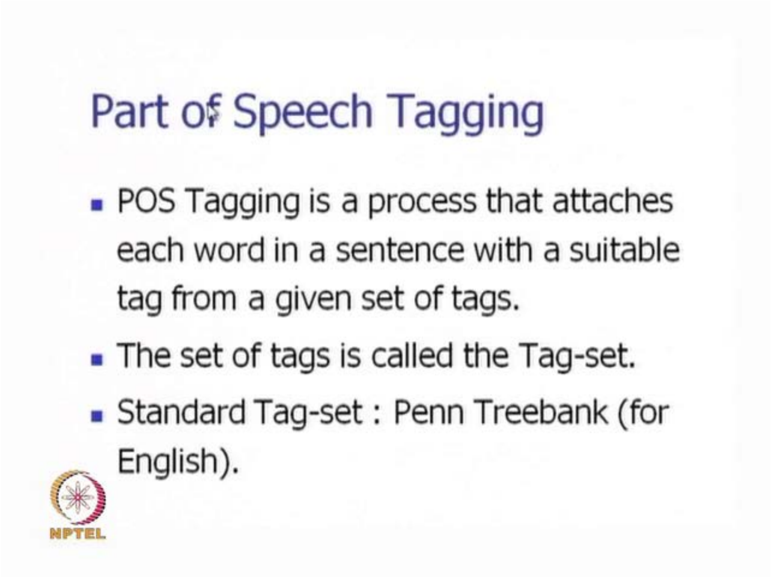
Natural Language Processing
Prof. Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture - 11
Part of Speech Tagging
Contd…

In this lecture, we continue with our discussion on Part of Speech Tagging of words in a sentence. We have remarked many times that part of speech tagging is the first fundamental operation on raw text, and subsequence stages of natural language processing, like phrase detections, parsing, semantic low leveling and so on follow after part of speech style. This is also a very nice problem to illustrate, statistical natural language processing, the application of machine learning on natural language problem alright.

(Refer Slide Time: 01:01)

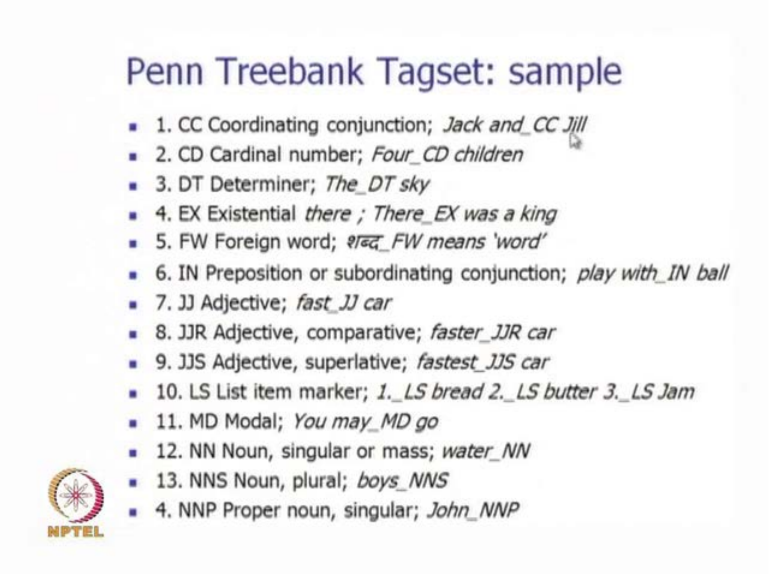


Part of Speech Tagging

- POS Tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags.
- The set of tags is called the Tag-set.
- Standard Tag-set : Penn Treebank (for English).

So, as is shown in the slide now, part of speech tagging is a process that attaches each word in a sentence with a suitable tag from a given set of tags. The set of tags is called the tag set and one of the very standard tags sets is the penn Treebank, which we show in the next slide. But before that just some small remarks, we are saying that part of speech tagging is the process of attaching tags on the words of a sentence. So, that thing to note here is this entity call tag, which is like a level on each word, what these labels are we are going to see in a minute alright. So, the words are given this levels and this are very informative, for subsequence stages of processing.

(Refer Slide Time: 02:07)



Penn Treebank Tagset: sample

- 1. CC Coordinating conjunction; *Jack and_CC Jill*
- 2. CD Cardinal number; *Four_CD children*
- 3. DT Determiner; *The_DT sky*
- 4. EX Existential *there ; There_EX was a king*
- 5. FW Foreign word; *शब्द_FW means 'word'*
- 6. IN Preposition or subordinating conjunction; *play with_IN ball*
- 7. JJ Adjective; *fast_JJ car*
- 8. JJR Adjective, comparative; *faster_JJR car*
- 9. JJS Adjective, superlative; *fastest_JJS car*
- 10. LS List item marker; *1._LS bread 2._LS butter 3._LS Jam*
- 11. MD Modal; *You may_MD go*
- 12. NN Noun, singular or mass; *water_NN*
- 13. NNS Noun, plural; *boys_NNS*
- 4. NNP Proper noun, singular; *John_NNP*

Progressing with our discussion, this is an important slide which shows the Penn Treebank tag set, it is a sample there are about above 30 tags and we show here, a few of them may be above 10 and what is shown here these level. So, the first level here is CC, which is the short form for Coordinating Conjunction, the example of that is the jack and Jill. So, this is a phrase, jack and Jill may be they go up the hill and this CC is attached to the word and, which is a conjunction, joining jack and Jill. So, an underscore CC shows the part of speech tag of an.

The next one is CD or Cardinal number. So, if we look at this phrase four children, four is a numerical adjective qualifying the noun children and this is a cardinal number CD is attached here, if the word was fourth, the fourth child t h is other then the part of speech tag would be the tag for ordinal number. So, that indicates an order, so cardinal number tag would be place, DT is a determiner example the underscore DT sky, EX is Existential there as oppose to, an adverbial there here the example is, there was a king.

So, there underscore EX was a king, this there note is a pleonastic it is a pleonastic, it is a backward subject English sentence demand a subject to be compulsorily present. So, this there is given the tag example, EX existential as appose to the adverbial there. FW is the foreign word, here is a sentence [FL] underscore FW indicating it is a foreign word [FL] means word. So, we are explaining the meaning of the string [FL]. So, whenever the text of a language contains a string return in the alphabet of the foreign language, then we always place the tag FW; however, if [FL] was return as shabd. So, in this case these would be an absorption of a foreign word transliterated into the language of the text. In that the case, [FL] would be a noun and the noun tag would be placed on this.

So, note the difference between [FL] written in [FL] and [FL] written in English, when written in [FL] embedded in English text it is, in the foreign word underscore FW otherwise it is the part of speech that it has in the sentence. IN is preposition or subordinating conjunction. So, for example, play with ball, with underscore IN is the entity here and IN is the tag for with, this can also be used for subordinating conjunction and example, we can see later.

JJ stands for Adjective, many times people have wondered where this JJ is coming from adjectives, it could be abbreviated as Ad or Adj, but why JJ. So, the example that is first underscore JJ car, fast car it is an adjective, one conjunction is that adjective is open pronounced as adjective and it has two j sounds, adjacent to each other and most likely that give rise to this two letter tag JJ. JJR is adjective comparative, so fast car was the previous example, now it is faster car, fast car compare to another car and this is underscore JJR.

JJS adjective superlative. So, fastest car JJS fastest underscore JJS, LS is List item marker for example, somebody is proposing to buy this grocery items, bread butter and jam the listing is given here, 1 bread, 2 butter, 3 jam. So, 1 underscore LS, 2 underscore LS, 3 underscore LS, they are the tags for this list items markers. MD is the Model, this is a sentence here, you may go as appose to you go and may here is the model auxiliary and is given the tag MD with an underscore. NN is probably the most important tag, which is the noun tag, which is used for singular or mass.

So, water underscore NN, NNS is plural noun, boys underscore NNS. If you are wondering, why there is a separate tag for NNS note that, in English most nouns can be used as verbs and the third person singular form of a noun and the of a verb and the plural form of a noun are open identical. So, if I say place it is not clear is it multiple place, mini place in the sense of drama let us say or is it the third person singular form of the verb play.

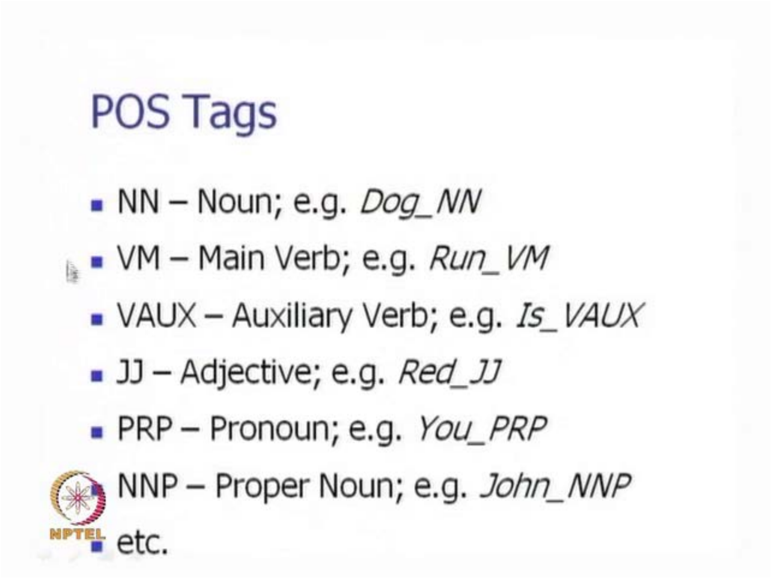
NNP is proper noun singular, so john underscore NNP is the proper noun. So, these illustrates some important points in this tag set, if you are pay attention to the tags then you would see that, lot of care has gone into deciding how to designate tags properly. For example, there is a very certain discussion on why, plural nouns should have a separate tag compare to singular noun. Singular noun is NN, plural noun is NNS and the reason for that as I said is that, almost all English nouns can be used as verb play can be a noun meaning a drama dramatic performance or a play could be the act of playing which is a verb.

Now, if we see an isolated word in the text, in the form of place than we would not know whether this is a plural noun or is a at third person singular form of play, the task of part of speech tag is to place the correct tag on the word. So, the part of speech tagger uses limited context to window and it places the tag on the target word. Now, if this distinction is not meet, between singular noun and plural noun and in the part of speech tag corpus, which is used for training if we have many typically have many singular third person singular number verb forms and we also have many, many plural nouns.

If noun and plural noun is not cleanly separated, then the part of speech tagger will get vary of an confused between the third person singular number form of the verb and the plural form of the noun, will analyze these certain issue with examples later. So, the point I am making is that when, a part of speech tag set is design, normally very experience people with lot of inside into how language operates. As well as how a computational system inspired of it is many limitations, is required to do leveling with this kind of experience people discuss part of speech tag design.

We in India, have come up with a part of speech tags set for pan Indian languages, languages of not the India in though area are family languages of southern India dravidian family, languages of the north last the Tibetan family and Austro-Asiatic languages namely munda and khasi. So, we have tried our best to come up with a tag set for India languages, which should be used for annotating data and subsequently doing national language processing. So, I hope you understand the importance of designing a tag set with lot of insight and correctness.

(Refer Slide Time: 12:31)

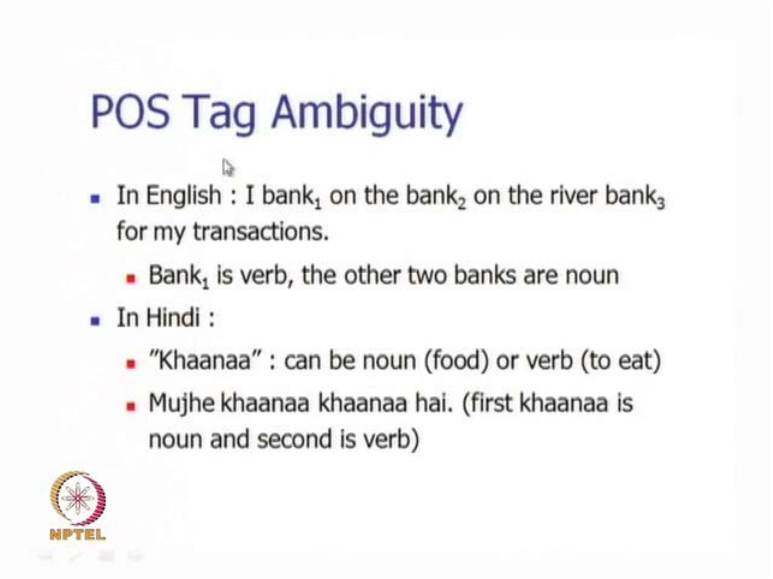


POS Tags

- 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 proceed, so we have already looked at a number of examples, with words and the part of speech tags.

(Refer Slide Time: 12:39)



POS Tag Ambiguity

- In English : I bank$_1$ on the bank$_2$ on the river bank$_3$ for my transactions.
  - Bank$_1$ 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, part of speech tag task is a essentially disambiguation task, look at this sentence here. In English, we have this sentence I bank, on the bank on the river bank for my transactions I concede that this is not a very natural sentence, which is a rather artificial sentence. But, it illustrates the points I would like to make, the first bank is a verb this means depend, I depend on the bank this is a noun, on the river bank this is also a noun, the third bank is also a noun for my transactions.

So, the first level disambiguation task which is execute it through part of speech tagging is the disambiguation of categories, this first bank is verb other two banks bank 2 and bank 3 or nouns. Now, this disambiguation is simple, but crucial for subsequence stages, another disambiguation which will have to may affected, is the disambiguation of second bank, bank 2, bank 3, the first bank is the place where financial transactions take place, the second bank is the embankment the land escaped by the side of the river.

So, these disambiguation is known as what sense disambiguation, it is possible to have same part of speech for a word, but the sentence can be different. So, the first bank is a verb, other bank is noun these ambiguity is not on the characteristic of English, we seat in Hindi and many other languages for example, the word [FL] it can be a noun, which means a food or a verb which means to eat. [FL] the first [FL] is noun and the second [FL] is verb.

(Refer Slide Time: 14:54)

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*

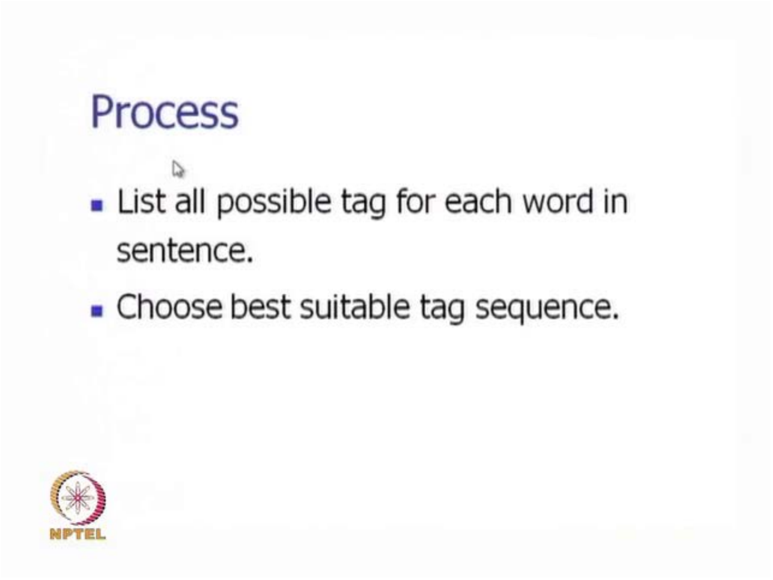
We, can also look at this nice example ram [FL] ram sings well and ram [FL] ram is a good boy. So, here you can see that the only word, which is different between the two sentence is [FL] and [FL] is a verb. So, [FL] is an adverb qualifying this verb [FL] is a noun, so [FL] here is a adjective for the noun, so ram [FL] and ram [FL] first [FL] adverb second [FL] adjective. So, the part of speech tagger has to disambiguation with this.

If you say that this disambiguation is not difficult because you see the adverb is followed by a verb, the adjective is followed by a noun. Then you might be miss late into what is call the false positive and false negative, which we have discuss last time in the lecture. I would just like remind you that between [FL] you can have some amount of text ram. [FL] So, this rule that and adverb should be follow by a verb, will not hold true here, there will rule will have to be made more robust.

Similarly, ram [FL] here are also, we can introduce a particle ram [FL] ram [FL]. So, you can have text in between adjective or noun and therefore, the rule will have to be more robust, we have more problem in this sentence, [FL] look at the word [FL] and ram [FL] the first [FL] is a helping verb, for [FL] it carries the tense information, gender number information also. So, ram [FL] if it was seta, [FL] so [FL] for would change if the tense was past tense ram [FL].

So, auxiliary verbs place in important role it carries, information about gender number person this auxiliary verb. Ram [FL] here, this verb is this [FL] is not a auxiliary verb it is not helping another verb, it is called copular verb, copular verb also carries the information of gender number person tenses, so on. So, it is very difficult to distinguish between VCOP and we VAUX unless the word morphologically changes in this two situations, from the context it can become very difficult many times, will see examples later disambiguation examples.

(Refer Slide Time: 18:10)

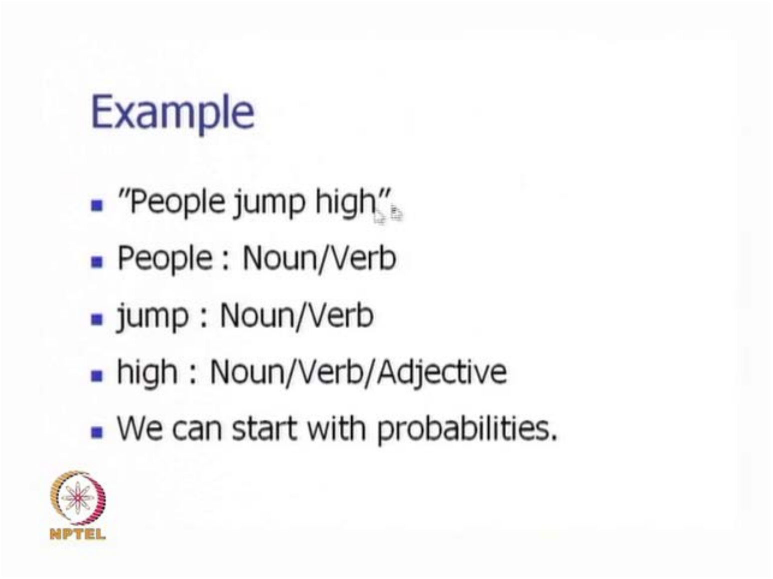


Process

- List all possible tag for each word in sentence.
- Choose best suitable tag sequence.

Now, we discuss the process of parts of speech tagging this is a computational process. So, you have understood what the problem is and now, where entering to the computational discussion, we would like to understand how one could make an algorithm to produce these labels automatically that is a task. So, that process is that, list all possible tag for each word in the sentence and choose the best possible tag sequence.

(Refer Slide Time: 18:40)

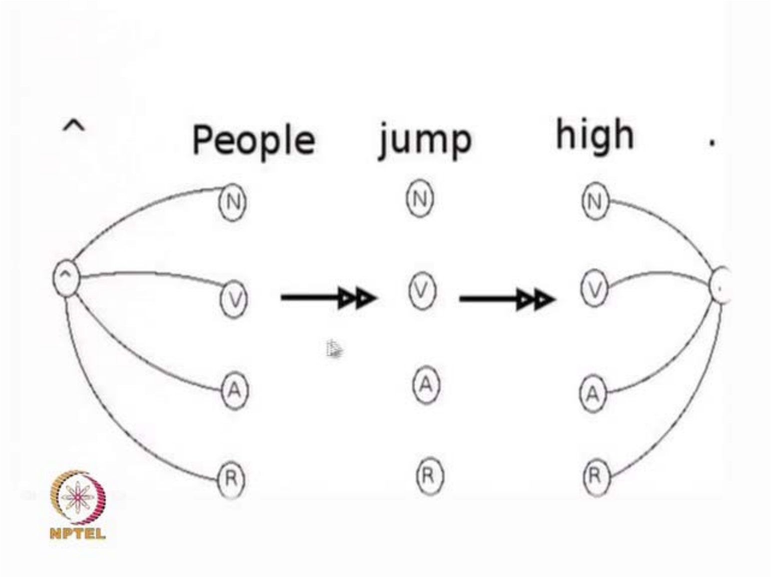


Example

- "People jump high"
- People : Noun/Verb
- jump : Noun/Verb
- high : Noun/Verb/Adjective
- We can start with probabilities.

We have an illustration here, people jump high is a sentence, people can be noun or verb jump can be also noun or verb, high can be noun verb adjective and we actually start with probabilities. So, people jump high will finally, get this level if the program is working correctly, it should get the level noun verb and adverb, actually high can be an adverb also noun verb and adverb. However, each word is multiple part of speech text and we will choose that tags sequence which is the highest probabilities, this is where the probabilistic approach. So, noun verb adverb tag will be the highest probability tag here.

(Refer Slide Time: 19:30)



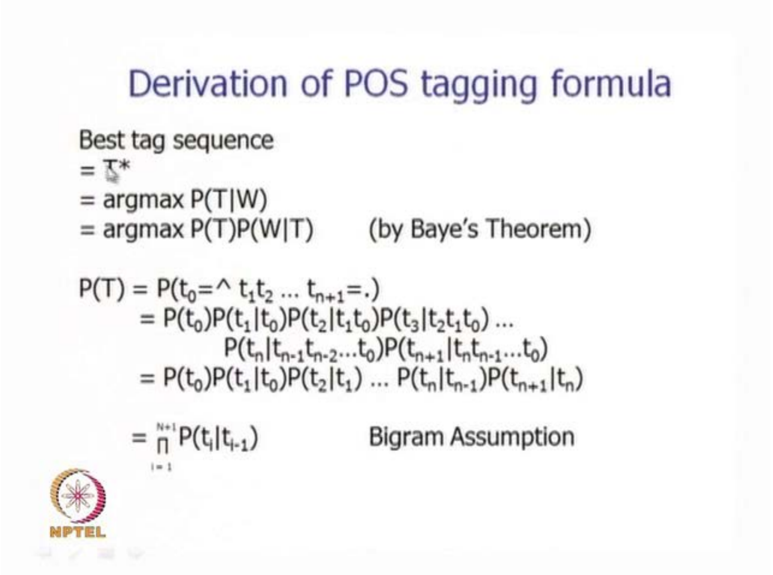
This is illustrated here, people jump high. So, this hat symbol is a important special symbol, which is at the beginning of a sentence dot is again a special symbol, which is at the end of a sentence also called pull stop and for each word we place all possible tags as columns of tags on the words. So, people can be noun verb, adjective adverb jump can be noun verb adjective adverb, I can be noun verb adjective adverb.

So, off course you will object saying that people can never been an adverb, jump how can it be adjective and, so on. But, this is only for the purpose of keeping the discussion simple and also illustrating, how a simple algorithm would work now, how a first cut algorithm would work. The first cut algorithm will not try to be intelligent right from the beginning, it will try to take all possible options or all possible tags for each word and then do a disambiguation or selections. So, if you look at this picture here, then for hat symbol conventionally the tag is hat itself. So, for the hat symbol hat, people it can have all this four possible tags, which is from the whole tag deprecatory of course, people can would be a adverb, so this will come out to be probability is 0, jump can have all this tags I can have all this tags. Now, when we connect all this tags together.

So, this hat is connected to 4 and is connected to the 4 tags after it v is to connected to all the 4 tags after it a similarly, to all the 4 tags or again to all the 4 tags. So, from here you can see 16 arcs go to this 4 tags for the next what. Similarly, from here there would be 4 into 4, 16 times going into the next words tags. So, this way we will define a graph and on part of speech tagging problem or part of speech tagging determination problem, becomes finding a path, through this whole graph.

And this graph is the set of levels, which should be placed on the words. Now, this graph traversal before we being discussing the mathematical aspects, this graph traversal would finally, find out the best possible tags sequence the best possible levels for the words, what does this best possible mean, the best possible means the highest probability path from the hat symbol to the dot symbol. So, these highest probability path taking only one level per word, is found out by the odd math's computation. So, the best possible path here means the higher probabilities path.

(Refer Slide Time: 22:57)



Derivation of POS tagging formula

Best tag sequence
$= T^*$
$= \text{argmax } P(T|W)$
$= \text{argmax } P(T)P(W|T)$ (by Baye's Theorem)

$P(T) = P(t_0{=}\wedge\ t_1 t_2 \ldots t_{n+1}{=}.)$
$= P(t_0)P(t_1|t_0)P(t_2|t_1 t_0)P(t_3|t_2 t_1 t_0) \ldots$
$P(t_n|t_{n-1} t_{n-2} \ldots t_0)P(t_{n+1}|t_n t_{n-1} \ldots t_0)$
$= P(t_0)P(t_1|t_0)P(t_2|t_1) \ldots P(t_n|t_{n-1})P(t_{n+1}|t_n)$

$= \prod_{i=1}^{N+1} P(t_i|t_{i-1})$ Bigram Assumption

So, this mathematics is based on an argmax computation, on which we have already spend quite in amount of tag. And you will see that, this fundamental ideas or broad to be a here. So, the best tag sequence we call as T star, T star is the best possible tag sequence, T star is found out from argmax of P T given W where, W is the word sequence and T is the tag sequence. So, if we please the tags on the set of words and we have the sequence T and the word sequence is W.

Here, we apply baye’s theorem and we convert this expression into P T into P W given T. Now, this is also an old point discuss many times in previous lectures that it is the problem, which determines whether we should apply baye’s theorem or not and there by, adopt a generative approach or a discriminative approach. Now, in this case the part of speech tag is determine through a generative approach, P T given W is converted to P T the prior probability of the tag sequence into P W the likely would probability of the word sequence given the tag sequence.

So, this particular part may sink counter intuitive or one intuitive, one would think what is this P W given T because our very natural thinking about this problem is that given the word sequence, we would like to find out the tags sequence. However, has we remark before, we convert this probabilities in to this two parts, prior probabilities and the likely would and the probability, acts a nice filter, to eliminate bad possibilities. So, these tags sequence P T is a representation for highly likely tags sequences as learnt from the corpora.

So, this point I suppose is clear to you, we have the prior probability which many times acts as a model an ideal model, for the label sequence that we reproduce. So, the P T part in the formula is a filter, which helps isolate bad tag sequences and which gives weight ages to good tag sequences, we understand this in a minute the only counter intuitive part there is the P W given T. And that came because we wanted to apply baye’s theorem and we will have to make some engineering judgments, take some engineering steps to make use of this expression P W given T.

So, let us go ahead with a mathematical formulations P T is written as P t 0 equal to hat followed by t 1, followed by t 2, followed by t 3 and, so on. Until we have t n plus 1 which is equal to dot. So, we have here this tag sequence t is t 0, t 1 up to t n plus 1, these now can be broken down into a number of expressions, number of probability values by applying what is call the chain rule, which is a fundamental operation in a probability.

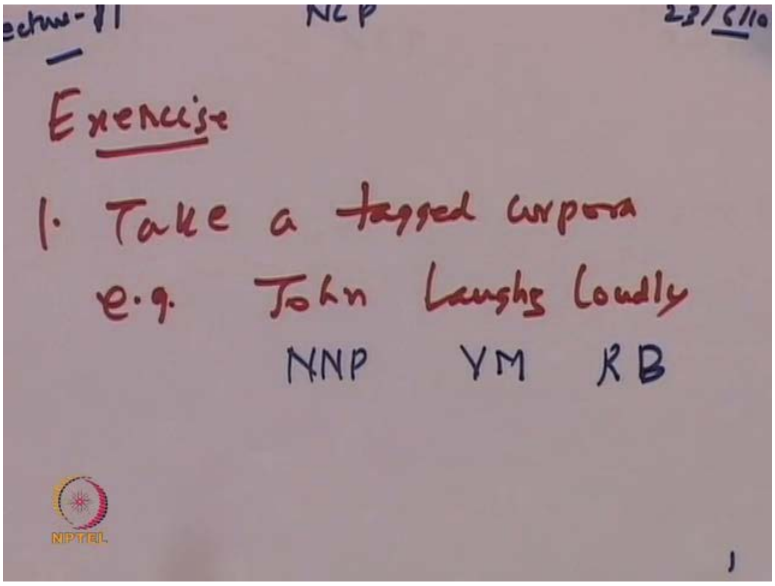
So, what you have done is that we have apply baye’s theorem, first fundamental operation and second fundamental operation is this chain rule, which is P t 0 into P t 1 given t 0 P t 2 given t 1 t 0 P t 3 given t 2 t 1 t 0 and, so on. Until we meet our last expression which is P t n plus 1 given t n t n minus 1 up to t 0. So, this whole probability of the sequence is converted into probabilities of single levels, given the presiding tag levels alright.

So, baye’s theorem applied, chain rule applied now we apply what is called the markovian assumption. And the expression that we have written here, in the last line P t i given t i minus 1 i going from 1 2 n plus 1, this actually is a bigram markovian assumption, what does it say, it says that fine apply the chain rule get the expressions sub expression the constituent triviality values. However, this terms very complicated to compute.

So, P t n given t n minus 1 t n minus up to t 0, we have to deal n terms here, in the conditioning point what the markovian assumption is saying is that, this regard anything which is very distant from the current tag, if the current tag is extremely faraway it does not have influences on the tag at the current position. So, P t 2 given t 1 t 0 is we are saying that at the second position, the tag is t 2 depending on the two tags, just before it and anything beyond that is not useful to compute to consider.

So, if I take P t 5 for example, the tag at the fifth sequence, the only influencing factor on this is t 4 and t 3 the only the first two previous tags. Here, we have given a very simple expression, we have made what is call the bigram assumption, we are saying that each tag depends only on the previous tag, on the tag coming before it which is the condition a part. So, I would present you with a small exercises and request you to do this to be convinced of the bi gram assumption or the markovian assumption.

(Refer Slide Time: 29:49)



ecture-11 NLP 23/6/10

Exercise

1. Take a tagged corpora

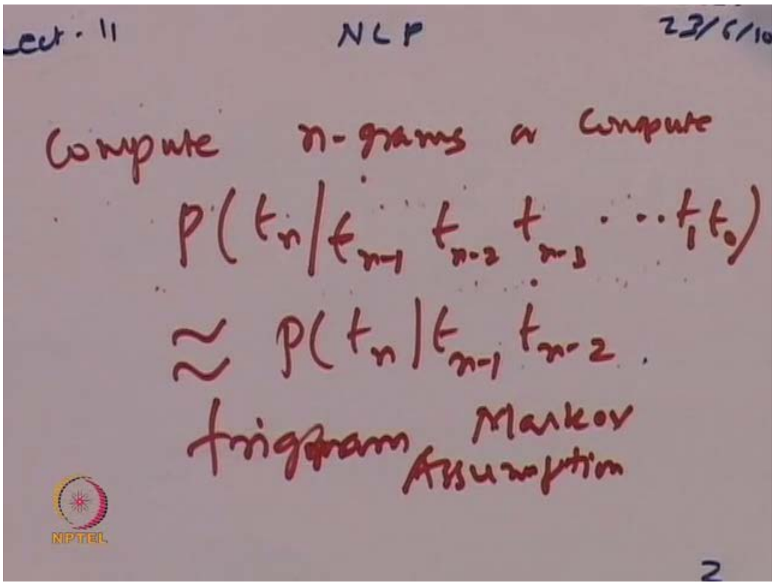
e.g. John Laughs Loudly

NNP VM RB

1

So, the exercise is this take a tagged corpora for example, john laughs loudly, so here the tags are john NNP laughs is a verb, which is a main verb VM, loudly is adverb RB. So, take it at corpses you will have this kind of tags along with the words.

(Refer Slide Time: 30:23)



Lect. 11 NLP 23/6/10

Compute n-grams or Compute

$P(t_n / t_{n-1} t_{n-2} t_{n-3} \cdots t_1 t_0)$

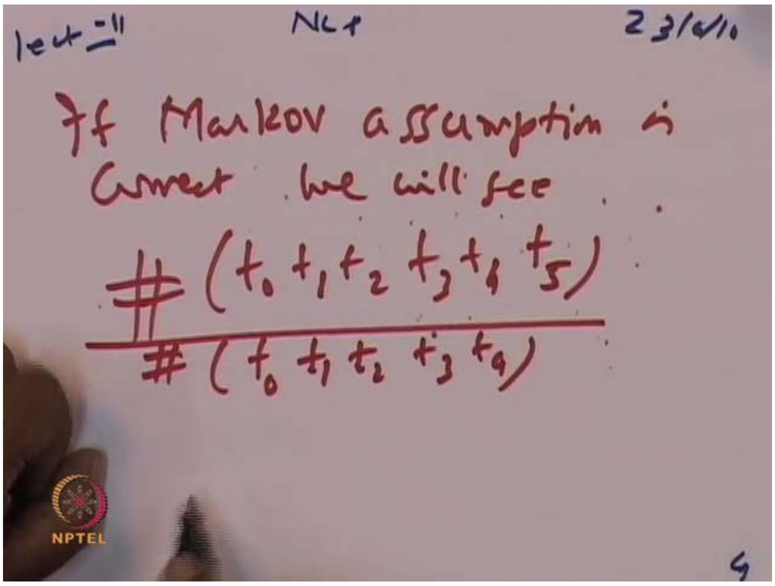
$\approx P(t_n | t_{n-1} t_{n-2}$

trigram Markov Assumption

2

After this from this tag corpses, compute n grams or compute P t n given t n minus 1 t n minus 2 t n minus 3 up to t 0, this we are saying is almost equivalent to P t n given t n minus 1 t n minus 2 this is trigram mark of assumption.

(Refer Slide Time: 31:09)



lect-11 NLP 23/8/10

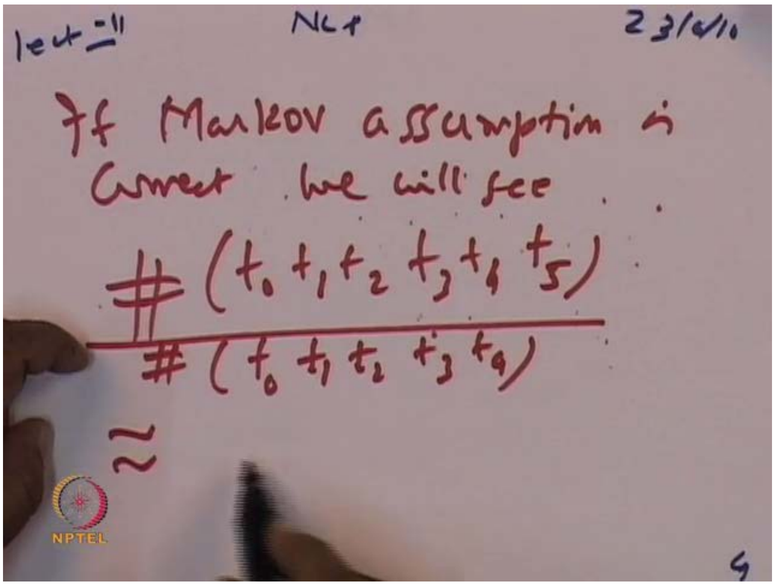
If Markov assumption is correct we will see

$$\frac{\#(t_0\,t_1\,t_2\,t_3\,t_4\,t_5)}{\#(t_0\,t_1\,t_2\,t_3\,t_4)}$$

5

So, what I request you to do is take P t let say 5 and t 4, t 3, t 2, t 1, t 0 which you know, is equal to count of the pattern t 0 t 1 t 2 t 3 t 4 t 5 divided by count of t 0 t 1 t 2 t 3 and t 4.

(Refer Slide Time: 31:48)



lect-11 NLP 23/8/10

If Markov assumption is correct we will see

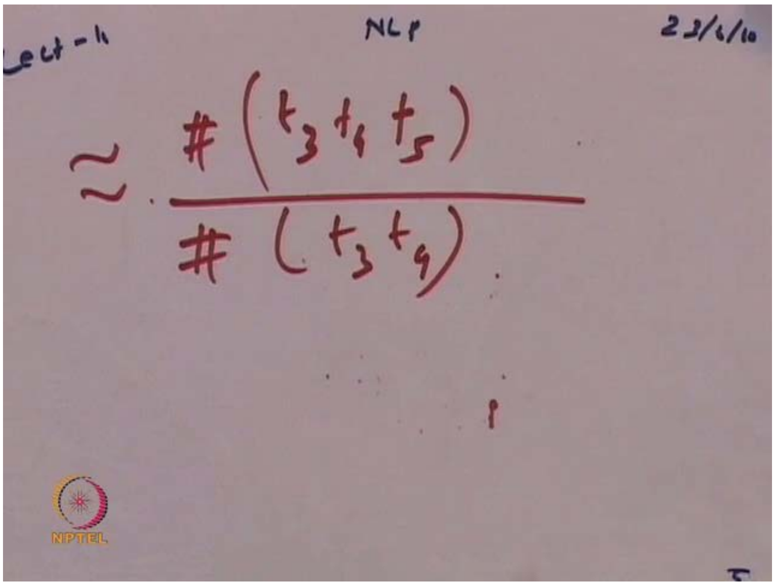
$$\frac{\#(t_0\,t_1\,t_2\,t_3\,t_4\,t_5)}{\#(t_0\,t_1\,t_2\,t_3\,t_4)}$$

$$\approx$$

5

If our assumption is correct if the mark of assumption is correct, we will see count of t 0 t 1 t 2 t 3 t 4 t 5 divided by count of t 0 t 1 t 2 t 3 t 4 will be approximately equal to we take it on the next page.

(Refer Slide Time: 32:20)

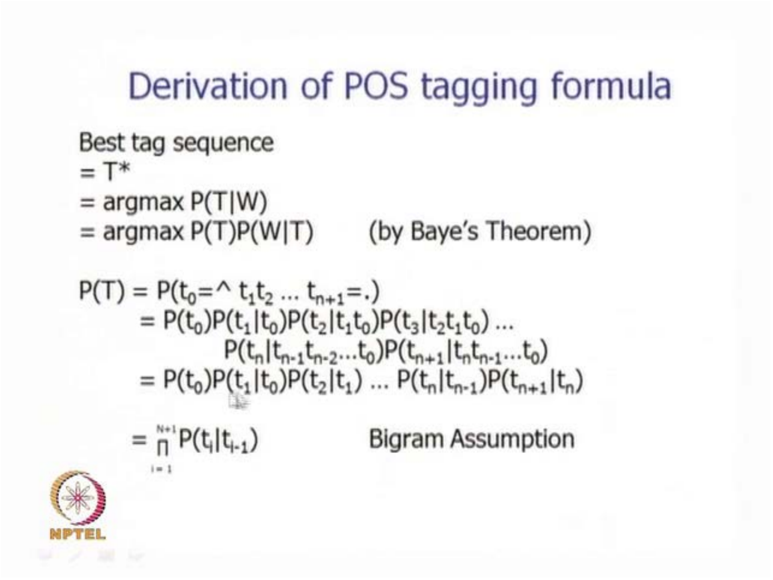


Lect-11 NLP 23/6/10

$$\approx \frac{\#(t_3 t_4 t_5)}{\#(t_3 t_4)}$$

T 5 that count divided by count of t 3 t 4 alright. So, this whole sequence t 0 to t 5 divided by t 0 to t 4 is approximated as t 3 to t 5 when t 3 to t 4. So, that is the bigram assumption and if are markovian assumption is correct, this to count should come out to be approximately equal telling us that mean for a particular tag at a position or for a tag at a particular position, we did not consider any word which is beyond a certain distances that is the meaning. So, this you should verify from a training corpuses, we will mention important corpora tag corpora towards the end of the lecture and it should be possible for you to verify this quite easily, but you should do this to be convinced of the operation of the markovian assumption alright.

(Refer Slide Time: 33:26)



Derivation of POS tagging formula

Best tag sequence

$= T^*$

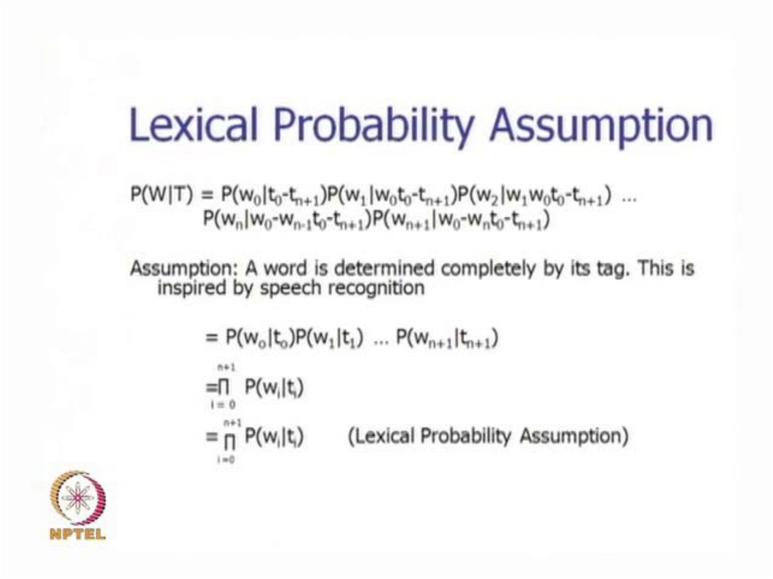
$= \text{argmax}\, P(T|W)$

$= \text{argmax}\, P(T)P(W|T)$ (by Baye's Theorem)

$$P(T) = P(t_0 = \hat{}\ t_1 t_2 \ldots t_{n+1} = .)$$
$$= P(t_0)P(t_1|t_0)P(t_2|t_1 t_0)P(t_3|t_2 t_1 t_0) \ldots P(t_n|t_{n-1}t_{n-2}\ldots t_0)P(t_{n+1}|t_n t_{n-1}\ldots t_0)$$
$$= P(t_0)P(t_1|t_0)P(t_2|t_1) \ldots P(t_n|t_{n-1})P(t_{n+1}|t_n)$$
$$= \prod_{i=1}^{N+1} P(t_i|t_{i-1}) \qquad \text{Bigram Assumption}$$

Proceeding further we see that P t is equal to this expression and after making bigram assumption it comes out to be equal to P t i given t i minus 1 i going from 1 to n plus 1. So, this first part of the expression the prior probability comes out to be product of bigram entities P t i, t i minus 1 is nothing but count of the number of times t i is followed by the t i minus 1 divided by number of times t i minus 1 appears.

(Refer Slide Time: 34:09)



Lexical Probability Assumption

$$P(W|T) = P(w_0|t_0\text{-}t_{n+1})P(w_1|w_0 t_0\text{-}t_{n+1})P(w_2|w_1 w_0 t_0\text{-}t_{n+1}) \ldots P(w_n|w_0\text{-}w_{n-1}t_0\text{-}t_{n+1})P(w_{n+1}|w_0\text{-}w_n t_0\text{-}t_{n+1})$$

Assumption: A word is determined completely by its tag. This is inspired by speech recognition

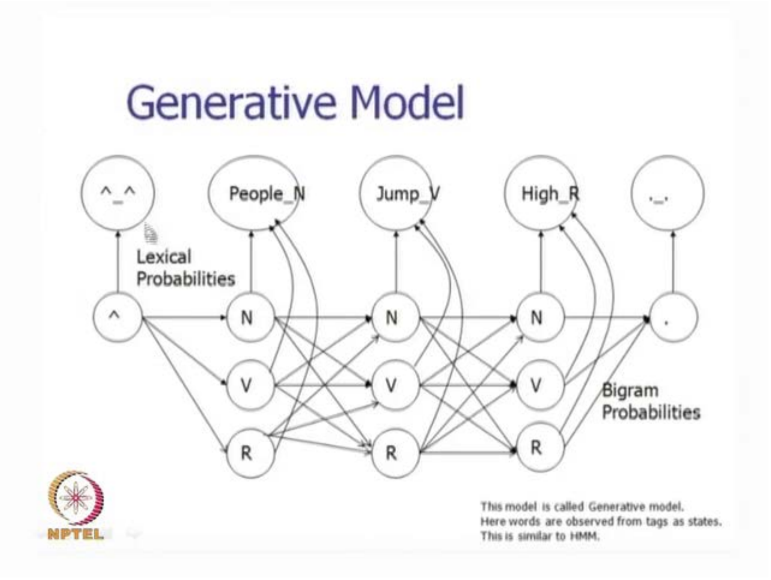
$$= P(w_o|t_o)P(w_1|t_1) \ldots P(w_{n+1}|t_{n+1})$$
$$= \prod_{i=0}^{n+1} P(w_i|t_i)$$
$$= \prod_{i=0}^{n+1} P(w_i|t_i) \qquad \text{(Lexical Probability Assumption)}$$

Alright we move on to the next probability, which is the lexical probability. So, P W given T is shown here, which is equal to P W 0 given t 0 to t n plus 1, t 0 to t n plus 1 is nothing but capital T, P W 1 given W 0 and capital T. This the chain P W 2 is nothing but P W 1 W 0 and the whole tags sequence finally, we have the two expressions here, P W n given W 0 W n minus 1 and the whole tags sequence t 0 t n plus 1, the last expression is P W n plus 1 which is condition by W 0 to W n and the whole tags sequence t 0 to t n plus 1.

Now, we make an assumption which does not have much of linguistic support, it is more like an engineering decision for the purpose of computation. However, the results of part of speech tagging shows that this is a not a bad assumption after all, the assumption here is that a word is determine completely by it is tag, this is inspired by speech recognition. So, I have attired a set of words now, I know what the part of speech tag of the next word most likely to be and given that is the part of speech tag which will appear after the sequence of words, what is the probability of a particular word coming there.

So, if I have sentence I see a black and now, it is clear that the most likely after black is a noun, so now, what is the probability of attiring I see a black cow or I see a black dog or I see a black umbrella. So, dog, umbrella, cow all this are nouns which are lexical items, what is the probability they can appear at that position the last word position, given that the tag there is a noun. So, this kind of word prediction problems are quit common in daily of speech recognition and the lexical probability assumption is inspired by. So, if you look at the expression here, t w given t has been converted in to a chain rule expression, after that by making this assumption that a word is dependent only on the tag at the position, we obtain P W 0 given t 0 into P W 1 given t 1 up to P W n plus 1 given t n plus 1. So, this comes out to be equal to i equal to i going from 0 n plus 1 P W i given t i.

(Refer Slide Time: 37:35)



Generative Model

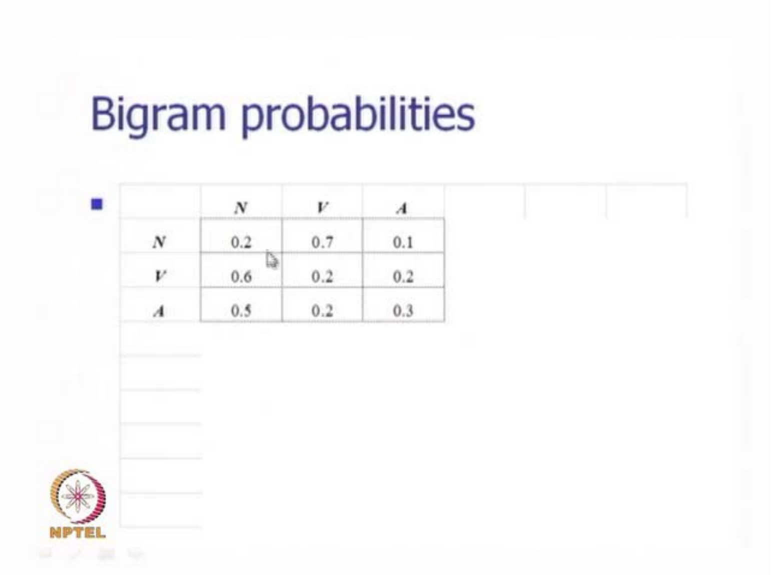
This model is called Generative model.
Here words are observed from tags as states.
This is similar to HMM.

Now, this is giving rise to what is called a generative model, if we have this sentence people jump high preceded by sentences begin a hat and the sentence end there dot. Then we have here, the tag hat for the first symbol which is hat and the dot tag for the last symbol dot, in between we have noun verb adverb for people noun verb adverb for jump, high verb adverb for high. So, you can see that only one tag is the correct tag in the context of the sentence, people is a noun, this noun will be chosen, jump is verb we chose this verb, high is an adverb we chose this adverb which qualifies the verb.

So, people jump high would get the tags N V R preceded by hat followed by dot. Now, this model is called generative. Because, you can see that from the automaton, we find the words are generated by the tags it is as if the tags generate the words that‘s why it called a generative model, the second part of the probability expression the likely would probability P W given t, which is converted in to product of P W i given t i is the generative model part of the computation, as if the words are generated from the tag.

So, we go through this automaton and we have the probabilities of the arcs from tag to tag, this is known as the bigram probability P t i given t i minus 1 and we have the lexical probabilities, probability of people given n, probability of people given verb, probability of people given adverb, this are the lexical probabilities and the R probabilities between two tags is call the bigram probability are the transition probability alright. So, when we make a traversal from hat to dot, we find out for very word a tag and the tag sequence we get is the best possible tag sequence, in the sense of being the highest probability path.

(Refer Slide Time: 40:16)

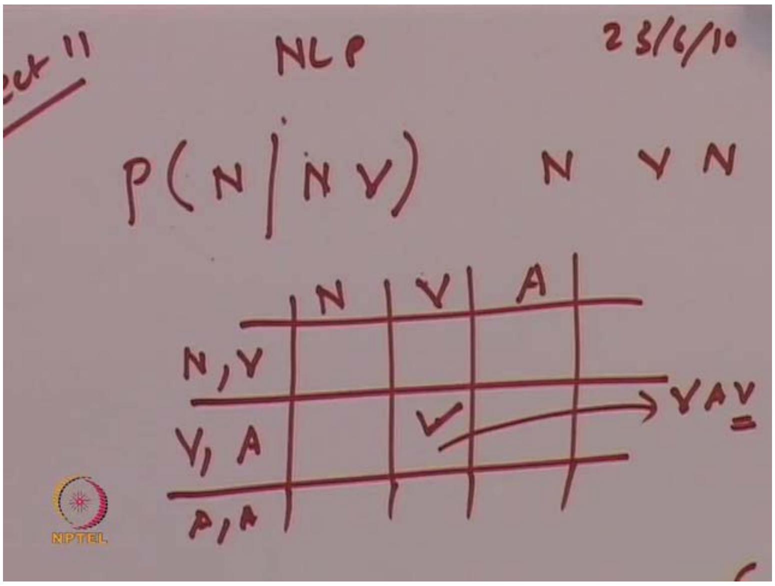


Bigram probabilities

	N	*V*	*A*
N	0.2	0.7	0.1
V	0.6	0.2	0.2
A	0.5	0.2	0.3

So, this here shows how the bigram probabilities are represented is an important data structure, which it is very useful to remember you see on the column we have the tags N V A noun verb, adjective let say in the row also we have N V A. Every cell in this matrix denotes the transition triviality, since it is a bigram probability situation, where we say that every tag depends only on the previous tag; we have a matrix where the column or tags and rows are also single tags.

(Refer Slide Time: 41:10)



Lect 11

NLP

23/6/10

$P(N|NV)$ N V N

	N	V	A
N, V			
V, A		✓	
A, A			

→ VAV

If it was a trigram probability situation consider a trigram probability situation. Suppose we have the trigram probability situation and we have P of a noun given that the previous two tags were noun and verb. So, the situation is N V N, so when you have this kind of trigram situation, we again have a matrix of transition probabilities noun verb adjective; however, on the row we have the conditioning part, which is a pair of tags. So, would have N comma V and then we have V comma A let say we can have A comma A.

So, what is the meaning of this the meaning of this suppose, I take up this cell supposes this cell is our cell of attention, here will the place probability of a verb being preceded by a verb and adjective that is this sequence V A V. So, the row will have tupelos instead of single tags, if is a bigram then you have single tags on the rows. So, is it clear the columns will always have tags, single tags, rows will have pairs triples quadruples, entaple, depending on the markovian assumption we make.

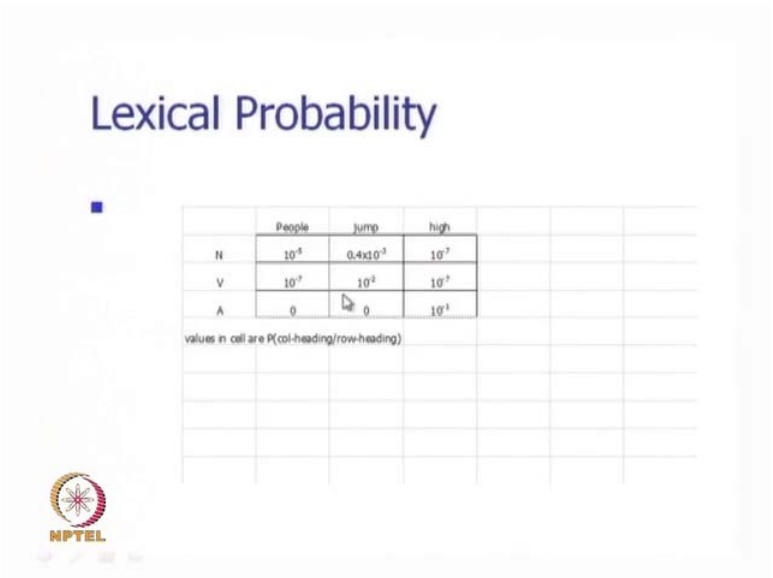
If the assumption is a quadric gram assumption a tag depends on the previous three tags. So, the whole thing is a four tap let, three previous t tags than on the row we will have three symbols triples, so this is about the transition probability, the lexical probability table also we look similar it will be a matrix. So, in the situation we have let say noun verb and adjectives, which are on the rows on the column we will have actual words.

So, how many columns will this matrix have this matrix will have as many words as there are in the language, are if we want to be more conservative we can say that the number of columns will be equal to the number of distinct words the corpus. So, now, if you look at the cell n and people, this is 10 to the power of minus 5, this means what is the probability of the word people given that the tag is went. That means, at a particular position, what is the probability that the word is people given that the tag at the position is noun.

Now, this how will you compute you will clearly compute this, by finding out how many times the noun tag appears, have it in the denominator and let it divided how many times does people appear in those noun positions. So, probability of people of given n, so how many nouns are there in the corpora and out of them how many are people that ratio is the is the probability value. So, this shows the 10 to the power of minus 5, which means there are many, many nouns in the corpus.

And people appears one once out of 10 to the power 5 nouns, for every 10 to the power 5 nouns one is a people. So, this the meaning of this probability, we can see some zeros here what is the probability of people given adjective, so are what is the probability of jump given adjective. So, this kind of probabilities are very low almost nil jump has never possible been used in that purpose as an adjective.

(Refer Slide Time: 45:27)



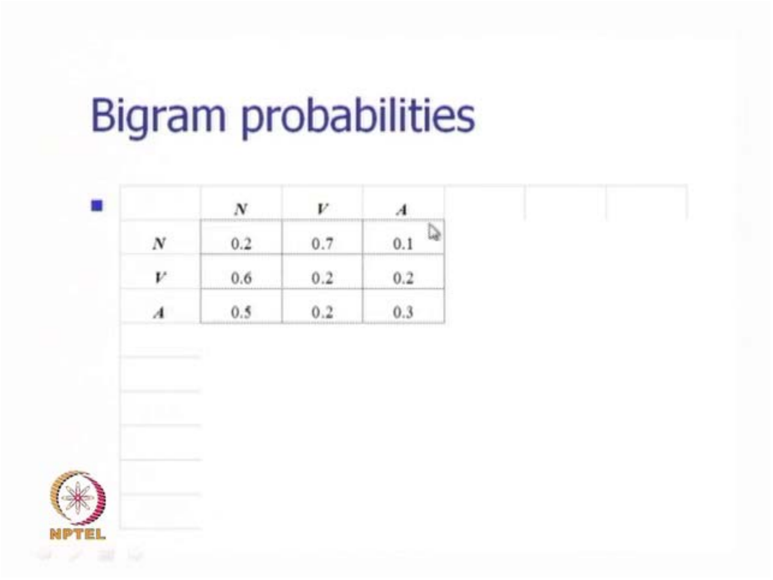
Lexical Probability

	People	jump	high
N	10^{-5}	$0.4x10^{-3}$	10^{-7}
V	10^{-7}	10^{-2}	10^{-7}
A	0	0	10^{-1}

values in cell are P(col-heading/row-heading)

So, I suppose this is clear, so let me just repeat this two data structures once again, they are very impotent for our understanding lexical probability is given by the lexical probability matrix.

(Refer Slide Time: 45:36)

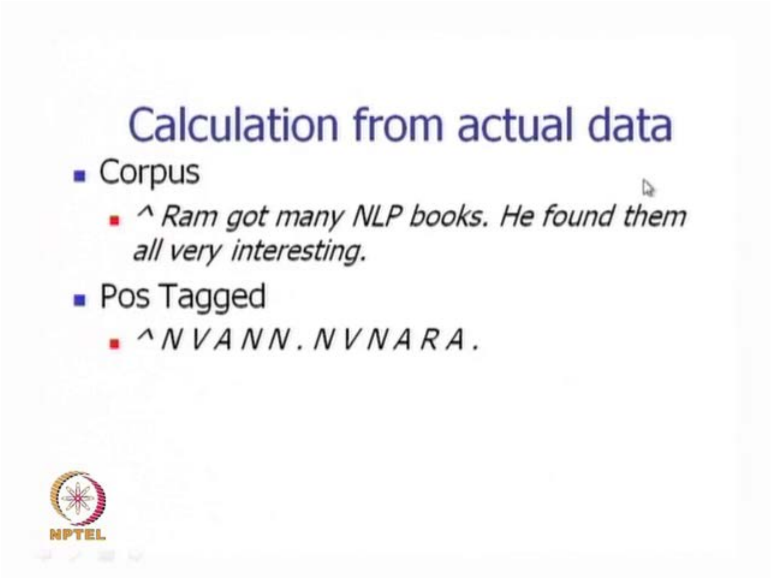


Bigram probabilities

	N	*V*	*A*
N	0.2	0.7	0.1
V	0.6	0.2	0.2
A	0.5	0.2	0.3

And in the previous slide, bigram probability is given by this transition probability matrix.

(Refer Slide Time: 45:44)



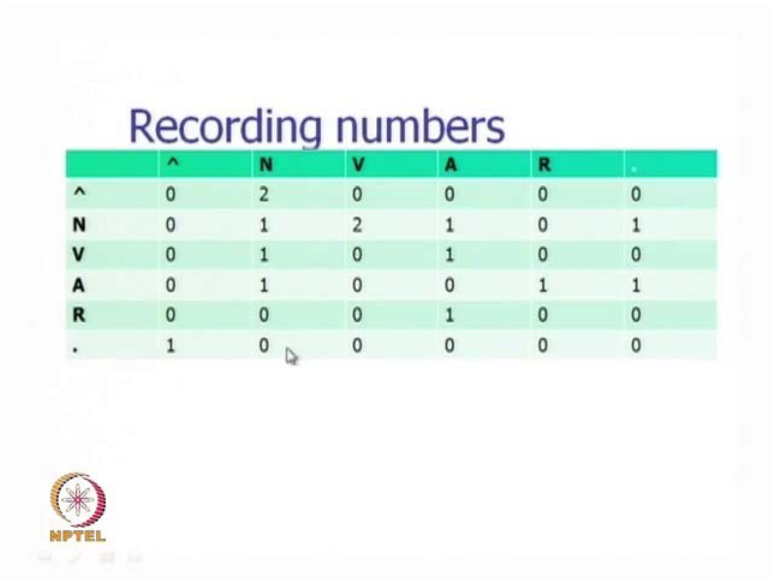
Calculation from actual data

- Corpus
  - *^ Ram got many NLP books. He found them all very interesting.*
- Pos Tagged
  - *^ N V A N N . N V N A R A .*

Now, how do we calculate this probability values is from the actual data, what have we done, so for just recapitulation what we done, so for is that we have set the part of speech time problem as a sequences leveling task used or max computation. Apply baye's theorem divided in to two parts probability of probability of the tag sequence, multiplied by the likely would probability of likely would of the word given the words sequence given the tag sequence, we applied markovian assumption.

And then we obtain P t i given t i minus 1 which is the bigram assumption, then we obtain probability of w i given t i which is the lexical probability assumption. So, markovian assumption preceded by baye’s theorem application chain rule, know this probability gave rise to the transition probability table, also the lexical probability table. And how do we do the calculation from the actual data, this will discuss in the next class, but you can see, this is a raw piece of text and the whole thing is pos tag.

(Refer Slide Time: 47:03)



Recording numbers

	^	N	V	A	R	.
^	0	2	0	0	0	0
N	0	1	2	1	0	1
V	0	1	0	1	0	0
A	0	1	0	0	1	1
R	0	0	0	1	0	0
.	1	0	0	0	0	0

From the pos tag we record the numbers and then compute the probabilities, we will do the calculation in the next lecture.