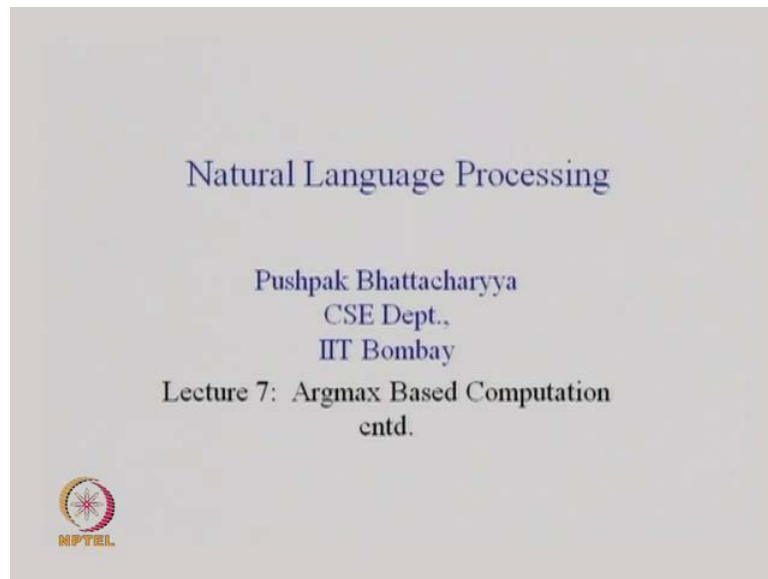


**Natural Language Processing**  
**Prof. Pushpak Bhattacharyya**  
**Department of Computer Science and Engineering**  
**Institute of Technology Indian, Bombay**

**Lecture - 7**  
**Argmax Based Computation (Contd)**

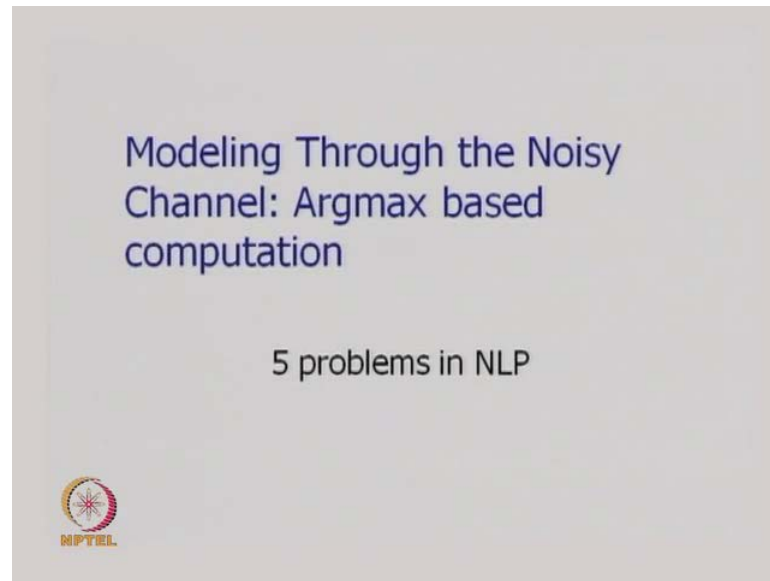
We, continue with the lecture in natural language processing.

(Refer Slide Time: 00:20)



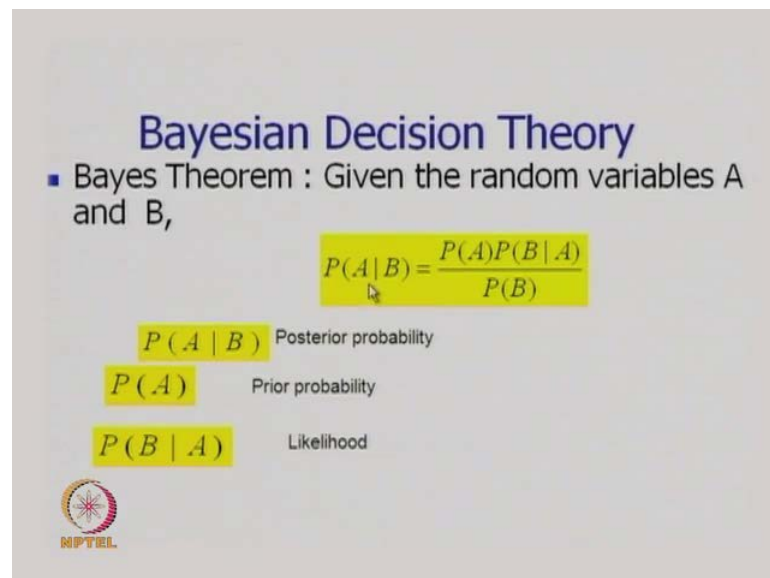
And, we continue our discussion on Argmax based computation which we are remarked is the foundation for many, many task seen statistical natural language processing. Given a problem, many times, we formulate this as computation through an argmax function ok.

(Refer Slide Time: 00:50)



So, we proceed and we are discussing five problems in natural language processing, which are modeled through the noise channel and argmax based computation.

(Refer Slide Time: 01:01)



We also, remarked that Bayesian decision theory is at the heart of these kind of computation given the random variables A and B.  $P(A|B)$ , which is the posterior probability of A given B is equal to prior probability of A  $P(A)$  into likelihood of B given A which is  $P(B|A)$  divide by probability of B. In argmax based computation when we are maximizing the function  $P(A|B)$  overall A is  $P(B)$  is generally

neglected, because it is not influencing the relative ranking of different possibilities for  $P(A|B)$  varying over  $A$ .

So, we have also remarked on, why we would like to apply the Bayes theorem and work with  $P(B|A)$  and  $P(A)$  instead of  $P(A|B)$ . A very, very compelling reason for this is we can make use of the prior probability  $P(A)$ , which open acts like a filter to eliminate bad sequences for  $A|B$  and it also is sometimes more convenient and more confidence boosting to work with  $P(B|A)$  ok. So, we will have examples to illustrate this point. So, let us get our fundamentals very clearly established. The reason for applying Bayesian decision theory is to be able to make explicit or clearly separate prior probability from clearly be in a position to make use of prior probability and the likelihood, ok.

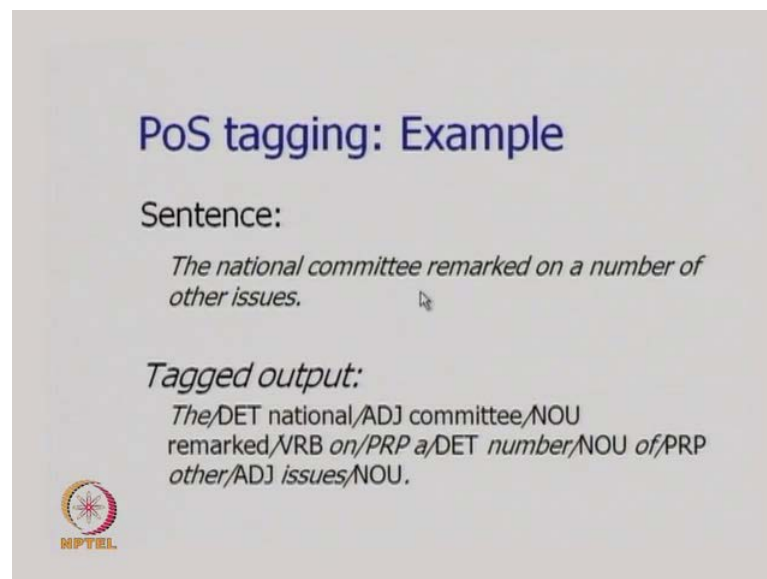
(Refer Slide Time: 03:04)



So, we already discussed in the last lecture, the part of speech tagging and this will; the whole mathematics and experimentation on part of speech tagging will be explicated in the next few classes. We will devote who at least two lectures on participation tagging to illustrate the methodologies of statistical natural language processing. Now, participating tagging was mention very briefly in the last lecture, to bring out the main use of the Bayesian theorem in participate tagging. And also to show an example of how a natural language processing task can be formulated as an argmax based computation.

The other, four problems would you like to discuss in this lecture are statistical spell checking; this can be formulated as an argmax based computation, automatic speech recognition namely: conversion of speech signals into linear text, probabilistic parsing how do you obtain, the best possible parse tree from a given sentence. And statistical machine translation is, how do we obtain the best score in translation from one language to another? All these are sequence leveling tasks, we will see how and all these are amenable to argmax base computation and use of Bayesian theorem.

(Refer Slide Time: 04:43)




**PoS tagging: Example**

**Sentence:**

*The national committee remarked on a number of other issues.*

**Tagged output:**

*The/DET national/ADJ committee/NOU  
remarked/VRB on/PRP a/DET number/NOU of/PRP  
other/ADJ issues/NOU.*




So, part of speech tagging the example shown was that a sentence has a number of words and these words are given part of speech tag levels. These levels are obtained from the properties of the word themselves and the context in which they are embedded.

(Refer Slide Time: 05:05)

POS Tagging

Best tag  $t^*$ ,  $t^* = \arg \max_t P(t | w)$

$$t^* = \prod_1^{N+1} P(t_i | t_{i-1}, t_{i-2}) P(w_i | t_i)$$

 MPTEL

So, the best possibility tag sequence  $t^*$  is obtained as argmax over  $t$  of probability of  $t$  given  $w$ . So, what is the input to the system the input  $t$  is nothing but, the word sequence  $w$  this  $t$   $w$  are not single entities but, these are sequences. Let us remember that  $t$  is the tag sequence  $w$  is the word sequence. So, given the word sequence, you would like to find the best possible tag sequence and these can be converted into probability of a word at a position  $i$ , given the tag at that position  $t_i$  into probability of  $t_i$  that is the tag at the position  $i$ . Given that the previous 2 tags are  $t_{i-1}$  and  $t_{i-2}$ .

So, these we remarked gives rise; to a try gram based computation and these kind of tagger would be called a try gram tagger, ok. And there is a very famous system, a very famous participate tagger which is based on hidden mark of model and makes use of diagram called the  $t_n t$  tagger,  $t_n t$  tagger. So, this is the heart of all these taggers and we are also see how these computation can be modeled, by the way the very well known and the famous combination of hidden mark of model and Whiter Be decoding. So the formulation is through the  $h_m m$  and Whiter Be computation.

(Refer Slide Time: 06:53)


Spell checker: apply Bayes Rule

$$W^* = \operatorname{argmax} [P(W|T)]$$
$$= \operatorname{argmax} [P(W).P(T|W)]$$

$W$ =correct word,  $T$ =misspelt word

- Why apply Bayes rule?
  - Finding  $p(w/t)$  vs.  $p(t/w)$ ?
- Assumptions :
  - $t$  is obtained from  $w$  by a single error.
  - The words consist of only alphabets

(Jurafsky and Martin, Speech and NLP, 2000)



Let us now, look at another example of us natural language processing, which is spell checker and in the spell checker problem also; we will apply Bayes rule. So, the problem is as follows  $W^*$  equal to  $\operatorname{argmax} P(W|T)$ , ok. So, many times we do not explicitly mention what the  $\operatorname{argmax}$  is over, because of the advantage of the notation shown here  $W^*$ , star is the best possible  $W$  star. Star indicates best so, best possible  $W$  is  $W^*$  and how to be do we obtain this? We obtained 8 from amongst a number of  $W$ 's and for each  $W$ , we can obtain the  $P(W|T)$  value, ok.

So,  $T$  is the misspelt word as I shown here and  $W$  is the correct word corresponding to  $T$ . So,  $W^*$  the best possible word corresponding to the misspelt word  $T$  can be found by an  $\operatorname{argmax}$  composition over  $W$  of, the probability of  $W|T$ . Now, we apply Bayes theorem here and we obtain the expression  $\operatorname{argmax} P(W|T)$  into  $P(T|W)$ . Where,  $W$  is the correct word and  $T$  is the miss pelt word as has been mentioned already. Now, the question that arises as before is, why should we apply Bayesian rule here and these falls down in part to answering the question. What should we be the finding  $P(W|T)$  or  $P(T|W)$ ?

So, here we would like to again remark that, this  $P(W|T)$  which becomes explicit thought the application of Bayesian rule indeed acts like a filter, ok

So,  $P(W)$  is a valid is a characters sequence producing a word and the probability of the character sequence goes high, when it is an actual word in the language. If the character

sequence does not correspond to an actual word of the language; the  $P(W)$  value is low. Thereby making that  $W$  score also low, that particular  $W$  score also low so therefore, what is happening is that, the words which are not the or the character sequence, alphabet sequences which are not words of the language are eliminated, because of the influence of the  $P(W)$ , ok.

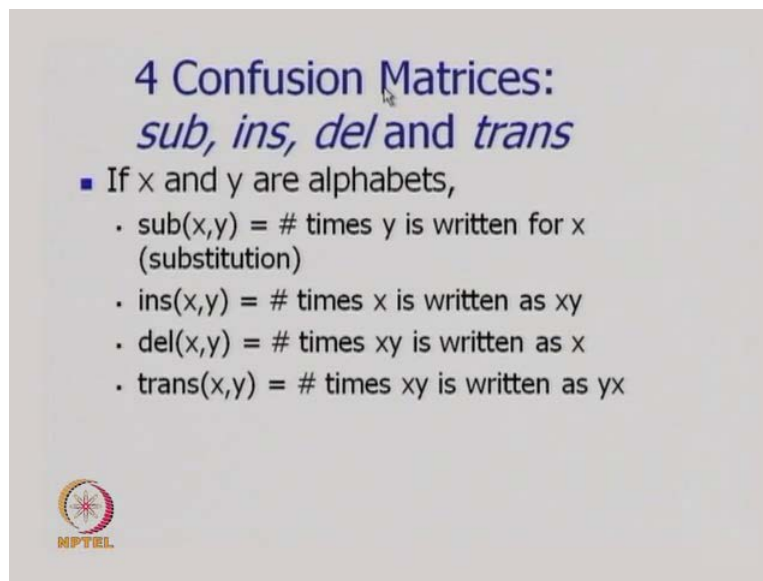
So,  $P$  the prior probability indeed acts like a good filter, ok, so the other question for this is easier to compute  $P(W)$ , given  $T$ . That means, probability of the correct word given the given the misspelt word or is it easier to compute the probability of misspelt word; given the correct word, ok. So now, we make some assumptions and we come back to that question little later. We make some assumptions, these assumptions are the following:  $t$  or the misspelt word is obtained from  $w$  the correct word by a single error, ok. By a single error and the words consist only of alphabets.

We are dealing with only the alphabetic sequence; which are words and this discussion is primarily from another famous text book of natural language processing, written by Jurafsky and Martin, Speech and natural language processing, which is a well known book, ok. So now, let us make some remark on the formulation of the problem. First thing to notice is that spell checking is looked up as an argmax computation, ok. Why argmax computation and why probability? Probability, let us understand is a very, very powerful mechanism to produce a ranking on a set of entities. The probability values associated with these entities, give a score for these entities and that inherently, produces a natural ranking for the entities and based on these rankings we choose a particular entity.

So, in this particular case of spell checking; given the misspelt word. There are a number of possibilities for the correct word, ok. If I look up on the misspelt word as a sequence of characters then, a changed character sequence is what we are looking for, ok. After all the corrected word from the misspelt word what is it? It is nothing but, a transformation of character sequence of the misspelt word into the correct word. And many options are available we have to choose one amongst them. How do you choose? We choose based on the rank. Where does a rank come from? The rank comes from the score associated with a character sequence and that score is nothing but, a probability value, ok. So, see the whole line of reasoning is a pretty intuitive sequence of steps.


(Refer Slide Time: 06:53) So now, if we look at the slide again we find that  $W$  star is obtained from  $\text{argmax } P W$  given  $T$ . Now, these assumptions that  $t$  is obtained from  $w$  by a single error and the words consist of only alphabets. These are the assumptions which essentially give rise to a method by which these probability values are calculated and the method by which the best probability character sequence is found; as a possible candidate, the correct candidate for the misspelt word ok.

(Refer Slide Time: 13:55)



**4 Confusion Matrices:**  
*sub, ins, del and trans*

- If  $x$  and  $y$  are alphabets,
  - $\text{sub}(x,y) = \#$  times  $y$  is written for  $x$  (substitution)
  - $\text{ins}(x,y) = \#$  times  $x$  is written as  $xy$
  - $\text{del}(x,y) = \#$  times  $xy$  is written as  $x$
  - $\text{trans}(x,y) = \#$  times  $xy$  is written as  $yx$



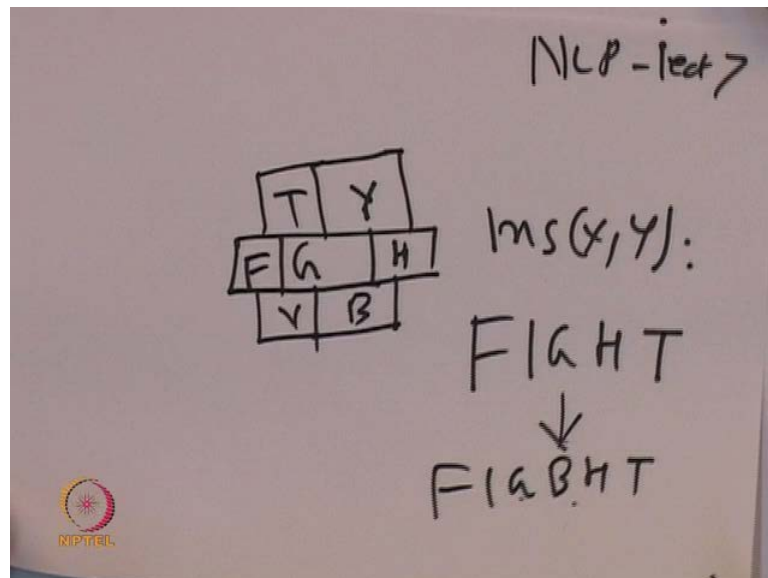
So, since there is only a single error in the miss pelt word ok, removing that particular error should give rise to the correct word and there are 4 kinds of errors giving rise to 4 valuable data structures called confusion matrices. So, if  $x$  and  $y$  are alphabets then we defined that function  $\text{sub } x y$ , equal to number of times  $y$  is written for  $x$ , this is substitution.  $\text{ins } x y$  is the number of times  $x$  is written as  $x y$ . So,  $y$  is inserted after  $x$ ; that is why it is called as  $\text{ins}$  function  $\text{ins } x y$ . The  $\text{del } x y$  is times  $x y$  is number of  $x$  is written as  $x y$  that mean, the number of times  $y$  is deleted, when it is preceded by  $y$ . And  $\text{trans } x y$  is the number of times  $x y$  is written as  $y x$ .

Now, these functions where do they come from? These functions come from the actual experience of people in spell checking situations. What happens is that the people make these kind of mistake of substituting a character  $x$  and alpha  $x$  by  $y$ , for this is common spelling mistake will take some example.



People also, make the mistake of writing y after x may be that some accidental press of the finger where, on a button which is adjacent to button for x. So for example, if I look at the keyboard I find that G and H are adjacent to each other; F is also adjacent to G before coming before G. G is again flanked by alphabets T Y V B and so on, these are neighbors of G, which are seen from the keyboard.

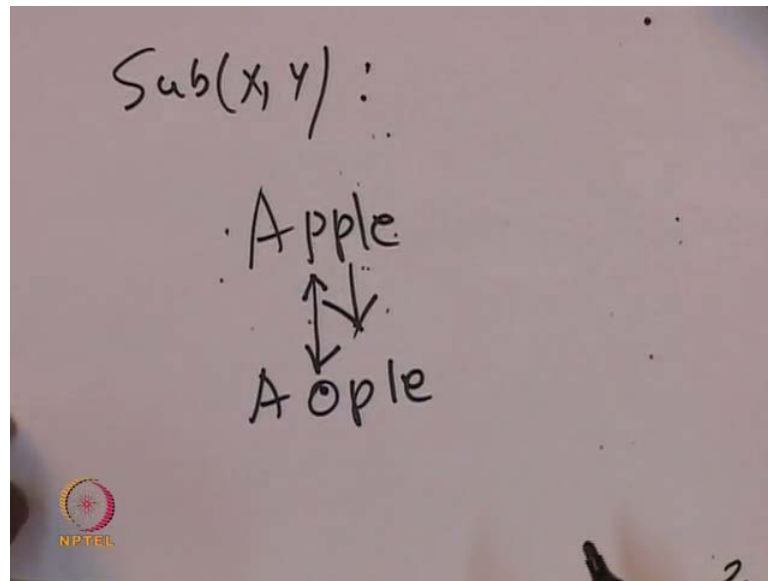
(Refer Slide Time: 16:23)



I will just draw it, for the sake of convenience of discussion ok, so I find from the keyboard that we have G here then G and then we have F here. And G has on top of H the alphabet T, it also has a part of the Y button. And then, I find that it has that these 2 buttons are V and B. So therefore, it is quite imaginable that when a person is typing G is in his finger can also the touch some finger can also touch T Y B H or V or B or F and therefore, an insertion can happen after G.

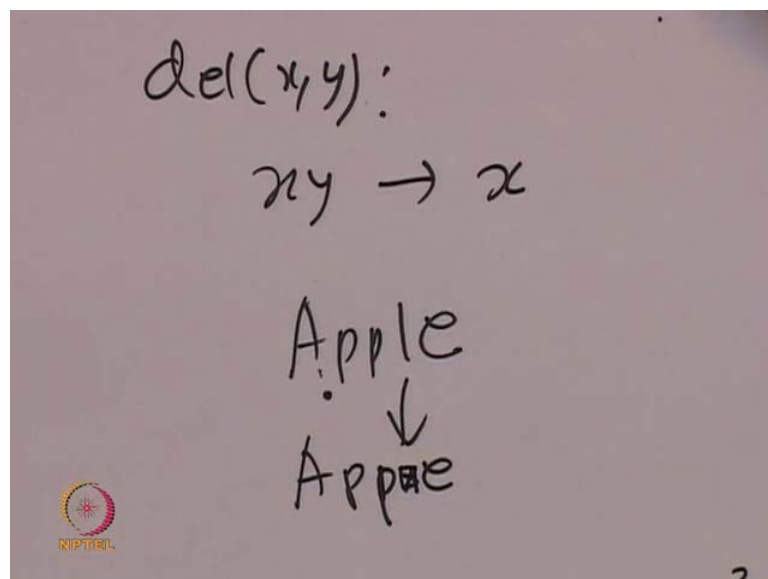
So, one might also question that why this is not happening before G, we can see that after or before does not matter because both the situations can be captured by either after or before ok. So, in insertion error Ins x y and example of these error would be somebody is trying to, let say type F I G H T. And instead of that I have said as F I G B H T, because B happens to be adjacent to very closer to B and a finger has touched B. So, B as coming this is the insertion error where x is replaced by x y so, G is actually replaced by G B. So, this is an insertion error.

(Refer Slide Time: 18:14)



Let us, take other examples where, other kinds error mentioned are present a very important error is sub X Y where X is substituted by Y, so x substituted by y. So, if I take a word apple for example, apple can be misspelt in the following way for P I might just type O O. So A O P L E, which is a wrong word, ok. So, this P has been substituted by O ok. And this is an example of sub x y substitution of x by y P by O.

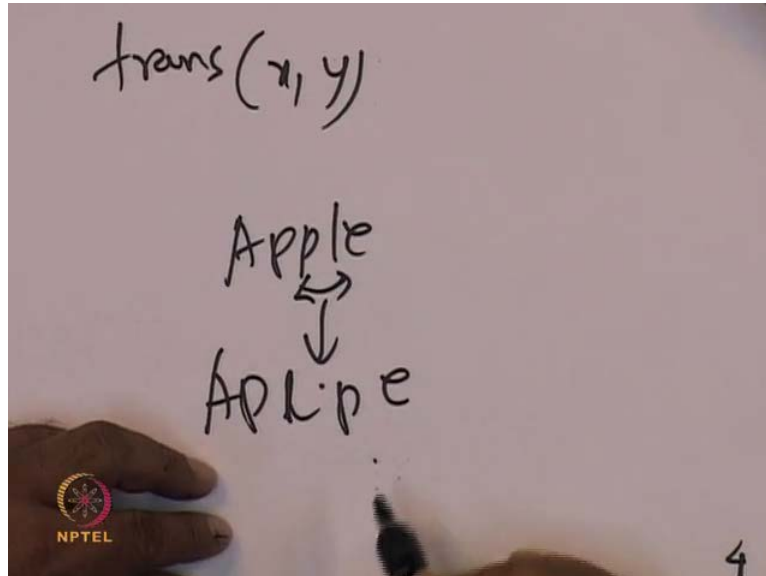
(Refer Slide Time: 18:59)



Then we have the error of deletion, which is a del x y and in this case the error is that x y is written as x, so a sequence x y has become x. So y got deleted, so instead of writing

apple A p p l e you might write this as A p p e where this l has been deleted. So, this is an example of del x y.

(Refer Slide Time: 19:38)




Another very, very common error is trans x y, trans x y the number times x y is written as y x or x and y are has transposed. So, for apple of course, transposing these 2 p's will not make any difference however. If you have A p l p e, here what is happening is that this p and l are getting transposed between them. So, p and l are transposed and therefore A p p l e has become A p l p e, which is an error. So, these are the different types of error for spell checking and these are the 4 errors which are assumed to be presents in obtaining a misspelt word.

This is, an assumption ok. We do not have 2 errors committed in the characteristic in when you typing a word you are making utmost one error; this is an assumption. (Refer Slide Time: 13:55) So, from these we come to these function, which are useful in our probabilistic computation. So, sub x y number of times y is written for x substitution ins x y; number of times x written as x y insertion, del x y; number of times x y is written as x deletion. trans x y; number of times x y is written as y .

(Refer Slide Time: 21:01)

### Probabilities from confusion matrix

- $P(t|w) = P(t|w)_S + P(t|w)_I + P(t|w)_D + P(t|w)_X$   
where
  - $P(t|w)_S = \text{sub}(x,y) / \text{count of } x$
  - $P(t|w)_I = \text{ins}(x,y) / \text{count of } x$
  - $P(t|w)_D = \text{del}(x,y) / \text{count of } x$
  - $P(t|w)_X = \text{trans}(x,y) / \text{count of } x$
- These are considered to be mutually exclusive events



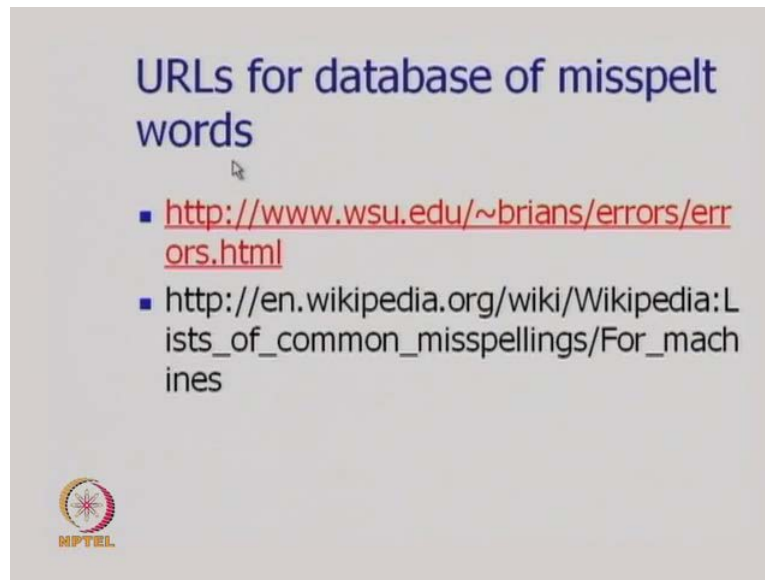
From these, confusion matrixes we could obtain the probability of  $P(t|w)$  given  $w$  in the following way.  $P(t|w)$  given  $w$  is equal to  $P(t|w)$  given  $w$  s that is substitution kind plus  $P(t|w)$  given  $w$  i that is insertion kind plus  $P(t|w)$  given  $w$  d deletion kind plus  $P(t|w)$  given  $w$  x that is the transposition kind. So, the probability of obtaining a misspelt word  $t$  from the correct word  $w$  is the probability of these mutually exclusive 4 events of: substitution, insertion, deletion and transposition, ok. Whether these events are actually mutually exclusive or not that question of course, arises in the mind.

But since, we are made that assumption that each misspelling is a result of one and only one of the 4 errors of substitution, insertion, deletion and transposition. We can write the probability of  $t$  given  $w$  is nothing but, the probability of  $t$  given  $w$  through substitution error plus probability of  $t$  given  $w$  through insertion error plus probability of  $t$  given  $w$  through deletion error plus probability of  $t$  given  $w$  through transposition error. So for  $P(t|w)$  a given  $w$  of s kind that, is a substitution kind is nothing but, substitution of  $x$  by  $y$  the number of times  $x$  is substituted by  $y$  divided by count of  $x$ .

Similarly, insertion probability: insertion probability is given as ins. The number of times  $x$  is written as  $x y$  divided by count of  $x$ . Probability of  $t$  given  $w$  through deletion nothing but, the deletion probability that is the deletion count of  $x$  comma  $y$  that is a number of times  $x y$  is written as the number of times  $x y$  written as  $y$  divided by count of  $x$ . And  $P(t|w)$  given  $w$  of transposition kind is the number of times  $x y$  as transpose,


number of times you write  $y$  in place of  $x$  divided by number of counts of  $x$ . So these are considered to be mutually exclusive events and through these substitution count and counts of  $x$  and so on, we obtain the probability values given here. And from this it is possible to compute the probability of  $t$  given  $w$ , ok.

(Refer Slide Time: 24:04)



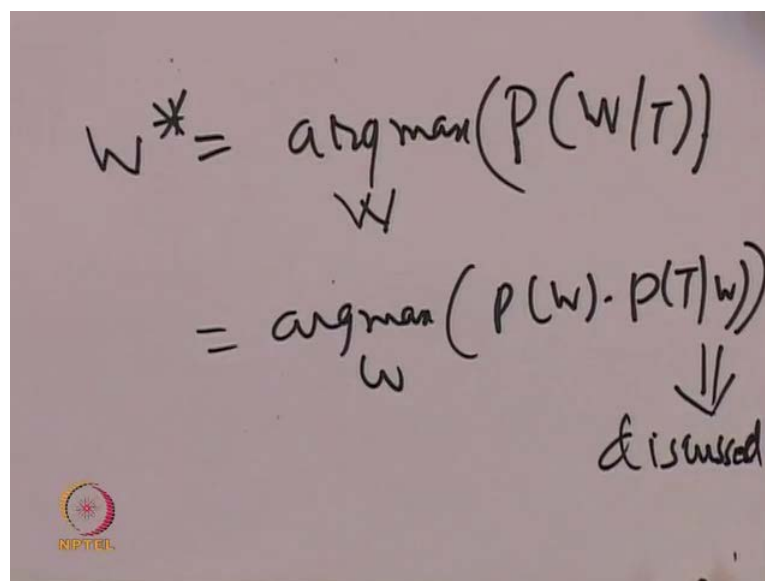
URLs for database of misspelt words

- <http://www.wsu.edu/~brians/errors/errors.html>
- [http://en.wikipedia.org/wiki/Wikipedia:Lists\\_of\\_common\\_misspellings/For\\_machines](http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines)




So now, (Refer Slide Time: 21:01) we come to an interesting question about the other part of the probability and we now, look at the expression through our writing.

(Refer Slide Time: 24:17)

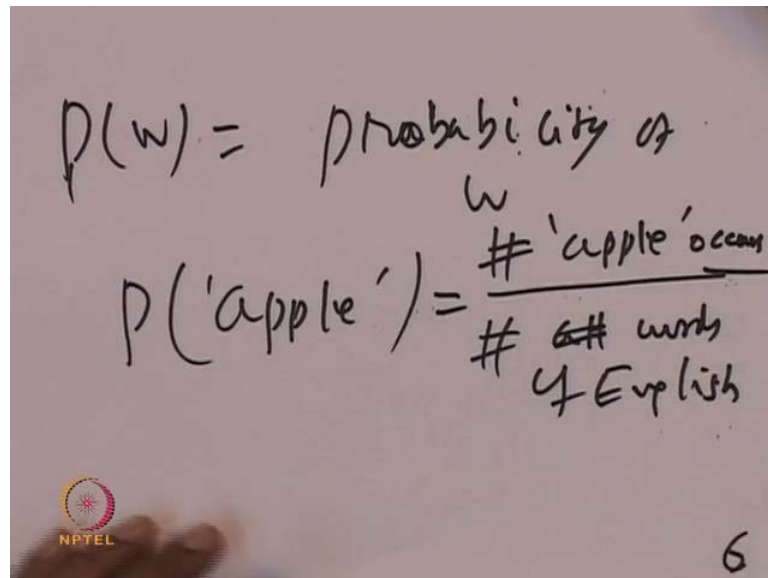

$$w^* = \underset{w}{\operatorname{argmax}} (P(w|T))$$
$$= \underset{w}{\operatorname{argmax}} (P(w) \cdot P(T|w))$$

↓  
discussed



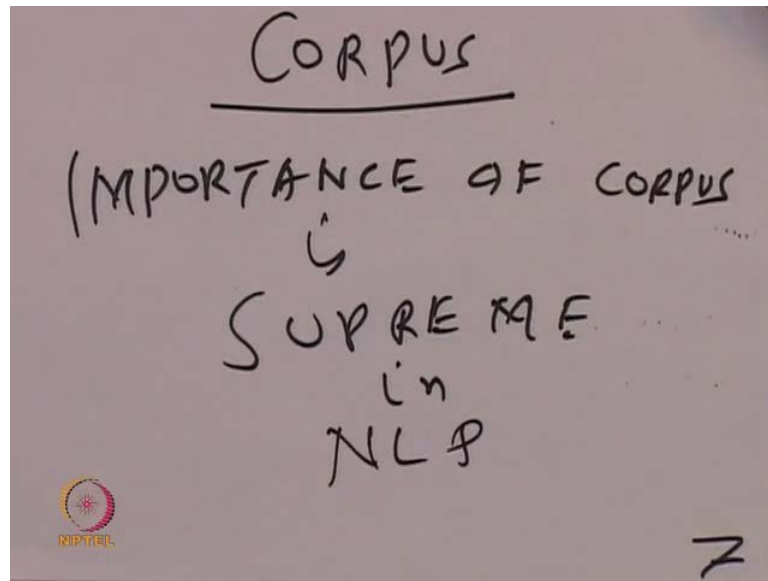
So,  $W$  star was written as  $\text{argmax}_W$  over  $w$ ,  $P W$  given  $T$  ok. So, now this was converted into this is converted into  $\text{argmax}_P W$  into  $P T$  given  $W$  over  $w$  ok. We have already discussed how to compute  $P T$  given  $W$ ; this is already discussed.

(Refer Slide Time: 25:00)


$$P(w) = \text{Probability of } w$$
$$P(\text{'apple'}) = \frac{\# \text{'apple' occurs}}{\# \text{ words of English}}$$

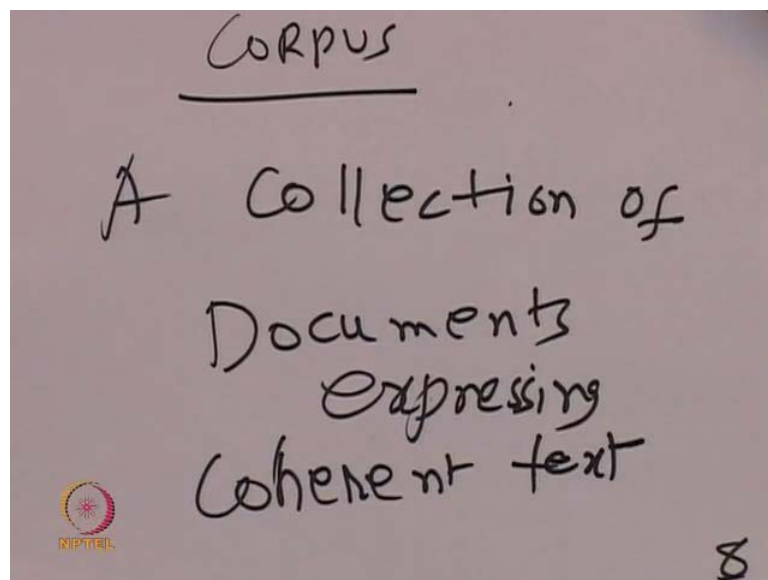
Now, we go on to discuss  $P W$  so,  $P W$  is nothing but, the probability of  $W$ . So, the issue here is, what is the probability of a particular word? So let us see the probability of apple for example, as a word is equal to the number of times apple occurs divided by a number of all words of English. This looks like a reasonable probability measured so, probability of apple is equal to the number of times apple occurs divided by total number of words of English, ok. Now, the moved question here is what is the meaning of number of time apple occurs? Where does it occur?

(Refer Slide Time: 26:06)



So, these brings us to a very important point about what is called the corpus. Importance of corpus is supreme in N L P so, natural language processing the importance of corpus is indeed supreme, it takes the center stage in all kinds of computation.

(Refer Slide Time: 26:30)

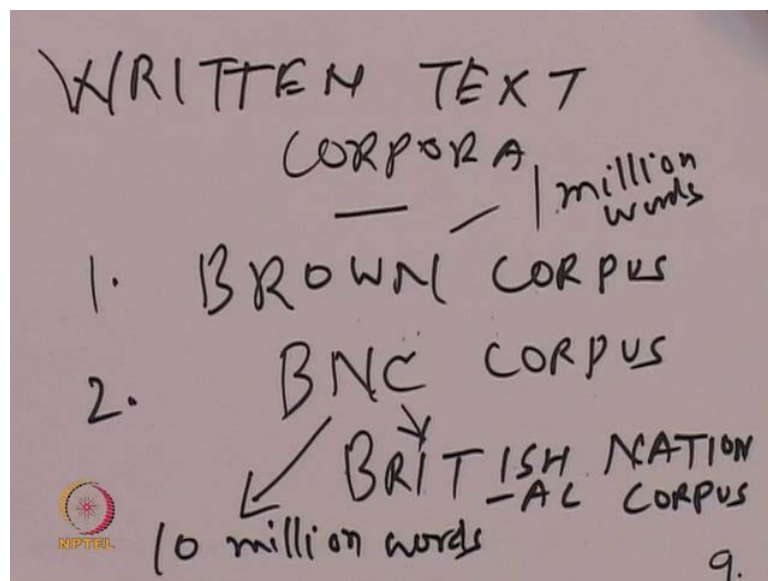


And, what is the meaning of corpus? Corpus is nothing but, a collection of documents expressing coherent text, ok. So, we collect documents in textual form on many topics from many sources and we put them together we make a collection of out of it.

So, this particular collection is called a corpus, corpus is actually very, very important in whole of statistical natural language processing. What kind of corpus? The more variety in the corpus we take corpus; we collect the corpus let us say with the sports domain, from the economy, to main in from history domain, from science domain, from technical domain. We collect them we put them together and this becomes a very, very important repository of language behavior of people, ok.

The corpus, can be spoken corpus where we collect the recordings of speeches, conversations, dialogues of people. And the corpus also could be in the form of written corpus which are material available in textual form on the internet, in libraries in the printed form or in some other collection. So, basically collection of all kinds of written text. So, written text produces: written text corpora, recording of speech dialogue conversation produces: spoken corpora.

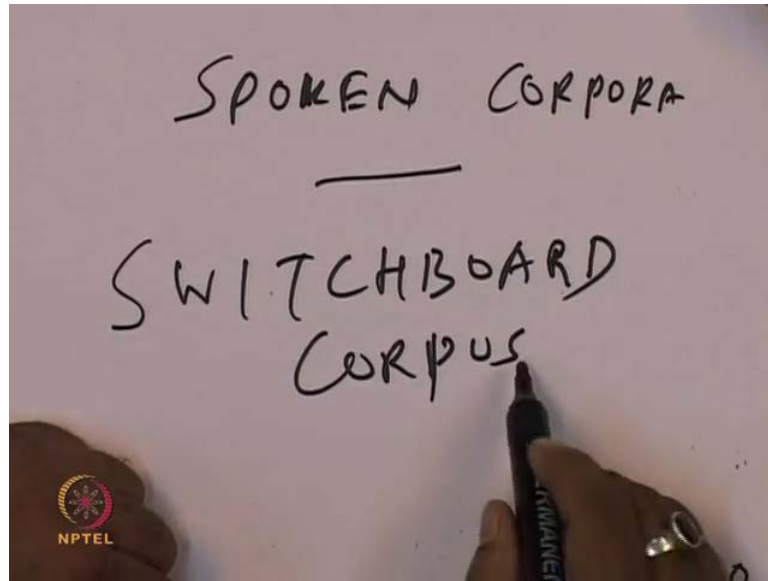
(Refer Slide Time: 28:18)



So, let me mention two very, very important corpora of English which are been used heavily in natural language processing written text corpora, we have what is called the brown corpus you can google on this and you will get all the information, Brown corpus. The other is B N C corpus this is the British National Corpus, ok. So, this are very famous corpora, these are about 1 million words and this is about 10 million words, ok.



(Refer Slide Time: 29:25)



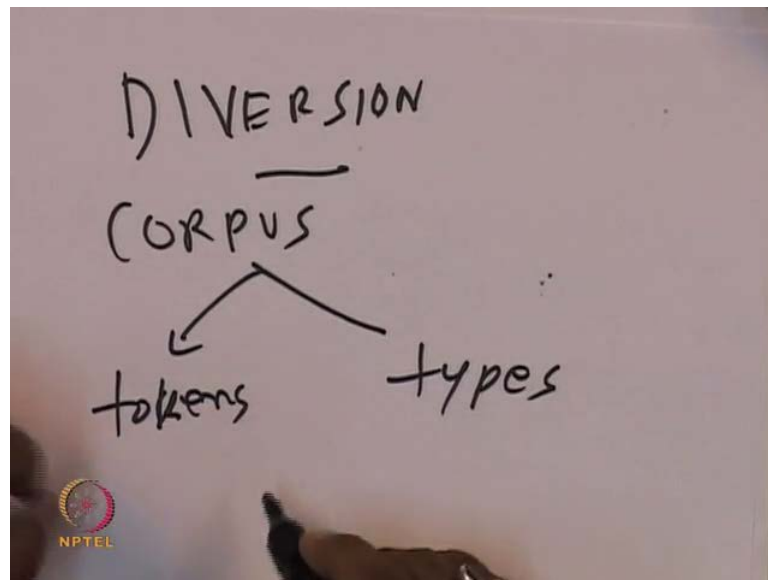
So, we mention these two very famous corpora and as far as spoken corpora is concerned there is a famous one spoken corpora, we have what is called switch board corpus. Again you can google on this switch board corpus. This is essentially the record of a telephonic conversation of people and these are many hours of recordings; many hours of recordings, they provide valuable information on the spoken behavior of people, ok. So I hope I have explained to you the meaning of corpus, which is essentially coherent pieces of text. Just a jumble collection of words is not corpus.

Corpus stands, for meaning full text people's ideas, thoughts, and people's mind are reflected in documents, textual documents. And collection of textual documents forms what is called the corpus. So, when a machine is shown called a corpus ok, we do something with the corpus through a machine. We essentially expose the machine to actual language behavior of the people; which is very, very important for natural language processing.

So, we come back to now, the question of spell checker and looking at the transparency. We have said that the probabilistic formulation of spell checking is this  $w$  star the correct word is nothing but,  $\text{argmax}_w P(w | t)$  given  $w$ . Now,  $P(t | w)$  we have extensively discussed  $P(w)$  we have say that number of times  $w$  occurs divided by total number of words. Now, total number of words where total number of words in the corpus and the number of times  $w$  occurs where, again in the corpus.

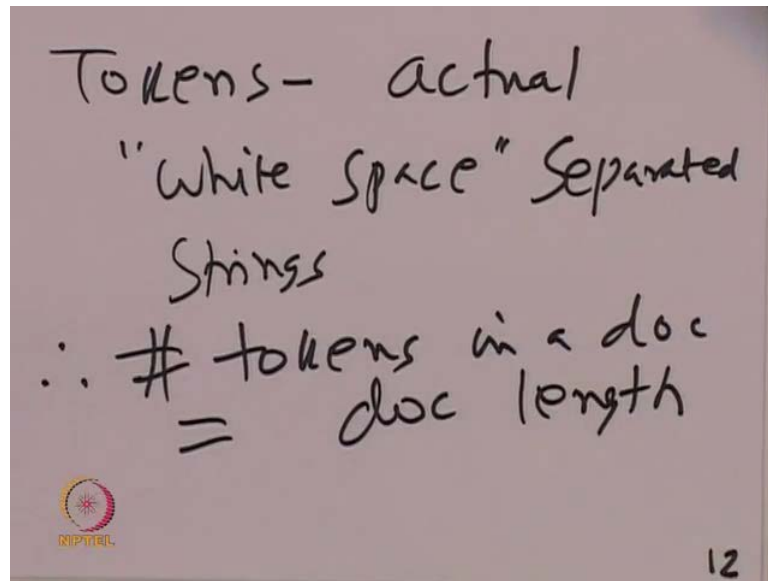
So, do you get the methodology? The methodology is that to compute  $P_w$  we look at a corpus substantially, large corpus and see how many times a particular word appears. So, the probability of apple can be computed as the number of times apple appears in the corpus divided by total number of words in the corpus. So, the question that might be arise our mind is when we say total numbers of words in the corpus. Are we talking about the unique appearances or unique number of words? No, we are talking about the document length the total number of words in the document and the total number of times word apple appears in the document. So, this celebrates us to another interesting discussion which is again a slight diversion.

(Refer Slide Time: 32:29)



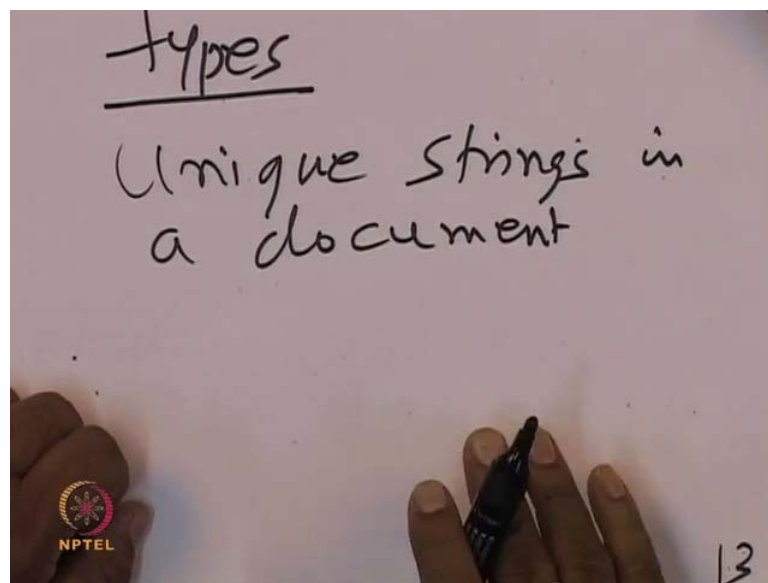
But, let me all these say mention this an interesting diversion, when we take a corpus. There are two concepts - tokens and types. Tokens and types. So, tokens and types.

(Refer Slide Time: 32:51)



Tokens is so, tokens are actual white space separated strings. Tokens are actually white space separated strings in a document, ok. And therefore, the number of tokens in a document gives us the document length is it clear, so tokens is the actual white space separated strings in a document. And therefore, the number of tokens in a document gives as the document length.

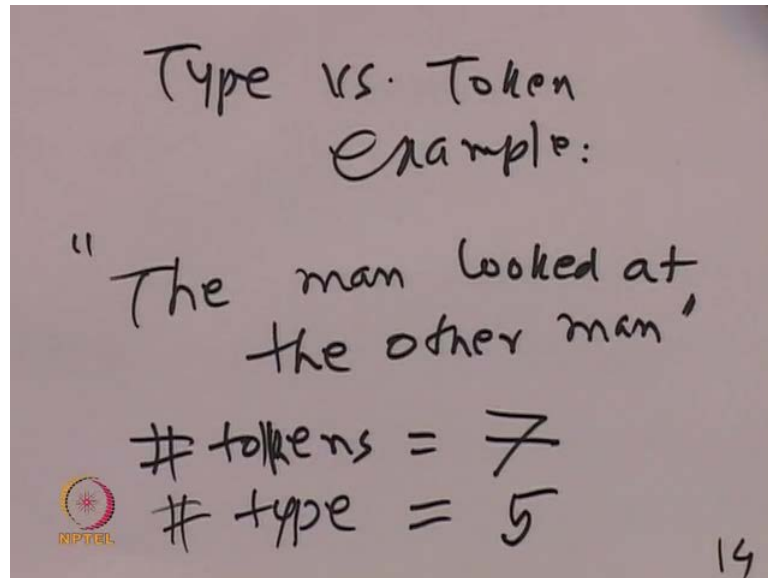
(Refer Slide Time: 33:32)



Having explained, token let me also tell you what types are. For a types, types are unique strings in a document, ok. So, this means that it captures the unique words or strings in a

document. So, let me take an example to illustrate this point, it is an important point in our computation.

(Refer Slide Time: 34:00)



So, type versus token example; suppose I take the sentence here: The man looked at the other man. This is the sentence the man looked at the other man. The number of tokens, the number of tokens here is equal to white space separated strings. So essentially the number of words here, so: 1, 2, 3, 4, 5, 6, 7 number of tokens is 7, which is equal to the document length because the document has 7 words 1, 2, 3, 4, 5, 6, 7. What is the number of types? That is the number of unique words here? The number of unique words will also be smaller than the number of words so, the number of types is always less than the number of tokens.

So, if you look at the sentence here, the man looked at the other man. The unique words are: the man looked at other, ok and next the next man they have already appeared before. So, number types is 7 minus 2 which is 5, ok. So, this document the man looked at the other man as 7 tokens and 5 types that is 5 unique words, ok. So, this was a diversion and we go back to the go back to the discussion on spell checker taken as some time but, it illustrate many, many fundamental and important concepts. So, in the slide we again see; the  $w$  star is nothing but,  $\text{argmax}_w P(w \text{ into } P_t \text{ given } w)$ , we were discussing how to compute  $P(w)$   $P(w)$  is nothing but, the number of times the word  $w$  appear in the corpus; the corpus we have taken to compute the probability divided by number of

tokens in the corpus, not types. The number of tokens in the corpus. That means the number of times apple appear divided by the document length. That is the number of; total number of words in the document.

So now, this makes a very clear how to compute  $P(w)$ . Now, what is the use of  $P(w)$ ? The use of  $P(w)$  is that through these computation through these argmax computation. The statistical system will suggest many possibilities of the corrected words after all it is a probabilistic situation and the corrected word can be any sequence of alphabets. Now, we will choose that particular sequence of alphabets as the possible correction, which has the highest probability. Now, since it is the probabilistic situation and we are computing with through  $P(t|w)$ , probability misspelt word from the correct word  $P(w)$  is acting like a filter ok.

So, the string apple a p p l e as a high probability in a corpus. The string a p l p e as lower probability, because this stream does not appear in the language. We are assuming that the corpus is a set of documents, where all the words are correctly spelt, ok. So now, we can see that  $P(w)$  is acting like a filter, if the word is an improbable string in the language  $P(w)$  value will be less and with that particular  $w$  will not appear in our suggested list of options, ok.


The probability value will automatically come down because of the  $P(w)$  quantity. Now, this  $P(w)$  as shown here is called a word model, word model. This models, the probability of an alphabet string being an actual of word of the language. The valid word of the language there by, it has this filter influence. So, how this two probabilities are actually computed? How they are broken down? How they are again put together to form a string all this our discussion; which will become clear when we do part of speech tagging. But, let us appreciate that we have under stood first through fast a computation how to formulated a problem as an argmax situation.

Second, we have understood why Bayes theorem may be applied. Bayes theorem makes a prior probability explicit and this prior probability acts like a filter, ok. And the likelihood probability is always computed by means of independent assumption and a product of probabilities of smaller units. So these are the principles.

(Refer Slide Time: 39:35)

## URLs for database of misspelt words

- <http://www.wsu.edu/~brians/errors/errors.html>
- [http://en.wikipedia.org/wiki/Wikipedia:Lists\\_of\\_common\\_misspellings/For\\_machines](http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines)



We proceed now, and we take an example which you can make use of as a possible assignment implementation. Here is a URL which is the database of misspelt words [www.wsu.edu/~brians/errors/errors.html](http://www.wsu.edu/~brians/errors/errors.html) or even [http://en.wikipedia.org/wiki/Wikipedia:Lists\\_of\\_common\\_misspellings/For\\_machines](http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines), ok. So, clearly this the second url has been put in place by somebody who has been that actually working on probabilistic spell checker. So, these are a resources by which a probabilistic system can be trained to detect spelling errors. In future and those probability to values can be obtained by processing the data is given here. This kind of probability value computation is also called the training.

(Refer Slide Time: 40:51)



**A sample**

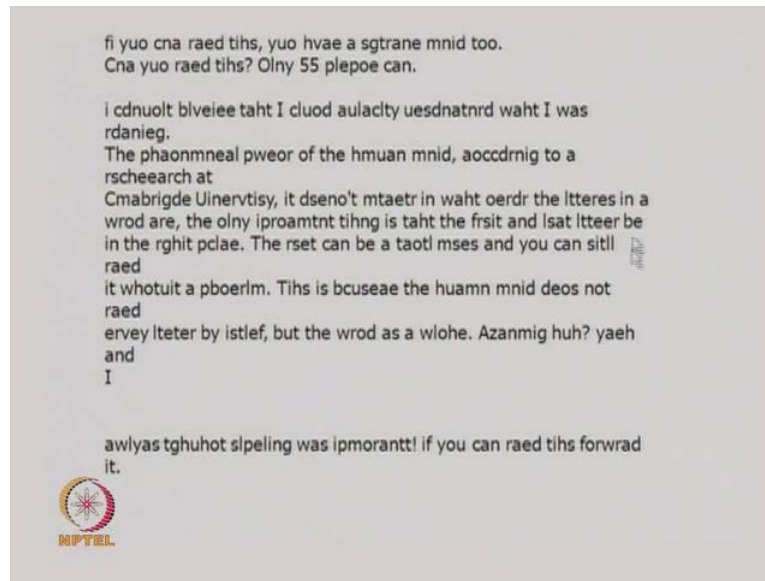
- abandonned->abandoned
- aberation->aberration
- abilities->abilities
- abilty->ability
- abondon->abandon
- abandoned->abandoned
- abandoning->abandoning
- abandons->abandons
- aborigene->aborigine



Here is a sample abandoned is written as abandonned here which is the spelling mistake n has been written as double n, may be the key has been pressed the n key the finger s press the n key twice. So this belongs to the error of insertion type so, n has been replaced has been written as double n, aberration a b e r a t i o n actually is correctly spelt as a b e r r a t i o n only single r has been placed.

That means, the error is of its deletion kind where double r has been written as r, abilities. Abilities is written as a b i l t i e s this is a deletion error, where i has been deleted from abilities. This ability a b i l t y again is of a deletion kind, these i has been deleted, abandon has been written as abondon, a has been written as o this is an error of transpose substitution kind, a has been substituted by o. Abandoned this is abandoned and again it is a substitution error, abandoning again a substitution error a is replaced has o, abandons again substitution, aborigine is actually written as a b o r i g i n e and we are seen a b o r i g e n e. So, this is an error of substitution type, this is substitution error. So, here we see mainly substitution insertion and deletion error, so we do not see an example of transposition error. But, if look at the data base there would be such examples also.

(Refer Slide Time: 43:03)



Here, is an interesting point I want to bring to a notice, so this particular text is doing the round in the internet I received these through email lot of people have obtained these through an email. Now this text is full of spelling errors there is not a single word which is not spelled incorrectly not a single sufficiently long word. I can see that i is correctly spelt we can match about the and it is correctly spelt i t it correctly spelt i n the correctly spelt but, there also misspelt. Now we see that we can read this text quite conveniently so what is the first sentence, if you can read these you have a strange mind two.

This is the way the sentence is let and this is full of spelling error but, looks like a we do not have any difficulty in reading the sentence, our mind is correcting the sentence as it goes on and reading it. So, feel I and f have been switched this is a transposition error, you y o u has been written as y u o again a transposition error, can c n a transposition error, read these transposition error this is again transposition error, you have a strange mind tool strange s t r a n g e.

So, here we see there are 2 errors here a s g t r a n e n g has been deleted from a error and g is placed there are there is an insertion error. So, this is the deletion and insertion both but, a mostly the errors are transposition kind. But, the remarkable think is that we can read the sentences with this if you can read this you have strange mind too. Can you read this? Only 55 people, see how people has been spelt. The people can I could not this is very, very wrongly spelt but, we can still reading I could not believe that I could actually



understand what I was reading, the phenomenal power of the human mind according to a research at Cambridge university. It does not matter in what order the letters in a word are the only important thing is that the first and the last letters be in the right place. So, this is very interesting point right, the rest can be a total mess and you can still read it without a problem.

This is because the human mind it does not read every letter by itself but, the word as a whole amazing huh? Yeah and I always thought spelling was important, if you can read this forward it. So, this is a very nice interesting example after says that the letters within a word we do not have methods as long as the first and last letters are properly kept. So, it is a very, very interesting issue there are cognitive science points involved here and with that we finished today's lecture we will take up other natural language processing problems. But, I leave it for you to reflect on how much powerful human mind is when it is comes to decide for the correct meaning in language situations. This is extremely fault or tolerant and extremely resilient to the noise.