


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


Lecture - 9
Brief on Probabilistic Parsing and Start of Part of Speech Tagging

(Refer Slide Time: 00:28)

Problem 5: Parsing



Source sentence Noisy Channel Target parse

$$\begin{aligned}
 T^* &= \underset{T}{\operatorname{argmax}} [P(T|S)] \\
 &= \underset{T}{\operatorname{argmax}} [P(T) \cdot P(S|T)] \\
 &= \underset{T}{\operatorname{argmax}} [P(T)], \text{ since given the parse the sentence is completely determined and } P(S|T)=1
 \end{aligned}$$


We will start the very important topic of Part of Speech Tagging. Before that, we finish what we left in the last lecture namely, the argmax spaced noisy channel computation of parse tree from a sentence. So, we said that the source sentence parse is through a noisy channel does an argmax spaced computation or phases in argmax parse computation and a target parse tree is produced. So, the expressions show here the theory and the computation, T star is the best possible tree in the sense of probability value of the condition probability P T given S and the conditioning or the argmax based computation is, who were all possible T's.


Now, we apply base theorem and convert this probability into P T into P S given T, P S given T is 1, because given the T, the sentence is completely determined without any uncertainty. The leaf nodes traversed in that order produces the sentence and P T is a prior probability, which is the probability of the tree. So, T star becomes a very simple expression now, it is argmax over all possible T's of P T, the probability of the parse tree T.

Now, this is a very intuitive when seemingly obvious expression where the best possible parse tree of a sentence is nothing but the tree which has the highest probability. So, that naturally leads to the question as to, how to compute the probability value of the parse tree, what is the meaning of the probability of the parse tree. So, this is a discussion reserved for the topic of parsing and probabilistic parsing, which we do later.

(Refer Slide Time: 02:25)

Probabilistic Context Free Grammars

■ $S \rightarrow NP VP$	1.0	■ $DT \rightarrow the$	1.0
■ $NP \rightarrow DT NN$	0.5	■ $NN \rightarrow gunman$	0.5
■ $NP \rightarrow NNS$	0.3	■ $NN \rightarrow building$	0.5
■ $NP \rightarrow NP PP$	0.2	■ $VBD \rightarrow sprayed$	1.0
■ $PP \rightarrow P NP$	1.0	■ $NNS \rightarrow bullets$	1.0
■ $VP \rightarrow VP PP$	0.6		
■ $VP \rightarrow VBD NP$	0.4		



But, I just want to give you the main idea behind, now probabilistic context free grammar forms the foundation of this discussion. Probabilistic context free grammar is just like context free grammar where, only the additional thing is that, with every rule or every rewrite rule or every production, we have a probability value. So, these symbols try to capture a small grammar of English, let us go through these rules one by one, S goes to $NP VP$.

That means, a sentence is composed of a noun phrase and a verb phrase, a simple example of that would be children play where, children is noun phrase, play is the verb phrase and every sentence has noun phrase and verb phrase. All sentences are composed of these two consequence, there is no other possibility for a sentence and therefore, the probability value is 1.0. This kind of certainty with respect to probability is not the case for other kinds of phrases, because NP which is a noun phrase can be constructed in many different ways.

Three ways I have shown here, NP goes to DT NN that means, a determiner and NN, this is a very common situation for a noun phrase, the why DT is the NN is why. NP goes to NNS, what is NNS, NNS is a plural noun and the convention comes from what is called the Penn tree bank, which is an important project for annotated co-procreation and they adapted this convention and it is shown here also. So, a noun phrase is a plural noun then the final NP production is, noun phrase is, a noun phrase followed by a preposition phrase.

So, this is a recursive definition with left recursion, NP is at NP followed by a preposition phrase. So, let us say, we invoke the NNS rewrite for this recursive NP, so suppose NP is now, NNS PP. So, we have to have, we are looking for an example where the noun phrase starts with a plural noun and has a preposition phrase. So, that is easy to find out, because preposition phrase P phrase what is the definition, the definition is that, it starts with a preposition and is then followed by a noun phrase.

So, suppose here also for noun phrase, we use plural noun and therefore, these now becomes noun phrase is a plural noun followed by a preposition followed by another plural noun. So, can you think of a construct of this kind, not very difficult, so this would be boys with footballs for example, this will follow the structure, boys is NNS plural noun, with is a preposition and with footballs is similarly a plural noun, boys with footballs were spotted on the ground, that is the sentence here.

So, noun phrase can be of different kinds, in this case there are three possibilities namely, determine a noun, plural noun and noun phrase followed by a preposition phrase, the boy, boys, boys with footballs. Preposition phrase is a preposition followed by a noun phrase, so things like with a binocular in the train by the people, for the people, of the people, all these are preposition phrases. VP is the verb phrase, the next important structure after the noun phrase and verb phrase can be again of different kinds.

These captures two different situations for forming a verb phrase, a verb phrase can be a verb phrase followed by PP, a preposition phrase, this is a left recursive production once again. And this is a basic rule, verb phrase is VBD, D indicates the past tense e d, reminiscent of the e d suffix and VBD indicating past tense is again from the Penn tree bank. So, here it is an example of a verb phrase, which is past tense verb followed by a

noun phrase and these actually captures the transitive verb situation that is, verbs which take objects.

So, for example, VBD could be played, NP could be football or saw the boy, NP is DT NN, VBD is saw, saw the boy and VP with PP, VP is VP and PP, a recursive definition, what could be an example for that. An example for that could be, saw with telescopes, people saw Everest with telescopes. So, saw Everest or saw mountains with telescopes, saw mountains with telescopes will become parse by this grammar. But, saw Everest with telescopes will not be parse, because we have not made prohibition for noun phrase going to a single noun.

But, it could parse things like saw the Everest, if you introduce the before Everest, that could be parse anyway. So, this shows how some segment of English, a few constructs of English are captured by production rules, which are context free grammar rules. So, there is nothing new in this, what is new are this probability values, we come in minute to the understanding of this probability values. Notice the probabilities here in the right hand side DT goes to the, with 1.0.

So, this is not a completely correct situation, because a determiner can be the, a, and an, so this is a limited view of DT, what DT is, DT is the we are ignoring here a and an. And NN is noun, which goes to gunman this means that, our machine has seen only the word gunman and another word of course, it is same this is building. And VBD, the past tense verb is sprayed, NNS is bullets, so this is an extremely artificial situation.

We know that, English languages many, many plural nouns, lakhs of nouns and there are many, many verbs, 30, 40000 verbs and if you also consider the past tense form, they would be equal in number. So therefore, this is a very, very artificial situation where, the machine has been exposed to only one sentence, can you imagine what the sentence is, the sentence is, the gunman sprayed the building with bullets. So, since the appears twice, it has to be taken into account only once in the production rule.

And we divide the probability equally amongst gunman and building, because the computation is probability of NN going to gunman is the number of times gunman appears divided by total number of nouns. The total number of nouns here singular noun of course, is 2, so this is 1 by 2, 1 by 2, 0.5 and there is only one pronoun, which is

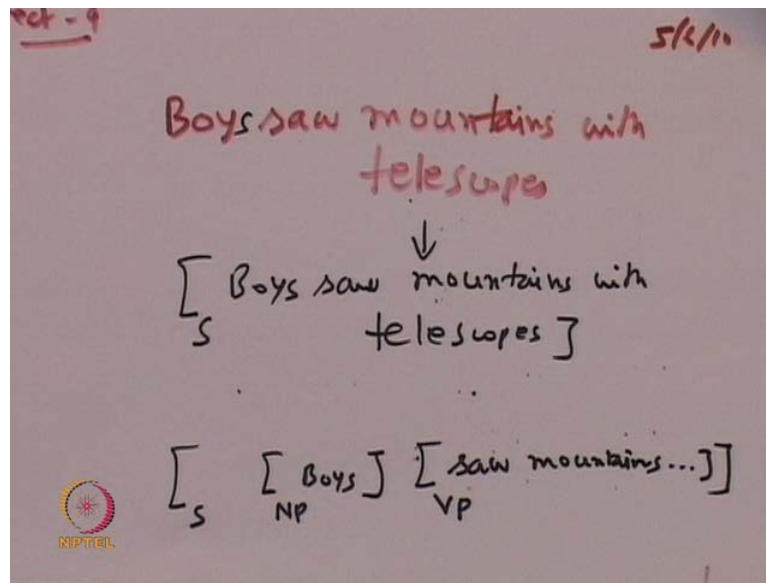
bullets and NNS goes to bullets with 1.0 probability. Now, one question that might arise in your mind at this stage is that, what is the point of keeping the plural nouns as bullets.

Let us say, one could always do morphology analysis or run a tool, which is called the morphology analyzer and that isolates the suffix S and says that, bullet is a singular noun, we will talk about all these things later. So, parsing typically is presided by a phase of morphology, presided by a phase of parsing for part of speech tagging, which is our next topic of our discussion. So, you just accept the following at this stage that, a probabilistic context free grammar is a set of production rules just like in context free grammar, but associated with probability values.

Now, let us quickly understand what this probability values mean, the probability values mean that, noun phrase being expressed as determiner and noun occurs 50 percent of the time in the corpus. How is the corpus created, corpus is created by lexicographers, linguists, language experts, who produce these kind of structure on the sentences. So, the number of times the structure is found in the corpus divided by total number of noun phrases in the corpus, it gives me the probability value.

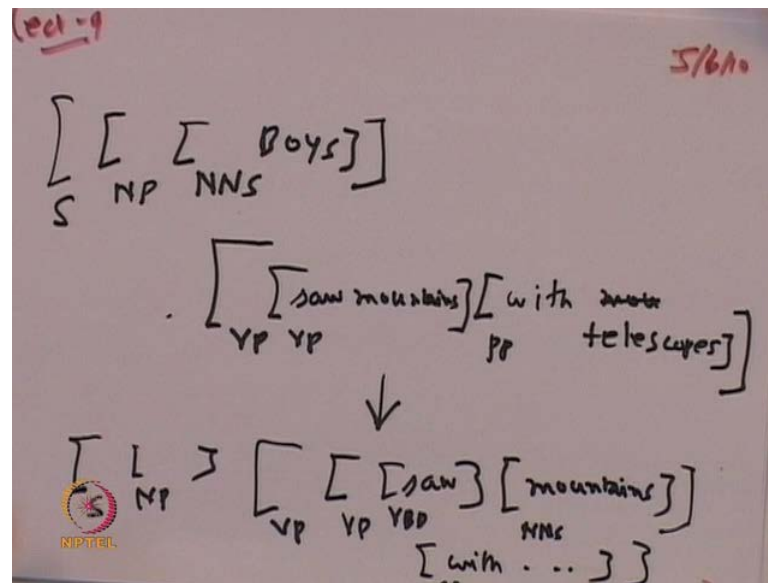
So, this shows that, in the corpus 50 percent of the time noun phrases appear, of the 50 percent time the noun phrase appear, it was DT NN. So, NP was DT NN 50 percent of the time, NP appeared as plural noun 30 percent of the time, NP appeared as noun phrase followed by a preposition phrase 20 percent of the time. Similarly, verb phrase had 60 percent, 40 percent distribution, so this is quite clear. Just let me give an example of, how the machine actually looks at the corpora and gets these structures, so let me take a very very simple example of how the corpora looks like.

(Refer Slide Time: 13:13)



So, if we have a sentence like boy saw mountains with telescopes, this is a sentence, so we begin with S goes to NP VP. So, in this case, the whole structure is S of course, boys saw mountains with telescopes, the whole thing is S. Now, we identify noun phrases and verb phrases in this and gives similar kind of bracketing. So, we are doing what is called the top down processing, now boys is the noun phrase and the whole sentence part, saw mountains, etcetera is the verb phrase. So, is it clear to you, how the bracketing is being done, boys saw mountains with telescopes is S and then boys is NP, saw mountains with telescopes is VP, what we are doing is essentially top down parsing. In the next step, we have to give finer structures to these.

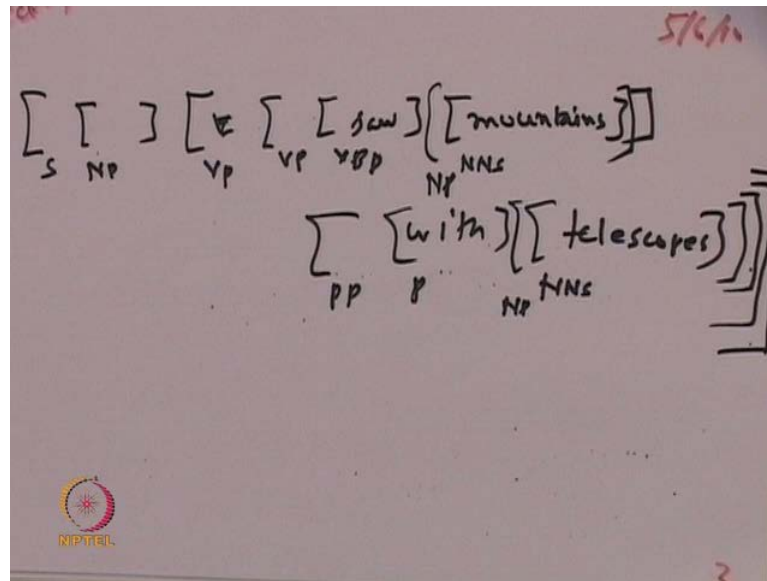
(Refer Slide Time: 14:47)



Boys cannot be broken up further, because NP goes to NNS is a rule which cannot be elaborated further however, NP goes to NNS, we have to give the closest tag to this, NNS boys. So, this finishes the story on boys, which is boys is a plural noun indicated by NNS as a bracket and this is also the noun phrase as indicated by NP. The processing of verb phrase is little more complex, because we have VP here, saw is also a verb phrase and it is not very obvious how we should group them or break them apart.

So, we have saw mountains with telescopes, with telescopes has to be a preposition phrase, so we put it down here, with telescopes and this is a PP, the whole thing is under a VP. And here, we have to invoke the rule, VP goes to VP PP, it is a left recursive rule, so saw mountains is another VP. So, you see how the structure is emerging, S is nothing but NP and VP at the top most level. NP in this case is NNS, NNS is boys and VP the verb phrase, saw mountains with telescopes is broken up into VP and PP, PP is with telescopes saw mountains. I will not expand the NP part, because that is same, it will continue to remain same and I will just say, this is the NP without showing the details below it. However, the VP has saw and saw, we know is a VBD, mountains is NNS. So, the whole thing is the VP and with telescopes remains the PP, we would like to break the PP now.

(Refer Slide Time: 17:50)



Proceeding further, we get S, NP is as before within VP, we have a VP with saw as VBD, mountains as NNS and with telescopes is a PP, with being a P and telescopes being NNS, which is also the NP. So, this whole thing finishes the PP, whole thing finishes the VP and whole thing finishes S. This large number of brackets makes the expression slightly complicated towards the N, but I guess you get the main idea. Now, what we can do is that, we can go whatever, we can go from the words and see how the final structure emerges.

So, telescopes is NNS plural noun, which again is a noun phrase, so NP goes to NNS is invoked, P with is a P, so P with NP, P and NP gives rise to PP, which is the higher level structure. So, the work of PP is over, now we come to mountains which is NNS, which again requires an NP to be placed, this is NP, VBD and NP together forms a VP, this is the rule. VP goes to VBD NP this is invoke now, the whole thing of the VP and the PP forms the VP, so VP goes to VP PP this left recursive rule is invoked, we have the NP here, NP and VP together forms S.

So, this illustrates how the bracket at corpus is created for training a machine to learn parsing and you must have seen, how complicated the process is. A process like this is invoked on the sentences and many times of course, a rule base parser can create this kind of structure and many times not, because the rule basis have their own limitations.

But, once the structure is created, it is now suitable for a machine to process and the machine can be trained to learn parsing.

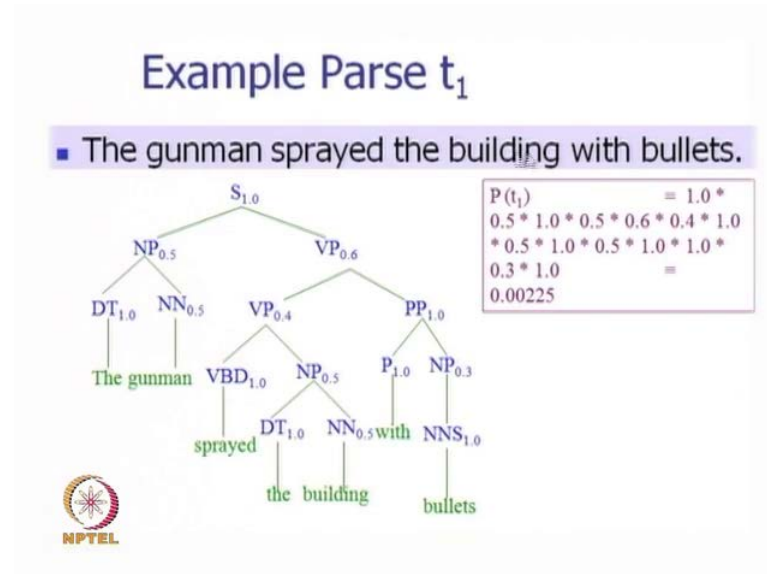
I draw your attention to the bracketing, now bracketings essentially capture the internal structure of the sentence and how smaller structures group together to form a higher level structure and the highest level structure is S. So, what should be noted here is, how the annotator is required to invoke the correct rule at specific points of processing of the sentence. So, the person has to be knowledgeable with respect to the syntax of the language, must have a very good idea of the properties of the words. So, telescopes for example, could be a third person singular verb also.

Telescopes as a verb and saw is a verb in past tense, but saw could also be a noun, an instrument for cutting wood, let us say. So, the lexicographer who is producing these kind of brackets has to know the structure of the sentences, the structure of the grammar and the properties of the words. So, this is by no means, a non trivial task, it requires lot of money and time to create this kind of resource, bracketing structure of sentences, but once created, it becomes useful for training a machine to learn.

So, we now of course, we again come back to the bracketed structure we created on the paper and now, I can explain to you how the probability values are calculated. So now, you see here, VP going to VBD NP, this structure appears once and VP not VP, but NP going to NNS was invoke twice for telescopes and boys, and trice actually, NP goes to NNS was invoke for boys, mountains and telescopes. So, what the machinery do is that, it will find that, NP goes to NNS was applied three times whereas NP, no in this case NP going to NNS is the only production which is used.

However, VP going to VBD and P, and VP going to VP and PP, these are the two instances of the verb phrase expansion. And since they appear with equal number, if this was the on the corpus available on the evidence available then you have no other choice than distributing the probability equally to VP goes to VP PP and VP goes to VBD NP. So, this kind of bracketed structure is looked at by the machine to compute the probability by counting, simply counting.

(Refer Slide Time: 24:03)



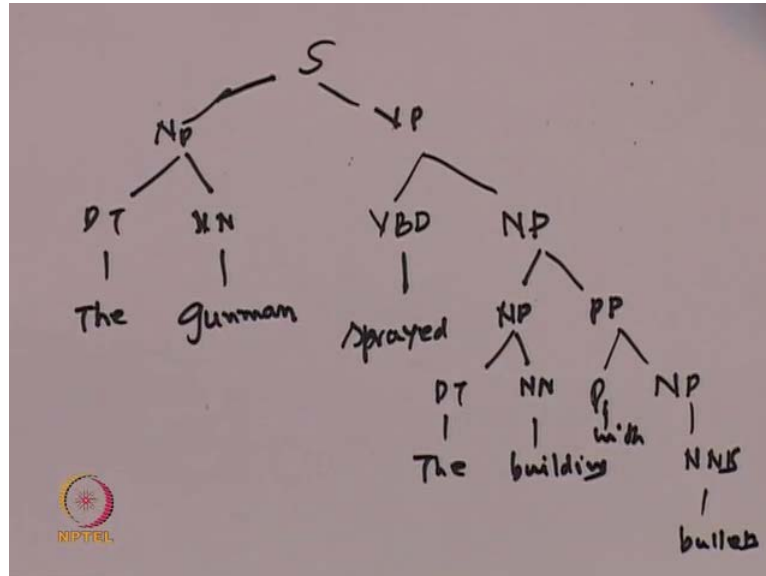
So now, let us understand what is the meaning of the probability of a tree? And how one has to choose the best possible tree in terms of its probability value. This sentence, the gunman sprayed the building with bullets is an ambiguous sentence, do not think of, do not look at the sentence from the word knowledge point of view, but simply from the point of view of structure. So, the gunman sprayed the building with bullets, in this case what is the status of this preposition phrase with bullets.

So, the immediate answer I suppose, that even make is, with bullets is the entity with which the gunman sprayed the building. So, with bullets goes with sprayed and therefore, this is logical structure. So, the sentence goes to noun phrase and verb phrase, a noun phrase is again determiner and noun, which is the gunman. Verb phrase goes to verb phrase and preposition phrase with left recursion, notice that preposition phrase being here makes it attached to the main verb, the main verb is sprayed, so spray with bullets is the reading.

VP going to VBD and NP which is sprayed, NP here is the building through DT and NN, PP is preposition at noun phrase and in this case, the noun phrase is a plural noun bullets, so with bullets. So, the point to be noted here is that, the PP is at the same level as this VP, which contains the main verb, this structure contains the main verb sprayed. So, this way of forming the tree is telling me that, with bullets is attached to sprayed, the gunman

used bullets to spray the building. Let us look at the other parse tree, the other parse tree let me draw on the paper.

(Refer Slide Time: 26:27)



The gunman sprayed the building with bullets, S again goes to NP and VP, NP goes to DT and NN which is the gunman. VP goes to VBD, NP this is the other invocation of the rule VP, so VP can be expanded as VP PP or VBD NP. Now, VBD becomes sprayed, NP on expansion becomes NP and PP, this NP becomes DT and NN and produces the building and PP is P NP, this P becomes with, NP is bullets. So, NP goes to NNS and bullets, so notice the difference between the two structures.

I will draw your attention once to the slide where, this PP is at the same level as VP or PP is at the same level as a structure, which contains the main verb. Whereas, if you look at the tree that I drew on the paper, here you find that the PP is not at the same level as the main verb sprayed. It is at a level below, at one level below, in a sense that, there is an NP and that NP contains the another NP and PP. So, this means that, the preposition phrase is attached to the noun phrase here that means, it is attached with the building.

So, as if the building has bullets in it, which building, that building which has bullets in it, the building with bullets. So, the gunman sprayed the building which has bullets, which contains bullets, so this is another possible reading of the sentence. The sentence is ambiguous, this is very similar to a very famous sentence in natural language processing. The sentence is, I saw the boys with telescopes and the ambiguity here is,

who has the telescopes. I saw the boys using my telescope or I saw a boy who had telescope, that is the meaning.

So, similar is the situation in this case, in one case with bullets is attached to building. In the other case, with bullets is attached to sprayed, that is the ambiguity, now the question is, which parse tree is more probable. This is founded by computing a probability values and the theory for this will be discussed later when we do probabilistic parsing in detail. But in this case, for the moment surprise it to say that, the probability of the parse tree is nothing but the probabilities of the rule applications.

So, this first 1.0 comes from S going to NP VP and that is a definite rule with no uncertainty and therefore 1.0, into 0.5 where does this come from, this comes from the fact that, NP has been expanded as DT NN. And that occurs 50 percent of the times in corpora with probability 0.5 therefore, this is the 0.5. Then again I have 1.0, this is the expansion of DT with probability 1.0, there is only one determiner which is the then we have a probability value of 0.5, which is noun going to gunman.

Assume, this is the probability value and then we come to this part of the tree, the multiplication factor is 0.6 here, which comes from this value 0.6 which means, VP is expanded into VP PP with probability 0.6, that is the 0.6 here. Then we had 0.4 which is the probability of this PP expansion, which is VBD and NP, that has a probability of 0.4, so this is 0.4. Then 1.0 comes from PP being expanded as P NP, PP is expanded only this way as given in the grammar, there are no other possibilities, so 1.0, so this is the 1.0.

Now, we have another 0.5 which comes from the NP getting expanded as DT and NN and then we have 1.0 with DT being the another 0.5, which is NN going as building. And then there is another 1.0 which is the preposition with, this is the only preposition in the grammar then NP expanded into NNS with 0.3 probability, which is 0.3 and NNS becoming bullet with 1.0 probability. So, this is the way, the probability of a parse tree is computed namely, the probability of rule application or the probabilities associated with the production rules, simply the product of probabilities.

Now, the question of why this is done, what is the rationale behind this, what is the theory behind this, that will be explained later when we do probabilistic parsing in detail. So, this is the probability, the probability comes out to be 0.00225, I am not showing the probability of the tree which is drawn in the paper. I am not showing that probability,

again one can record the probability values on the nodes for the rule application, this is given in the grammar.

Take the probability values take the product and you will find that, the probability value here will come out to be less than the probability value on the slide. So, this means that, this tree is much more probable than the tree I drew on the paper and this is because you might be tempted to say that, the gunman sprayed the building with bullets is a much more feasible situation. Then sprayed the building which contains bullets, which is fine as far as the word knowledge is concerned. But, the machine does not have that kind of common sense knowledge, it goes purely by counts.

And it so happens that, the production rules here are occurring more in the corpus, giving thereby the evidence that, the verb phrase indeed is expanded as verb phrase and preposition phrase many many more times. And preposition phrase at gets attach to the verb, also the verb phrase becoming VBD and NP is more frequent at deeper levels of parsing rather than other top level. So, this kind of phenomena exists in the corpora and the machine simply computes the frequency and obtains the probability value.

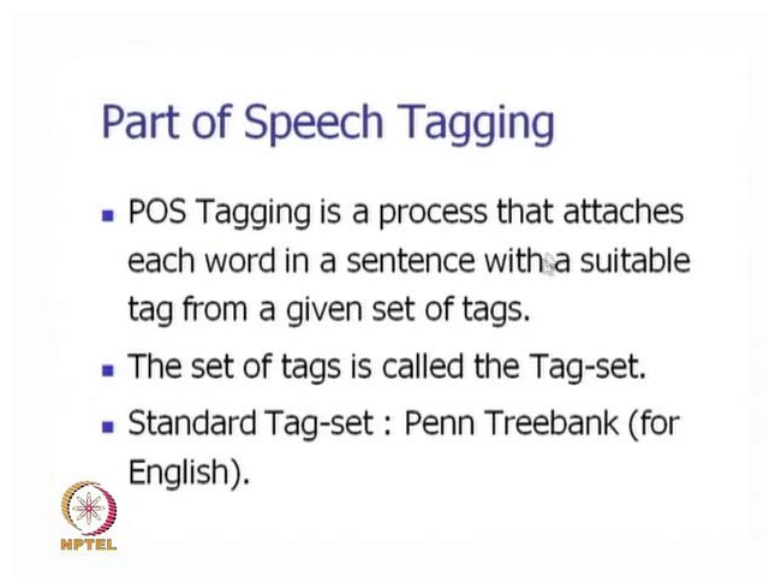
One should not be tempted to be leave that the machine has intelligence and word knowledge, but the point is that, one knowledge on intelligence can be almost simulated. I am tempted to say, if there is a very, very large amount of corpus and people's actual language behavior, written sentences, spoken utterances can be captured and there we will find the evidence for preposition phrase getting attached to verb more frequently than to the noun, so that is the point. So, from this the main conclusion we draw is that, it is possible to obtain the parse tree of a sentence through argmax based computation. And in the argmax based computation, the tree which is chosen finally as the output, is the tree with the highest probability. The probability is found simply by taking the product of production rules, the theory will be done later.

(Refer Slide Time: 35:32)



We now move on to a very important topic, a very, very fundamental basic topic of natural language processing called POS tagging, part of speech tagging. The part of speech is something, which you have done before in your high school, the very famous part of speeches are top level categories noun, verb, adjective, adverb, adjectives qualify nouns, adverbs qualify verbs. So, there are verbals and nominals, so these are basic grammatical categories, which come from our high school grammar. So, one might be wondering, what has POS tagging got to do with natural language processing, this is a very important point to discuss, so let us go ahead and make those issues.

(Refer Slide Time: 36:21)



Part of speech tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags. What are these tags, we will see in a minute through examples, the set of tags is called the tag set, there are many standard tag sets. So, Penn tree bank is the most famous one for English and some of examples of this, we will see very soon. So, POS tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags, that is the main point.

Now, the point is that, who creates these tags, how is the set of tags defined, lot of intricate linguistic and computational concerns go behind the design of a tag set. We will see as we discussed POS tagging that, certain things are not possible when we are at the level of POS tagging without invoking syntax and semantics of the sentence. The higher level structure of the sentence or the meaning of the sentence, but the point is that, if we do that then we are putting the cart before the horse.

The horse has to pull the cart and therefore, it has to be behind the horse which means that, we are doing an activity which is supposed to be done later. So, syntax analysis, semantic analysis require as a fundamental step, the part of speeches of the words constituting the sentence. So, how could we assume in a syntactic or semantic knowledge for POS tagging when those activities actually follow POS tagging, this is the point. So, at the level of part of speech tagging, we will not be able to produce certain tags. And therefore, those tags will not be useful in the tag set, because those tags cannot be produced. And therefore, these kind of considerations go for designing the set of tags, the tag set.

(Refer Slide Time: 38:38)

POS Tags

- NN – Noun; e.g. *Dog_NN*
- VM – Main Verb; e.g. *Run_VM*
- VAUX – Auxiliary Verb; e.g. *Is_VAUX*
- JJ – Adjective; e.g. *Red_JJ*
- PRP – Pronoun; e.g. *You_PRP*
- NNP – Proper Noun; e.g. *John_NNP*

 etc.

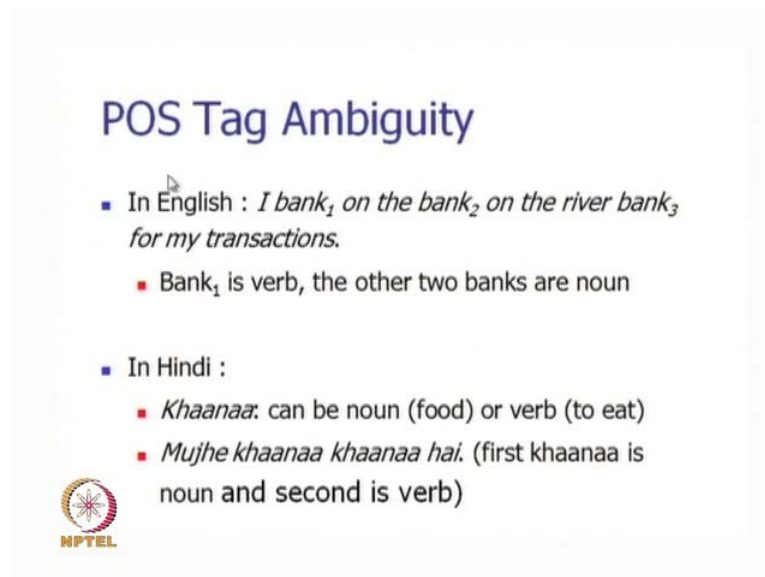
Let us look at some example of POS tags, NN is the most important and most frequent POS tag, this stands for common noun. For example, dog NN, so look at the convention here, this is a ((Refer Time: 38:56)) accepted convention where, the word is followed by an underscore and followed by the tag. So, this means that, dog underscore NN, here dog is a noun. Now, what is an example of this, I saw a dog here, dog is a noun, so dog will be tagged as underscore NN.

VM or the main verb is possible in the next most important entity and here the example, I have chosen is run. So, if we have dogs run, dogs run or a dog runs and so on, so let us not worry about the morphological changes here dogs, runs, etcetera. So, run is the main verb, run underscore VM is the main verb. VAUX is the auxiliary verb for example, is, am, are these are auxiliary verbs, in this case, is is the example, so is underscore VAUX is the auxiliary form.

JJ stands for adjective, so one might wonder where does JJ come from, this is from the penn tag set, it is most likely adjective is pronounced that way, the d has a palatal sound ja, so that is why, this gives an impression of double J and that is why, JJ is kept here. So, for example, a red ball, here red is an adjective, so red will be tagged as underscore JJ. PRP is a proper noun, so I, you, we, she these are proper nouns, in this case I have taken the example of you, which is you underscore PRP.


So, if the sentence is, you laugh then you underscore PRP, laugh underscore VM finishes the tagging of the sentence. NNP is proper noun for example John, John underscore NNP is the proper noun. So, one could tag a complete sentence by means of these kind of tags, this is not the full tag set by any means, that is why there is an extra here. You could go to the site of Penn tree bank, go to Google, type as a query Penn tree bank and you would be lead to the home page of Penn tree bank project. And then you will see all these tags there, so I suppose the meaning of these tags and their purpose is quite clear.

(Refer Slide Time: 41:37)



POS Tag Ambiguity

- In English : *I bank₁ on the bank₂ on the river bank₃ for my transactions.*
 - Bank₁ is verb, the other two banks are noun
- In Hindi :
 - *Khaanaa*: can be noun (food) or verb (to eat)
 - *Mujhe khaanaa khaanaa hai.* (first khaanaa is noun and second is verb)



Now, why is parse tag a difficult problem, the problem is difficult because of these very basic or common situation prevailing in natural language pressing, namely that of ambiguity. So, POS tag ambiguity is rampant in language, in English we know that, almost all nouns can be used as verbs. So, I take this very interesting sentence though improbable, I bank on the bank on the river bank for my transactions. So, for my financial transactions, I bank on the bank on the river bank, so the word bank appears three times here.

And therefore, this linguistic convention is followed which is the lower suffix, so bank one is the first appearance of the word bank, bank 2 is the second appearance, bank 3 is the third appearance. So, for financial transactions, I bank on the bank on the river bank, the first bank is the verb meaning depend, so I depend on the bank on the river bank for

my transactions. The other two banks are noun, but you can see that, here is a situation for word sense this ambiguity.

The second bank is the actual bank, from where money is withdrawn or into which money is put in, so this is where financial transactions are carried out. Whereas, this bank is a place, river bank is a place, so words in this ambiguity will be called for, to get the correct sense of the two banks, the first bank of course, is a verb. Now, the question is, when the POS tagger is called to put levels on the words of this sentence, I will be leveled as PRP and bank will be noun on will get the tag for preposition, there will get the tag for determiner.

This bank now will have to get the tag for noun, the first bank has to get the tag for verb, VM this is the main verb, so this is noun NN. On the tag for preposition the, tag for determiner river, tag for noun and bank again the tag for noun, for the tag for preposition my, the tag for pronoun and transactions the tag for noun. So, I and my are pronouns here and this bank is the only verb, prepositions there are three of them on, on, for and the nouns are bank, bank, transactions, river.

So, this shows that, POS tag can be facing the challenge of ambiguity, so this is not the situation only in English, in other all languages the POS tag ambiguity exists. From the context, one can make out the actual part of speech, but sometimes it may not be possible. Now, here is an example in Hindi where, we deal with the POS tag ambiguity, [FL] can be noun in the sense of food or it can be a verb to eat, which is the infinity will form. So, [FL] or [FL], this is like the previous sentence in English, I bank on the bank on the river bank, here we are saying [FL]. So, the first [FL] is noun, I need to eat food and second [FL] is verb in the inimitable form.

(Refer Slide Time: 45:44)

For Hindi

- *Rama achhaa gaata hai*, (hai is VAUX : Auxiliary verb); *Ram sings well*
- *Rama achha ladakaa hai*. (hai is VCOP : Copula verb); *Ram is a good boy*



There is another example here which is interesting, Ram [FL], [FL] is auxiliary verb and the sentence means, Ram sings well. Ram [FL], here [FL] has a different grammatical role, what is called copula verb, VCOP copula verb. So, the symbols also are different VAUX and VCOP and the meaning of the sentence is Ram is a good boy, so Ram sings well and Ram is a good boy. So, I draw your attention here to two problems, one is [FL], so this [FL] has the auxiliary role, [FL], [FL] has the copula role.

How will the POS tag in system find out, one is auxiliary the other is copula, think about this. The other problem is [FL] appears at both places, Ram [FL], this [FL] is an adverb, it is qualifying the action of singing. And [FL], here it is qualifying the noun [FL], a particular quality of the [FL], so this is an adjective. So, [FL] in the first sentence is adverb, the second sentence [FL] is adjective, [FL] in the first sentence is auxiliary verb, [FL] in the second sentence is copula verb.

So, when the POS tag works, it has to correctly produce the tags, in the first case adverb, in the second case adjective and this is a disambiguation task, it has to do it correctly. One might be advancing a simple rule saying that, if a word which can be both adverb and adjective, is followed by a noun then it is actually an adjective, if it is followed by a verb it has to be an adverb. But, though this rule is simple, this rule is not all encompassing, it can fail for situations which we will explain in the next class.