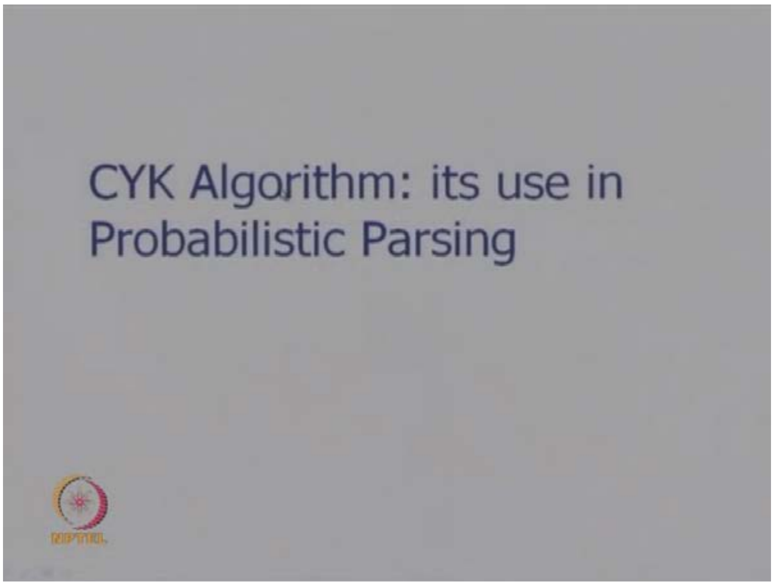
Natural Language Processing
Prof. Pushpak Battacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture - 40
Probabilistic Parsing Algorithms

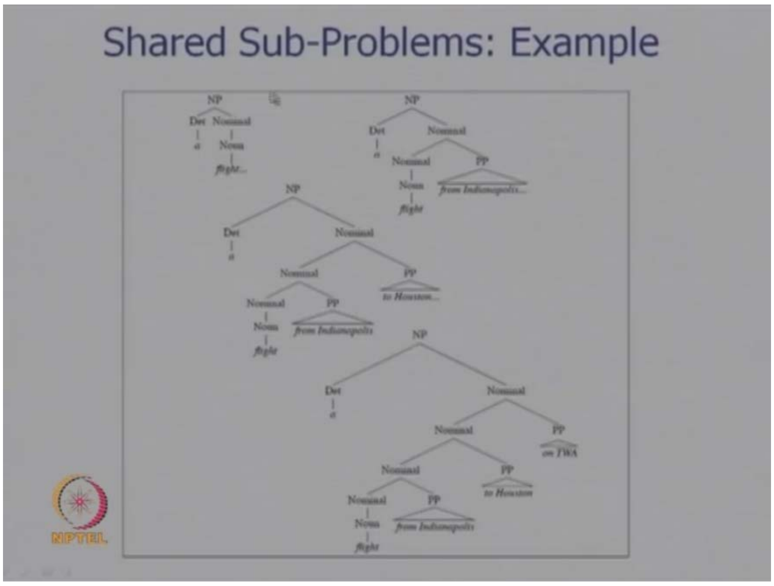
In this lecture, we will go a deeper into probabilistic parsing, and will finish the course on Natural Language Processing, with some summarizing observations, and an overview of what has been done, during this course of 40 lectures. So, as we said Probabilistic Parsing is needed, so that we can have a principled scoring mechanism, for multiple parse trees in case of a ambiguous sentence. If a sentence is not ambiguous, and has a single parse tree, then of course a score is 1.0, if there is a multiple parse trees, then depending on the frequency of the constituents of the parse tree. There is a weight age mechanism, which reveals how probable the parse tree is given the sentence, and as evidence in the corpora. So, we will begin this discussion on Probabilistic Parsing, and go to the slide.

(Refer Slide Time: 01:35)



But before that we will take up a very important dynamic programming, based deterministic algorithms, the CYK algorithm which is also used extensively in probabilistic parsing by making use of the beta or inside probabilities, so all this is coming little later.

(Refer Slide Time: 01:57)

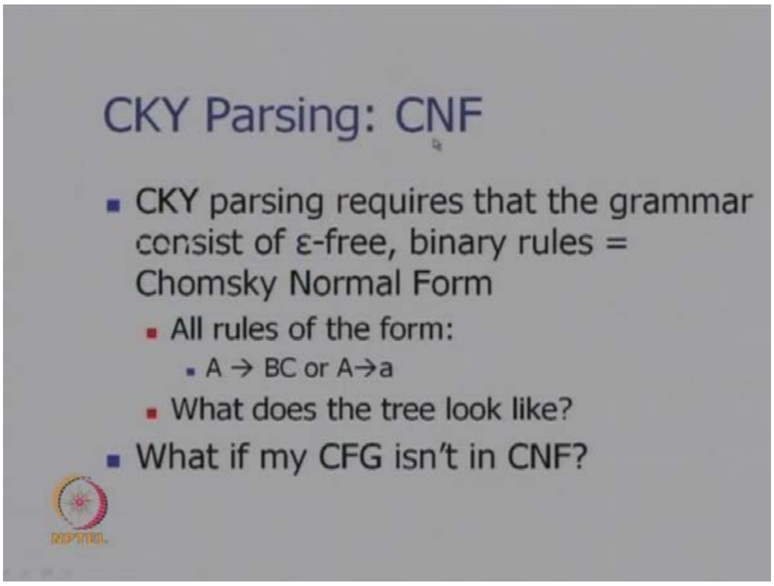


Now, when we look at the parsing problem, we see that lot of work on the way towards, building the complete parse tree can be reused. So, for example if we have this phrase a flight, here this is a noun phrase, is a article, and flight is a Noun and the Nominal, finally, giving rights to a noun phrase. A flight from Indianapolis, from Indianapolis a preposition phrase and flight is a Noun, and the whole thing, a flight from Indianapolis makes a Noun phrase but the point is that this flight going to Noun to Nominal to Nominal again, and then combining with a determiner, and producing a noun phrase is showing that, the previous work on NP can be reused. Now, you have 2 bigger structures, a bigger phrase a flight from a Indianapolis to use turn.

So, in this case 2, we find that this whole work on a flight from Indianapolis can be done for building the parse tree for the bigger phrase, a flight from Indianapolis to use turn. And finally, these larger phrase tree for the longer phrase, a flight from Indianapolis to use turn on TWA, these can make use of the work already done for flight from Indianapolis to use turn, which in turned can make use of the work done for, a flight from Indianapolis, which in term makes use of the work done for a flight.

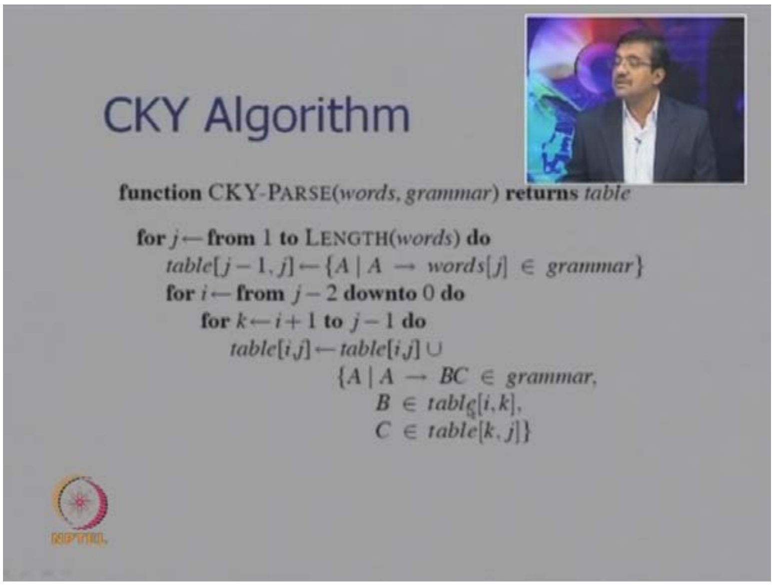
So, this kind of shared sub problems is a very common theme in building a parse trees is can make use of in the work done for a sub parse trees, and can sub parse trees for constructing a bigger parse tree, this quite common on in computer science, we have a mechanism to handle the situation. We can make use of the work done for smaller problems, and can a create the solution for a larger problem, and that the platform which is a eminently suitable for these is the dynamic programming platform. We use d p dynamic programming in parsing to and we have a very famous well-known algorithm, for this called CYK algorithm.

(Refer Slide Time: 04:40)



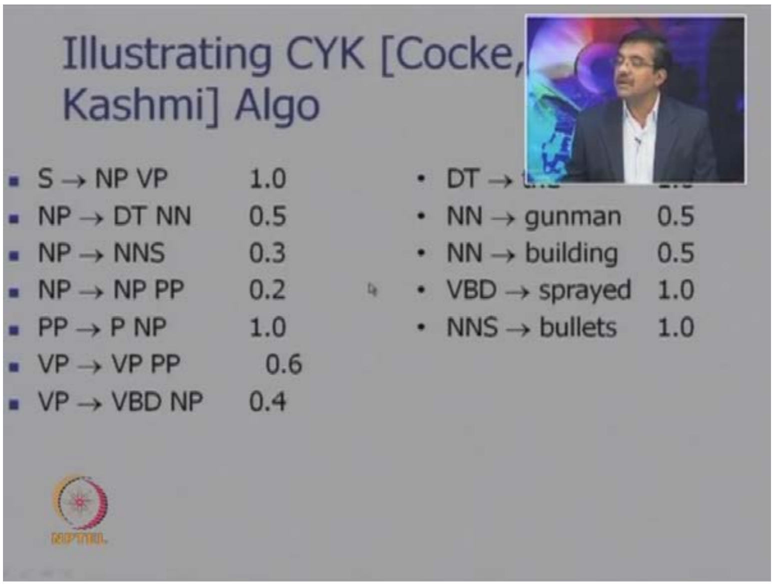
So, CYK parsing requires that, the grammar is consisting of excel entry binary rules, and all the rule should be Chomsky Normal Form. There is a, A goes to BC that means to non-terminals on the right hand side or A goes to small a, which is a terminal now, what if the CFG is not in CNF.

(Refer Slide Time: 05:05)



Then of course, we can do some preprocessing can we can converted into CNF form, the CYK algorithm is a very small recursive algorithm, dynamic programming based, the algorithm is based described by means of an illustration.

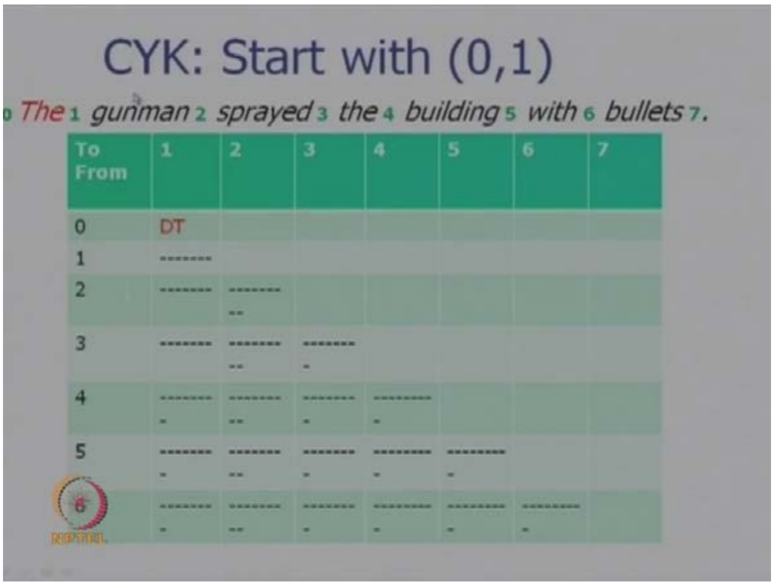
(Refer Slide Time: 05:20)



So, this algorithm is Cocke, Younger, Kashmi Algo and very interestingly, this algorithms is not the joint work of these 3 researchers, it is not as if together produce a paper describing algorithm, what is interesting has said they independently worked with other collaborators possibly to produce this algorithm, independent of each other findings. Since, there were almost discovers simultaneously, the nlp field the parsing field has taken the practice of has adopted the practice of calling this algorithms CYK algorithm, after the name of this independent researchers, Cocke, Younger, Kashmi algorithm.

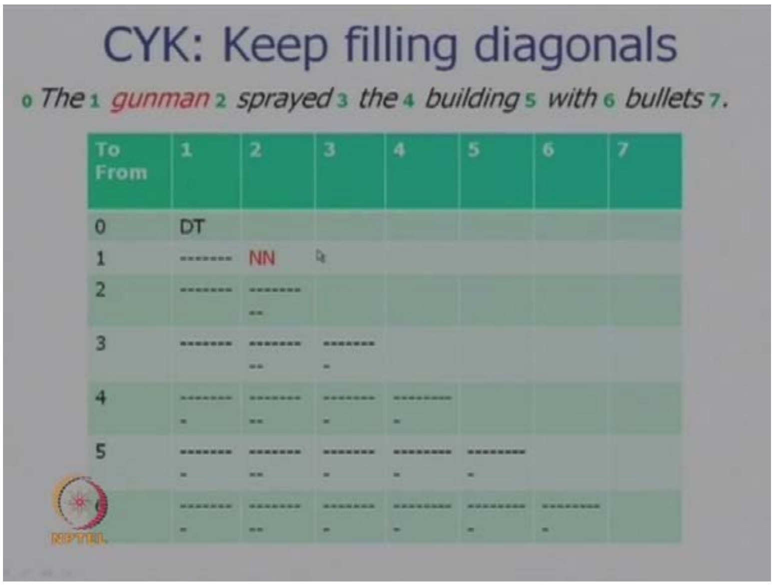
We work with our already familiar example as go to noun phrase, verb phrase, noun phrase goes to determiners noun, noun phrase goes to plural noun, noun phrase goes to NP NP P, PP goes to PNP, VP goes to verb phrase preposition phrase, verb phrase goes to past tense, verb and noun phrase with different probabilities. In take care of the fact at all the rules, which have the same non-terminal on the left hand side, have their probabilities coming up to one, and we have say that the probability value indicates, what percentage of time, these particular rule is applied in the corpus, to produce the sentences. On the right hand side, under lexical probabilities the probabilities are obtaining the vocabulary words.

(Refer Slide Time: 06:58)



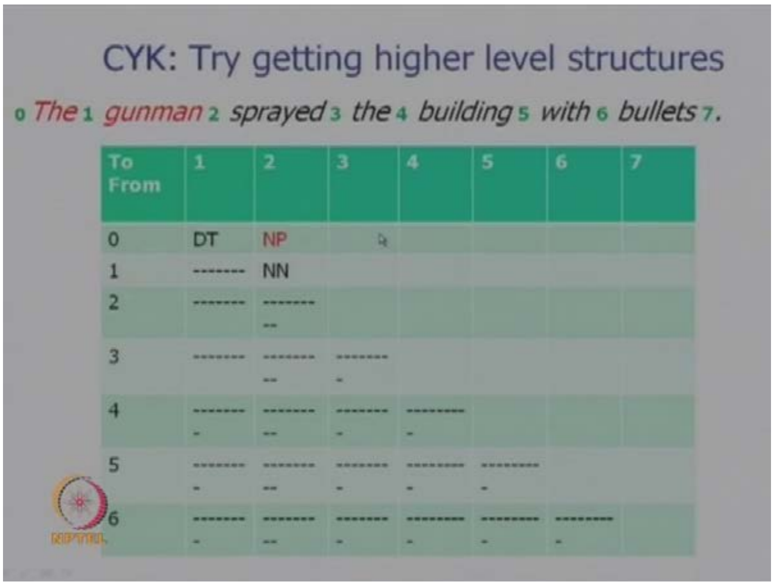
Now, we discussed algorithm for this sentence, we ever seen before the gunman’s sprayed the building with 6 bullets, we saw that these, particular sentences to parse trees in 1 case, with bullets is the preposition phrase, attaching to spray, another case the preposition phrase attaches to building, as if the building has bullets in it. And in this case bullets are the instruments, by which the spraying is done or others the bullet are the objects, which are sprayed. Now, the algorithms proceeds with finding non-terminals for between what positions for example, the word the gives rise to the non-terminal DT between 0 and 1. So, on the column, we produce the indices starting from the first word to the last word 1 2 3 4 up to 7 and on the rows, we have the preceding positions 0 1 2 3 up to 6. So, DT a determiners if found between 0 and 1, so these from an 2 means from 0 to 1, we have a determiner.

(Refer Slide Time: 08:12)



Next stage, we have found another non-terminal noun between 1 and 2.

(Refer Slide Time: 08:19)

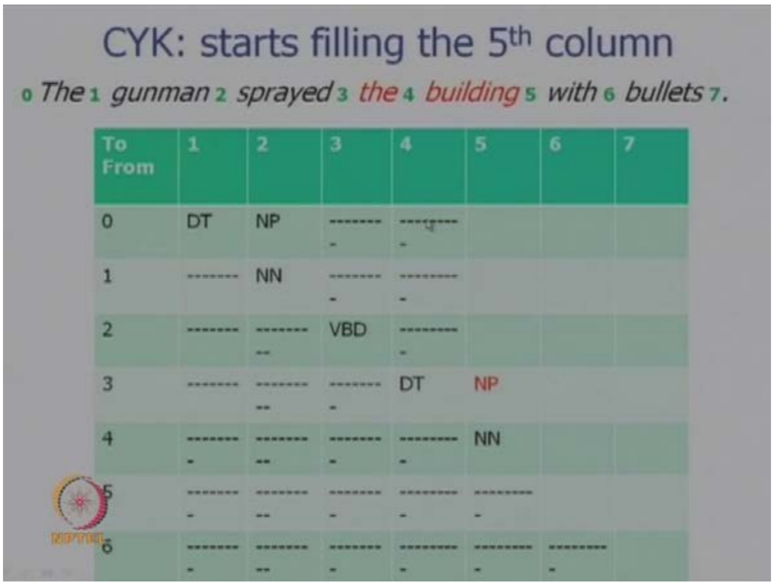


Next stage, we can combine this DT and NN to produce the noun phrase, between the positions 1 and 2, we have a noun. But between 0 and 2, that is spanning this whole range 0 to 2, we have a noun phrase which is in it through the gunman is determiners noun combination. And therefore, this is a noun phrase as for the rule Grammatical rule. Next between 2 and 3, we have been able to find a verb in past tense form VBD.

(Refer Slide Time: 08:52)

Proceeding further, we see that we cannot produced any bigger phrase at this location which is 2 and 3 and which is 1 and 3, so between 1 and 3 I cannot produced any bigger phrase. Because noun and in the past tense together do not form any phrase, between 0 and 3 the gunman sprayed, again know phrasing is possible. Now, between 3 and 4 we have found a determiner, this is the, we do not find any phrase, which hence in the, that is impossible so that is why all these columns are kept vacant, there is no phrase which hence in position 4. Then we find a noun between 4 and 5, which is the building and now we can see that.

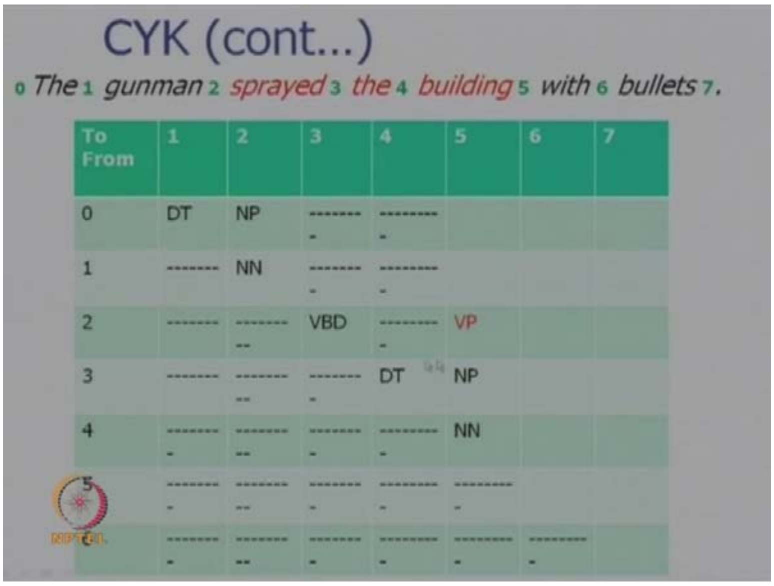
(Refer Slide Time: 09:47)



To From	1	2	3	4	5	6	7
0	DT	NP	--------	--------			
1	-------	NN	--------	--------			
2	-------	--------	VBD	--------			
3	-------	--------	--------	DT	NP		
4	--------	--------	--------	--------	NN		
5	--------	--------	--------	--------	--------		
6	--------	--------	--------	--------	--------	--------	

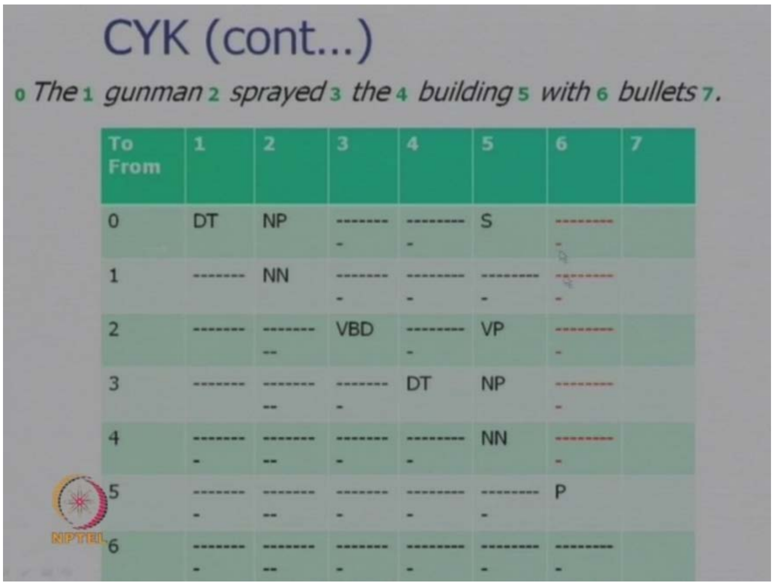
We can keep on finding phrases, so between 3 and 5 there is a noun phrase, the building is indeed a noun phrase, produced from the combination of DT and NN.

(Refer Slide Time: 09:58)



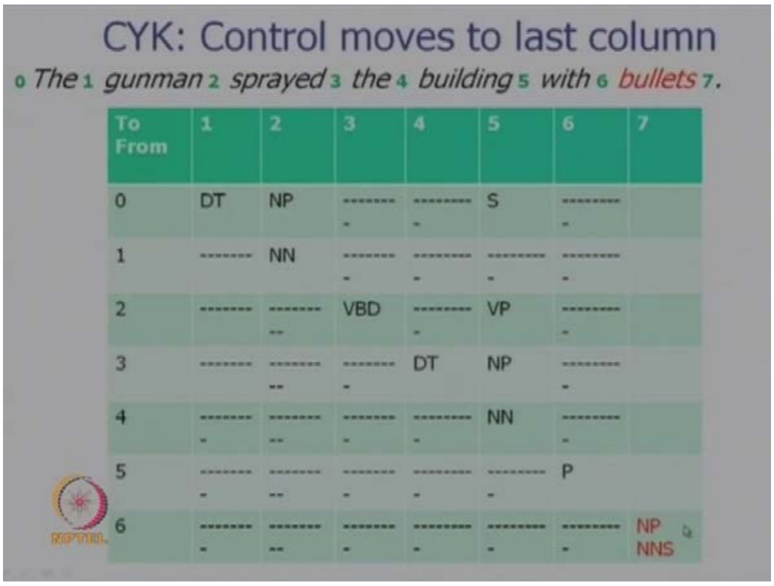
Then between 2 and 5, from 2 and 5 we can find your verb phrase, why sprayed the building, this is the verb phrase, where the verb phrase comes from VBD and NP, this there is a grammatical rule VBD and NP together can produced a VP. So is the contiguity satisfied yes, because you see between 2 and 3, we have a VBD and verb and between 3 and 5, we have a noun phrase VBD NP together can produce a VP, which is between 2 and 5.

(Refer Slide Time: 10:37)



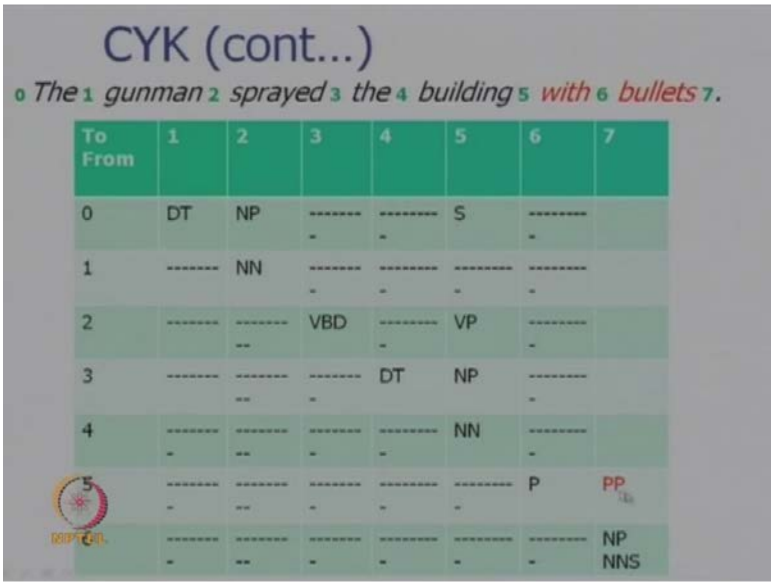
Next, we have a P preposition between 5 and 6, and now nothing can no phrase can end in a preposition, with so that why we find that from 5 to 6, there is on the column from 4 to 6 on this column, there is no phrase on top of P, because nothing can end in a preposition.

(Refer Slide Time: 11:00)



Next, we see that a NNS is found with bullets between 6 and 7.

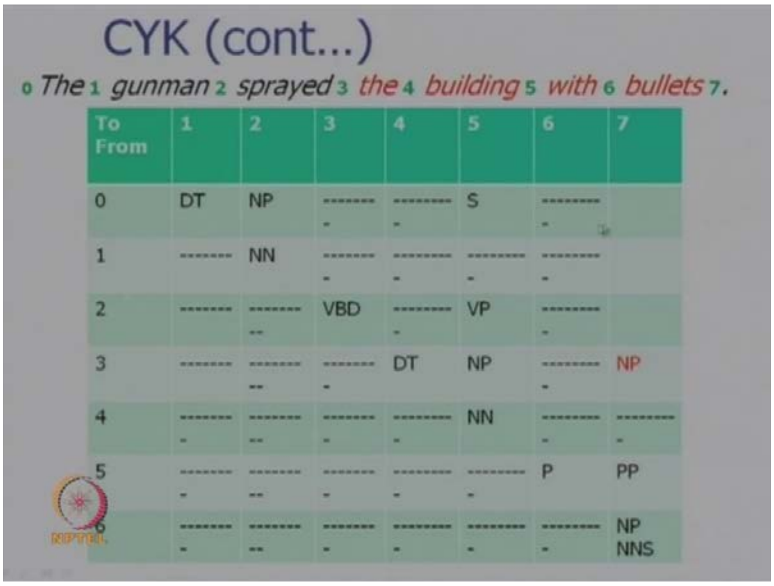
(Refer Slide Time: 11:08)



To From	1	2	3	4	5	6	7
0	DT	NP	--------	--------	S	--------	
1	--------	NN	--------	--------	--------	--------	
2	--------	--------	VBD	--------	VP	--------	
3	--------	--------	--------	DT	NP	--------	
4	--------	--------	--------	--------	NN	--------	
5	--------	--------	--------	--------	--------	P	PP
6	--------	--------	--------	--------	--------	--------	NP NNS

So, these produce a preposition phrase with P and NP NNS nothing but NP so there is a little bit of violation of Chomsky Normal Form here or rather, we have converted the rules into Chomsky Normal Form. NP was going to NNS, we are replaced NNS by NP and thereby converted all the rules in a Chomsky Normal Form. So, between 5 and 7 we have the preposition phrase.

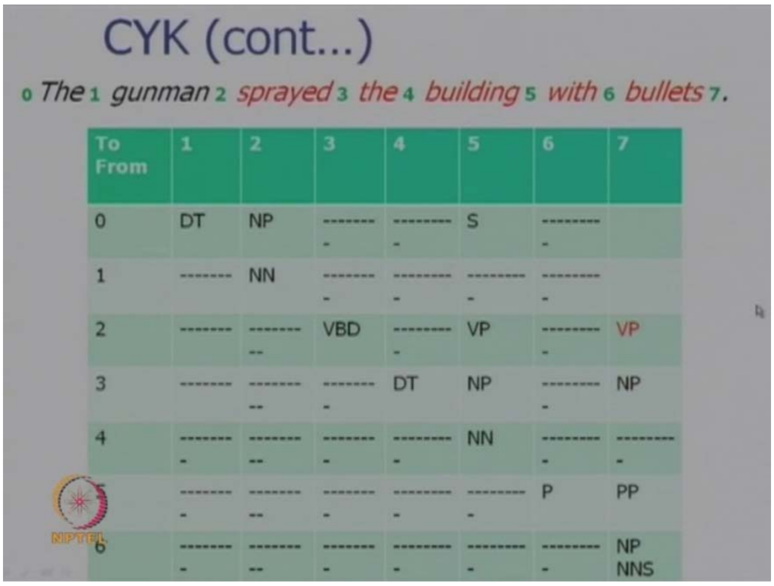
(Refer Slide Time: 11:33)



To From	1	2	3	4	5	6	7
0	DT	NP	--------	--------	S	--------	
1	--------	NN	--------	--------	--------	--------	
2	--------	--------	VBD	--------	VP	--------	
3	--------	--------	--------	DT	NP	--------	NP
4	--------	--------	--------	--------	NN	--------	--------
5	--------	--------	--------	--------	--------	P	PP
6	--------	--------	--------	--------	--------	--------	NP NNS

Then between 3 and 7, we have a noun phrase, which is the building with bullets.

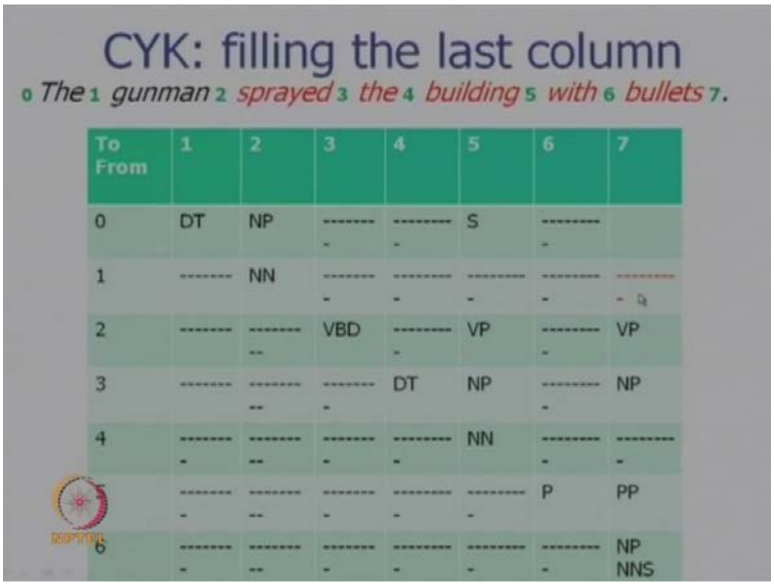
(Refer Slide Time: 11:41)



To From	1	2	3	4	5	6	7
0	DT	NP	--------	--------	S	--------	
1	--------	NN	--------	--------	--------	--------	
2	--------	--------	VBD	--------	VP	--------	VP
3	--------	--------	--------	DT	NP	--------	NP
4	--------	--------	--------	--------	NN	--------	--------
5	--------	--------	--------	--------	--------	P	PP
6	--------	--------	--------	--------	--------	--------	NP NNS

Then between 2 and 7, we have a verb phrase, which is consisting of a verb in past tense and a noun phrase.

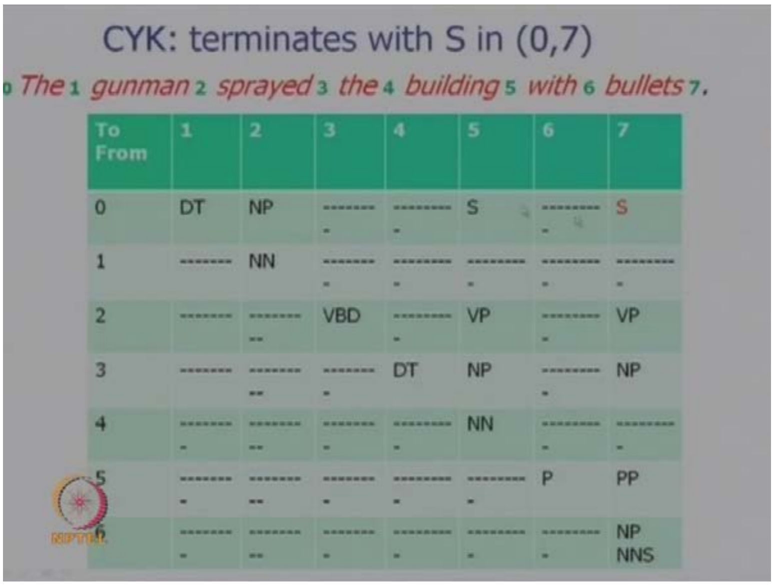
(Refer Slide Time: 11:51)



To From	1	2	3	4	5	6	7
0	DT	NP	--------	--------	S	--------	
1	--------	NN	--------	--------	--------	--------	--------
2	--------	--------	VBD	--------	VP	--------	VP
3	--------	--------	--------	DT	NP	--------	NP
4	--------	--------	--------	--------	NN	--------	--------
5	--------	--------	--------	--------	--------	P	PP
6	--------	--------	--------	--------	--------	--------	NP NNS

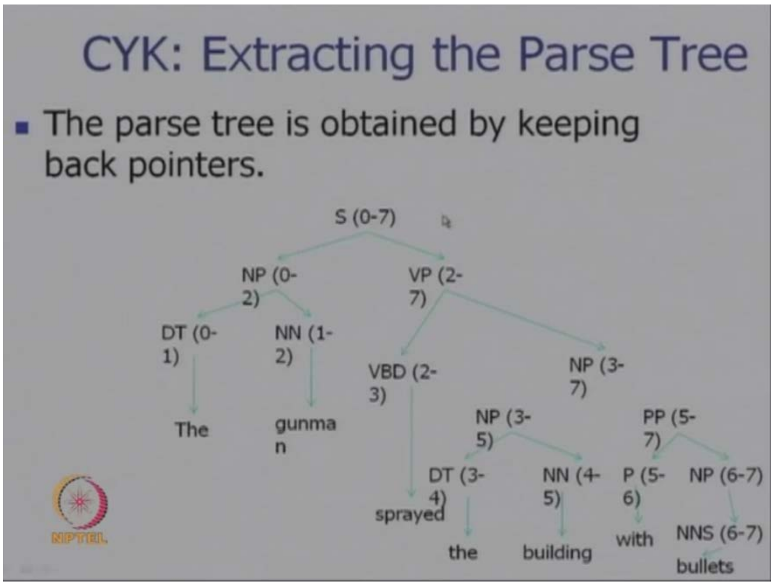
Then there is nothing between 1 and 7, because we do not have any phrase, which begins with a noun, noun phrase does not begin with a noun in singular form.

(Refer Slide Time: 12:06)



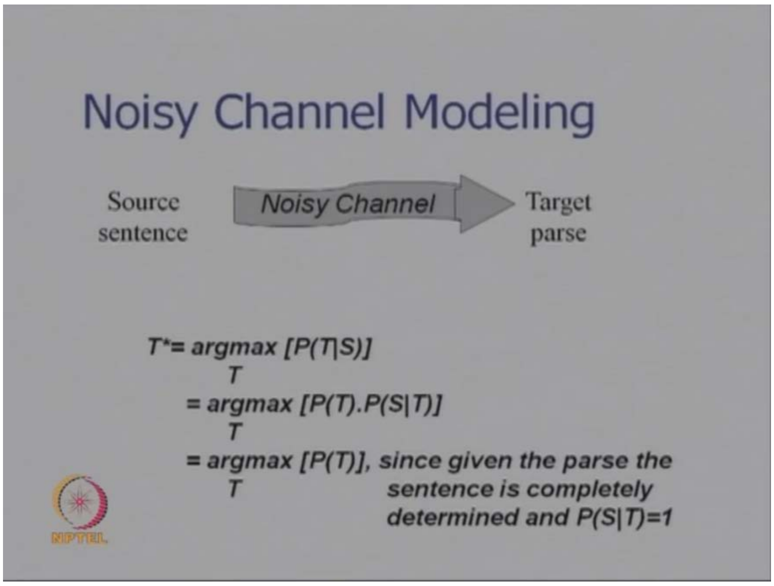
And finally, we have the symbol S, between 0 and 7 the whole thing is resolve then to S. So, this is how the CYK algorithm words is a very allegiant algorithm, and relies on the fact that the grammatical rules are in Chomsky Normal Form the, on the right and side you have, we have exactly to non-terminals or a single terminal. The single terminal takes care of a feeling out the cells with non-terminals, attached to all single terminal and columns and rows, can be combined keeping the contiguity information to produce phrases, these are CYK works.

(Refer Slide Time: 12:51)



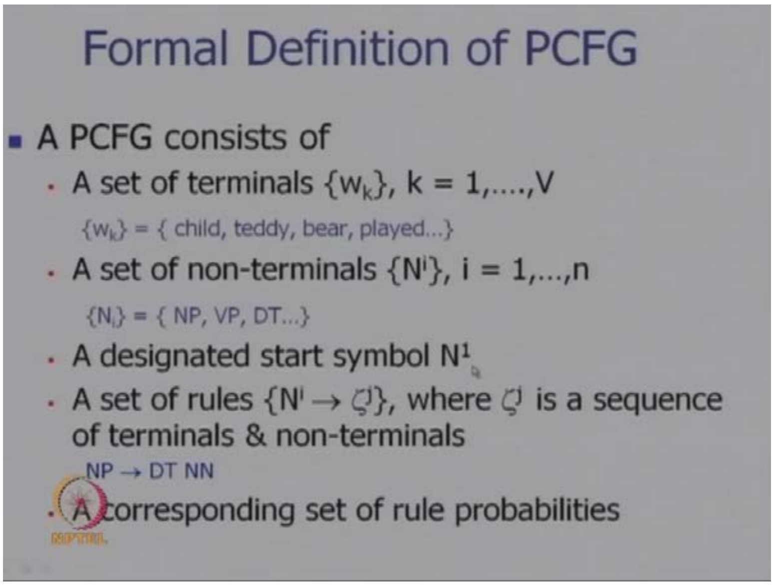
The parse tree can be obtained by keeping the back pointers so for example, NNS is between 6 and 7. So, bullets is resolve to NNS, which again is NP, this is a Non-Chomsky Normal Form construct but it is there for the purpose of understanding between 6 and 7, we have noun phrase then between 5 and 6 there was a PP between 5 and 6 there was a preposition. So, these 2 can be combined 5 6 and 6 7 can be combined to produce a preposition phrase between 5 and 7. And then a similarly, building and determiner is combined to produce the noun phrase, noun phrase and preposition phrase combine to produce another noun phrase between 3 and 7. So, this is the way the parse tree is obtained by keeping a back pointer, which makes use of the table, which is just now completed.

(Refer Slide Time: 13:51)



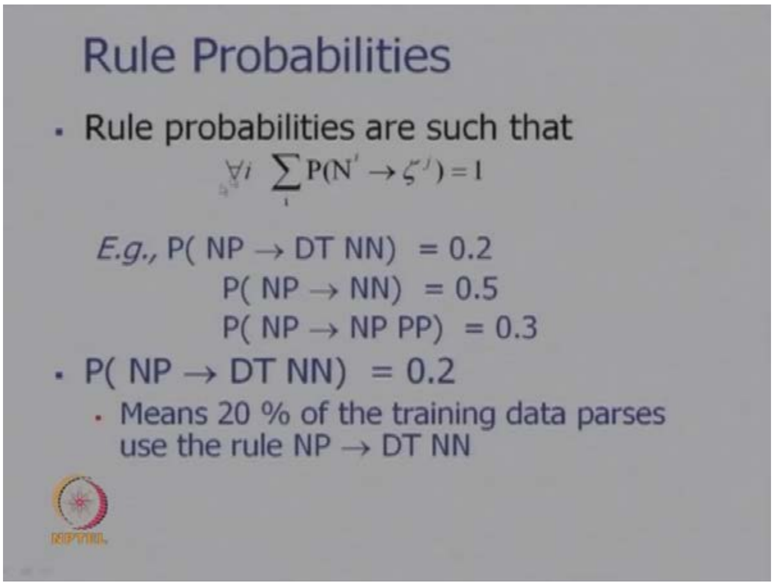
Now, this same CYK algorithm is now used, to obtained the parse tree of a sentence in probabilistic frame work also will see how but before that let us go back to probabilistic parsing with it is definitions and terminology. So, we have already say that the best possibility is the one, which maximizes the probability of the parse tree given the sentence, and in the arg max is over all possibilities, when we used the base theorem the get PT into PS given T and PS given T is 1. Because given the tree, parse tree sentence is completing the determining probability is 1.0 and finally, we have the probability PT coming here.

(Refer Slide Time: 14:36)



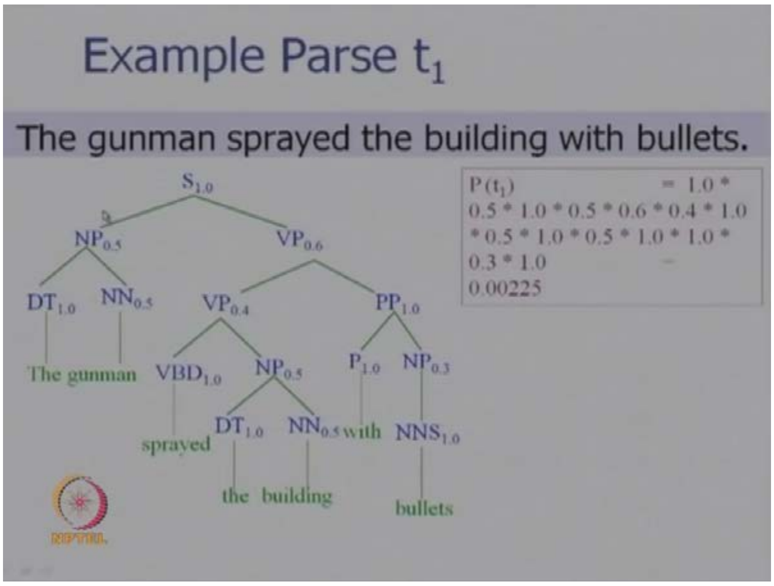
So, now the probabilistic context free grammar as discussed, in the last lecture consist of set of a terminals and non-terminals, start symbol which is spatial status symbol is set of rules only new thing is the rules, have probability values associated with them.

(Refer Slide Time: 14:52)



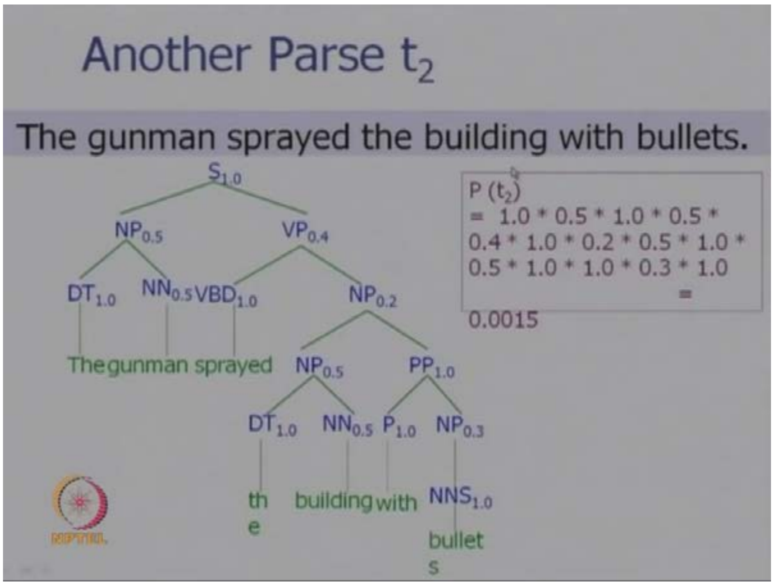
The rule probability are such that, the probability of the some of the probabilities of all those rules, which have the same left hand side should be equal to 1.

(Refer Slide Time: 15:01)



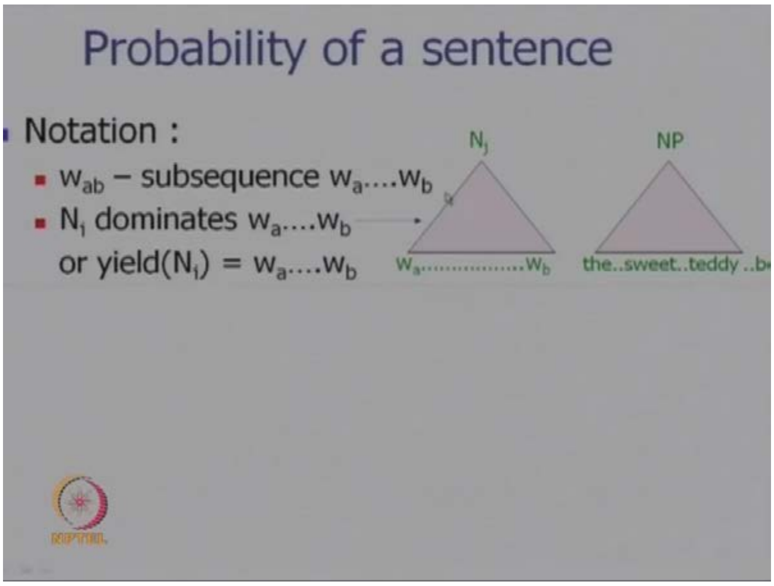
Now, this is the grammar and then these grammar produces for a sentence the gunman sprayed the building with bullets at 2 trees, in 1 tree the preposition phrase is attached to verb.

(Refer Slide Time: 15:15)



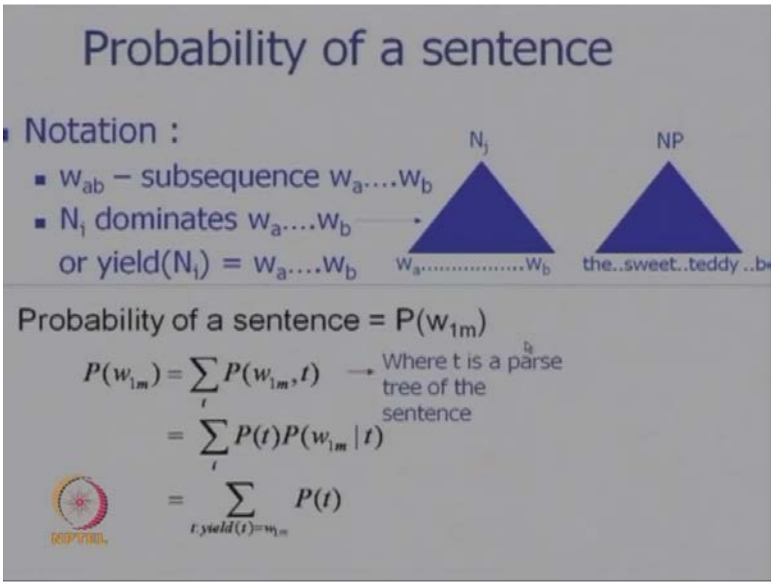
So bullets were used to sprayed the building or in the other case, there is this unlikely parse for with bullets being attached to building, which means it is a building, which contains bullets. These interpretation is unlikely but it is possible to have these kind of attachment, an example of that would be, I the gunman is sprayed the building, with we man in it. So that would mean the building has, we man in it, with the man in it is attached building is say it is that building which has we man in it. Now, in this between these 2 parses seems the first parse is semantically more plausible, we find that it is probability value is higher than the other parse tree, which is lesser plausible, and the probability values here is 0.0015 probable parse tree. And the more plausible parse tree has probability values 0.00225, which is more and how is this probability calculated is an our next concern.

(Refer Slide Time: 16:23)



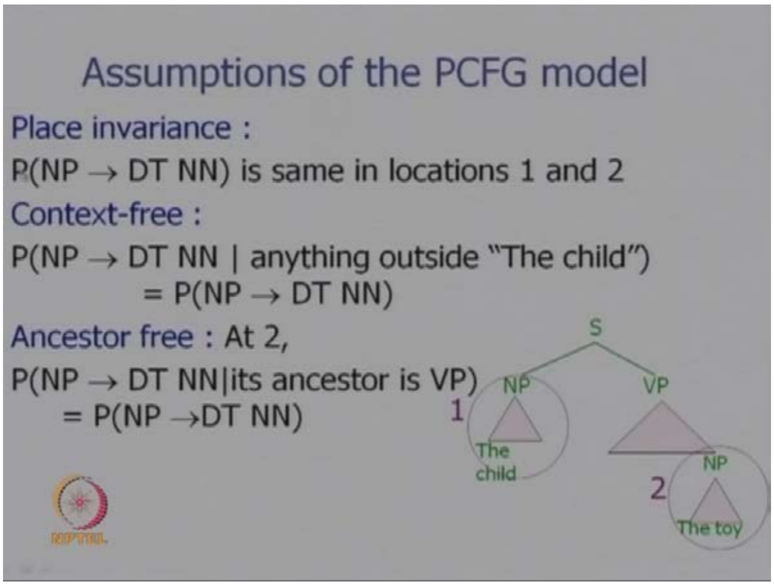
We have to understand these through the discussion of the meaning of the probability of a sentence, then notation is shown here W ab is a subsequence, W a W a plus 1 W a plus to up to W b. Now, these picture shows that Nj is the root for the sub parse tree, for the sequence of words from W a to W b. Now, since Nj generates the W a to W b you we say Nj dominates W a to W b. So, in computer sciences parlance, we would say Nj generates W a to W b and in linguistic problems, we would say Nj dominates W a to W b are in the parsing parlance and the yield of Ni should be N I, is the sequence Wa to W b for example, this sweet teddy bear is the yield of the noun phrase NP or you say NP generates the sweet teddy bear or NP dominates the sweet teddy bear.

(Refer Slide Time: 17:31)



The probability of a sentence, how do you compute is probability of a sentence with words going from W 1 2 W m is nothing but probability W1 to W m comma t, these marginalization overall possible parse trees of W 1 to W m, the sequence W 1 to W m. Now, these joint probability is broken up, which is Pt into P w m t, this is nothing but some sigma P t. So, the probability of the sentence is nothing but the probability of all those parse trees is who is yield in the sentences itself. Now, P W1 to m given t is nothing but the probability of the sentence, given the parse tree this is equal to 1.

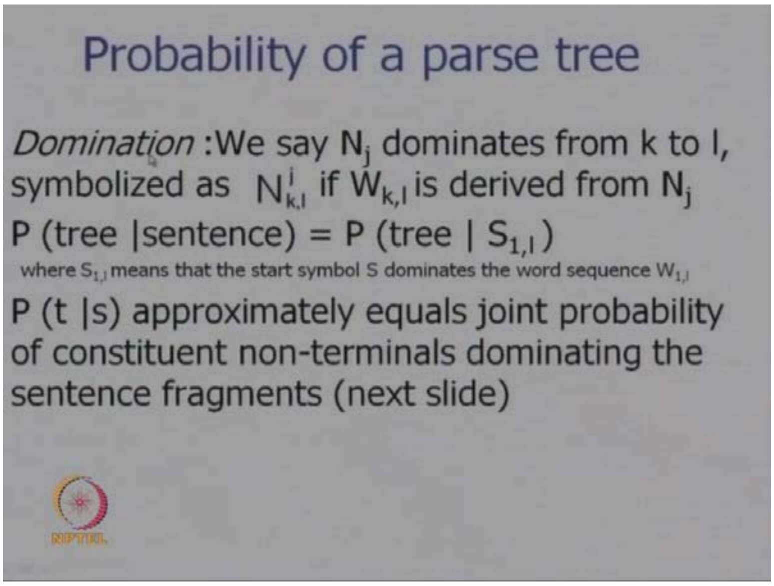
(Refer Slide Time: 18:19)



So, now the probability of contents tree grammar model as 3 assumptions, fundamental assumptions, place invariance, probability of noun phrase going to DT NN if same in locations 1 and 2. So, wherever this rule appears NP going to NDT NN the probability is the same, it does not depend on the place, context freeness says that probability of NP going to DT NN, given anything outside the child is probability P NP goes to DT NN.

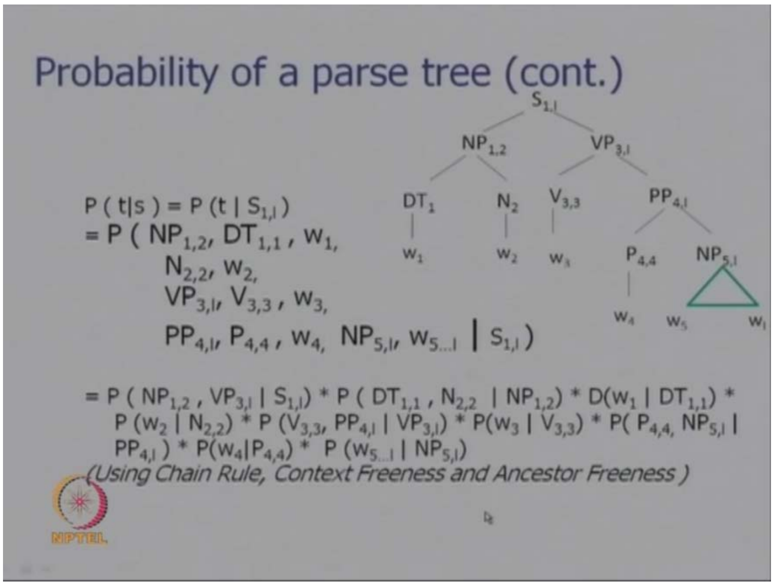
So, this means that the probability of these rule is independent of anything, that happens outside these sub tree NP going to DT NN, ancestor freeness is that a probability of NP going to DT NN its ancestor is VP the conditional part does not matter. So, whatever with the ancestor of NP goes to DT NN as the same probability, so these are 3 assumptions of p c f g, which are used for calculating the probability of a sentence, calculate a probability of a tree and so on and so forth.

(Refer Slide Time: 19:31)



So, probability of the parse tree for this we first understand the notion of domination, we say that Nj dominates from k to l symbolized as Nj k l, if W k l is derived from N j are w k l is yield of N j. So, probability of a tree given in a sentence is probability of the tree given that S dominates 1 to l, the start symbol S dominates the word sequence 1 to l, so P t s is approximately equal to the joint probability of constitution a non-terminals dominating the sentence fragments.

(Refer Slide Time: 20:06)



So, let us understood this by taking on this example, here we have the word sequence W 1 to W l W 1 W 2 W 3 W 4 to W 5 to W l. So, we have a noun phrase between 5 to N, this is indicated here, we have a preposition phrase from 4 to l, where between 4 and 4 4 and 4 is slightly different notation, there is a preposition phrase, then the verb preposition phrase obtains verb phrase, and then noun phrase and verb phrase obtains a sentence yes. So, DT N combination gives satisfy to NP, now the probability of the tree given the sentence is probability of t, given that sentence dominates S dominates 1 to l. So, here this probability of the t is even a S 1 to l is nothing but the joint probability of dominations by various non-terminals. So, probability of the parse tree t given S 1 to l is the joint probability of a noun phrase, which dominates the preposition 1 to 2 a determiner a at position 1.

Then the word W 1, then a noun dominating the position 2, then a word W 2 then the a verb phrase dominating a sequence from 3 to l evolve at 3 3 in word W 3 and so on. So, all these are the domination situations, where the word dominates itself, the non-terminal dominates sequences, now what you do is that, we take these joint probability and a apply conditioning on this and then make independents assumptions. So, for example, NP and VP isolated a given S. Now, we isolate DT and N given the NP, we also have to bring in the VP in S but because of the context freeness, assumption this VP has no influence on the probability of these NP going to DT and N.

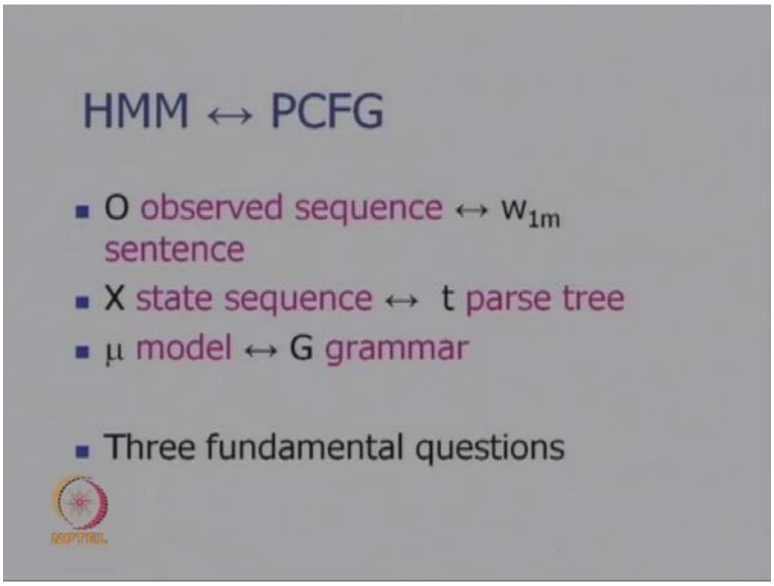
So, I can draw up VP and S here similarly, probability of W 1 given DT is independent of NP here, because of ancestor freeness assumption, and these VP here again, because of context freeness assumption. So, anything that happens outside is separates a material, even the ancestors influences not felt, so that is why we have probability of W1 given DT. So, we use chain the rule, context freeness, and ancestors freeness and thereby obtained the fact, that the probability of tree given the start symbol S is nothing but the product of all possible rule applications, product of the probabilities of the rule applications.

So, this is the main theory behind calculation of the probability of a parse tree, we just detect are note the probability of various dominations, that means a probability of various sub trees, which obtained in a tree and then take the product of for probability but why product because of the fact that, first we convert that tree into the probability of a tree given S to joint probability of all the sub trees is that, that is because you see the domination by S is nothing but the joint domination of non-terminals, in the sub tree. So, these we have to accept because the domination of S is nothing but result of domination of individual non-terminals on different parts of the sentence, that is that and after that, after we obtained the joint probability we applied chain rule. And when we applied chain ruled, we isolate those sub trees and the root of the sub tree, which are dictated or which are entailed by the grammatical rules.

And then we applied chain rule, and when which applied the chain rule, is all those variables which are outside the sub tree or in the ancestors adopt, because of ancestors freeness assumption, and context freeness assumption and invariance of course, soles because of wherever a particular sub tree, which same left and side, and right and side of obtains the probability values is same. So, this is a very instructive is slide, which shows what is it, what is the theory behind the calculation of probability of the parse tree. So, we proceed further, and if you is this theory then we can compute, the probability of this pass tree, we can say why it is coming out to be 0.0015 S goes NP VP as probability 1.0.

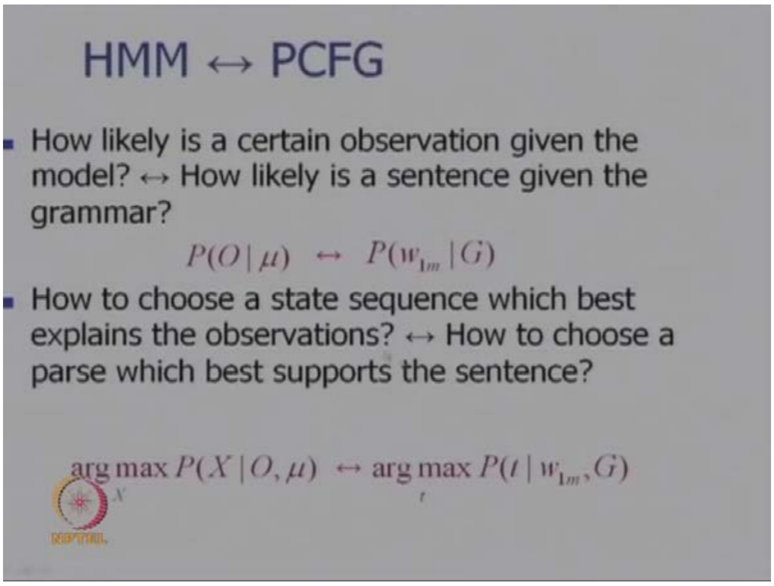
So, this is 1.0 NP goes to DT NN probability 0.5, so this is taken here DT goes to the probability 1.0, find NN goes to gunman probability 0.5, VP goes to VBD NP 0.4 is the probability here is 0.4, VBD goes to sprayed 1.0 here it is, NP goes to NP PP with probability 0.2 here it is, NP goes to DT NN probability 0.5 here, PP goes to PNP with probability 1.0 yes 1.0 p goes to with that is 1.0 here, then NP goes to NNS with probability 0.3 here, NNS goes to bullet with probability 1.0 here and when, we multiply this values we get 0.0015.

(Refer Slide Time: 26:29)



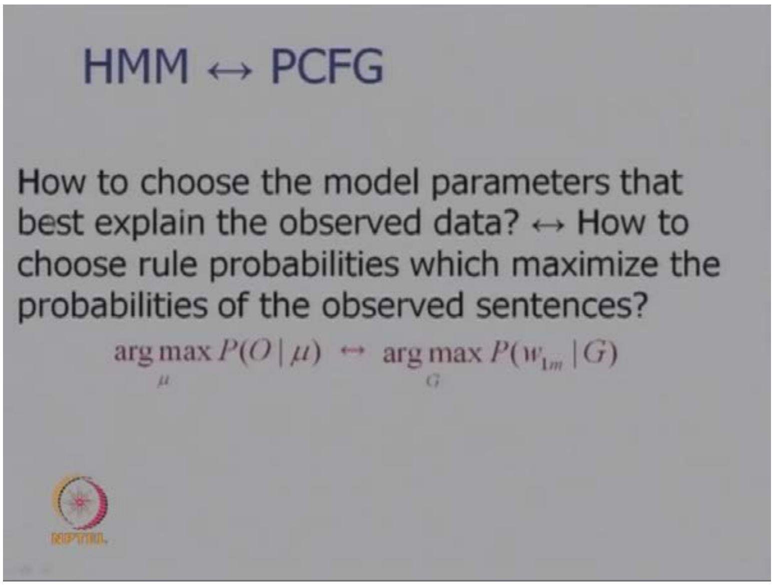
Now, we have been remarked before that Hidden Markov of Model and PCFG have lot of correspondence Hidden Markov of Model is on the linear sequence of words, probabilistic context free grammar on the tree. So, we apply the same theory very similar theory, for tree as supposed to a sequence now in HMM, we have 3 important algorithms and situations. So, we see here in HMM we have the observed sequence, the corresponding thing in PCFG is the sentence, X is the state sequence in HMM t the parse trees is the corresponding entity, and PCFG mu is the model for the HMM, G is the grammar for the corresponding entity for PCFG.

(Refer Slide Time: 27:21)



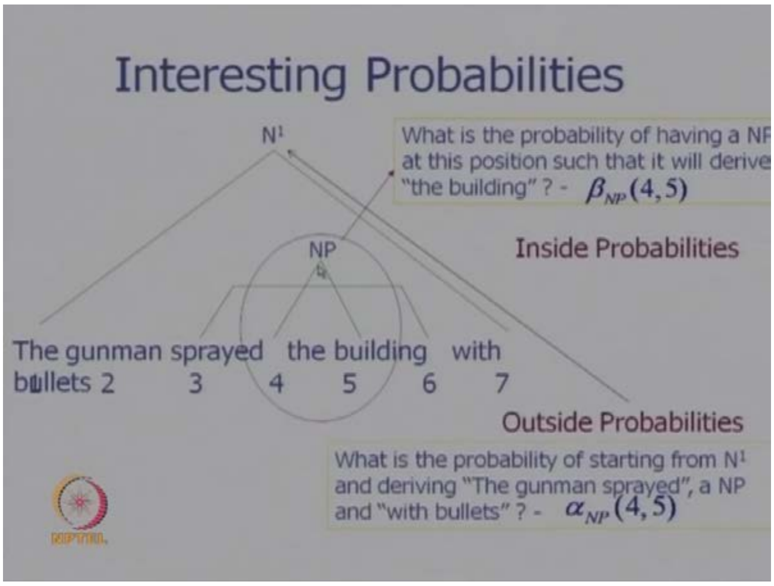
Three fundamental questions has addressed in HMM, how likely is a certain observation given the model, how likely is the sentence given the grammar is the corresponding question. So, P O given the model is equal to P sentence, given the grammar how to choose a state sequence, which best explain the observations, this corresponds to the question of how to choose a parse tree, which best supports the sentence. The corresponding expressions are shown here in Hidden Markov of Model, what is the probability of state sequence X, given the observation sequence O similarly, arg max over all possible parse trees, when the word sequence, here we have arg max over all possible state sequence, sequences given the observation sequence O.

(Refer Slide Time: 28:07)



How to choose the model parameters that best explain the observed data? This equation in HMM, how to choose rule probability which maximize the probabilities of the observed sentences? This is the PCFG question.

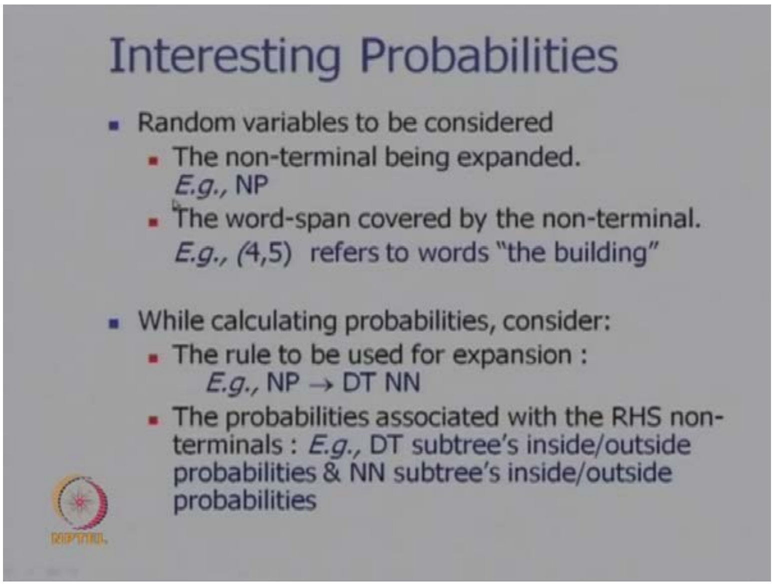
(Refer Slide Time: 28:18)



So, now we take up this question up, how we construct the probabilistic parse tree, the following interesting probabilities, if you look at the slide here, we take this sentence the gunman sprayed, the building with bullets.

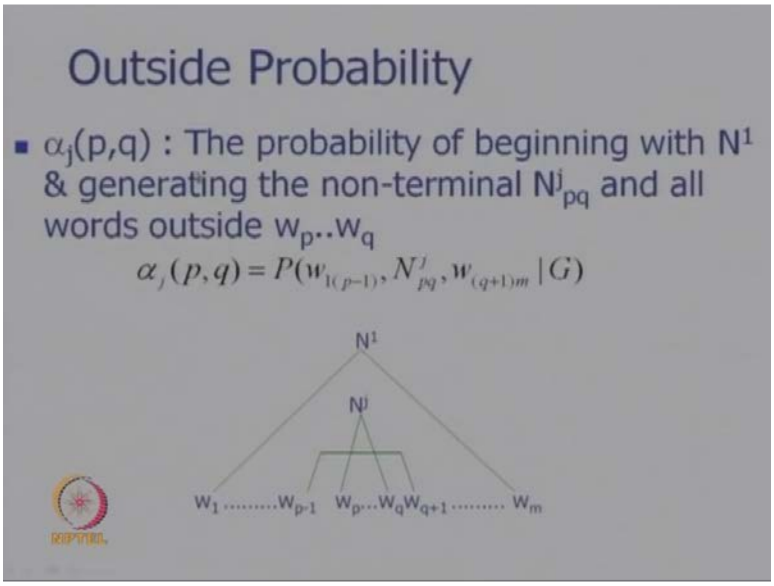
So, the building is the noun phrase, sprayed the building is the verb phrase, the gunman sprayed the building with bullets is generated by the start symbol N1. So, what is the probability of having a noun phrase at this position, such that it will derive the building, this is known as the inside probability divided by notation beta. So, what is the probability of finding a noun phrase, between the position 4 and 5 is beta? And what is the probability of starting from N1 and deriving the gunman sprayed in noun phrase and with bullets? So is known as the outside probability the bullets is having it is own parse sub tree root an NP. So, the alpha probability of this whole sequence the gunman sprayed, then a non-terminal NP appearing.

(Refer Slide Time: 29:29)



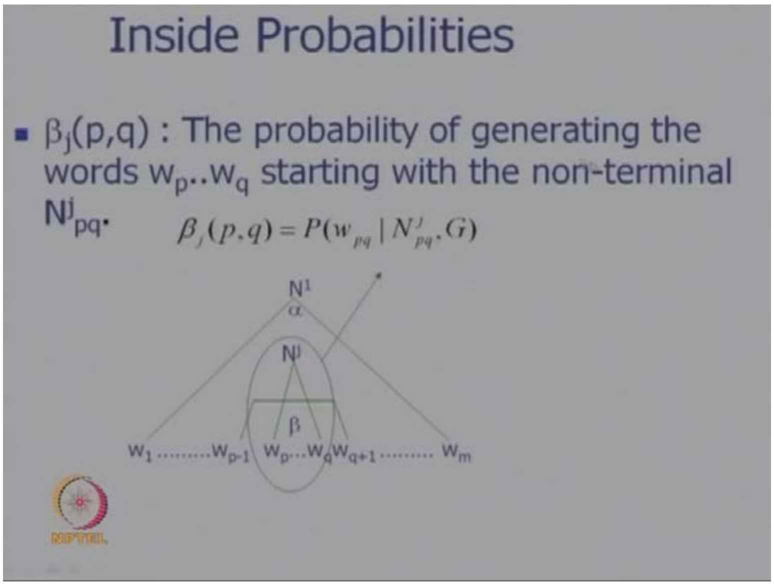
And then a sequence with bullets coming after that the interesting probabilities are as follows, the random variables are to be considered are the non-terminal being expanded the word span covered by the non-terminal, an while calculating probabilities, we consider the rule to be use for expansion.

(Refer Slide Time: 29:41)



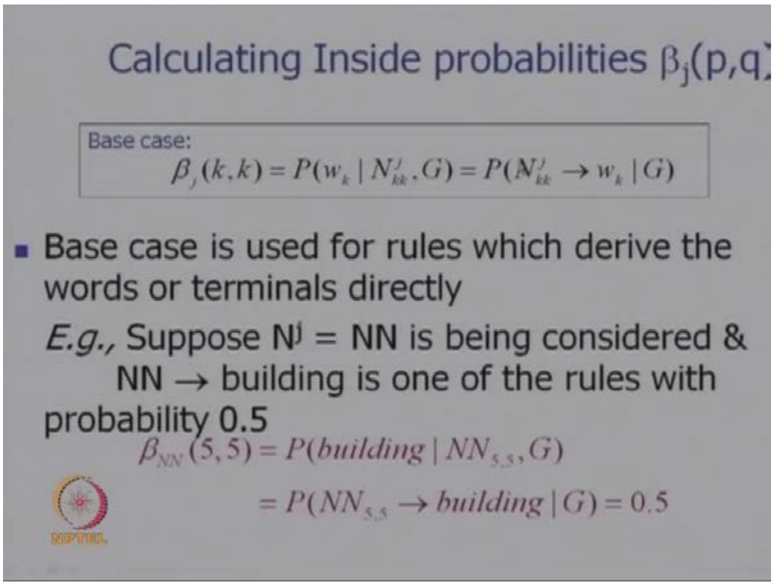
Outside probability is a better illustrated in this slide, the probability of beginning with N1, and generating the non-terminal N j p q and all words outside W p to W q. So, alpha j p q is denoted this way, the joint probability of the word from 1 to p minus 1, then Nj dominating the subsequence p to q and the sequence W q plus 1 to m.

(Refer Slide Time: 30:09)



Beta j p q is probability of generating the words W p to W q starting with the non-terminal N j p q, so beta j p q is nothing but probability of W p q given that N j dominates the sequence subsequence W p to W q.

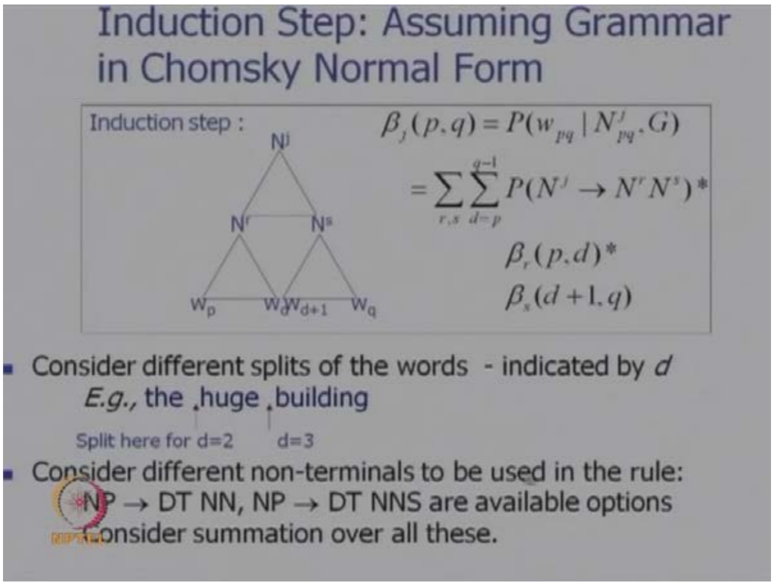
(Refer Slide Time: 30:28)



Now, calculating the inside probabilities is again and dynamic programming based a parsing algorithm. So, we have already seen the CYK algorithm for the deterministic situation the grammar is in Chomsky Normal Form, and each non-terminals gives rise to 2 non-terminals side-by-side or a produces a terminal only. Now, in the dynamic programming based algorithm, a table is maintained a matrix is maintained, starting from position number 0 to the last position, and in the column, we have position starting from 1 to last parse 1 position, and then between any 2 position, we try to find a non-terminal given the word, and then be try to combine this non-terminals to form a bigger phrase.

So, beta j k k the base case, the notation is slightly different here now, this is nothing but the probability of word k, given that N j dominates these k position. So, probability of N j k k going to W k the base case is use for rules, which derived the words are terminal directly to suppose N j equal to NN is being considered, and NN goes to building is one of the rules with probability is 0.5, then beta NN or 5.5 probability of building, with NN in noun dominating position 5 so probability of NN going to building given G the probability is 0.5.

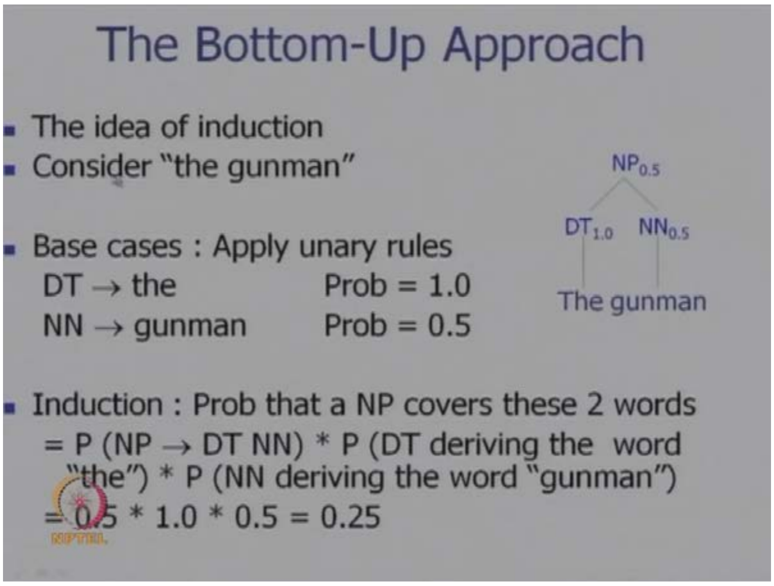
(Refer Slide Time: 32:02)



The induction step is most interesting step, we assume grammar in Chomsky Normal Form and beta j p q is probability of word sequence is p to q given Ng dominates it. Now, this is nothing but a marginalization case we vary the position d and which will also vary the non-terminals, which dominate the sequence from d to q and from p to d from d plus 1 to q, from p to d. So, we have 2 non-terminals Nr and Ns, now when r n s varies and d varies from p to q minus 1, then we get this expression here by applying various kinds of independents assumptions, and ancestor freeness and context freeness and joint probability, to this you can say a nothing but domination of N r and N s, so this whole expression can be written at as probability of W N r N s given N j.

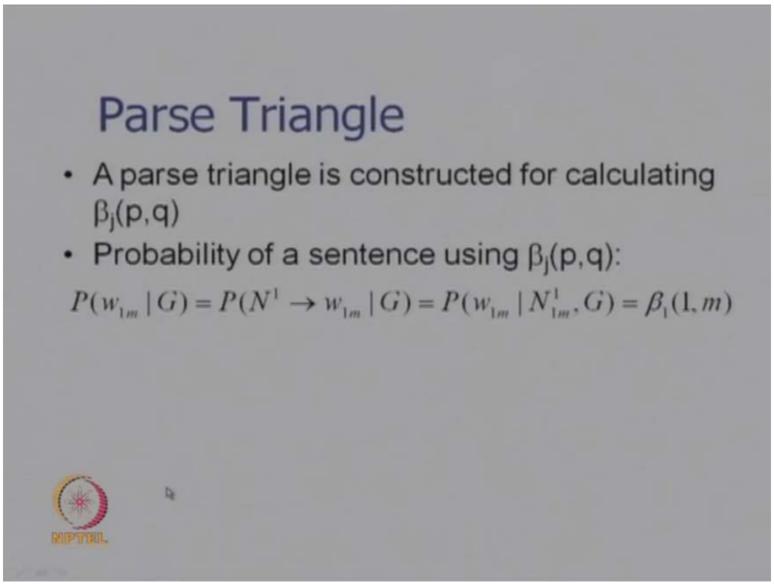
Now, N r N s can be isolated after that, when we have n r going from p to d and then N s going from d plus 1 to q, we have these to beta probabilities beta r p to d and beta 8 as d plus to q, and probability of a n j going to N r N s. So, this is a simply the probability of the rule, and the beta values for p to d and d plus 1 to q have to be reconcilably found it. Now, we can understand the inside algorithm, inside probability algorithm by taking these simple example of a small phrase, the huge building. So, we split here for the d is equal to 2 and d equal 3 and so on.

(Refer Slide Time: 34:05)



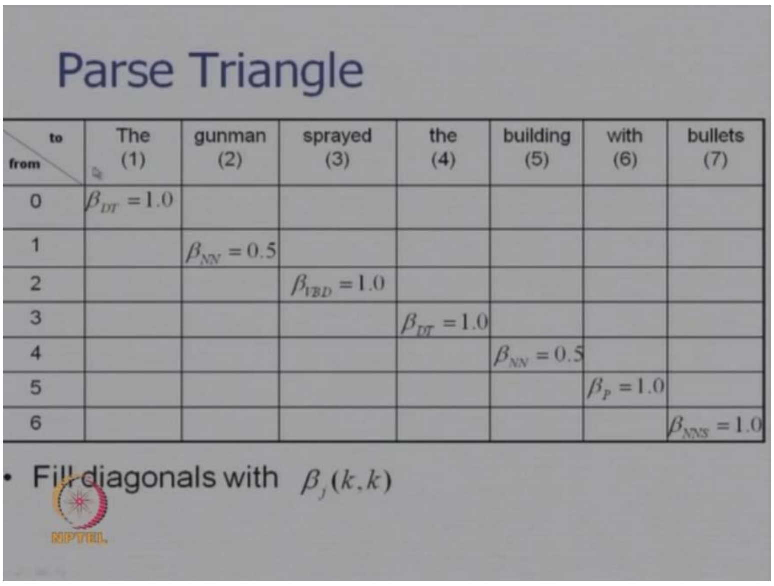
And we consider different non-terminals which can be used in the rules, and the algorithm is a bottom-up approach, the idea of induction is consider the gunman we apply unary rules DT goes to the NN goes to gunman, probability is 1.0, probability is 0.5, so DT goes to the 1.0 and NN goes to the 0.5, NP goes to DT NN with probability 0.5. So, induction is that probability that in noun phrase covers these 2 words P NP goes to the DT NN and P DT deriving the word, the P NN deriving the word gunman is equal to 0.5 into 1.0 into 0.5 is equal to 0.25.

(Refer Slide Time: 34:43)



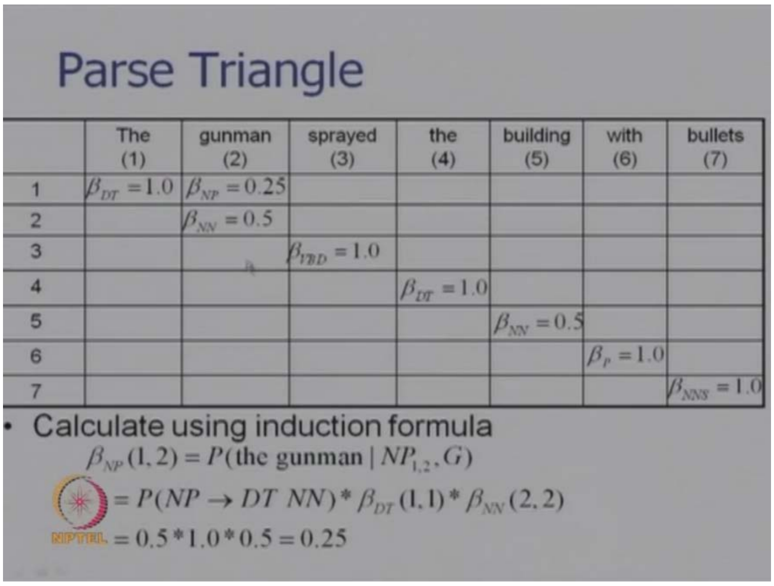
Now, just like CYK algorithms a parse tree, a parse triangle is a data structure, which help this operation very well, the parse triangle is constructed for calculating beta j p q, the probability of a sentence can be found out, again from the beta probabilities. So, probability of w 1 to m given G is nothing that probability of N 1 dominating w 1 to m, and this is nothing but beta 1 1 to m that means N1 dominating the sequence of words from 1 to m.

(Refer Slide Time: 35:18)



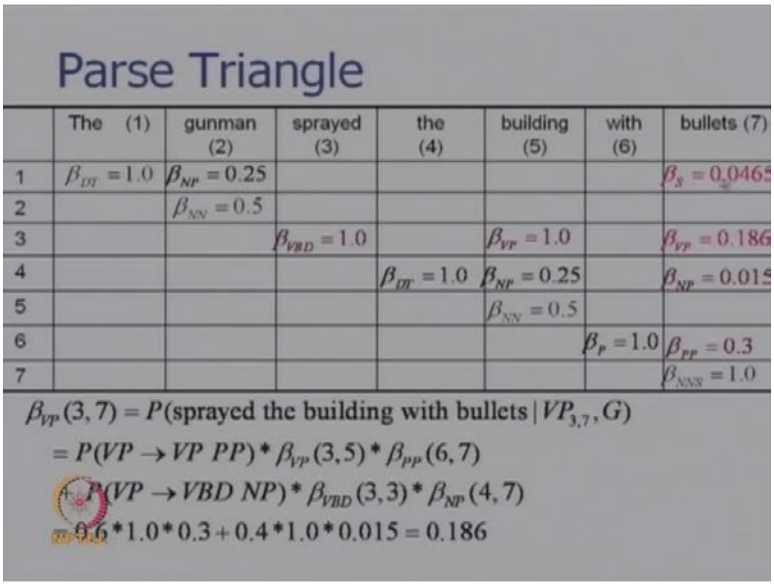
So, just like CYK algorithm, we first feel the diagonals with beta j k k values, here is the first location between 0 and 1. So, the reason though there is so there’s we find beta DT is equal to 1.0, because the probability of DT going to the, is equal to 1.0, then at gunman we find a noun. So, the probability of that is 0.5 we record this beta NN, between 1 and 2, then beta VBD is sprayed probability of the word going to sprayed is 1 beta DT is 1.0, beta NN is 0.5 like before beta p is 1.0 preposition goes to with the probability 1.0, beta NNS equal to 1.0 that is because NNS going to plural word, the probability is 1.0 first this diagonals are filled.

(Refer Slide Time: 36:25)



Now, we try to see, how he could obtained the probabilities of the phrases. So, we see here NN and DT can combined to produce a noun phrase, and we see how beta NP 1 to 2 is obtained, probability of the gunman, dominated given that there is an NP between 1 and 2. So, this is probability of NP going to the DT NN which is 0.5, and then probability of DT going to the which is 1.0, the probability of NN going to 1 1 which is 0.5, so the probability comes out to be 0.25.

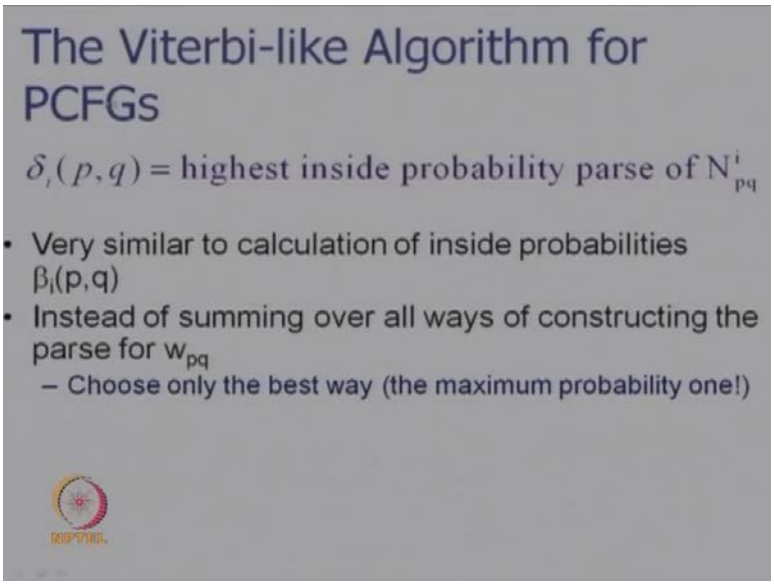
(Refer Slide Time: 37:09)



This is the way, the parse tree is produced, we let us see a few more entries, let us see beta VP between 3 and 7, that is a VP, now here there is an ambiguity consideration Sprayed the building with bullets, with bullets should be attached to spray or building, so since it is a probabilistic framework, we have to sum up the probabilities of both the parse trees, and this can be obtained by beta VP 3.7 3 to 7. So, probability of sprayed the building with bullets, given that there is a verb phrase between 3 and 7 here, there should be a verb phrase. And how do you obtain this probability of verb phrase is going to be VP PP, this is preposition phrase attachment to verb into beta VP 3 to 5, beta PP 6 to 7 plus probability of VP going to VBD NP.

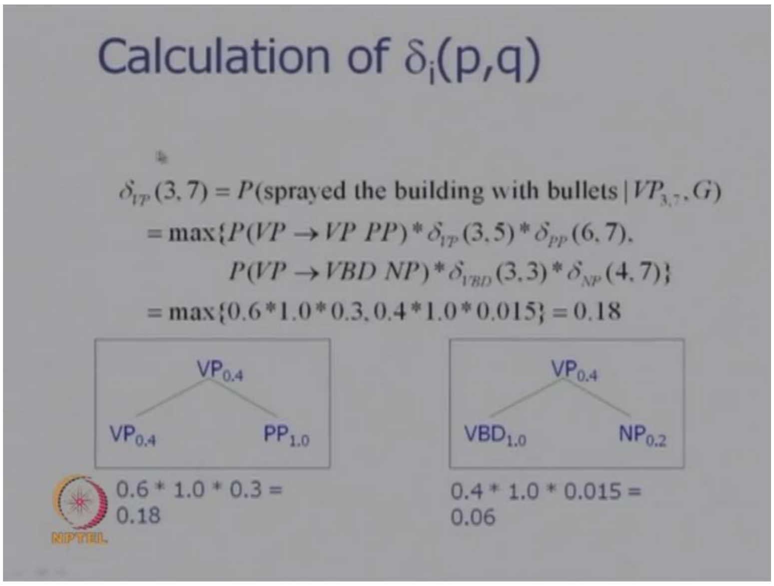
This is noun attachment of the preposition, beta VBD 3.3 and beta NP 4 3 to 3 and beta NP 4 to 7, so 0.6 1.0 0.3 plus 0.4 1.0 0.015, the beta value for VP here 3 to 7 comes out to be 0.186. So, this is the way 1 goes on filling the entries in the cell and finally, the probability of the tree comes out, from the beta value at this location, which is the location for S. So, this is nothing but CYK algorithm as you can see is nothing but a CYK algorithm which is dynamic programming, based only thing is that the probability values are used, and they are combined together, obeying the theory which is used for computing the probability of the whole tree. So, given a sentence what is the probability of the sentence is nothing but some of the probability values of all the trees, which can come for this sentence so between 3 and 7 we can have 2 parse trees that is why 2 probability values expressions are taken in sometime.

(Refer Slide Time: 39:18)



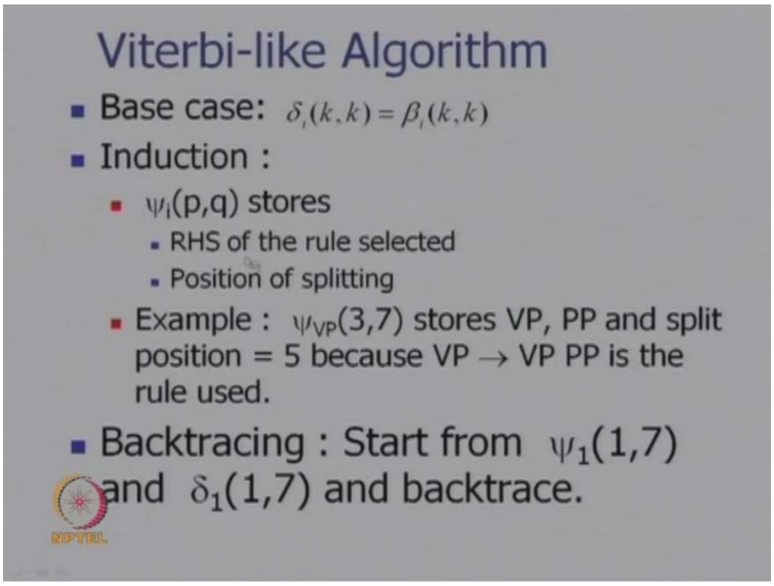
So, now there is Viterbi like Algorithm for probabilistic context free grammars, it makes use of this delta probability, and this is the highest inside probability parts of N i p q. This is very similar to the Viterbi Algorithm for the best state sequence, for an observation sequence, linear observation sequence.

(Refer Slide Time: 39:39)



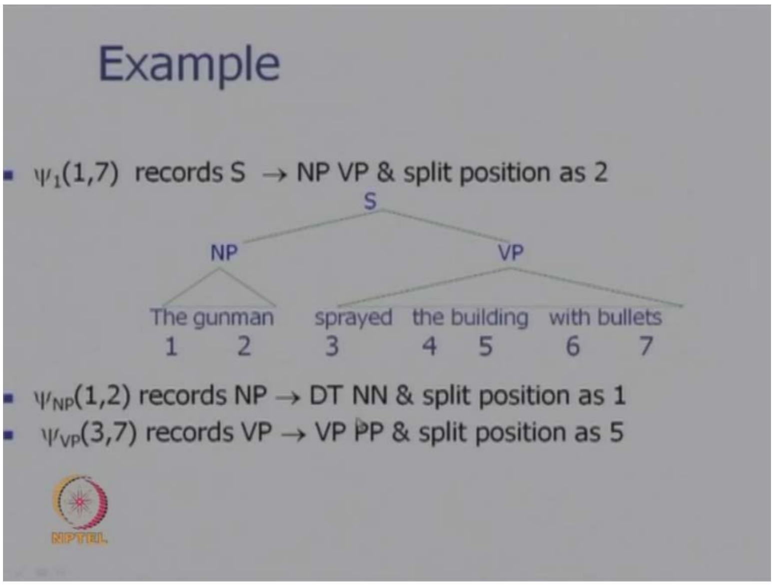
And the same dynamic programming, like approach is used for finding the best possible parse tree. So, we see here again in the with respect to the example, probability of sprayed the building with bullets, even that there is a verb phrase, between 3 and 7 is maximum of the 2 probability values, what 2 probability values, 1 is PP attachment to verb, that is PP attachment to the noun. So, here the probability comes out to be 0.18, the probability will be comes out be 0.06, so will add up to these sub tree, as the tree for the verb phrase. And this is obtained by keeping track of delta, which is the maximum of the, of all the parse trees.

(Refer Slide Time: 40:23)



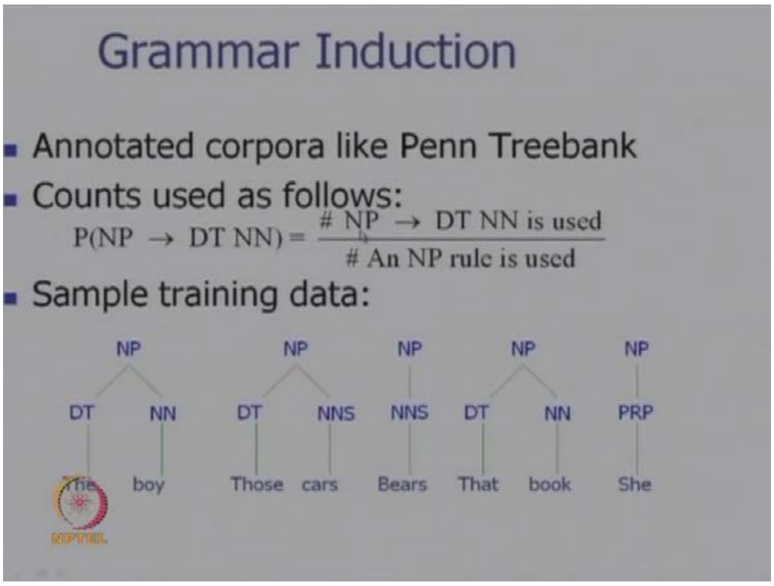
So, Viterbi-like algorithm with backtracing and so on.

(Refer Slide Time: 40:27)



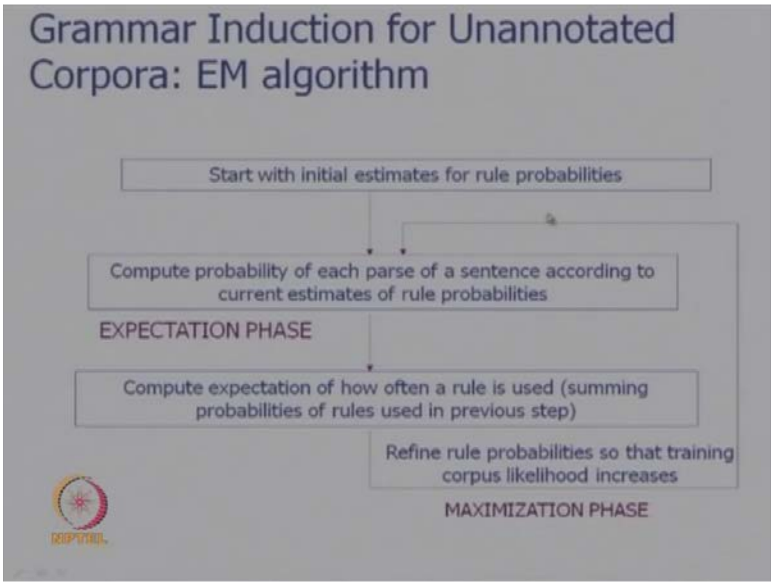
This is an efficient algorithm, to find out the probability of the parse tree.

(Refer Slide Time: 40:32)



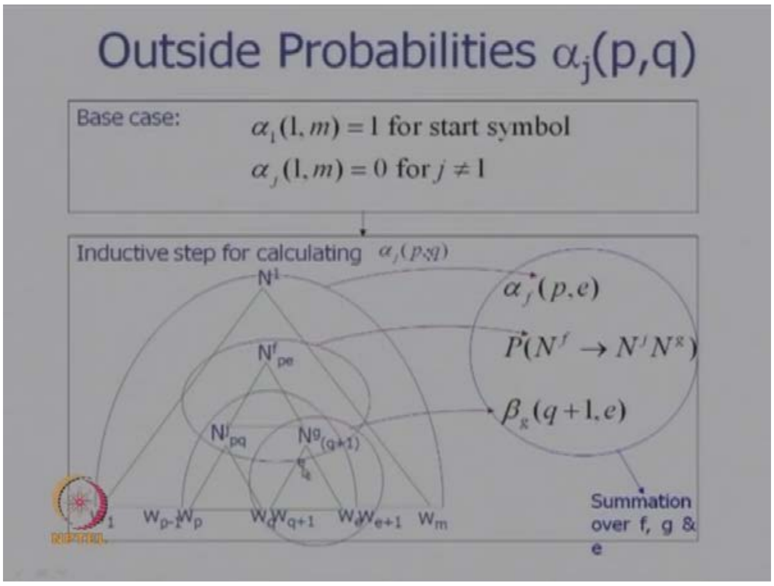
Grammar Induction also can be done on the parse trees, and we use the annotated corpora, like the pen tree bank.

(Refer Slide Time: 40:40)



And the algorithm proceeds in terms of an expectation maximization, deterioration between expectation and maximization. So, in the expectation phrase, we start with initial estimates of the rule probabilities, we compute the probability of each parse of a sentence according to the current estimates of the probability. We compute the expectation of how often a rule is used, summing the probabilities of rule used in previous step, we refine the rule probabilities. So, that the training corpus likelihood increases, this is the maximization step.

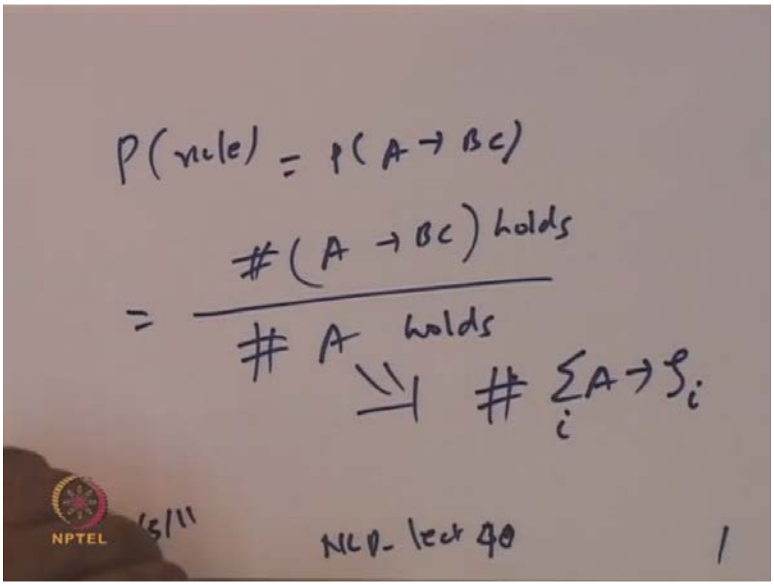
(Refer Slide Time: 41:17)



So, this continuous oscillation between expectation and maximization, gradually leads to the stabilization of the probability values for the rules, outside probabilities also used and probability of a sentence can again be computed by means of outside probability. So these then finishes the discussion on the probabilistic parses, and what we have done so far is that, we gave a theory of how to use probability in probabilistic parse.

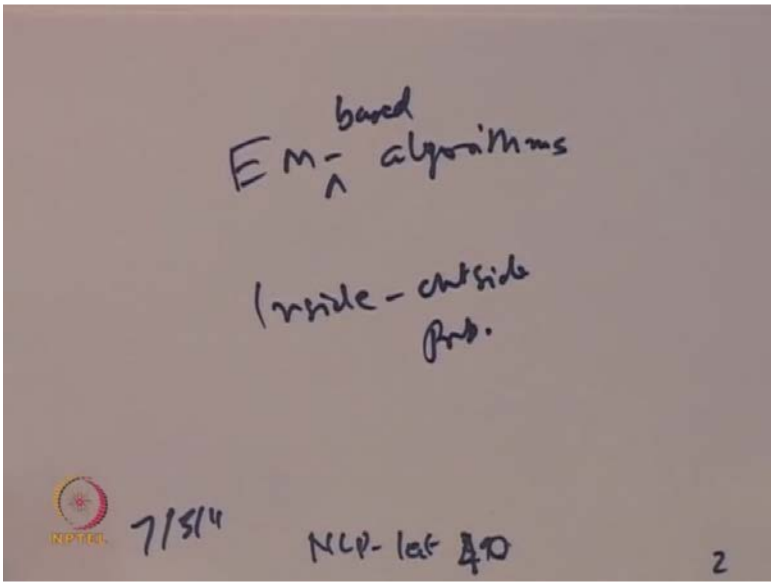
Now, the calculation of this probability values for the rules, obtained from the training corpora. So, the training corpora can be either marked with bracketed structure, which leads to the situation of supervise learning or the training corpora could be without any bracket marking. So, this is a case of unsupervised learning, when we get the rule probabilities, from bracketed training corpus, then we can make use of a frequency-based approach, a simple frequency-based approach. We simply calculate, how many times a rule has been used, compared to other situations, where the same non-terminal of the rule is used but not the same right hand side. So, from this calculation, we obtain the probabilities of the rules.

(Refer Slide Time: 42:52)



So this part I could, write probability of a rule which is nothing but probability of A goes to BC is nothing but number of times A goes to BC holds divided by number of times A holds, so this is nothing but number of times sigma A goes to zeta i overall possible i's, so this is nothing but a frequently is approached probability calculation. So, probabilities of the rules are found this way, however if the, if this calculation is not possible, because we do not know, how many times this rule is applied.

(Refer Slide Time: 43:49)



We can obtain the probabilities by the EM-algorithm, EM based algorithm which is the inside outside probability, now it is time to summarize the course, I will just go over them, when we started the course, we first discussed a number of ambiguity examples, which obtains in case of natural language processing. Natural language processing is divided into number of stages, which are classically accepted, phonology, morphology, syntax, semantic fragmenting this courses and so on, at every stage there are ambiguities. So, this ambiguity resolution happens either by rule-based method, which is the knowledge-based classical approach to n l p or it happens by machine learning based method, which is statistical in nature.

Then, we moved on to shallow parsing, where we discussed part of speech tagging, this required understanding a very important machine learning algorithm, Hidden Marco of Model, and the associated algorithms. So, this was covered in Hidden Marco of model, we covered Viterbi algorithm forward backward algorithm, and the Baw moist algorithm, and their usage in estimating the sequences, it is main usage is in shallow parsing. We also dwelled in Tibetian to Indian language part of speech tagging, which is a very challenging and important issue, and it is usage is supreme in the context of various machine learning, machine transition and cross search projects going on in the country.

Then, we explored a bit the information retrieval topic, and this relationship with natural language processing was addressed, then we moved to a very important topic namely the topic of words and disambiguation both unsupervised and supervised algorithms are covered knowledge-based algorithms also were looked at. But we gave lot of emphasis on the less algorithm, this is overlapped based algorithm, and it is usage was explored in detail, followed by supervised and unsupervised and semi-supervised methodologies. After finishing words and disambiguation, we started discussing parse version disambiguation occupied a major chunk of lectures, probably about 10 lectures. Then in parsing we used about 5 to 6 lectures to cover deterministic algorithms, classical algorithms. Then we move to probabilistic parsing, in probabilistic parsing the inside outside algorithm, the inside probability the CYK algorithm and how it influences probabilistic parsing too were covered.

So, in particular I must say that, I enjoy teaching this course by covering both the knowledge based techniques in natural language processing, and the machine learning probabilistic based method. These two together form the core of natural language processing in modern times, we did not cover large applications like summarization machine translation. We did a bit machine translation in the course, but not a whole lot information extraction, these were not covered, these are a subsequent advanced level course. I hope the techniques covered in this course will be useful to you, in your future studies of language processing, which is a very exciting field of artificial intelligence.

Thank you.