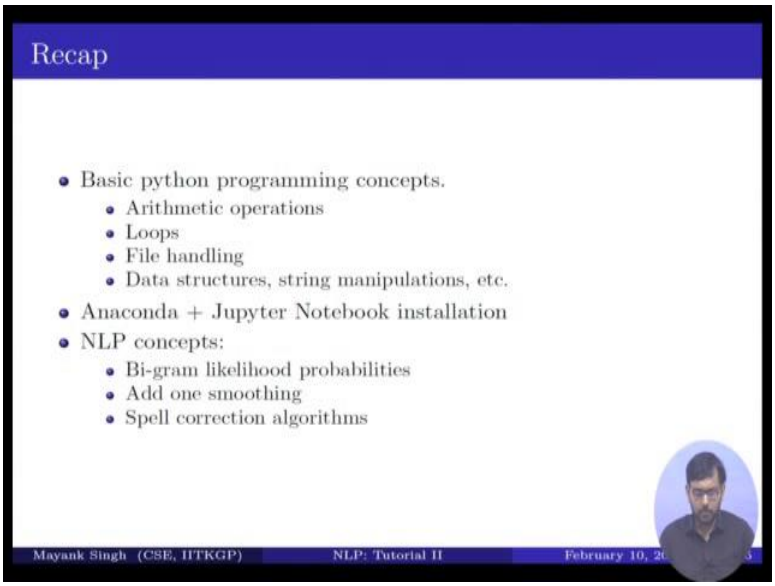
Natural Language Processing
Prof. Mayank Singh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 57
Tutorial

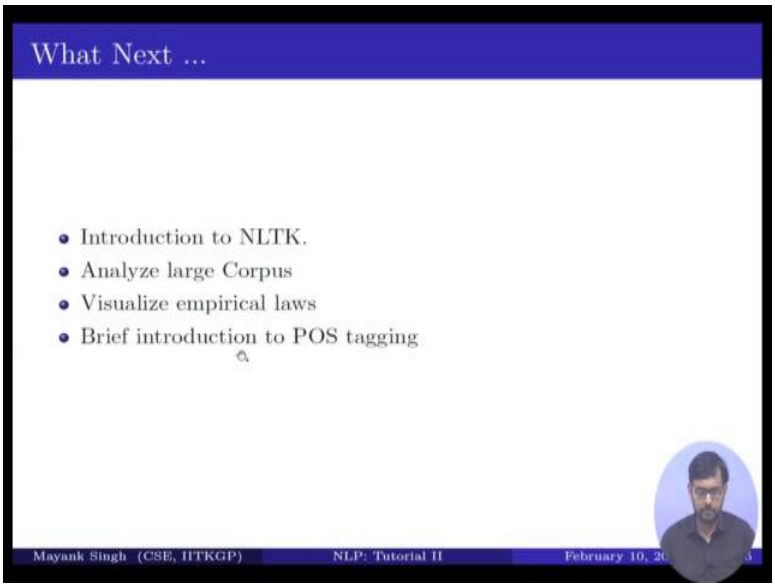
Hello everyone, my name is Mayank Singh. So, today I will be going through the second tutorial for this course.

(Refer Slide Time: 00:25)



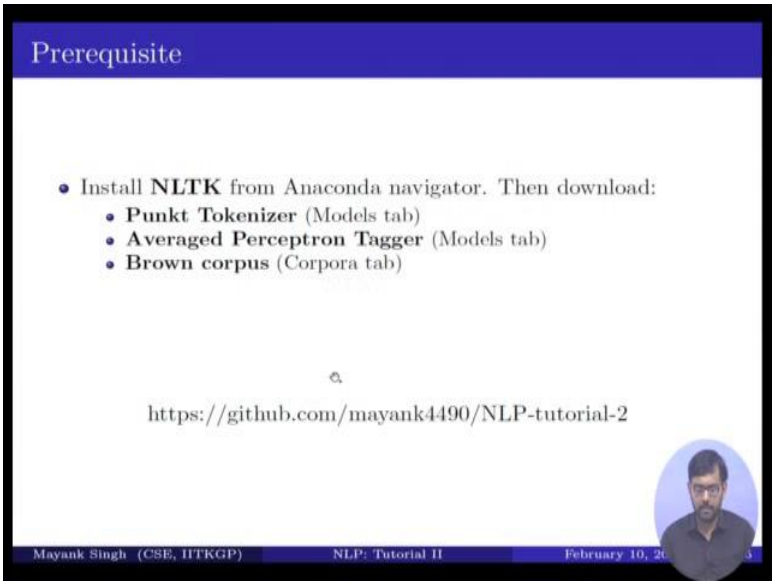
In the previous tutorial we have gone through the basic python programming concepts for example arithmetic operations, loops, file handling, data structures, string manipulations, etcetera. We have also shown steps to in install anaconda and Jupyter notebook installations along with that we have shown some NLP concepts for example, how to compute bigram likelihood probabilities, how to do add one smoothing and how to correct spellings using 2 famous spell correction algorithms.

(Refer Slide Time: 00:53)



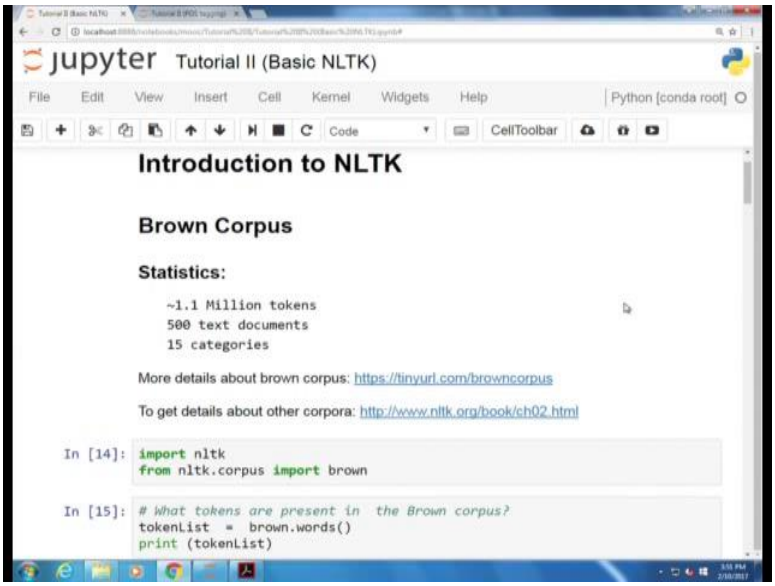
What are we going to do in this tutorial? In this tutorial, I will start with the brief introduction to NLTK. NLTK is a python toolkit for natural language and processing tasks. We will use NLTK to analyze large corpus and how to how we can visualize some empirical loss. Towards the end I will also give you a brief introduction to pos tagging.

(Refer Slide Time: 01:19)



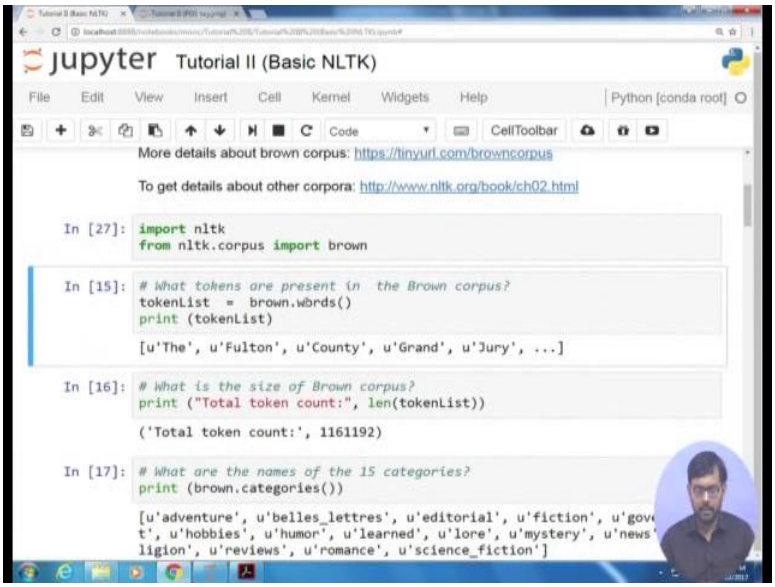
But before you start programming for this tutorial, you have to install NLTK from anaconda navigator. Once NLTK is installed, you have to install and download Punkt Tokenizer and averaged Perceptron tagger from the model step and brown corpus from the corpora tab. For more information about the NLTK and the installation part, you can click on this link.

(Refer Slide Time: 01:57)



I assume that you can successfully install all these dependencies. So, let us move to the coding part, yes. So, this tutorial is divided into 2 python note books, in the first note book I will give you a brief introduction to NLTK and how we can process large corpus using NLTK, as an intuitive use case we have used brown corpus. So, in brown corpus is a very old English corpus that consists of 1.1 million tokens from 500 text documents all these text documents are divided into 15 categories. For more information about brown corpus you can visit this link.

(Refer Slide Time: 02:35)



There are many other corpora in different languages in NLTK. So, you can get details about those corpora also on the second link. So, let as first set up our environment. So, first of all, we need to import NLTK. So, importing NLTK means that we are going to load all the dependencies of NLTK using import NLTK. In the second line we are importing brown corpus using NLTK corpus. So, let us run this, yes. So, now, our initial set up is done our libraries and the brown corpus is loaded into the python note book.

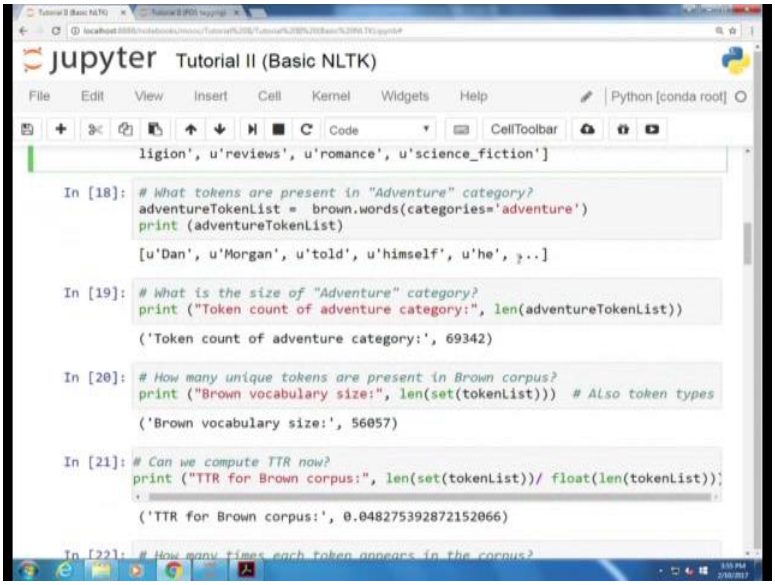
Now, first question is what tokens are present in the brown corpus? So, what does it mean? So, brown corpus is a very large corpus it consists of 1.1 million tokens, but what are those tokens. So, brown dot words function will give you the list of the words or tokens that are present in the brown corpus. What I am doing here is I am storing all these tokens into a list called token list and then printing out that list. So, if I am doing this, I am getting the output as the Fulton, County, Grand, Jury, etcetera that is these are the first 5 first 6 tokens from the brown corpus.

Now, the next question is what is the size of the brown corpus? To get the size of the brown corpus, we just need the length of the token list. So, I am using the python length function to get the length of the brown corpus. So, what I am doing is length token list will print out the length of the token list and that is also equal to the size of the brown

corpus. So, if I run this I will get that token total token count is around 1.1 million tokens. So, as I have already mentioned that the brown corpus is categorized into 15 different categories. So, what are the names of those categories? So, if I run the function brown dot categories, it will list the names of different categories that are present in the brown corpus.

For example, if I run this, it gives the category names as adventure, belles, lettres, editorial, fiction, government, hobbies, humor, learned, lore, mystery, news, religion, reviews, romance, science, fiction. So, similar as before, we can ask a very similar question that what are the tokens that are present in our adventure category that is given a particular category what are the tokens that are present?

(Refer Slide Time: 05:30)



Again we will use the same function as before that is brown dot words, but this time the parameter for this function will be the name of the category.

For example, for adventure category the parameter will be categories is equal to adventure. So, the output will be a list of tokens and we are storing that list in a variable called adventure token list then we print the adventure token list variable. So, you can see the tokens are Dan, Morgan, told, himself, he and so on and so forth. Now what is the

size of the adventure category? This is again similar to the size of the entire brown corpus.

What I am going to do is I will again find the length of the list of adventure tokens. So, this is what I have done here I have just use the len function on the parameter that is adventure token list, please see that here if you see the names here like in 18 cell, the names are the tokens actually are Dan Morgan which is a name then, told, himself, he, all these tokens are preceded by a character u. So, u here means that all these tokens are actually Unicode strings; these are not ASCII strings fine. So, till now, we have done, we have got the list of the tokens that are present in the brown corpus, we have got the size of the brown corpus, we also know that the different categories of the brown corpus then we can know the different words that are present in the brown; different categories of the brown corpus and finally, the length of or the size of the each category.

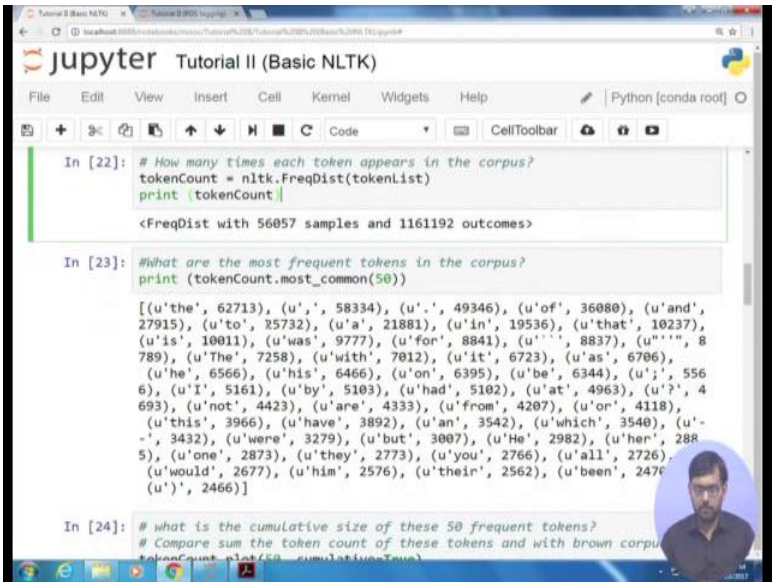
Now, the question is how many unique tokens are present in the brown corpus? So, for this we are using the set function that is a part of python library. So, what is set function? Set function will actually convert a list into a set as you know that in a set all these elements are unique. So, what I am doing here is first of all I will type cast that token list into a set. So, set will have the unique elements and then I will find the length of that set fine. So, if I run this, I will get the set sizes around 56000 that is the brown vocabularies size is 56000. So, these are the unique number of tokens that are present in the brown corpus.

Please note that here I have not converted any of the tokens into lower case. So, please if you want to convert these tokens into lower case then the total vocabulary size will be even lesser. Now can we compute TTR? Now if you remember the TTR was discussed in the first week of the lectures. So, TTR is a ratio of the number of unique elements upon total number of elements. So, here we have just computed the number of unique elements or the tokens and we have computed here the number of total number of tokens.

Let us compute TTR. So, the TTR value here is 0.0482 which means that on an average each token is 21 times present in the corpus. So, till now, we have not done any token

wise analysis so; that means, we do not know how many times each token appears in the corpus.

(Refer Slide Time: 09:27)



It may appear in the corpus or we may talk about the category also like how many times each token appears in a particular category, to get that information we have used here the NLTK function frequency distribution and the parameter for this function is that token list. So, token list if you remember is actually the list of the tokens that are present in the brown corpus.

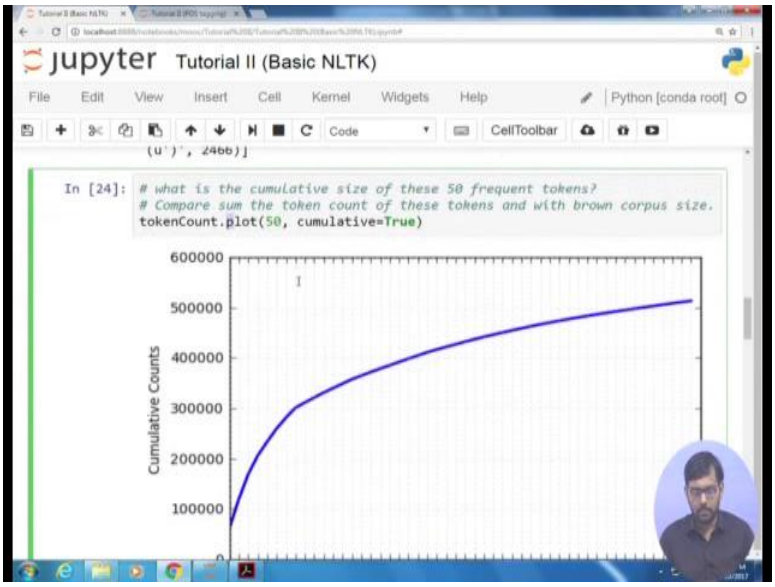
What I am doing here is I am passing the list of the tokens that are present in the brown corpus to the frequency distribution function that is a part of NLTK. So, these functions written that dictionary of tokens with their counts so, the dictionary token count will have the key as the token name and the value will be how many time that token occurs in the corpus. So, now, we have the information stored in the token count that how many times a particular token has occur in the corpus. So, can we get the most frequent tokens in the corpus? Yes, we can use the most common function that is a part of the NLTK.

What we are going to do if we write token count dot most common 15? So, it wins out 50 most common tokens from the token count dictionary. So, as you can see here it is it

has printed the list of first most frequent 50 tokens. So, as you can see the has been the most frequent token then by then comma then full stop and of and to a in that etcetera. So, most of the words here are almost all the words here are the function words. So, this is as for our knowledge that functional words are the majority of the words in any of the corpus.

Now, what we wanted is, what is the cumulative size of these 50 frequent tokens? That is we want to compare the sum of the token count of these 50 frequent tokens with the entire brown corpus size. So, for that I have used a plot function.

(Refer Slide Time: 11:51)

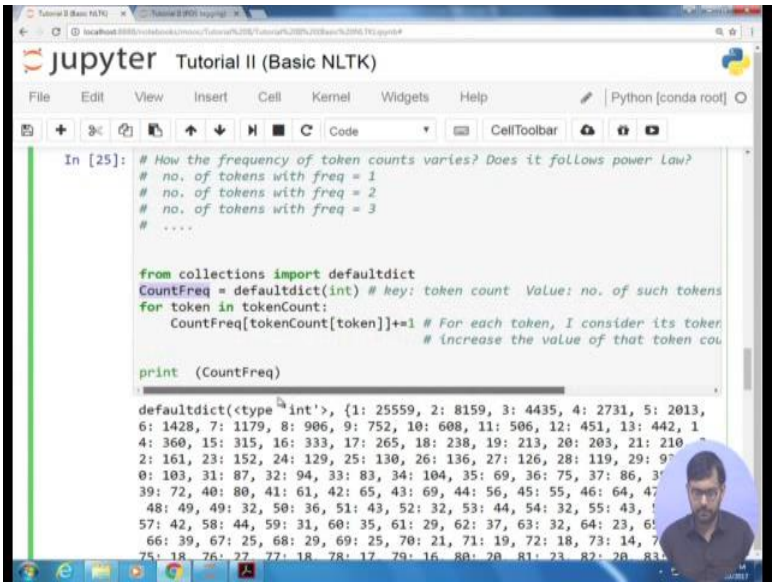


What I am plotting here is I am plotting the token count of 50 most frequent tokens and cumulative wise that is on the x axis we have the tokens and on the y axis we have the cumulative token counts. So, we start from that and then from on onwards we add the previous token count and we get the cumulative value.

The main observation here is that the size of these 50 tokens is almost half of the size of the brown corpus that is these size; these tokens cover almost half of the brown corpus finally, in the end to in the end of the analysis part, how we ask very important question that is how the frequency of token count varies and does it follows power law that is

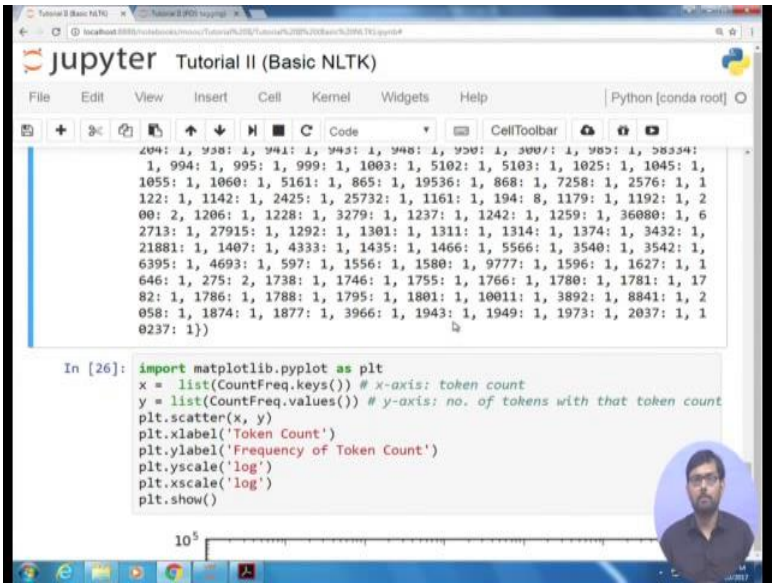
what we wanted to ask is what is the number of tokens with frequency 1? What is the number of tokens with frequency 2 and so on?

(Refer Slide Time: 12:59)



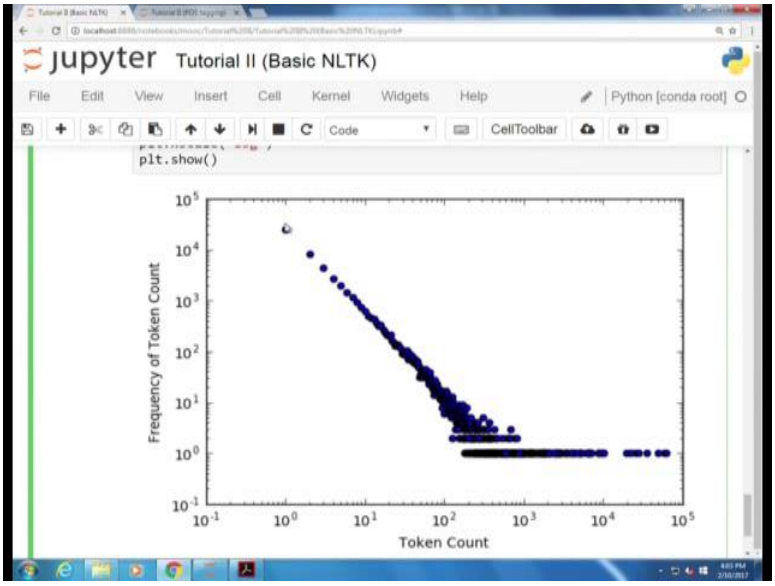
For this, we initialize a dictionary called Countfreq. So, Countfreq will store the frequency of the token counts that is the key will be the token count and the value will be the number of such tokens with that particular token count we iterate over the token count dictionary to populate count frequency dictionary. So, what we are going to do is for each token in count token count I will consider it is token count and increase the value of that token count in count frequency dictionary. So, this is what I have done here and then I have printed count frequency dictionary. So, as you can see here it prints large number of token counts along with their frequency.

(Refer Slide Time: 13:37)



What it says is there exists around 25000 tokens which have just 1 frequency, similarly there exists around 8159 tokens which occur twice in the corp brown corpus and so on, now can we plot this distribution? Yes, we can plot this distribution using Matplotlib. Matplotlib is a very useful tool for plotting in python.

(Refer Slide Time: 14:17)



What I am doing here is on x axis? I have stored the token counts and on the y axis, I am storing the number of tokens with that particular token count then I am just plotting a scatter plot x comma y and this plot for this plot, we have scaled x and y axis on the log scale. So, as you can see, this is a very famous power law plot and this gives the t log that power log. So, on x axis we have the token count on y axis we have the frequency of the token count. So, you can see. So, this actually concludes our first part of the tutorial that is the analysis of corpus using a NLTK.

For more information, you can question on the you can click on the link that I have provided in the slide and you can for any installation query also you can question on that same name now going to the second part of the tutorial that is a basic introduction to pos tagging. So, first of all what am doing here is I have imported a NLTK. So, let us first import NLTK that is the basic requirement for the second tutorial. This NLTK is now given a sentence can we pos tagging. So, the sentence here is this is a basic tutorial on pos tagging.

First of all, we will tokenize this sentence and then we will pass it to the pos tagger. So, what we are going to do here is NLTK dot word tokenize sentence. So, word tokenize is the function of NLTK that will tokenize the sentence and the tokenize sentence will be pass to the pos tagger. So, the tokenize sentence is stored in a Tokenizedsent variable and we have print the Tokenizedsent variable here. So, Tokenizedsent consist of nine tokens.

Now we have then we have directly pass Tokenizedsent into the pos tag function which is actually a pos tagger of NLTK and we have printed the output. So, the pos tagger has pos tag the Tokenizedsent list. So, let us see the result. So, this here is the determiner is here is third person singular present, a is again determiner, basic is adjective, tutorial is singular now, on is preposition is proper noun singular tagging is singular noun and then we have the period. So, here we have given a example sentence, but we can also do same for the brown corpus.

First of all, let us import the brown corpus. So, we have imported brown corpus as we have done before say from NLTK dot corpus import brown now what are we going to do

next is how to get sentences from the brown corpus. So, brown dot sentence actually gives you tokenized sentences from brown corpus. So, if I write brown dot sent and we print the result we get the sentences. So, the first sentence is the Fulton county grand jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place and then sentence ends.

Similarly, we have the other sentence and the sentence (Refer Time: 17:50). So, we directly pass each of the sentence into the pos tag function and we get the pos tagged output. So, what we are doing what we have done here is we have passed the first sentence of brown corpus into the pos tagger and we have got the result.

So, with this I would like to conclude for the tutorial and in the next tutorial we will be talking about some more advanced techniques of NLTK.

Thank you.