Business Analytics And Text Mining Modeling Using python

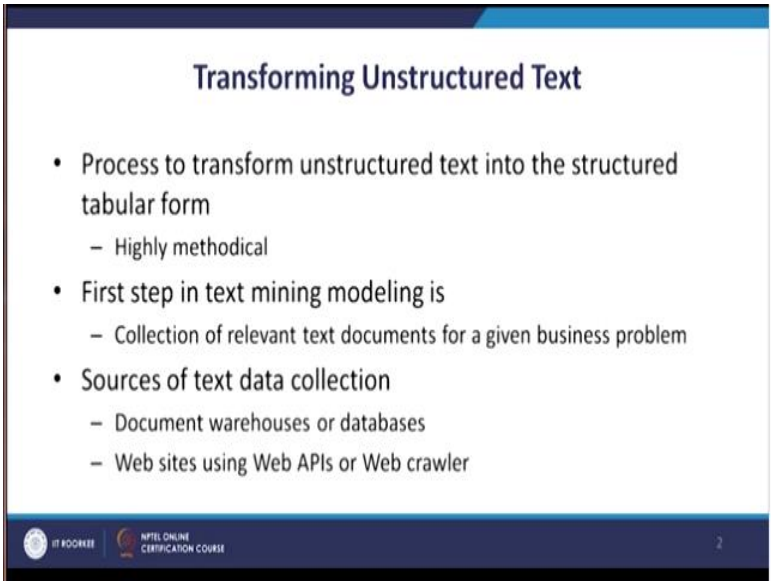
Prof. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Lecture-36
Text Collection And Transformation-Part I

Welcome to the course business analytics and text mining modeling using python. So, in previous lecture we completed our discussion on the most common python modules and again in a good amount of expertise in python environment. And we also started about discussion on the some of the you know important text mining modeling aspects. So, first thing is about transforming unstructured text, so we test upon a few points in the previous lectures.

So, we will do a small recap and then we will pick up from the point where we stop in the previous lecture. So, as we discussed you know that in the initial few lectures that the differences between text mining modeling and data mining modeling and how we will have to deal with the unstructured you know text, so few points we will do a recap.

(Refer Slide Time: 01:15)



Transforming Unstructured Text

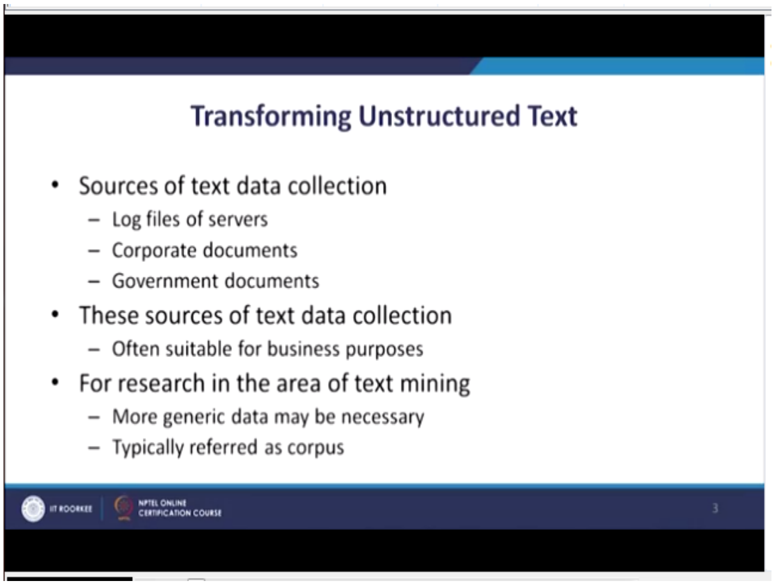
- Process to transform unstructured text into the structured tabular form
  - Highly methodical
- First step in text mining modeling is
  - Collection of relevant text documents for a given business problem
- Sources of text data collection
  - Document warehouses or databases
  - Web sites using Web APIs or Web crawler

So, first step is step recollection of you know relevant text documents, so we talked about different sources of you know these documents. So, it could be document warehouses or databases if your organization is you know has invested a lot in the information systems computer-based systems. They might be having those servers where various repositories of you know rules, guidelines, technical documents you know knowledge transfer documents many other kinds of documents could be available.

So, all those documents you know could be you know sources of text data collection. Then we have websites also, so web APIs like you know twitter provides web APIs to extract tweets, web crawler program could also be used to extract you know text content from various websites. So, all these all are important sources of text data collection, then we have log files also.

So, in our operational systems there are so many you know things so many operations related activities tasks and various other business related activities you know they might be being logged in our you know servers in a business organization server. So, those log files itself could be an important collection of text document because we might like to gain certain insights from those you know logs and we like to help our you know business objectives with those you know insights that we generate.

(Refer Slide Time: 02:42)



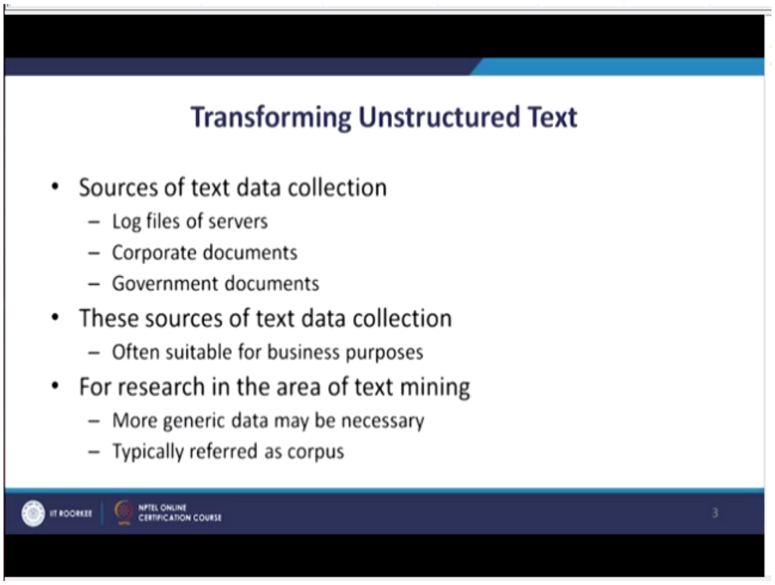
Transforming Unstructured Text

- Sources of text data collection
  - Log files of servers
  - Corporate documents
  - Government documents
- These sources of text data collection
  - Often suitable for business purposes
- For research in the area of text mining
  - More generic data may be necessary
  - Typically referred as corpus

So, similarly corporate documents and government documents so many you know public and private organizations. Even if they have not been really into you know investing in computer systems and formation systems. And do not have digitized versions of documents, they might still have a number of documents and different sources in their personal computer in their you know hard copy format, you know different files they might have their paper-based system.

So, all those are also if they could be digitized and you know they could also become sources of you know text data collection. Some of these documents will be it might be related to policy, rules, regulations and many decisions, many meeting minutes that could be there. So all these documents could also be sources of text data collection. So, this is about different sources of text data collection.

(Refer Slide Time: 03:31)



Transforming Unstructured Text

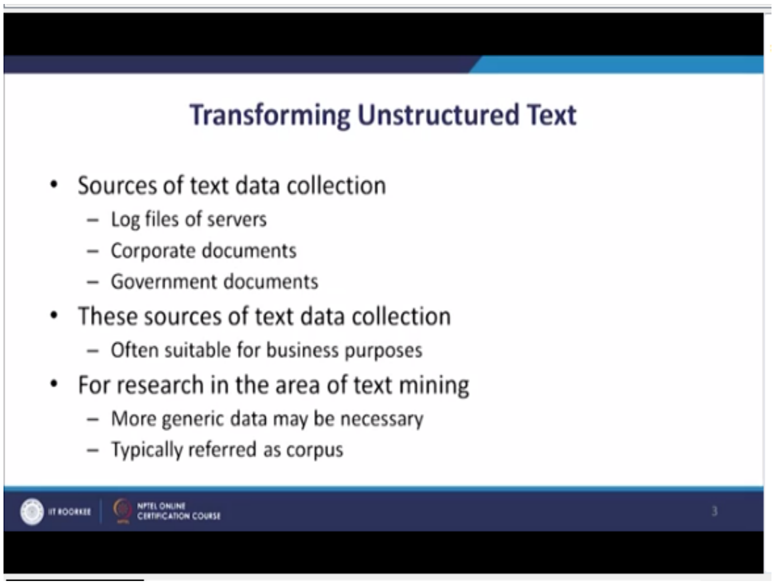
- Sources of text data collection
  - Log files of servers
  - Corporate documents
  - Government documents
- These sources of text data collection
  - Often suitable for business purposes
- For research in the area of text mining
  - More generic data may be necessary
  - Typically referred as corpus

So, typically if we look at these sources they are suitable for business purposes for research in text mining typically these sources are slightly different because we would be requiring more genetic you know data. Because if we talk about a particular public organization or private organization the documents would be related to the you know the functioning of that particular organization. So, there would be very specific, so specific to that particular organization.

So for improving the working and helping the business for that organization we would like to mine even those documents. But those that would go that would be more the job of the analysts which are working in that particular organization. However they might not be really useful for the research purposes for researchers working in universities.

So, for that they try to develop different for research purposes, they try to develop different kind of you know text collection of text documents which are more generic in nature. Because you would like to generalize some of the findings that we have you know after applying you know text mining modeling techniques.

(Refer Slide Time: 04:55)



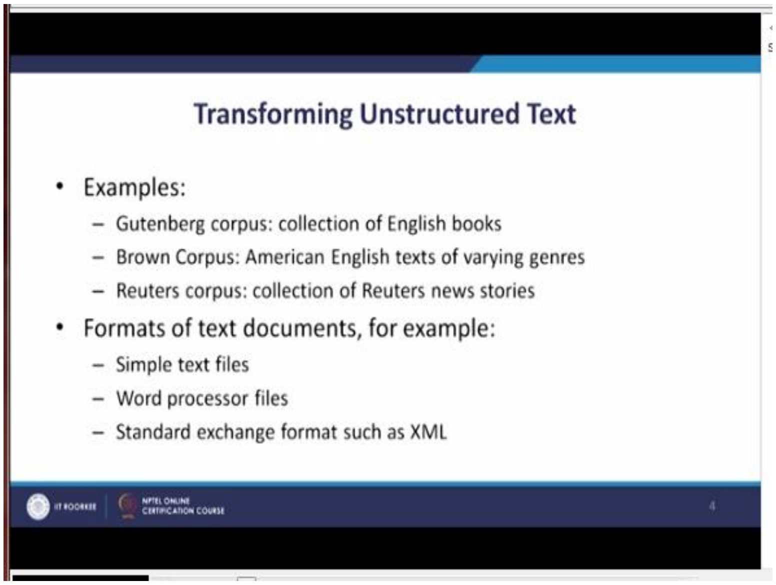
Transforming Unstructured Text

- Sources of text data collection
  - Log files of servers
  - Corporate documents
  - Government documents
- These sources of text data collection
  - Often suitable for business purposes
- For research in the area of text mining
  - More generic data may be necessary
  - Typically referred as corpus

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
3

So, these collection of documents are typically referred as corpus we call them text corpus.

(Refer Slide Time: 05:02)



Transforming Unstructured Text

- Examples:
  - Gutenberg corpus: collection of English books
  - Brown Corpus: American English texts of varying genres
  - Reuters corpus: collection of Reuters news stories
- Formats of text documents, for example:
  - Simple text files
  - Word processor files
  - Standard exchange format such as XML

IIT ROORKEE
NPTEL ONLINE CERTIFICATION COURSE
4

So, few examples or for example a Gutenberg corpus collection of English books, round corpus American English text of varying journals then we have reuters corpus collection of readers news stories. So, you can see different you know kind of text corpuses that have been typically used for research purposes. So, these are some of the corpuses you know that are easily available to us.

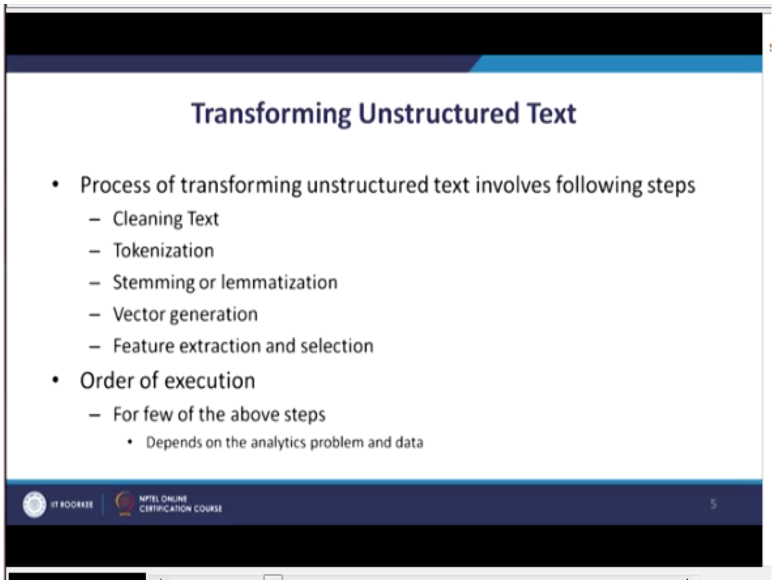
So some of the demonstration, some of the you know examples exercises that we would be hands-on exercise that we would be going through in this particular course. We would be using some of these you know text corpuses and they would be really useful in the context that because they are generic, so they could be used for research purposes. So, therefore you know they would also help us you know understanding learning about you know it will make it easy for us to work with documents specifically related to a particular organization.

Now let us talk about the formats of text documents, so there could be, so essentially textual data would be present in you know a number of you know different formats. So, it could be in simple text files format also, it could be you know word processor files you know microsoft word files or simple text files, it could be standard exchange format such as xml. So, as we understand you know most of the data you know if we look at the content publically available sources of data.

So **so** major chunk of that data actually is coming from internet that is worldwide web. So, from internet mainly from the websites world wide web, so they are these markup languages html and xml they are the more common format because of the you know portability they provide because of the platform neutrality of internet and world web that they facilitate.

So, standard exchange format so much of the textual data is also available in these formats also. So, you know some of these are you know some of the important file formats and sources of data collection that we could actually use. Now let us move forward now we will talk about process of transforming unstructured text and this requires a number of steps.

(Refer Slide Time: 07:19)



Transforming Unstructured Text

- Process of transforming unstructured text involves following steps
  - Cleaning Text
  - Tokenization
  - Stemming or lemmatization
  - Vector generation
  - Feature extraction and selection
- Order of execution
  - For few of the above steps
    - Depends on the analytics problem and data

So, some of the common steps are for example cleaning text, tokenization, stemming or lemmatization and vector generation, feature extraction and selection. So, you know cleaning text is about you know for example if we are using html or xml files there are going to be so many you know markup tags, so we would like to get rid of them and just would like to have the text content over there.

We have the text files it might be they might be having certain comments there, so we would like to get rid of that. So, the cleaning text particular step is actually you know about removing the you know undesired content from the those text files, those text formats. So this is step is about that. Once we have got the you know the text document number of text documents mainly containing this textual content that we require that we would like to analyze, that we would like to mind to gain you know insights.

So, then will once we have that we can go ahead and execute the next step that is tokenization. So, in this particular step you know different types of tokenization can be performed, for example we can consider a text document comprising of paragraphs and further sub components could be sentences and words. So, depending on the kind of you know component that we want we can perform different kind of tokenization.

So, typically it is sentence tokenization or word tokenization that is typically performed and word tokenization is the more common because it could be really useful for some of the analytics you know tasks where would like to focus. So, tokenization is important step, so essentially we are trying to convert a chunk of text either into sentences, separate sentences.

So, those separate sentences are going to be called tokens sentence tokens or we would like to convert those you know those chunks of in that chunks of text or paragraphs of text into words and in to you know separate words and those words could be taken as tokens. So, this tokenization processes about transforming making this transformation happen from a larger chunk of text to smaller chunks of text which are actually you know which could be actually sentences or words.

Now the next is stemming or lemmatization, so in the stemming or lemmatization we are essentially we are actually looking to analyze the stem of those words. So, some of the words that we might be using you know they could be different forms of different forms of the same root word. So, we would like to identify those root words because you know that would be more beneficial for us to create a tabular format.

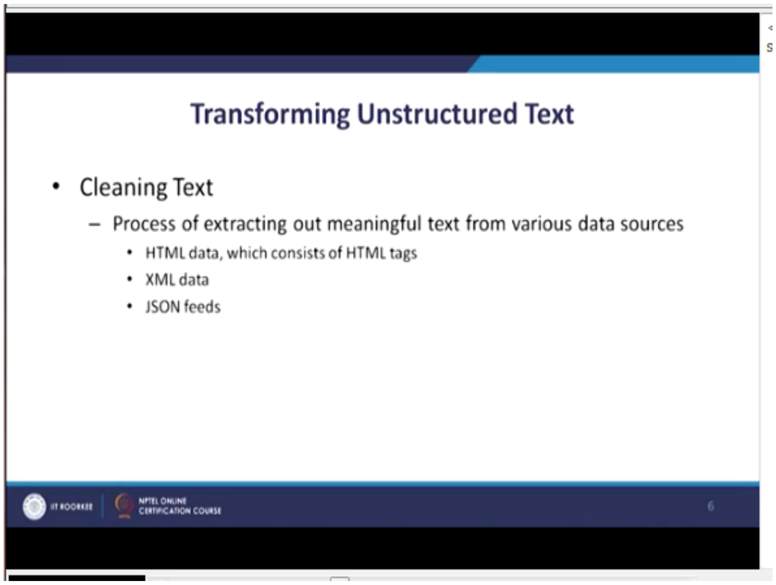
Because you know typically techniques deal with the frequencies of those root words, so therefore if many forms of if there are many words are there which are coming from same root word, so they will all count into the will be counted at you know come increase the frequency of that particular root word. So, that would be really beneficial for the text analytics you know purposes, so stemming and lemmatization could be useful for that.

Then vector generation, so from after stemming a lemmatization we would like to further process the data the tokens that we might have there and based on that we would like to create you know vectors. More on this we will be discussing in coming lectures, then the next step is about feature extraction selection. So, this is step is about the you know features which will be finally taking into the analytics which will finally taking into the text mining modeling.

So, whatever we are going to finally take into the modeling, so extracting and selecting those features, so this is step is about that. So, all these steps will be discussing one by one in the coming lectures. In terms of order of execution as you would not understand you know with our discussion on these some of these steps that some of these steps you know there is no there might not be a particular order that is must to execute some of these steps .

So, that order might depend on the analytics problem and data at hand if you have a slightly refined data then probably you might skip some of these steps and just go ahead with the you know core processing that is required. So, that is also an important to learn you know looking at the problem, looking at the data, what steps might be required, what steps might not be required. So, all those things are also important for us to understand in this you know text mining modeling context.

(Refer Slide Time: 12:48)



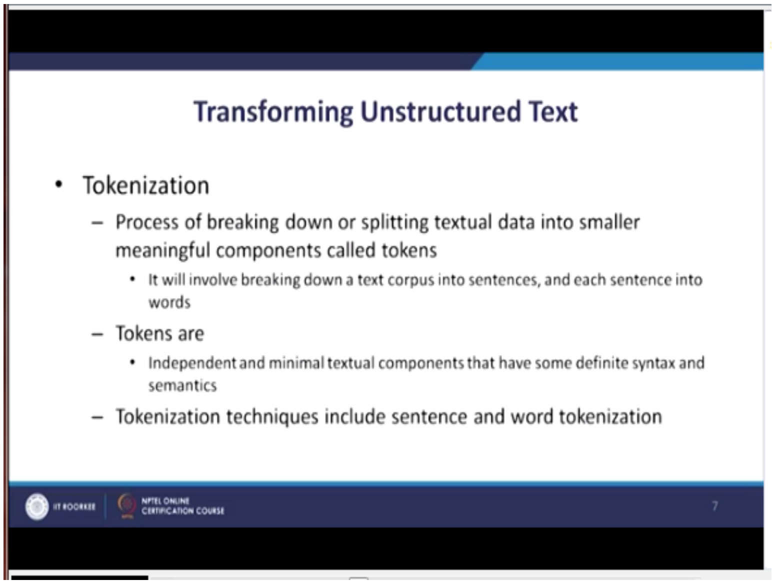
Transforming Unstructured Text

- Cleaning Text
  - Process of extracting out meaningful text from various data sources
    - HTML data, which consists of HTML tags
    - XML data
    - JSON feeds

Let us move forward, so first step as we talked about is about cleaning text, so in this step it is all about the process of extracting out meaningful text from various data sources. So, we talked about various file formats where the data could be stored using those formats. So it could be a stemmer data, so it could consist of html tags, xml data, xml tags or JSON feeds. So, you know some of these formats require other kind of you know meta information and other things.

So, we would like to attract you know only the meaningful text, text that we would like to mine, so the cleaning text step is about you know performing this particular process.

(Refer Slide Time: 13:37)



Transforming Unstructured Text

- Tokenization
  - Process of breaking down or splitting textual data into smaller meaningful components called tokens
    - It will involve breaking down a text corpus into sentences, and each sentence into words
  - Tokens are
    - Independent and minimal textual components that have some definite syntax and semantics
  - Tokenization techniques include sentence and word tokenization

The next step is about tokenization, so essentially as I discussed tokenization this is going to be a process of breaking down or splitting textual data. So, as I said chunk of data we might have paragraphs of you know text we might have. So, we would like to break it down or split that textual data into smaller meaningful components called tokens. So, typically meaningful you know components are sentences words in the analytics context.

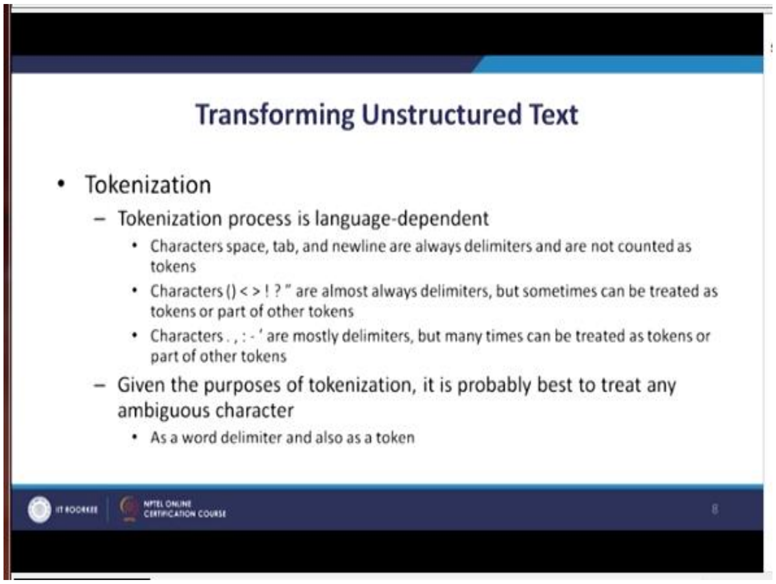
So, we would like to have sentence token or word tokens mostly world tokens. So, this tokenization process is actually achieving this, splitting, breaking down and splitting textual data into you know meaningful components like tokens. So, what it involves it will typically involve breaking down text corpus into sentences and each sentence into words, so this is how the typical this tokenization is actually process is actually achieved.

If we look at the tokens and how we can define it, so as I talked about the root word, so if you keep that into the mind that essentially we are in the lemmatization and in the another step that we talked about here as you can see stemming and lemmatization. There we might be focusing on this stem all emma's, the root you know root form of words there. So, keeping that in mind tokens are supposed to be independent and minimal textual components.

Because essentially we are suppose to perform another steps will essentially we have vowed this. So, if any way we are able to reach much closer to those root forms, so those tokens that tokenization process would be consider more complete more comprehensive. So, essentially tokens can be defined as independent and minimal textual components that have some definite syntax and semantics.

So, that syntax and semantics should not be changing, so that would really very close to the root form of the word or stem or lemma different you know terms that we will be using for this and discussing in coming lectures as well. So, we would like to generate these tokens from the textual data that we might have and tokenization techniques typically it involves sentence and word tokenization.

(Refer Slide Time: 15:58)



Transforming Unstructured Text

- Tokenization
  - Tokenization process is language-dependent
    - Characters space, tab, and newline are always delimiters and are not counted as tokens
    - Characters () < > ! ? " are almost always delimiters, but sometimes can be treated as tokens or part of other tokens
    - Characters . , : - ' are mostly delimiters, but many times can be treated as tokens or part of other tokens
  - Given the purposes of tokenization, it is probably best to treat any ambiguous character
    - As a word delimiter and also as a token

Now if we look at the tokenization process essentially when we talk about text, so text is typically going to be in a certain you know natural language, it could be English or you know any of the local languages also. However if we look at the you know most of these you know packages and platforms that we have for analytics, they are morally they have been mainly developed for European language English another European languages.

However if you look at the tokenization process, this is a language dependent process because you know the kind of punctuation that kind of you know the syntax and semantics that could be use in one language might vary from one language to other language, so the whole process going to be language dependent. However be you know like to focus on few aspects of tokenization process.

So, this could be about for example characters, for example space, tab, new line, so they are typically always delimiters and not counted as tokens. So, they are just to separate out different words in English language and in many other languages also. So, typically we would like to exclude these characters, strip of these characters space, tab and new lines.

If we talk about other characters for example parentheses you know less than, greater than, you know symbols, exclamation mark, question mark all these in symbols. There you know not always well almost always you know delimiters, almost always are being used as delimiters. But sometimes they can also be treated as tokens or part of other tokens, they could be part of other tokens.

Sometimes we might be defining a terms between these you know using parentheses or some of the other characters. So, and you would like to consider you like to emphasize that term using these you know marks using these characters punctuation marks. So, for that you know we might have to use them as part of other tokens or you know they themselves can be treated as tokens depending on the problem at hand depending on the data that we might have but almost always they could also be you know treated as delimiters.

If we talk about some other characters for example dot, comma, colon and dash they are also mostly delimiters but they have more scenarios where they can be treated as tokens or part of other tokens for example dash and colon. They might you know really emphasize on some of the words and therefore might really useful to have them as part of tokens, how would we look at the whatever we are discussing till now that is quite you know coming from our knowledge of English language.

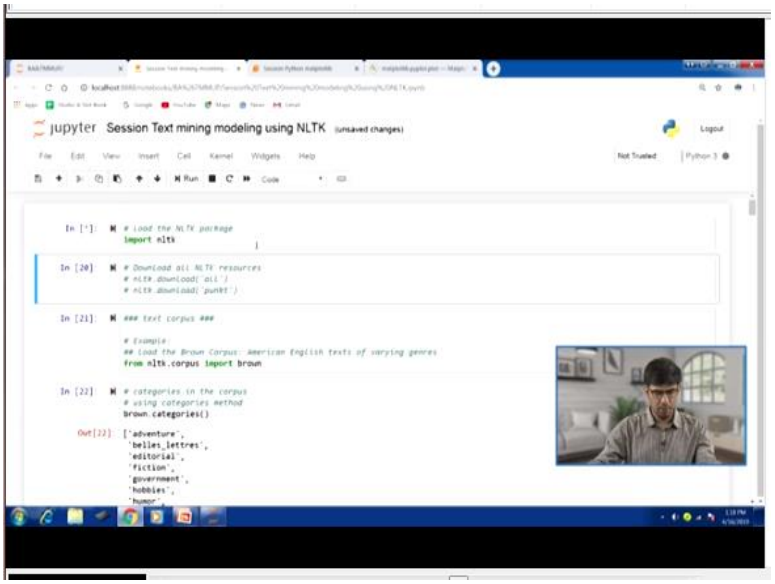
So, the whole process the analysis that we just did, these point that we just talked about, these are all language dependent because this all is coming from the syntax and semantics of those languages. So, the whole process the tokenization process is actually you know language dependent. If we try to you know extract some general principle rules from some of these observations that we talked about.

So, you know we can it would be probably best to treat you know any ambiguous character that we come across both has a delimiter and as a token. So, we can keep it in both the places and of course no one stops us from building a number of candidate models. So, we can have a candidate model where we are using some of these you know these characters as tokens and 1 model where we are we have excluded them stripped of them.

So, we can always compare performance, so there is no point in you know focusing too much on this. However we should have a consistent we should always try to find out a better you know principle which will work most of the time. So, if we find something which is slightly ambiguous we can treat is treated as a both as a delimiter and also as a token. So, with this what we will do we will move to you know Jupyter platform.

And we will talk about some of the things, some of the concept that we have discussed through few exercises and then we will move to the next step in this you know transformation process.

(Refer Slide Time: 20:39)


jupyter Session Text mining modeling using NLTK (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Run Code
Logout
Not Trusted
Python 3

```
In [*]:  # Load the NLTK package
         import nltk

In [20]: # Download all NLTK resources
         # nltk.download('all')
         # nltk.download('punkt')

In [21]: ### text corpus ###

         # Example:
         ## Load the Brown Corpus: American English texts of varying genres
         from nltk.corpus import brown

In [22]: # categories in the corpus
         # using categories method
         brown.categories()

Out[22]: ['adventure',
          'belles_lettres',
          'editorial',
          'fiction',
          'government',
          'hobbies',
          'humor',
```

So, one thing that we did not touch upon is the you know the kind of package that we will be using to perform some of these task. So, the NLTK you know natural language toolkit you know that we have this is typically the most popular package available in python platform that is used for text analytics text mining modeling and you know natural language processing as well. So, we would be using this package to perform some of these steps that we just talked about.

So, first thing that we need to import this NLTK, so before that we need to download you know many NLTK resources. If you have done your installation of python environment using you know anaconda package and the default complication that we did for this course, then NLTK would already be installed NLTK package would already be available there in your setup. So, if I run the this first line here then you can see that we are importing the whole NLTK here.

So, it will take slightly more time you can see asterisk sign is still there, so NLTK as I am telling you this package is already there so we would be able to import that. But the next step is you know depending on if you are doing this process so for the first time then there are certain NLTK resources also that we would be using in you know you know exercise that would be forming.

So, you can see now in 1 is there that means NLTK has been loaded into this python environment, Jupyter notebook environment. Now the next step is actually about downloading the required and NLTK resources, so this also includes a number of text corpuses that we might have.

(Video Starts: 22:38)

So, we would like to install this as well so what I will do I will comment out I will remove the comments from some of these comments here. So, we would like to use this NLTK.download all. If you just require only you know few resources you can specify them also. But for the purposes of our course will go with the NLTK.download all. So, I will run this, so once we perform this it will start downloading the resources in the environment, in the environment that we have here.

So, before we move ahead and use this you know particular package for our text analytics you know text mining modeling that we are going to use in this course we need all these resources. So, this will include some of the corpuses that we talked about as you can see in the next line of code round corpus another things you can see here. We would be loading the brown corpus which we discuss in the slides.

For example this is American English text of varying generous so you can see, this download process here. So, once this is installed we will actually go ahead and take the example of this round corpus, so you can see will be from NLTK.corpus will be importing this one brown corpus. And then we will start our discussion on the you know various aspects of this collection of text documents.

So, essentially it is the text data that we will be dealing with typically most of the text corpuses they have you know certain categories in the corpus. So, because the text can be you know organized in various categories and there would be may various files under those categorie4s. So, we can consider it like a you know tree kind of structure where you have the name of your corpus let us say brown and then would be certain categories and within each categories there are going to be a number of files.

So, it is you can you know think about these corpuses in that sense, so typically first thing that we do is once we have a corpus typically, first thing that we do is we try to have a look at the kind of you know categories that might be there and then we can process further. So, let us wait for this download process to complete and then we will move ahead. As you can see till this a strict sign you can see there that means some processing is going on.

Once it completes the processing completes a number is going to be assigned at this in place of this state just like here. We got 1 because we were able to import NLTK package because it came with the anaconda, so it was there it just imported. Now this is being downloaded NLTK.download all, this is still going on. So, as you can see that after running this command NLTK.download all now you know looking the whole download process is complete.

Now we have access to all the NLTK resources that we might be using in this particular course, so let us move ahead. And the next thing that we would be doing as we discussed that will talk about the text corpus in this case, so we will start with the brown corpus that is that we have just downloaded. So, as we discussed that this is about American English text or varying generous.

So, first thing now once downloaded now we would like to import this into the python environment here, so for this from NLTK.corpus import brown.

So, we will run this then we will have access to this particular corpus, now as we discuss that you know we can think about these text corpus as a kind of a tree kind of a structure where the corpus corpuses might be divided into a number of categories. So, those could be different branches of the tree starting from root node and then within those you know branches we might have a number of files you know within those categories.

So, let us look at the categories in this particular corpus that are there, so for this we can use the categories method. So, we can call this you know this corpus object brown, so in this case brown object brown dot categories, so this program method will give us the list of categories that are there in this particular text corpus. If I run this we will have this list as you can see.

So these categories are their adventure you can see editorial is also their fiction is there government is there humor and many other categories are part of this particular corpus. So, under these categories there are going to be a number of files in this text files in this corpus. So, let us move forward let us have a look at first 5 sentences in this corpus, so to read those sentences we can use the sense method here and a list of lists of strings is going to be written here.

So, we call brown dot sense then we can have the all the sentences read into a objects and sense variable that we have here. And then we can have a look at we can use this indexing mechanism 0 to 5 so we can print those sentences here. So, if I run this in the output you would see that a lists of strings is presented here. So, if you start you can see that we have you know double brackets there the first sentence is starting and different words you can see part of that first sentence.

And if I just scroll here in this output you can see here in this place a dot you can see and the bracket is there then a new bracket is starting. So, it is a list of list of the strings and within each list you can see number of strings and you can see even the punctuation marks. If you look at the output here, now we will scroll back to the top and you can see even the punctuation marks for example the space and different kind of you know characters they have also been you know split it and they are being treated as a separate kind of you know words there.

So, this is how the output is going to be displayed in this when we use the sense method. So, easily we can read different various sentences that are part of the corpus. Now if you would like to you know concatenate these strings and I would like to have slightly more readable format. So, that we can have a look at some of the sample sentences from that corpus, so for that we can use the join method which we have discussed in our you know our discussion of you know python.

So, we can call this you know a join method here we can write this list comprehension and for each words in the in sentences we can use the join and concatenate these strings. And then we can have a look at you know again first 5 sentences in this fashions if I run this then you can see in the output 6. We have got these sentences here you can see and the Fulton County Grand Jury said Friday and investigation of this and that.

So, these kind of you can see you know each sentence if you observe this output one sentence is complete and then after comma a new line a new sentence is starting. Similarly if you look at first and then you know second sentence here and then these the jury further said the second sentence then the September/October term the next sentence. Then next one start from only a relative handful, the next one is start from the Jury said it did fine.

So, in this fashion you can see thus that you know those 5 sentences have been displayed there. Now this is about from the corpus now we just want to focus on a particular category in that corpus so that can also be done. So, for that we can again use the you know sense method there but within the parentheses we can pass on the categories keyword argument here where we can specify the category where we are interested in. We can of course as we learn in python we can of course pass on more than one category here.

So, we are going with the categories government here, so the files from this product you know category they are going to be passed for this processing and the sents method. So, if I run this and then in the same in this particular example we are again using the join method also to concatenate these strings that we get from the you know sense method.

So, then we look at the first 5 sentences, so if I run this so as you can see if I run this so these 3 lines of code if I run this and have a look at the first 5 sentences, you can have a look at the output. Now this output is actually coming from the you know this particular category government and you can have a look at the content also here. So, first 5 sentences you can see they are coming from that category the office or business economics of the US Department of commerce.

So, you can see these are these documents relate to the government, so this is how for the corpus also and for the particular category also, this is how we can read sentences. And have a look at the sample you know sample sentences from the corpus to understand what kind of textual data we might we are dealing with here. Now we have file IDs also in the corpus because corpus is about a number of collection of you know text documents.

So, these documents they also have their corresponding file IDs there. In the so those file IDs could also be used to actually access these you know text files. So, for that we have the file IDs method, so for this corpus brown if I call this file IDs method, I will have the list of files that are part of this corpus. So, let me call this brown dot file IDs if I run this in the output 11 you can see the file names a list of these file names has been produced in the output.

So, these files are part of this process, so if I am interested in finding out about the files in a particular file IDs in a particular category, so that can also be performed. So, in this case you know we can call this file IDs method here again and there we can pass on these categories are women categories you know an humor. So, this further humor categories we would like to have a look at the file IDs that are there number of files and you know filenames that are there.

So, if I run this you can see now we have got file IDs for humor, so of course you can see in the peep from the previous output on this output, these file IDs are different CR01, CR02, CR03. In the previous output the file id started like GCA01, CA02, so of course they were started there those were from different category and this is from the you know category humor.

Now earlier we looked at the you know text coming from a particular you know in the whole corpus and the category. Now we will look at you know the text coming from a particular file, so for that again we can use the file IDs argument in the sense method. So, we can call this brown dot sense method and in the within the parentheses we can use this file ID the argument and specify the list of files that we would like to you know pass here.

So, let us say CA18 and then we can again use the join method to concatenate those strings you know there. So, then we can have a look at first 5 sentences, if I run this you can have a look at the output you can see the Night in New Orleans. So, you can see that, so this purple sample text you know 5 sentences is actually coming from this file ID. And of course it will belong to you know one of the categories in the corpus that we have.

(Video Ends: 36:49)

So, different text output we have got because different we are different file we might be accessing. So, this was about the you know brown corpus and in this we were able to you know demonstrate how we can read the you know textual content that is there using different methods sense and the join method to concatenate and file IDs argument categories argument. So, different ways we can have a look at the sample data from a text corpus.

This is going to be really important in our further processing and also you know modeling. Because some of the you know insights that we can have by looking at the you know sample text is also going to help us in those steps. So, we would like to stop here and we will continue our discussion on this transforming text. And the next thing that we will pick up is Guttenberg corpus and there also will do some of the similar steps and then we will move forward to the tokenization process, thank you.

Keywords: structured and unstructured text, Argument, delimiters, text corpus, concatenate.