LECTURE 32 : Text Pre-processing

Hello learners, in the previous lecture I have discussed about the regular expression and how we can use the regular expression to match the pattern in the text dataset. So in this lecture I will be discussing about the pre-processing of the text data which is a very important step whenever you are starting any of the analysis on the text data. So the pre-processing of the text data is done because of several reasons depending on the type and purpose of the text data. So some of the common pre-processing of the text data is highlighted in this slide. The first is the removal of the noise from the text data. Many times it has been seen that the raw text data contains some URLs and other irrelevant information that is not so important in the context of our analysis.

So we need to remove all those kind of text from the raw data. Then we remove the punctuation from the text data. So you know these punctuation like comma, semicolon, hyphen, etc. are just the connecting symbol which are dividing the sentences and the phrases.

And these connecting symbols are just for structuring the text for reading purpose, does not carry any of the semantic meaning. So we remove these punctuation from the text data. So then we do the lower casing of the text data. So the lower casing is done to map the similar word in the same group. So say for example there is one word is like Mohit is in the lower case and another word is Mohit which is in the upper case.

So these two Mohit conveying the same meaning that there is a person whose name is Mohit but one is written in lower case and another is written in the upper case. So we need to map these two words in the same group that these two words one is written in the lower case and another is written in the upper case but they are the same word. So next we do remove the stop words. So stop words are the most occurring word in the text data like a and the and etc. and all.

After that we remove the numbers like some numbers are there, some digits are there. So we remove those things from the text data and in the final step we do stemming and lemmatization. So stemming and lemmatization are the two key technique in the analysis of the text data which map the words to the base or the dictionary word. What exactly stemming is and what exactly lemmatization is we will discuss when I will be discussing particularly stemming and lemmatization. So these are some of the things what we do in pre-processing.

Again I will mention that the pre-processing of the text data depends on the type of data and the purpose of your analysis. So in this lecture we will be taking up the text of scientific publication and we will be doing some of the pre-processing on this data. So let's move to our R studio where we will be importing the data and then we will be doing the pre-processing of that data. So before doing any of the analysis the very first thing what we

do we need to know that what exactly our directory is and how we exactly check our directory using the function getworking directory. So if I do like this it will give me that my directory is user del documents and whatever the directory you are in whenever you are doing the analysis and if you want to change the directory like my current directory is document but if I want to change it to download so I can change it using the set working directory.

So I have kept my data this particular data in the documents directory so I don't have to give the whole path I have to just mention the file name. So this is a data so this is a dummy data which I am taking up for this analysis. So what this data is this data is the 50 documents which are published in journal of cleaner production so we will see it. So this is my data set it's in documents so I am reading it and I am assigning to an object df jcp. So this is my data set and my data set is imported after reading the data what next we will do we will see what exactly that data is so how we will do that we will view it so we will use the function view and if I run it.

So this is my data set is it's a all scopus data which have all the details like authors and page count citations affiliation and all but for this text mining thing we are only focused on title and abstract. So we will be considering this title part and then our abstract part. So this is our abstract so from our data set we have lot many columns so if we have to see what are the number of columns are there in our data set what we will use will use ncol and we will give the name of our data set. So there are 34 columns are there but we are focused only on the two column what are those two columns are one is the title and another is abstract. So how to access the title and abstract will use the $ sign.

This is my data set and if I write the $ it will automatically give me the name of the column so here I am taking the title and another I will be taking the abstract. So if I see first title of my data set so this is how I can check it so this is the title of that particular paper and then if I have to see the abstract in the similar way I can check the abstract of our data set. So now we have a data set we have a data set of 50 publication published in journal of cleaner production now we have to do the pre-processing on this text data before going for the final analysis of topic modeling. So this title is in one column and then abstract is in another column so what we will do we will combine it and make it as a single column because we are considering both title and abstract for our analysis so what we will do we will just simply combine this thing. So for combining we have a function in R which we call it as paste let us see how paste works first so this is a paste function and if you want to know about paste so what we will do we will use the this question mark symbol and if I run it so basically this paste function concatenate the strings.

So and how we have to call it we have to call it in this way the examples are also given. So let us take one example how this is connected so let us say for example we have one text Mukesh Vaira and another text Mohit Gurd so if I have to concatenate those two things okay so how we will do we will use this paste function and then we will give the this is a text one

and this is a text two and combine these two string only thing is that separated by a comma if I do like this so my output is this so my two strings are concatenated they are joined together then if I have to like give extra space or something like that I can assign here and if you see here the difference between that these two joined string okay so the first one is that only comma is there but in the second output you see the comma and then space is also there okay so in similar way we can join our two strings of this title and abstract also so we will combine these thing so this is our whole title data and this is our whole abstract data so we will use this function in this way that this is my title and this is my abstract and it should be separated by either we can give the full stop or comma whatever like we can give so this is my function paste which I am calling it here and after pasting it what I am doing I am also making it as a data frame I am making one more column of that data frame okay so now I will run this and assign to this object called dfjcptiab so if you see I have just given the name related to what exactly my data set is okay so whenever you are creating an object and you are doing analysis always try to give some kind of words which are related to your task okay so if we see this particular object is showing that this is a data frame then jcp is the short form of general of cleaner production and what I am saying is ti is title and ab is for abstract okay you can give any other name like you can just assign ti or a also okay ta okay so in this way also you can do so but I am going in this way so if I run this so my data frame is created so what exactly we have done here is so this is my title data is there okay and this is my abstract data and then I have combined it and it should be separated by dot and after combining it I am making a as a single column of a data frame okay like say for example if I do like this one okay so what exactly it is doing it is combining title and abstract and if I run this so this is my title drivers for and barriers to low energy buildings in Sweden and from here my abstract is there okay so if I have to see my title so this is my title and you have to see the first step and this is my abstract okay that is what exactly we have taken up the whole old 50 titles and all 50 abstract and then we have combined it and then we made a one single column okay so now our data set is in this particular object okay so from here we can see the column names so if you see here this particular column name is showing that what exactly we have done here we have pasted the df jcp title and df jcp abstract so we have combined the both title and abstract and then we have separated by this dot okay so what data frame does so data frame takes it as the column name and assigned to that particular data okay but this particular column name is too big so what we will do we will make it as a short column name we will change the column name of this particular data frame and how we will change the column name of that data frame so we will just simply use the col names function and this is this is my data frame and here I am just giving the text okay so if I do like this and now if I see df jcp text is there okay so now my column names are not too big so it is extra so this is just a optional way of doing but if you don't want to change the column name and if you want to go with this column name it's absolutely okay it won't make any differences in analysis okay so just to do the things easier we have changed this column name okay so if I do here so this is my data set and if I see my first title and abstract of the publication so this is the thing okay so this is my title and then it's starting from here abstract okay now we will do start our the pre-processing on this text data so the very first thing what we will do we will check whether any irrelevant

information is there okay so what we will do we will see the data so if we see here the first and and just give a reading to this particular data first title and abstract then if we see again here so in these two text you will see that this in the end this is coming okay so in the first title and abstract this was the common thing it is coming in the end of the abstract and then in the second one it is coming this copyright 2015 elsewhere limited alright results okay so we don't require this thing for our analysis it doesn't make any sense if we include this kind of text in for our analysis of topic modeling on this particular text data so we will remove it and how we will remove it we will use the rejects we will make some regular expression to match this pattern and we will remove all the this publisher by elsewhere or old right result from the text data okay so before going ahead let let us first see is there any other way this particular text is written okay because if you see here in our first F track it is written like copyright 2014 published by elsewhere limited okay but in the second case it was written 2015 elsewhere limited alright results there is no that publish what is there okay so let us see if third is also like that so in third we have copyright 2015 elsewhere limited we don't have that all rights are reserved okay what we have in this second one okay and now we will see for the fourth one also so here we have okay so from this we get to know about that there is something like the same text is a written in different way so we have to take all these three patterns and then we have to make the regular expression in such a way that whatever the way this elsewhere thing is written in that particular abstract we should remove it okay so if you see here so this was the string in our one of the abstract then this was the string and then this was the string okay so all these string have common is that one is copyright so our regular expression must include this okay so that is one thing is clear to us then we have another thing is that in all the three string there is a pattern of like here is given and here is in four digit okay so 2014 2015 and 2015 okay so these are of four digit so we want this symbol and after that what we want we want four digit okay so this is pattern is there after that what we have we have two instances where else we are limited is there and in another instance it start with published by okay so what we can do is we have to make rejects in such a way that it should match this also this also this also okay so if you see here I have made already the regular expression for matching this strings so what I am saying here is that so this is C so it is matched to this one then I am saying that digit four okay we have already discussed what exactly this is this showing that digit four digit so whatever the occurrence of four digit so four digit is there then I am saying then published okay so it matched to published also then I am saying that this this small s plus what exactly it is saying it is saying that one or more white space so this particular white space is matched by this s plus smallest plus okay then I am saying by so it matched to by then again I am using this one or more white space so this is the white space then elsewhere is there elsewhere mesh and then one or more white space so this is white space and then Ltd one of my regular expression is done and then we have like for second one we can do like this like so this thing is common in all the three patterns but here if we want like here we want elsewhere limited elsewhere limited but in this one we want elsewhere limited old right reserve also okay so these are the three pattern we have identified and we have defined the three rejects to match the these pattern okay so if you see here so this is my one regular expression this is my another regular expression to match this one so this one is for this one

then this is for matching two and this is to match this three number okay now we will move to our and then we will see whether it is matching or not so we have already discussed about this particular function graph which help us in matching the pattern and so to remove these things from the text we will be using a another function which known as the G sub okay so what G sub is if we have to see G sub will use the question mark and we will see what exactly G sub is so it's a pattern matching and replacement okay like what we used to do is like control H which exactly find and replace okay the same thing is done in here okay so this is my first pattern so we have defined there so if we see here let us use this graph function let us see where exactly our particular pattern is there so if we see okay there is a one occurrence of this particular pattern if we see here this one we have these like 19 20 21 22 23 24 25 26 27 28 29 30 31 okay so we have 31 abstract data where it is matching this and then if we check for this okay so it exactly 49 okay 49 these abstract are there where we are matching this elsewhere LTD okay so this elsewhere is LTD is matching this one also okay so if we count so for this particular pattern we have 18 and for this we have 31 abstract and for this we have only one okay so that's our data set is we have only 50 abstract so out of those 50 abstract we have this one abstract then we have 18 abstract for this and we have 31 abstract for this pattern okay so we will remove these things from our text data will make the pattern so there are like two way of removing this particular string from that text data okay how so one is ways that we will remove first from the first where the published by elsewhere limited is there then we'll go ahead where the 18 those patterns were matched and then we'll go to the 31 okay but this is like a two like too many steps process will be there but we can call all these pattern in a single way and then we can like remove all the occurrence of any of the patterns from the text data how we will do that here we have another operator known as the pipe so what pipe does pipe just like we have Boolean or operator so it is like we are giving all the three rejects and wherever you find any of the rejects you just matches and replace that particular thing with whatever the things we want to replace it okay so we are defining this is our pet pattern is there so this is our first pattern then we had put the pipe and then again we have another pattern and then we have put the pipe here and then we have here this again the pattern is there okay so if you have noticed that in this particular thing I have haven't used the question mark before pipe but here I have used the question mark twice once is here and once is at this part okay so if you seen that there may be the possibility that after LTD there is a no chance of this like some may have the period sign and some may not have the period symbol okay so this question mark exactly does that okay if there is no occurrence you consider this but if there is a occurrence of this period you consider that and remove that particular string from the data okay so this is my pattern is there and now what I will do I will consider the G sub function so as soon as you put the cursor here it will show what all things you have to give so you have to give the pattern then you have to give the replacement okay what exactly you wanted and X is the data where exactly what the data you are considering okay so for my this particular pre-processing I have the pattern my pet so I am giving the my pet as a pattern then replacement I am just giving an blank okay so I if I want to put it like a coma or anything or plus or anything I can put it but for the time being I am just removing this and replacing with the blank and then this is my data okay so if I do like this and now if I see this

so that that thing is removed that publish by elsewhere if I see okay so in this first abstract this copyright 2014 published by elsewhere limited was there but if we if you see in this removed abstract we have removed it okay so we have removed the first string from our text data which is like unnecessary for the kind of analysis we are doing similar way if we can check for the second one also like if we check second so there is no that else we are in copyright symbol is there I can check for three third also okay and the similar way we can check for another abstract also that whether that elsewhere is there or not okay so our first pre-processing of the text is done so many times we have some other strings like a URL or email ID in the abstract okay so we can remove that also so let us check do we really have these URL or email so in our abstract okay so how we will do so this is the pattern so what exactly this pattern is that any occurrence of HTTP because the URL can start with HTTP or HTTPS or FTP or maybe the WW dot so I am matching if there is any such occurrences is there in my data set integer 0 so what exactly integer 0 means is that there is no such occurrence of any of these URLs are there in that particular abstract okay so we don't have to call this gsub function to remove this URL because we don't have the URL in our abstract text data okay then we can similar in similar way we can check for email also like for the time being I am taking an email as at the rate but wherever at the rate symbol is there between the two phrases of words it will be an email ID so we'll check whether we have this kind of thing in our data set so again it is showing that there is a integer 0 that means there is no such pattern of email in the abstract okay so we don't need to run this if we have the occurrence of any of the email and URL we have to run either this or maybe this one okay for URL but though we don't have these things so we are not running these URL removal and email remover now the second things in the pre-processing is the lower casing of the character okay so lower casing is that there may be a possibility that the same word which convey the same meaning may be written in different form of words okay say for example you take this Mohit this Mohit and this Mohit and this Mohit okay so these all four words are conveying the same meaning that this there is a person whose name is Mohit okay so there is a thing which is named as Mohit okay but these are written in different way so here there is a one capital letter is there then all lowercase is there then all uppercase is there and then we have a mixed lowercase and uppercase okay so we need to standardize these words to a single kind of a form so what we will do we will put a lower casing so what exactly lower casing after lower casing it will be like Mohit it will it is already Mohit so it will remain Mohit in this way it Mohit and okay so when you count the number of words in like this way we have already seen the table function we'll see in our how exactly but whenever we will be doing the tabling of this the output will be like Mohit there is a occurrence of one then Mohit there is occurrence of one then we Mohit occurrence of one and then Mohit occurrence of one okay but when we run this table function on this one it will give me the output Mohit four okay so that is why the lower casing of the text data is a necessary step in the pre-processing of the text data okay we have this as a object where we have these different way of Mohit so if I run it and now if I table it just to check the occurrence of each word so you see that Mohit there is one though these all six Mohit are the same thing but it has been counted at one one one to each one okay so now what could be the possible function name to transform the lower casing of the words okay so just you post

the video here and just think what could be the possible function name okay so the function name to change the lower casing of the these words are too lower okay so this is a function which transform these all kind of words to a lower cases is the function we can check it by using the question mark sign before this and what exactly does so it change the cases okay so there is a another function called to upper we change the uppercase so we will see both of them also if you write any function name like if you write to lower it will automatically show you that to lower is your function which is a function inside the base package okay if you have like different function it will you have different package okay say for example if we use mean okay so it is automatically showing you that mean is a function in base package okay if we see here like there is another function called mean CL bot but this is a part of ggplot2 package okay so if you don't have ggplot2 package in your system you won't able to see this function okay so that is again you we need to know that what are the function we are using what exactly the package where it is okay so there is a so this is another thing we need to remember is that whenever we are using any of the function okay so from where this function is coming okay what exactly that package is okay for base package we don't have to call each time library is that thing okay but we if we have another package like ggplot2 another thing is a bibliometrics so we have to call those libraries using the function library and then we can able to use those functions okay so if I do like this and now if I do table it's showing there is a moit which is occurred by six time okay in the similar way we can do the two upper also so to upper and now if I do like this okay so there is a word called moit which is in uppercase which is occurred six times okay so this is the way you can do this casing of the text so here we are changing the text to a lower casing so how we will do this is our data set so whenever you are doing analysis step by step you must remember that what object you have created okay so now my process data from the first step one where we have removed this else we are is this rampart okay it's not that df jcp ti ab okay dot dollar text because we have made the first pre-processing and now our data is this so if you see here after pre-processing of step one we have this data okay we don't have to use this okay because this is the pre-processed data which has the elsewhere and all those things okay but this is the object where we have the data which is after removing these elsewhere or alright results okay so we'll call our object here we have data is from step one where we have pre-processed now I will assign this lower casing of this particular data set to a new object which I named is at df lower okay so if you see this particular data so this was my earlier data after step one where we have removed elsewhere so if you see here we have different words like e1 is there then moreover is there then if you see here like from is there this from is there and then we have from here also okay so in one all the letters are in the lower case and where is in another from we have the first letter as a uppercase okay so if we see output of this now if you see this abstract you will see and there is no uppercase is there all this from is there and then if you see below this from is there okay so both are in the lower case okay so this is how we can change the casing of our text data so until now we have done the two step of pre-processing so one is that we have removed the unwanted text like published by elsewhere or alright result and in another step we have done the lower casing of the text data okay now we will move to the third part of the pre-processing is the punctuation removal okay so what punctuation are so punctuation like period full stop

comma colon semicolon hyphen etc are the connecting symbol and that generally used in dividing the sentences and the phrases and punctuation are only used for structuring the text for reading purpose and does not carry the semantic meaning so we will remove all the punctuation so let us take the example of hair that this there is a punctuation marks has been assigned to each of the word so we have Mohit comma then we have Mohit this then we have Mohit semicolon so we do like this so now our object is this to remove the punctuation we need to have a package but that package is so there are different kind of packages there but whenever we are doing this text mining we have a separate package called as TM so TM has a function called remove punctuation if you don't have installed the package TM in your machine it will be displayed here that package TM and other required but not installed but if it is installed so we have to call that library so we'll call it so our TM package is called now there is like function is there which is used for removing the punctuation okay so what exactly that function name is so what we have to do we have to remove the punctuation so the function name is also remove punctuation okay so if I go here as soon as I write here remove it will also so remove punctuation and this function remove punctuation is a in the TM package okay so this you have to remember that okay this is the function which is exactly the part of TM package because why it is important because whenever you are calling it any function and you haven't installed or you have not called that library okay so in that way it is very important to know and if I have to see how to use this remove punctuation will use this question mark for seeking the help so if I do like this and we'll see here the arguments the types of argument we have to give and the example how we have to call it so our data is this a already created this object so if I simply run this okay so all punctuation are removed so the same thing we will be doing on the data set what we have received after doing the pre-processing at the step 2 so what exactly the object name but we have got it from after step 2 is DF lower now it's not rampant okay rampant was a data set but we have got from after pre-processing at the step 1 okay now I am calling this and assigning to a another object DF 1 1 okay now my punctuations are removed and now if I see okay my punctuation have been removed so if you see here all doors and other things have been removed but here one thing is happened is that when we have like sometimes we have these dashes between the words okay so it has also removed the dashes particular between the words okay so if you see here this low energy in the earlier our data set this was a dash was there okay so if we see here our data set maybe rem DF lower so if you see here we have dash here life cycle then we have dash here also then we have this is energy efficient energy efficient is there then we have this in low energy okay if you see here this low energy it's a one word is there now but we want like after removing the punctuation in such a case there it will be two words low and energy so we don't want these two words as a like a single word but we want as a two separate words okay so how we will do we will use the our earlier function g sub okay so in that case our g sub is there and this is our punctuation rejects so now what I am doing I am saying that in g sub wherever you see the punctuation okay you just replace it by a extra space okay so I am here putting the extra space okay and this is my data set so if I do like this and now if I see the output of my data set so you see here that so it's low energy now the dashes is removed but extra space has been added between the two words so that is why for this removing the

punctuation it's a like better way to go in this way sometimes we have different words written separated by dashes and also in that way we can do a effective analysis okay you can use the remove punctuation and there is like another kind of argument is there but you can specify is that that preserve intra word dashes okay so it has like if you put it true and if you do like this so it will remove the punctuation wherever punctuation marks are there but it will preserve these dashes so life cycle energy efficient so this is there the dash is there then we have low energy this is there this dashes are preserved okay so what exactly this it means that you do the remove punctuation operations on this data set but preserve the dashes between the words okay so by default it is always false okay but you can call it like in this way to have those dashes between the words okay or you can use the earlier gsub function to remove the dashes and replace it by a space or any other symbol you want it there okay till now we have done the three step in our pre-processing so the first step what we have done we have removed those LCR strings in the second way we have changed the lower casing of the our text data and the third step what we have done we have removed the punctuation now we will move to a next step in our pre-processing which is a stop word so what exactly stop words are stop words are the generally the most occurring word in the our text data set okay like a da is all those words are generally called as a stop words so which occur frequently in the text data set but have only little semantic meaning okay so to do an effective analysis we remove all such stop words so that we can focus on the most informative words in our text data set okay so we will see how to remove the stop words for our data set in R so let's do that so like as I discussed about there is a function for remove punctuation okay then the similar way there is something called stop word so we need to see what exactly is this stop word is so we'll put the question mark here and we'll see so like in TM package we have stop words so it's like what exactly it is showing that stop word kind of English language okay so we can call it as in this way top words kind if I run it so these are the list of stop words okay what those stop words are like I me my myself we our ours yours and these many stop words are there okay but we won't be using this function what we will be doing is we will be using the function called remove words and inside the removed words we'll call it as stop words and what all the stop words we will be using will be using the stop words given by the smart information retrieval system okay so this smart information retrieval system is by the Cornell University and we will be using those set of words to remove the stop words from our text data set okay our data set after third step was bf1 it is neither rampant or DF lower so our data set is bf1 and this is my function remove words so I am giving remove words this is my data set and stop words smart okay so you have to see what all those smart stop words are I can simply check it so if you see there these are the number of stop words are there okay this towards smart has much higher list of stop words compared to the just simply if we are running this particular thing okay if we are running this particular thing okay so that is why we are using the smart stop word list so if I run this okay now if we just see the data this was our data after lowering the case this one is after removing the punctuation now if you see here that this in is and all now if I see here bf stop you see that all those stop words are removed okay that is why this extra spaces you can see here okay so you can define your old stop words related to our your study and then you can remove from that X data set okay so like these authors have

identified these many stop words for doing the topic modeling okay so you can define your own stop words and then you can remove it okay so how you have to do it you have to make a list of stop words and instead of stop word smart you have to give the this that object which is storing your all those defined stop words okay in similar way you can remove these stop words related to your study from the text data set and I request you that please define some of the stop words related to your study and try to remove some of the those stop words from the text data set whatever you have and do let us know in the discussion forum that you have identified these many words as a stop words in your particular text data example okay so if you see here after removing the stop words there is some extra space is there okay so we can remove this extra space so to remove the extra space from the text data we have this function called strip white space so it is removing the white space wherever it is available in the text data set and if we have to check what exactly that package is where this trip white step is there so strip white space is a function in that TM package okay it's showing here okay and we can check the help also for this so we see here it is showing that strip white space from a text document and how we have to call we have to just call this function and we have to give the X means the data set we have what we have okay so here we have the DF stop so I am just calling DF stop and assigning to again to DF stop okay so what I am doing here is I am just cleaning up all the white space from my earlier data set but I have got it from the removing the stop words okay so if I do like this and now if I see my first step that you see here like extra white space has been removed so if you see here extra white spaces were there but now the extra white space have been removed okay the another step of pre-processing is that removal of number so but like it depends on kind of analysis you are doing so sometimes like you want to remove the numbers and sometimes we may not consider to remove the number so it totally depends so if you want to remove the numbers so how we will do in the same way the way we have removed the words so we have this function called remove numbers so if I call it like this so my numbers will be removed okay whatever number like 1 2 3 4 any numbers are there okay so it can be removed okay let us now call it as DF rem num and okay so if I do like this numbers will be removed okay so now after this we have a step called doing either lemmatization or stemming so what stemming and lemmatization are so stemming and lemmatization both are the two important technique to transform words to their base or the root form so in stemming there is a generally the suffix are removed but in lemmatization the words are mapped to the dictionary word okay then stemming sometimes generate the known valid words while lemmatization generate the dictionary correct word okay so let us understand this with an example so if you are doing the stemming for running okay so the output will be run okay and if you are doing the stem if you are doing running so if you are doing lemmatization on running it will be run okay but in that case whenever you like do stemming on houses so if you are doing stemming it will take all this part okay so now the word will be this house okay but whenever you will be doing lemmatization on this it will be mapped to house okay so sometimes the stemming give you the valid word but sometime it may not give you do the valid word so this is the difference between the stemming and lemmatization and it again totally depends on the study and the kind of analysis you are doing whether you want to do stemming and or

maybe this lemmatization okay so but the lemmatization is like considered as the preferred practice because it map to the dictionary correct word okay so there are some lemmas are there and it map the word based on those lemmas okay but lemmatization is a kind of like computational expensive effort like when you have a big data set it takes a little more computation compared to stemming okay so let us understand this with an example in our okay so take this example where I am I have a string called I cooking Brianna which cook cook but I ran away from home when running run better okay so this is my text okay now for this again we need a package and for that we are using the package called text them so if you haven't installed first you install the package and how you will install the package you go into this interface tools install package or calling the command okay so either way you can install the package so after installing we'll call this package for lemmatization we have this function called lemmatize strings okay and we have we can see how exactly we can use lemmatize strings using the question mark so here we have to like this is my data is there and then dictionary I have to mention what dictionary that has lemma is there so just I'll call it this so if you see my string was I cooking Brianna so I cooking Brianna which cook cook but I ran away from home when running run better okay so cooking becomes cook then cook cook cook cook then also this ran become run then again running run is become run run and better becomes good okay but as soon as we will do the stemming of this particular data let us see what it becomes okay so this is okay I is okay cookies okay which is okay this cookies also this is also okay this is also okay but you see here we have ran away so it has ran the same word okay so after stemming the word becomes same ran only and away become the Y is removed and I has been inserted okay and the similar way if you see here better remains better only so that is why you know generally it remove the suffixes okay within the word but lemmatization just map the words to the dictionary words okay so here I am considering that I am doing the lemmatization so this is my data set so this was my data set after the stop words removal but I have removed number also so this is my data set now I will do like this so if I see the output after running the lemmatization strings you see okay so if I see my earlier data set you identified promising these were the words were there so these words were mapped to identify promise okay so this is what lemmatize does with the text data okay so that is why we have done the lemmatization of our text data but if you want to do the stemming you can do the stemming also so let us see if like this was my data and if I do like this DF stem now if I see okay so the large become L A R G then if you see companies ES is have been removed then in identified ED is removed promise E is removed so we are not taking up with the stemming in this analysis we are considering the lemmatization okay but if you want to check it with the stemming you can do with stemming also okay so this is all about the pre-processing of our text data so what we have done at very first step we have removed the elsewhere in the end of the abstract we have the elsewhere and we have removed it in the second step we have done the lower casing of the our data set because like some of the words may be written in different way then we have removed the punctuation and after removing the punctuation we have removed the stop words so that some of the common words can be removed and after removing the stop words we have also discussed that we can make our own stop word list also and calling the remove word function we can remove those stop words also then we have removed the

numbers we have removed numbers like different numbers can be there in the text data set so we have removed it and in the last step we have discussed about that either we can do lemmatization or stemming and we can considered the lemmatization how lemmatization mapped to the dictionary word and stemming just remove the suffixes okay whatever the text processing we have done we have done step by step okay so we have first removed one thing then we have removed one thing and then we have moved to another part okay but like whenever we have to do load many text analysis on a regular basis so what we can do is here so we can make one kind of function we can combine all these operations in a single function and then we can create our own function and those function can be called whenever we are doing the analysis at some other time okay so how we will create the function so to create a function I will just request you that you post the video here and just based on this much like what are the step we have followed you create your own functions and then see whether you have created the right function okay so to create the function will call this our function name so this is our function name so I am giving it like week 9 clean LM because I am using lemmatization and I am using this function X so why I am using X because it's a single argument okay so my data is in only a single argument I am passing it to single argument we have seen the different kind of function so either the function where no argument is there we have created a function called Mukesh where no argument was required then we have seen the function where two argument is required like add them okay but this kind of like analysis we need to create one argument function where we will supply the text data and then it will pre-process the data okay so if you see here in this one son we haven't used these two operations because we don't have any of the URLs or emails in our data set but if your data set have URL and email so then you have to use these two operations also also if you remember that when I when I was talking about function if we just simply call this operation one by one the output will be whatever in the last okay so to get the output after each operation we have to assign it so here if you do like this okay and if you do do not use this X here and this one so the output will be whatever the last operation is okay so in the better way what we have to do is like we have done this okay and we are assigning to that object okay so whatever like my text data is there so first thing it will be removed the URL then the email it will be removed then that string elsewhere so after removing the this string we will be doing the lower casing of our data set after lower casing we will remove the punctuation so we will not use the remove punctuation because we want to have those two words low energy as two separate word rather a combined word so we are using in this way then we are using the stop word smart list for removing the stop words so we'll call it like this way then we'll remove the white space because stop words removal brings some white space in the text data set so we'll strip that white space and after that we'll remove the numbers and in the finally we'll do the limit is enough strings okay so whatever the text string we have we will limit it okay and finally after doing all those things will return it X okay so whatever my data final data is there it will be returned okay so I'll just create this function so my function is created now if you want to use like instead of limitization you want to have timing so here the rest things are same you can instead of limit eyes you can use the timing okay so I am using this one now we don't have to go step by step all those things my function is created resting so this is my data set and

this is my function so what I will call it here is that that this all those pre-processing of our text data should be done on this particular thing okay so if I run this now I will see here okay so it start from driver 4 and barrier to low energy buildings in Sweden so this was our title but after cleaning it we have this one okay so now our title becomes driver barrier low energy building Sweden okay because other things have been removed so this is our clean data set okay whatever the analysis we will be doing we will be doing on this clean data set okay so what we have done until now is if you see this was our title then we have this whole abstract is there and this if you see this was the line what we have removed very first thing then we have like this lower casing like this from was there and then and another from where it is so another from is this okay so we map these two lower casing and after that we removed the punctuation all those punctuation we have removed and while removing the punctuation what we have done we have just simply maintained these dashes okay so wherever the dashes are there we have put the white space so we have put the white space here then here also then here also okay then after that we have like removed the numbers in this particular thing we don't have numbers and we have also removed the stop words so these stop words like these all those stop words have been removed then and the finally we have done the lemmatization so this was the identified promising companies these have been mapped to the root dictionary word okay so this was our row data set now our after doing the pre-processing we have this data set okay so our whole data set is in the lower case so lower casing is done then we have removed the punctuation also while removing the punctuation we have maintained that low energy or this this one energy efficient the dashes here earlier dashes were there but now those dashes have been removed and we have this white space after this we have removed like stop words so we have removed the stop words so the common words have been removed and then removing the stop words we have removed the extra white space there is no extra white space is there between the words and we have also removed the numbers this abstract doesn't have numbers so we have removed the numbers and also the finally we have done the lemmatization okay where lemmatization so this was company identify promise okay earlier this was identified promising so this is all about the pre-processing of the text data I will be sharing this all code in on our github repository where you can use this and then you can do this pre-processing on a text data set okay we will also provide you with some sample data set where you can just give a practice and then you can have you these text pre-processing on your own data set okay so in the next lecture I will be talking about the topic modeling how we have to do the topic modeling on this pre-process data set and what are like how to interpret those topics and all okay so see you in the next lecture thank you