

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

Lecture - 12
Part of Speech Tagging contd...
and
Indian Language in Focus; Morphology Analysis

Today, we will continue discussing Part of Speech Tagging, which we have remarked many times is a very important step in Natural Language Processing, and is the first annotation task, which produces level on the raw text. The focus today is also on processing of Indian languages, which are rich in morphology, and therefore we have to understand the morphological processors for Indian language.

(Refer Slide Time: 00:57)



Process

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




We will take two examples that of Hindi and Marathi, continuing with a discussion on part of speech tagging, we have seen that the process is listing all possible tag for each word in the sentence, and we choose the best suitable tag sequence. Now, when we say list all possible tags this is only one of the options, there are other ways of starting the part of speech tagging process, where we take only the feasible tags for a word. A word exists in the lexicon and the lexicon records possible categories for the word, so one could begin with the tags which are placed in the lexicon, the method which is described

here is a simple one, where we produce all the tags for a word or we start from all possible tags for a word and we disambiguate from the context.

(Refer Slide Time: 01:54)

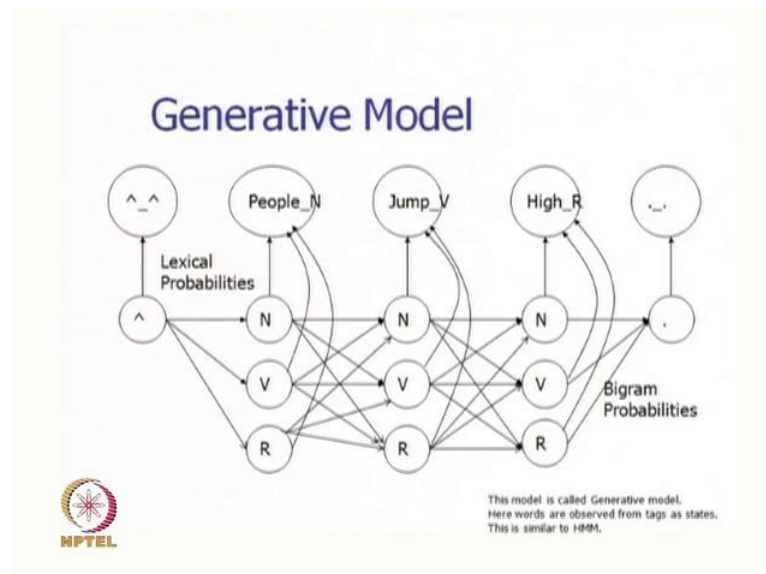
Example

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



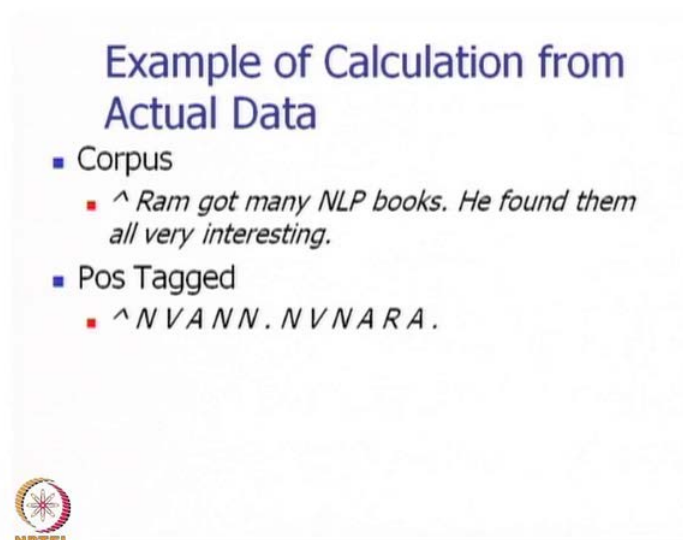
Here, is an example people jump high, people can be both noun and verb, jump can be noun and verb, high can be noun and adjective, so here we have the possibility of two tags for every word. And therefore, there are there are a number of tag sequences which are possible to be specific, there are 8 possible sequences for this sentence and we have to find out the best possible sequence probabilistically speaking.

(Refer Slide Time: 02:34)




The generative model shows that we can produce the tags for the word sequence by constructing a machine, which is shown in the diagram here, an automaton to be more specific. The starting of the sentence is indicated by the hat symbol, and the tag for the hat symbol is the hat symbol itself. Similarly, the sentence ends with a full stop and the tag for the full stop is also the full stop, when we come to the words people we have seen can be noun or verb. And R stands for adverb jump can be noun or verb R stands for adverb, again high can be noun or verb R stands for adverb and what we have here is a set of hypothetical tags for these words and we have to choose the best possible tag sequence.

(Refer Slide Time: 03:48)



Example of 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, in our previous lecture, we saw that the mathematics of part of speech tagging can be looked up on as an argmax computation, just to remind you we can look at the formula.

(Refer Slide Time: 04:08)

To find

- $T^* = \underset{T}{\operatorname{argmax}} (P(T) P(W/T))$
- $P(T).P(W/T) = \prod_{i=1 \rightarrow n+1} P(t_i / t_{i-1}).P(w_i / t_i)$
- $P(t_i / t_{i-1})$: Bigram probability
- $P(w_i / t_i)$: Lexical probability

Note: $P(w_i / t_i) = 1$ for $i=0$ (^, sentence beginner) and $i=(n+1)$ (., fullstop)



We have here the expression T^* equal to argmax of $P(T) P(W/T)$, the original expression is probability of the tag sequence given the word sequence, now we maximize this over the possible tag sequences. The expression $P(T) P(W/T)$ comes after the application of the base theorem, which converge $P(T) P(W/T)$ to $P(T) P(W/T)$. $P(T)$ is the prior probability and $P(W/T)$ is the likelihood probability. So, this if you remember came from the desire to score the tag sequences, the tag sequence which is the best possible tag sequence comes from the argmax computation.

Now, $P(T) P(W/T)$ after mathematical operations, that in the last class comes out to be equal to product of $p(t_i / t_{i-1})$ into $p(w_i / t_i)$. This came from the derivation that $p(t_i)$ can be modeled as a by t_{i-1} it could be diagram also in, which case this expression would be $p(t_i / t_{i-1})$. And the lexical probability $p(w_i / t_i)$ comes from the assumption of lexical independence, which is that at a position the word is determined completely by the tag at that position.


So, this part of the formula says that a tag at a position depends only on the previous tag, and this says that the word at a position is a function of only the tag at that position, so $p(t_i / t_{i-1})$ is the bigram probability, and $p(w_i / t_i)$ is the lexical probability. So, we note that $p(w_i / t_i) = 1$ for $i = 0$, where the lexical item is the hat symbol, and in it indicates the sentence beginner and at $i = N + 1$, that means the last position the tag is the dot and the lexical entity there is full stop.

So, this particular derivation was done in two classes if you remember, and the application of basin theorem converted p_t given w into 2 probabilities, the prior probability p_t into P_W given T , so this was treated for bigrams and lexical probabilities and finally, we got that product expression. So, if we look at the formula we find that we need to compute the probabilities p_{t_i} given t_{i-1} and probability w_i given t_i , these probabilities are obtained from the corpora, we will see very soon how to do it. Now, these corpora are all pos type corpora the annotated corpora and they produce the accounts for computing these probabilities.

(Refer Slide Time: 07:51)

Example of 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 .*



So, we take an example here example of calculation from actual data, suppose our corpus contains only one sentence, which is ram got many NLP books, ram is the name of a person got many NLP books, NLP stands for Natural Language Processing, he found them all very interesting. Now, in this case when we produce the tags which a very simple minimal set of tags, so hat corresponds to the tag hat ram is a noun, it is a proper noun, but we are using the general symbol for noun here noun got is a verb.

So, we many is an adjective, so a NLP is the name of the subject, so this is N books is also noun which is N full stop the tag is full stop, he is a pronoun, but for simplicity we look up on this as a noun. So, noun found is a verb, so V then is a pronoun we put this an noun all is A is an adjective, so A very is an adverb, so R and interesting is again adjective, so A. So, this whole sequence is tagged as hat N V A N N dot N V N A R A


and then dot, so suppose, this is the corpus and we need to find out. The probabilities $p(t_i | t_{i-1})$ and probability of w_i given t_i ; that means, the probability of a word even the tag at a particular position and the probability of a tag even the previous tag, so question is how do we find this out?

(Refer Slide Time: 09:53)

Recording numbers (bigram assumption)

	^	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

^ Ram got many NLP books. He found
 them all very interesting.
 Pos Tagged
 ^ NVANN. NVNARA.




When you go the next slide we find that we can record the numbers with respect to the bigrams, so we have the entities here, which are hat noun, verb, adjective and adverb, which are placed on these columns, 1 2 3 4 5 6 columns. Hat is the sentence beginner N is the noun, V is the verb a is the adjective R is the adverb dot is the full stop, so same tags are placed on the column hat noun verb adjective adverb and dot.

So, this table can be used, this matrix can be used to record the bigram frequencies, the meaning is as follows what is the number of times that a tag in the column is preceded by a tag which is show in the row. So, rows are preceding tags, columns are the following tags, so how many times is tag preceded by a tag 0 tag, in the corpus we do not have any evidence of tag being a hat being followed by another hat, how many times is noun preceded by a hat symbol, let us go into the corpus.

(Refer Slide Time: 11:26)

Example of 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 .




So, we find that here we have the hat symbol which begins the sentence, and after that we have a noun all sentences implicitly or explicitly have the hat symbol. So, hat a noun here, similarly here is a noun which comes which begins a sentence this is implicitly preceded by a hat symbol, we have not shown it here. So, there are two cases where the noun is preceded by hat therefore, we have recorded in the table.

(Refer Slide Time: 11:59)

Recording numbers (bigram assumption)

	^	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

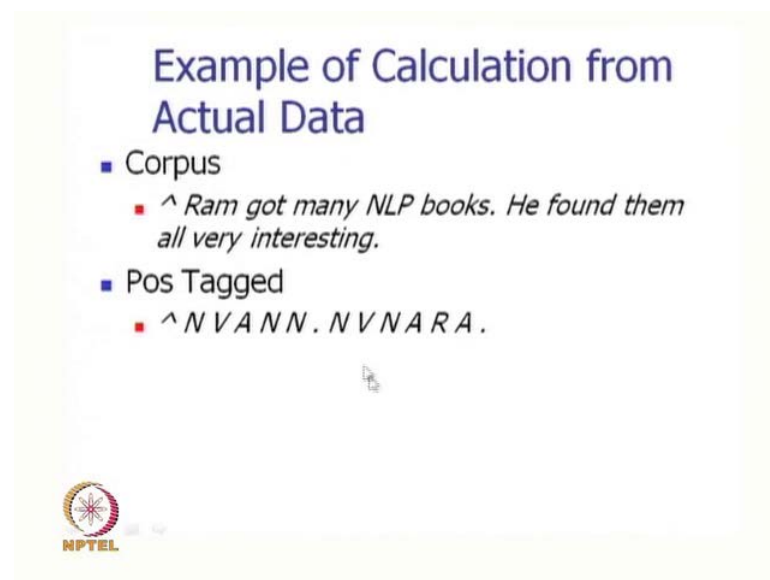
^ Ram got many NLP books. He found them all very interesting.
Pos Tagged
^ N V A N N . N V N A R A .



N in the in the cell for N as the column and hat as the row 2, how many times do we have a verb preceded by a hat symbol; that means, is there any example of a verb beginning a


sentence, in the corpus there is no such example. So, therefore, V preceded by a hat is 0 similarly no adjective begins a sentence in the corpus, so a preceded by hat is 0, R preceded by hat is 0, and is there any empty sentence no. So, that would be indicated by a dot immediately coming after a hat symbol that is there is no such example, which is indicating that the cell has value 0. How many times is hat preceded by noun? So, that is not possible because we are always ending the sentence with a dot and therefore, we cannot have any example of a hat preceded by either noun verb adjective or adverb. So, all these are 0, how many times is hat preceded by a dot there is only one such example namely the second sentence here.

(Refer Slide Time: 13:13)



Example of 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 .*




He found them all very interesting, here he starts a sentence and therefore, we see that we have a hat symbol implicitly not shown here and that is preceded by dot.

(Refer Slide Time: 13:29)

Recording numbers (bigram assumption)

	^	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

^ Ram got many NLP books. He found
them all very interesting.
Pos Tagged
^ NVANN. NVNARA.



So, we have here in this cell hat preceded by dot there is only one example of this, when we come to the noun column, how many times is noun preceded by noun? Let us see in the corpus noun preceded by noun, there is only one such example N which is NLP books NLP books. So, therefore, we have one such example we have recorded in the cell that noun is preceded by noun only once, how many times is noun preceded by verb only once, how many times is it preceded by adjective only once is it true.

So, we look at the corpus noun preceded by a verb is here found them, noun preceded by adjective also there is an example many NLP books. And therefore, this is recorded in the table, how many times is verb preceded by hat 0, we have already remarked verb preceded by noun two times from the corpus, rest of them are 0. Adjective preceded by hat is 0, adjective preceded by noun only once, adjective preceded by verb only once, again adjective preceded by R or adverb is only once.

So, this is the case of very interesting which is showing the number 1 here, is there any example of adjective preceded by noun, let us see the corpus adjective preceded by noun. Yes, here we have this and this is the case of them all, all is adjective it is preceded by noun N and so on. Similarly, when we go to the column for R we make this count R is preceded by adjective in one case, and then when we go to the dot symbol dot is preceded by noun once, and dot is preceded by adjective once. Is it true?

We see it is true, because we have two dots here in the corpus, in the first case dot is preceded by books, which is a noun, in the second case dot is preceded by interesting which is an adjective. So, this is the way we have tag the corpus and we are able to produce the entries in the table, so this is a very, very important step in the whole processing of the part of speech tagging.

So, this tag corpus produces the counts, and this counts are regarded in a bigram table or trigram gram table, we have already remark last time. If you see the table once again if the assumption is bigram, then we have columns and with these columns have corresponding rows, rows have only signal entities. If it was a trigram assumption then columns will have the tags as before; however, rows will have topples, because we have two preceding tags, which have to be considered.

Similarly, if it a quadrigram assumption then the rows will have trigrams, so in case of bigram we have columns and the rows are unigrams. So, this is an important discuss, this is the training phase for the part of speech tagger we have the corpora, and the corpora are marked with part of speech tags. Since, the whole scoring is based on an argmax computation, the quantity which produces the score is a probability expression there are two components in the probability expression the bigram probability for tags, and the lexical probability for words and tags. What we have discussed so far is the bigram probabilities, and they can come only from the tagged corpus, and I illustrate the process quite elaborately namely, we saw the sequences of noun followed, noun preceded by adjective or noun preceded by noun and so on.

(Refer Slide Time: 17:38)

Probabilities

	^	N	V	A	R	.
^	0	1	0	0	0	0
N	0	1/5	2/5	1/5	0	1/5
V	0	1/2	0	1/2	0	0
A	0	1/3	0	0	1/3	1/3
R	0	0	0	1	0	0
.	1	0	0	0	0	0

^ Ram got many NLP books. He found them all very interesting.
Pos Tagged
^ NVANN . NVNARA .




Let us move on to the calculation of probabilities, we have got the frequencies, which we have seen.

(Refer Slide Time: 17:42)

Recording numbers (bigram assumption)

	^	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

^ Ram got many NLP books. He found them all very interesting.
Pos Tagged
^ NVANN . NVNARA .




In the last table how do we get the probabilities?

(Refer Slide Time: 17:46)

Probabilities

	^	N	V	A	R	.
^	0	1	0	0	0	0
N	0	1/5	2/5	1/5	0	1/5
V	0	1/2	0	1/2	0	0
A	0	1/3	0	0	1/3	1/3
R	0	0	0	1	0	0
.	1	0	0	0	0	0


^ Ram got many NLP books. He found them all very interesting.
Pos Tagged
^ NVANN.NVNARA.



For the probabilities we for example, have to find out the probability of hat preceded by hat; that means, probability of a hat given that the previous tag was hat. Since, the count is 0, so the probability comes out to be 0, let us move on to some more interesting columns. So, we have here the nouns as the columns and we need to find out, what is the probability of noun given that the previous tag was hat, so how do you find it, let me write down the expression and this will be clear from that.

(Refer Slide Time: 18:34)

NLP 11/10

$$\begin{aligned} P(N/\wedge) &= \frac{\#(\wedge, N)}{\# \wedge} \\ &= \frac{2}{2} = 1 \end{aligned}$$


So, we have to find out probability of noun given hat, so this must be equal to number of times, we have the sequence hat and noun and number of times there are hat symbols. This we know happens two times noun is preceded by hat two times in the corpus hat, also appears two times therefore, this is equal to one. So, this is the way it is calculated, so we will remember this formula and we can look at the table now.

(Refer Slide Time: 19:09)

Probabilities

	^	N	V	A	R	.
^	0	1	0	0	0	0
N	0	1/5	2/5	1/5	0	1/5
V	0	1/2	0	1/2	0	0
A	0	1/3	0	0	1/3	1/3
R	0	0	0	1	0	0
.	1	0	0	0	0	0

^ Ram got many NLP books. He found
 them all very interesting.
 Pos Tagged
 ^ NVANN. NVNARA.



So, probability of noun preceded by the hat is 1 fine, how do we get this number 1 by 5, for this cell this is nothing but probability noun given that the previous tag was noun, so this is shown to be equal to 1 by 5. Why is it? So, we have to compute probability of N given N which is nothing but the sequence number of times the sequence N comma N occurs divided by number of N's. So, in the corpus you can see that N appears five times N N N and then N N.

So, there are 3 nouns in the first sentence 2 nouns in the second sentence, so 5 and the noun is following the noun only in one case namely this sequence. So, N N sequence appears only once, and there are 5 occurrences of noun. Therefore, the probability is 1 by 5, whatever the this cell, this cell is shown to be half which is probability noun given that the previous tag was a verb.

So, let us see this probability, we have to see the V N sequence, V N sequence appears only once and the denominator is V, how many times does a verb appear twice, the V N sequence appears only once namely this therefore, it is 1 by 2. What about probability of

N given that the preceding tag was adjective, what is this probability? This will be equal to number of times we have the adjective noun sequence, which is only once divided by how many times the adjective appears, the adjective appear 3 times A A and A therefore, we have 1 by 3.

So, I guess the method of calculating the probabilities is clear to you, even that the bigram sequence counts are there in the matrix, so from this it is easy to calculate the probability we simply take the account of the sequences and divide it by the number of times the unigram tag appears and from that we can compute the probability. So, may we should take a few minutes to discuss some points here, so it must be quite apparent to you that, the corpus is the most important entity in this whole calculation process.


The way we are doing the calculation is by looking at the tags as they appear on the corpora, for computing the probabilities of the bigrams of the tag t_i given that the previous tag was t_{i-1} . We simply obtain the bigram sequences divided by the unigram sequences, so a probability of N given V is nothing but the number of times V N sequence appears divided by the number of V's, so this was shown. Now, all these is completely driven by the tagged corpora moving forward, we also have to look find out the probability of the word given the tags.

(Refer Slide Time: 22:45)

Lexical Probability

	People	Jump	High
N	10^{-3}	0.4×10^{-3}	10^{-7}
V	10^{-7}	10^{-2}	10^{-7}
A	0	0	10^{-1}
R	0	0	0

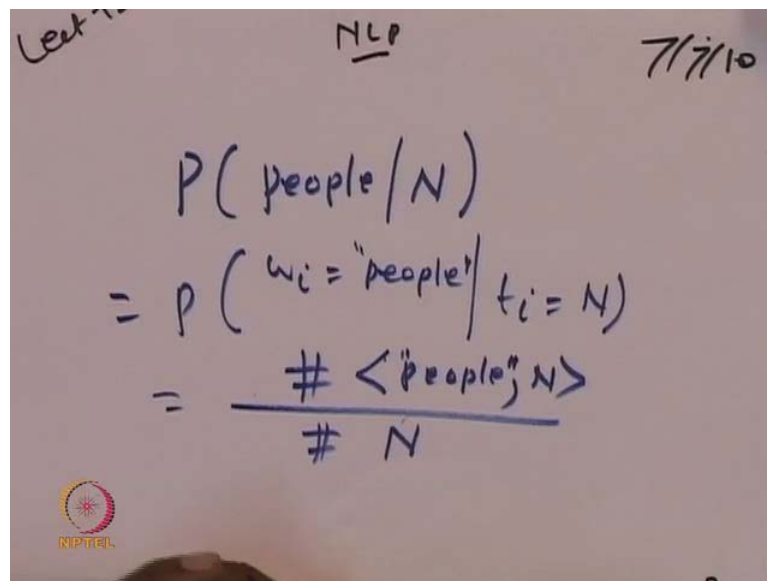
values in cell are P(col-heading/row-heading)



So, bigram probability was discussed, now we have to discuss the lexical probability, so we have to find out the probability of people given that the tag at that position is N, given

that the tag at the position is verb, given that the tag at the position is adjective or adverb. So, let us look at this column we are computing probability of w_i given t_i , similarly for jump and high if you look at this column, this cell is probability of people given that the tag at that position is N. This is shown to be equal to 10 to the power minus 5 this also we can compute from the corpora, how do we compute this from the corpora we will look at them at critical expression we will write it down.

(Refer Slide Time: 23:40)



Handwritten mathematical derivation on a whiteboard:

$$\begin{aligned} & P(\text{people} | N) \\ &= P(w_i = \text{"people"} | t_i = N) \\ &= \frac{\# \langle \text{"people"}, N \rangle}{\# N} \end{aligned}$$

Additional text on the whiteboard: "Lect 10", "NLP", "7/7/10", and the NIPTEEL logo.

So, we want to compute $P(\text{people} | N)$ more accurately this is probability y_i equal to people given that t_i equal to N . So, this is simply number of times we have people and noun occurring together divided by the number of nouns, so in the corpus we will record how many nouns appear that will give us the denominator. And out of these nouns how many times, we have at that position the word people, so out of this noun how many are people simply put that is the question.


(Refer Slide Time: 24:30)

Lexical Probability

-

	People	Jump	High
N	10^{-3}	0.4×10^{-3}	10^{-3}
V	10^{-3}	10^{-3}	10^{-3}
A	0	0	10^{-1}
R	0	0	0

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

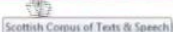



And this is computed by counting the frequency and we find that in the slide, the number is shown to be 10 to the power minus 5; that means, if there are 10,000 nouns in the corpus 1 in 10,000 nouns is the word people. This is the way it is computed and suppose it is quite clear, so at each cell we compute the probability by making observations on this count.

(Refer Slide Time: 24:52)

Some notable text corpora of English

- [American National Corpus](#)
- [Bank of English](#)
- [British National Corpus](#)
- [Corpus Juris Secundum](#)
- [Corpus of Contemporary American English](#) (COCA)
400+ million words, 1990-present. Freely searchable online.
- [Brown Corpus](#), forming part of the "Brown Family" of corpora, together with [LOB](#), Frown and F-LOB.
- [International Corpus of English](#)
- [Oxford English Corpus](#)
- [Scottish Corpus of Texts & Speech](#)

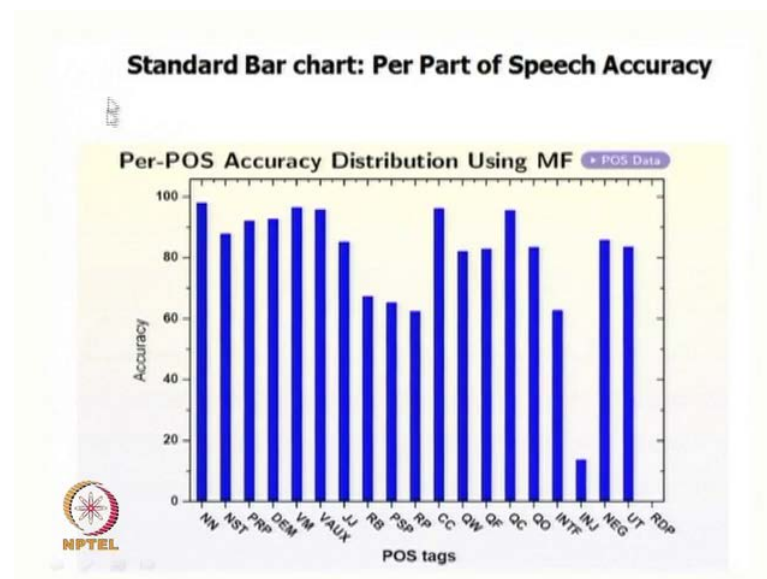


Now, we have been mentioning many times, that we have to make use of the annotated corpus, and some of the well knows corpora for English are American National Corpus,

Bank of English, British National Corpus is very well known. These records the English British English, American National Corpus records American English Corpus, Juris Secundum Corpus of contemporary American English, which is about more than 400 million words 1999 to present time freely searchable online.

Brown Corpus forming part of the Brown Family of corpora, International Corpus of English Oxford English Corpus Scottish Corpus of Texts and Speech many of these are very well known. All languages are creating their corpora and also annotating them with part of speech and soon on, in India there is a massive drive by The Ministry of Communication and Information Technology to create Indian Language Corpora in all official languages of India, these are being done at various institutes of the country. And these corpora are completely parallel, which means that sentences in different languages are align according to their meaning these are also being tagged by a set of part of speech tags. So, when these corpora are available, we will be able to compute the probabilities and then from that we will be able to create a part of speech tagger easily.

(Refer Slide Time: 26:34)



So, now, we say few words about the evaluation of part of speech taggers and the concerns, they are in whenever we build a part of speech tagger we have to make sure that the part of speech tagger operates with high accuracy. Part of speech tagging is a very important step in whole of natural language processing, so the part of speech tagger had better be highly accurate. Now, when we build the part of speech tagger, and we run it on

a piece of text which was not part of the training corpus we have what is called it a test corpus.

So, on the test corpus we run the part of speech tagger and we record, how many times did we get the part of speech tag right, and how many times was it wrong. So, this graph here shows for different part of speech tags the accuracy values, we need not be concerned with what the corpus was. This corpus actually was some tourism documents in Marathi and the part of speech tagger is a Marathi part of speech tagger, it produces part of speech tags on Marathi corpora.

These tags are some standard tags adopted for Indian languages by the concision of machine translation between one Indian language to another. So, this x axis shows the part of speeches N N is a noun, N S T is a noun of space, and time PRP is a pronoun, then dem is a demonstrative, V m is main verb, P aux is auxiliary verb, A J is adjective, R B is adverb, P S P is post position R P is a particle C C is a conjunct Q W is a quantifier, which is of W wage kind and so on.

So, these part of speech tags have been decided by a body of Indian natural Language Processing Researchers and the y axis shows the accuracy on each of this part of speech tag. How the accuracy is computed? We will discuss this either in this class or in the next for this moment, let us assume that by accuracy we understand how correct the part of speech tag is.

Of course, there is a very precise way of computing the accuracy, which we will discuss a little later, now we find that for noun the accuracy is close to 100. So, this pos tagger is doing well for noun which is a very, very frequently occurring category, in any text, N S T is the noun of space and time which is words like [FL] and so on. So, [FL] is a temporal word after something [FL] is above or on k [FL] and k [FL] is down or under, so these are the words on which accuracy is below 90 percent.

So, the system is not doing very well on the nouns of space and time PRP is pronoun which is having accuracy more than ninety percent, but less than noun. Demonstratives are things like, these that etcetera and their accuracy is comparable to pronoun, in one of the previous classes we have remark that demonstration and pronoun can be confused with each other the part of speech tagger can make mistake with respect to these two categories.

Main verb is being identified with quite lot of accuracy, slightly less than the accuracy of noun similarly the auxiliary verb which is the helping verb, for that also the accuracy is close to 90 percent. Adjective accuracy comes out to be less than 90 percent adverb accuracy is not very high in fact, it is quite low it is below 70 percent and so on. So, this part of speech tag accuracy is always recorded, this way one shows the part of speech tag accuracy per part of speech, the accuracy per part of speech what is the reason for doing?

This when we do this we know which are the tags, which are not behaving very well and therefore, we have to employ the machinery, we have to tune the machinery we have to tune the part of speech tagger, so that the difficult tags are treated well and they give rise to higher accuracy. So, the adverb accuracy we have seen is low which is just close to 70 percent, and this accuracy is slow accuracy can bring down the accuracy of the overall system therefore, we have to gear our machinery towards correcting this error we will see later how this can be done.


(Refer Slide Time: 32:24)

Standard Data: Confusion Matrix

+ POS Data

	NN	NST	PRP	DEM	VM	VAUX
NN	49988	18	92	2	167	4
NST	33	507	9	0	3	0
PRP	145	3	8071	312	8	5
DEM	3	0	231	3002	2	1
VM	225	1	4	9	17078	347
VAUX	10	0	1	1	257	6025

Table: POS Confusion Matrix with MF



So, this bar chart for per part of speech accuracy is a very valuable instrument for implementing part of speech tagger and making its accuracy high, another very useful data structure or information is this confusion matrix. So, here we show the confusion matrix structure, essentially the idea is we place we take a two dimensional table, we place all the tags on the columns and repeat them on the rows. The meaning of this

structure is as follows look at the cell here under N N and N N, N N means noun, so it says that actual noun was tagged as noun 49,988 times.

So, the noun cases in the test corpus are, so many this plus this plus this plus this plus this plus this. So, the sum of numbers on a row tells us how many nouns were there in the corpus, this is the gold standard information, and the column tells us how many times those nouns were tagged as noun and as other tags.

So, if I look at this here we see that the number of nouns is about 50,000 and out of them 49,988 times the noun has been tagged as noun, which explains the close to 100 percent accuracy on the noun tag, which was shown in the bar chart. Noun has been tagged as N S T 18 times noun has been tagged as pronoun 92 times, noun has been tagged as demonstrative two times, and noun has been tagged as main verb 167 times noun has also been tagged as auxiliary verb 4 time.

So, this is a very valuable piece of information, and we find that the biggest error or noun comes from noun main verb confusion. So, once it is 67 times we are producing the main verb tag per noun, if you are wondering why should it be possible that a non is tagged as per nouns and verbs are very different entities. How can they be tagged how can they be confused how can a noun be confused as verb. So, these for Indian languages is a slightly tricky issue, but for English it is a very simple problem in English most nouns are used as verbs, since the part of speech tagging is being done in an environment for a particular position we will look at the previous tag and we will also look at the word at that position.

So, these values give rise to probability of word given the tag or probability of the tag given the previous tag, so these probability values are used to compute the score of the whole sequence. So, in English it is quite possible to tag a noun as a verb depending on the context, we will look at the table once again and we find that the demonstrative and pronoun can be quite easily confused. So, if you look at the PRP row the pronoun row the pronoun has been tagged 145 times as noun has been tagged 3 times as N S T, it has been tagged as PRP 8071 times; that means, it got its own tag 8071 times.

It has been tagged as demonstrative 312 times, which is a large number, it has been tagged as main verb 8 times and has auxiliary verb 5 times. So, we roughly have about 850 PRP's in the corpus out of that 8071 times the PRP has been correctly tagged other

times other tags have been placed for PRP which is wrong. So, PRP has been confused as Dem 312 times whatever Dem, Dem has been confused as PRP 231 times, so now, these numbers are important for us let us discuss this a bit more closely we will do a bit of a calculation.

(Refer Slide Time: 37:12)

PRP → total occurrence
 $= 145 + 3 + 8071 + 312 + 8 + 5$
 $= 147 + 8071 + 325$
 $\approx 8071 + 475$
 ≈ 8550

So, we find from the table that the PRP tag total occurrence is equal to 145 plus 3 plus 8071 plus 312 plus 8 plus 5, so this comes out to be 147 plus 8071 plus 325. And this is roughly equal to 8071 plus 475 which is roughly equal to 8550, so this is 8550 roughly.

(Refer Slide Time: 37:59)

Standard Data: Confusion Matrix

POS Data

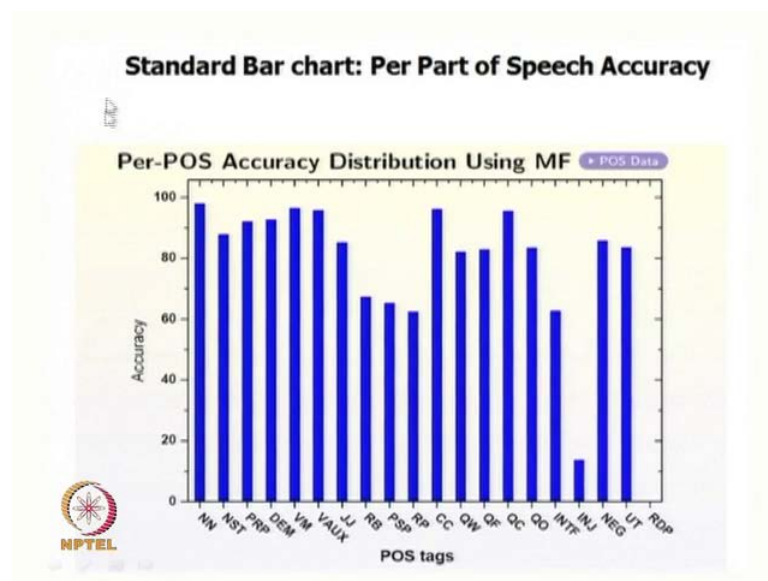
	NN	NST	PRP	DEM	VM	VAUX
NN	49988	18	92	2	167	4
NST	33	507	9	0	3	0
PRP	145	3	8071	312	8	5
DEM	3	0	231	3002	2	1
VM	225	1	4	9	17078	347
VAUX	10	0	1	1	257	6025

Table: POS Confusion Matrix with MF

Now, if we look at the table once again look at the PRP row out of 8550 times 8071 times, so you have got that tag correctly about 500 times we have gone wrong, and who is the main culprit? The main culprit is noun and demonstrative PRP has been confused with noun 145 times and with demonstrative 312 times, so if these can be reduced then we are able to increase the accuracy of the whole system.

So, now, this confusion matrix shows us, where should we invest our money when we are trying to improve the accuracy of the system we have got all these accuracy figures or confusion matrix figures. Now, the confusion matrix is tell will tell us that PRP error is coming mainly for from confusion with Dem and also confusion with 145 or noun. So, now, from this it is clear that if we can reduce the confusion between PRP and Dem, we will be able to increase the of the accuracy of the system.

(Refer Slide Time: 39:24)



So, confusion matrix that way is a very, very valuable tool let me just repeat this point once again for you, this kind of bar chart for per part of speech accuracy depiction.


(Refer Slide Time: 39:30)

Standard Data: Confusion Matrix

+ POS Data


	NN	NST	PRP	DEM	VM	VAUX
NN	49988	18	92	2	167	4
NST	33	507	9	0	3	0
PRP	145	3	8071	312	8	5
DEM	3	0	231	3002	2	1
VM	225	1	4	9	17078	347
VAUX	10	0	1	1	257	6025

Table: POS Confusion Matrix with MF



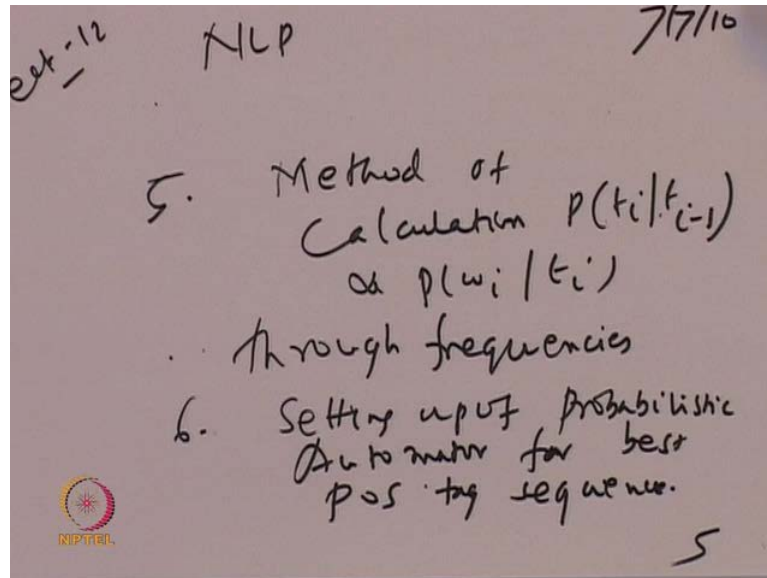
And the confusion matrix which shows, which tag has become which other tag how many times, so these two pieces of information are extremely valuable for increasing the accuracy of the part of speech tagger. So, let us make some summarizing remarks now, we will now take up the discussion on morphology because Indian language part of speech tagging requires morphological analysis in detail, the words are complex forms we have to detailed the detail out the morphological information.

(Refer Slide Time: 40:16)

- lect 12
- NL8
- 7/7/10
1. Defined pos tagging
 2. Showed pos tagging is a disambiguation task
 3. Pos tag \rightarrow Argmax computation
 4. $P(T)$ or $P(W|T)$
 \Downarrow
 $P(t_i | t_{i-1})$ or $P(w_i | t_i)$
- 

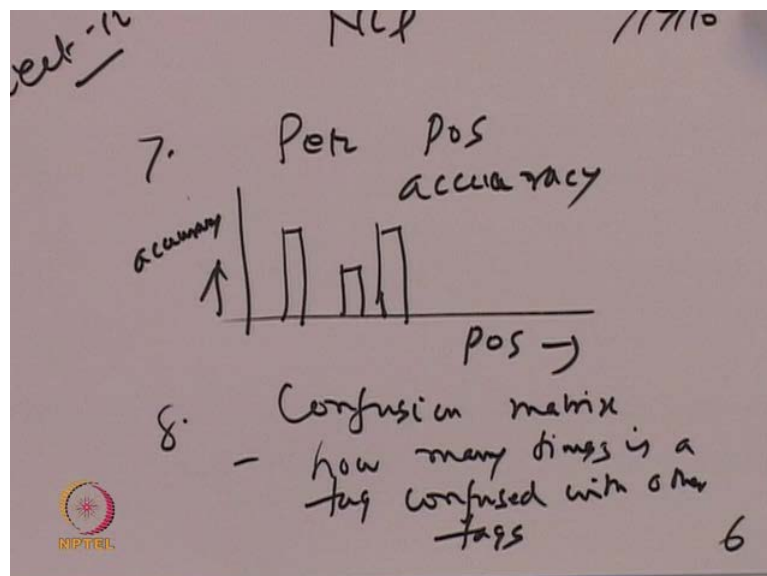
So, what we have done so far we will summarize, so we one we defined pos tagging, we showed pos tagging is a disambiguation task, 3 pos tag is nothing but argmax computation, we have to compute $P(T)$ and $P(W)$ given T which is computed by means of $p(t_i | t_{i-1})$ and $p(w_i | t_i)$.

(Refer Slide Time: 41:09)



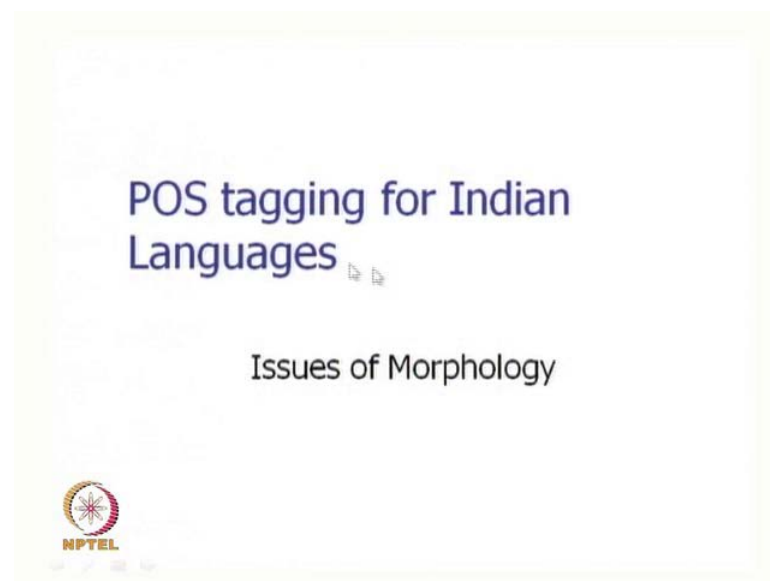
We have also done method of calculating $p(t_i | t_{i-1})$ and $p(w_i | t_i)$ through frequencies, then we did in the last class setting up of probabilistic automaton for best pos tag sequence.

(Refer Slide Time: 41:52)



Then we have also done today two very important things, which is we have discussed per pos accuracy which is in the form of a bar chart, looks like buildings, this is accuracy in the y axis, and pos in the x axis. We have also discussed the confusion matrix, how many times is a tag confused with other tags, so these are absolutely fundamental points about part of speech tagging, which have to be always kept in mind. And now the next topic will be how we actually calculate this probability of the best possible sequence, we calculate this by combining the individual probabilities, bigram probabilities, and lexical probabilities; this is done by means of hidden mark of model.

(Refer Slide Time: 43:11)



So, in the next class we will discuss hidden mark of model based computation for part of speech tagging now we will continue with the morphological analysis discussion. The goal of this discussion in is part of speech tagging for Indian languages, and since Indian languages are morphologically rich words transform into many different forms we have to discuss the issues of morphology.


(Refer Slide Time: 43:28)

MORPHOLOGY BASICS

Morphology is a two way process, consisting of:

1. *Morphology Analysis (MA)* : The analyzer MA produces from the word form (W_f) the root word (W_r) and feature structure 'F' (Assuming there is no ambiguity).
$$MA(W_f) = \langle W_r, F \rangle$$
2. *Morphology Generation (MG)*: Produces W_f from W_r and F.
$$MG(W_r, F) = W_f$$

This can be represented as (example word means "boy"):



W_r (लड़का)

\xrightarrow{MG}
MA
 \xleftarrow{MA}

W_f (लड़कें, लड़कों)

Here are some basic points about morphology which are very, very fundamental morphology is a two way process consisting of morphology analysis M A. The analyzer M A produces from the word form, W F the root word W R and feature structure f assuming there is no ambiguity. So, as if the morphology analysis produces a unique analysis for the word we assume there is no ambiguity, at this stage there are ambiguities which we will talk about later.

And so mathematically speaking MA, MA is a function work on W F the word form producing the root word W R and a future set for this analysis. Morphology generation M G is a reverse process it takes the future information and the root word and produces the word form. So, this can be represented here as an example the example word is [FL] which means boy, so W R is the root word [FL] when you generate morphology it can produce [FL] and [FL] going in the opposite direction during morphological analysis we get [FL] from [FL] and [FL] and also their features we will continue with this discussion in the next lecture.