LECTURE 33 : Topic Modeling

Hello learners, in the previous lecture I have discussed about the pre-processing of the text data. So we have pre-processed the text data using following multiple steps. So what we have done, we have removed the strings and then we have done like lowering the text, then we have removed the punctuation, removed the stop words and in the finally we have done the lemmatization. And in the last we have created one single function for pre-processing of our text data whenever we have to do the analysis we can call that function. So what was that function is? So this function was we have multiple operations and this particular example we have taken up because we are doing the lemmatization and if you are doing the stemming instead of this lemmatization you can do the stemming and then initially we have like removed the string called published by elsewhere limited, elsewhere limited, all right reserved and elsewhere limited. Then we have transformed the case of the text to lower case.

After that we have removed the stop words and then we have like removed the punctuation with this using gsub function and we have used gsub because of the reason that when we are using the remove punctuation function it was like removing the dash and there was no space was there, okay. So there was example called low efficient, low energy. So in on those cases that dashes were removed but there was no space, okay that become a single word. So that is why we have used this particular function.

Then we have removed the white space whatever the white space was there. After that we have done the lemmatization of strings, okay. And finally after all these operations we have written the final clean data, okay. And also this is the one of the thing what I have discussed about function is that that if you exclude all these things and just simply you call this, this, this, so it will give me the output of this only, okay. So if we want to run all the these operations, so we have to like whenever this operation is called it should be assigned to that data, okay.

So now we will do the topic modeling on this particular preprocessed text data. So what topic modeling is? Topic modeling is a kind of text mining that identify patterns in the text data. It is a statistical technique that process the text data and identify the topics in the collection of large volumes of documents. It works as unsupervised way. It helps in organization and retrieval of documents.

So it has various application in different domains like in libraries. It helps in analyzing books and journal content to organize them. Then for literature review we can use topic modeling for processing multiple scientific articles. Then for social media analysis where we can use topic modeling in identifying topics from the content posted on these platforms. So in our context we have title and abstract which we haven't analyzed.

Other than that whatever the data we have got from those bibliographic data sources we have analyzed. So these are the two things which are yet to analyze and by analyzing these title and abstract we can extract that what exactly that topic has been discussed in those content, okay. So to do the topic modeling we have different models but among these LDA is one of the popular models to use the topic modeling over the text data. So I will be discussing the LDA in this particular lecture. So what LDA is? LDA stands for Latent Richlet Location.

So what latent is? The latent means is that something exists but that is unknown to us, okay. So this latent means that is something exists but unknown to us. So in our context what is unknown? The topics. So the topics are unknown to us. Then the Richlet.

What Richlet means is? So Richlet means the distribution and the distribution of what? Distribution of words and the topic. So it is a distribution. So what allocation is? Allocation is based on distribution of topics and words allocating the topics to documents and words to topics, okay. So allocating of topics, okay. So it was given by David Blue, Andrew Engie and Michael Jordan in 2003.

So in simpler word we aim to identify the topics and words of each topics that are present in the text data and after identification we label the each document with the topics, okay. So there are two key fundamental principle of LDA is. So one is that each text document is made of mixture of topics. So you can take like we have this document 1, document 2, document 3, document 4, okay. So it has maybe topic 1, topic 2, topic 3, topic 2, topic 4, topic 6, topic 1, topic 2, topic 3.

Okay. There is some parameter is there based on that what we say that this particular document is related to this topic, okay. So maybe we can assign that parameter to this topic 1 here to 0.49, 0.

21, 0.30, okay. Based on this parameter what we can say that the document 1 is mostly related to topic 1, okay. Because it has that parameter value is 0.49. But that parameter is we will discuss, well we will be discussing more about the topic modeling, okay. Then the second principle is that every topics are made of group of words, okay.

So all these topics are made up of some group of words, okay. And those words like words 1, words 2, words 3, okay. Then this has some words, okay. And similar the way we have some parameter value for this topic 1 and document 1, in the same way we can have another parameter which define the like value of this words in a topic, okay. So let us now briefly see the LDA model.

So in the LDA model what we have is that we have this number of documents. Then n is the number of words in a given documents. Then alpha is the Drichlet parameter that is per document topic distribution. And theta is topic distribution for document. And this W is observed word.

And Z is this word topic assignment. After applying the LDA model we have the topics which are made of words and then we have this documents which are also made up of topics. So we can show that these are the document 1 where there this is topic 1, topic 2. So basically we get the distribution of topics per document, okay. So this whole structure is mathematical and quite complex.

So what we will do, we will understand this whole process of LDA with an example on our sample data set. So let us move to R studio where we have already preprocessed the text data. So after preprocessing, so this was our data set was there. We have already preprocessed it. First we will make the corpus.

You may face this kind of error. So what exactly it is? So basically we haven't called those libraries related to this, okay. So we will be using TEM topic models library. So now if I run this, so here what I am doing is I am just calling that this is my data set and make it a corpus, okay. So after creating a corpus we will create a document term matrix.

So what exactly this document term matrix is? So document term matrix is a kind of a matrix that shows the relationships of words and document. It shows the occurrence of a word in a document. So let us understand this document term matrix with a small example. So let us assume that we have these two documents, okay, where we have the text data that this is document one and then this is a document two, okay. Now before doing any analysis we will do the cleaning because like here if you see this good is in lowercase and this good is in uppercase.

So we will clean these two documents. So we will be using this particular like function we have already created. So I will just simply run this. So my documents are cleaned up. So this is first document and this is second.

Now we will create the corpus of these two documents. So we will make the corpus. So if we see the document term matrix of this corpus, so it created a matrix, okay. And if we have to see in the matrix form, let us say it is like, so this is a document term matrix is okay. If you see here on the left hand side here, this is exactly the documents, okay.

So we have document one and document two, then we have the terms here. So it is boy, cook, good, driver and then we have the occurrence of each of the word in that particular document, okay. So we have this boy. So it has one occurrence. So if we see our document, so this is our document.

So the term boy has been appeared in D1, document one and boy has also appeared in document two. So that is why we have written one one. Then if we see the cook, so cook has appeared in document one, but it is not appeared in document two. So that is why it is zero is written. Then good, good is appeared in both the document.

So this is also good and this is also good. So it appeared both here and then we have driver. So driver is appeared only on document two, but not in document one. So that is why zero is there. So this is what exactly document term matrix is.

So it shows that what exactly the term is appearing in that particular document. So if you see this particular output, it describe many things. So what exactly describe that this particular document term matrix has two documents and four terms. So what those four terms are one is boy, cook, good and driver. So if we see our clean document, this was first clean document and if we see our second.

So total terms are this is common. So one, two, three and four. Then it is saying that known sparse entry six slash two. So what six is representing here, six means that there are six known zero entries are in the matrix. And what two is showing two means that there are two zero entries in that particular matrix is. So if we see our matrix, we can easily say that there are six known zero, one, two, three, four, five, six.

So there are six known zero entry and there are two zero entries are there. Then sparsity, sparsity is 25%. So like total, it should be like eight should be there, but out of eight, two are zero. So it will be two by eight into 100.

So it will be 25%. So we have 25% sparsity, then we have maximal term length. So the longest word, which have the maximum number of letters, what is the length of that word, that length is six. It means that there is one word is there, which is the longest in terms of number of letters. So what that word is, where we have six letters, three, four, four, six.

So that word is driver. And after that it is showing that weighting is based on term frequency. So what this weighting is, so weight shows that how terms are weighted in the matrix. Weighting by term frequency represent the frequency of each term in the document. So here one represent that there is only one occurrence of that particular word in that document.

If there is two occurrence, so it will be two. Let us see like we count the frequency. So we'll count the frequency on this. And so what we have done here is we have selected the occurrence of all those four words in all the documents. So what we got it here is that the total occurrence of boy is two, the total occurrence of good is two and the cook and driver occurred once in each document.

So this is about the document term matrix. So now we will back to our sample of 50 publication of the JCP, what we have already processed and we have already created corpus. So this was our corpus. So we'll create a document term matrix. So to create a document term matrix, we have this function called document term matrix. So as practice what we used to do before using any function, we used to see that what exactly that function is.

So and how we will see, we'll use the question mark. Document term matrix. And if we search here, it will give us the old details. What exactly document term matrix this function do. And also we have the term document matrix also we can do that also.

So we create this. So now if I see this, what exactly this document term matrix output is showing that we have 50 documents. So that is all we have already seen that we are taking up the sample size of 50. Then we have number of terms is 1671. Then if we see here, what exactly it is showing here is that there are like 4434 non-zero entries are there in the matrix.

And rest 79,116 are zero entries in the matrix. So what exactly it is showing is it is showing that that most of the terms do not appear in most of the document. So and if you see our sparsity is 95%, then we have the maximal term length is 17. So we have a word whose length is 17 letters. So that much big letter we have in this particular document term matrix.

Then we have weighted based on that term frequency. Now we will see our top words of our this particular data set. So what we will do first, we will make it a matrix. So if I run it, so this is done.

Now if I will see the coalsum. So we have 1671 terms. So 671 words are already omitted to display, but the coalsum is created. So we will assign to this flip and if we see the, so we have total 1671 that is okay. We have already discussed. Now we will display only top 50 words, which are occurred most in our data set. So what we will do, we will just order that particular words in a decreasing order of their occurrence and then we'll select only the top 50.

So if I do like this, so these are the top 50 words in our data set. So the top most word is environmental, which is occurred 85 times. And we already know that how to export this particular data set.

So we'll use the function write.csv. We'll assign this to a value top 50 words and here if we run this, so like this, and if I run this like top 50 JCP word. So where this file will be created. So I request you all to pause the video here and just let us know in the discussion forum where this file will be located. So this file will be located in my current directory and what exactly my current directory is.

So we will check where my current directory is. So it is in documents. So this file is created in the document. So I'll go in documents. So these are the 50 words that file is created.

So now we will be doing the topic modeling. So for topic modeling, the first thing what we have to give is the number of topics. So let us assume that we are taking up the number of topics to 5. So we want to label all 50 documents in within these five topics. So to do the topic modeling in R we are using the package topic models and inside that we have the function LDA because we are using the LDA algorithm.

So again, we'll see how to do LDA in R. We'll use question mark. And it has all the details like how we have to call. So this X is that particular data set, whatever data set is then K. K is the number of topics.

So in our case, that number of topics, we have taken up to five. Okay. Then we have methods. So in methods, we have two methods. One is VEM and another is a Gibbs methods. So, but we will be using the Gibbs sampling method as it is a widely adopted one.

Then it has some control value. Okay. So we will be like calling this function. So this is our document term matrix. We have already created, then we have this topic num. So we have assigned these five number of topics. So here the K will be five, then method is Gibbs and then control is equal to list seed one, two, three, four.

Okay. So what this seed is, this seed is used for reproducible purpose. Okay. So in statistical analysis, it initialize the generation of random numbers that follow a specific sequence of numbers. Setting a seed to a value get you the same results whenever you run the code multiple times with the same seed value. Okay. So it ensure the LDA analysis consistent on multiple runs with reproducible output.

Okay. So we are setting up the seed one, two, three, four. And after running this LDA function, I am assigning to a object LDA JCP. Okay. So if I run this, so it is created.

So this LDA output, I give many things. So the first thing we have this beta. So what beta is, beta is exactly the per topic word distribution. So this is our object, which have LDA fitting on a JCP 50 publications. And now we are using the matrix beta.

Okay. And here we are using the tidy function. Okay. So if I run this, so our library is a tidy text. So we'll just call it. So this is the error. So now if I, okay. So these like a topic is there, then it has the term actor like these are the some different terms are there and it is given the beta score.

Okay. So we'll assign to object this topics dot JCP. Now, so we'll view this particular object. So it shows that there are total 8,355 entries are there and there are three total columns. Okay. So how this 8,355 entries came? So basically we have total terms 1,671 and over that we have five topics.

So it is computing the beta of all the words in all those five topics. Okay. So it is calculated like for actor for all the five topics. Okay. So if we see here, our length is 8,355.

Now, if we want to see that actor is related to which topic. Okay. So here we have the data set of actor term with five beta score in five different topics.

Okay. So if we run this, we get this score. So actor is mostly related to topic four. Okay. Because it has the highest beta score. So if like, if I sort it down here, it is displaying that this is the highest. So where this value was, this value was in four topic four.

So this particular actor term is related to this topic four. Okay. So this is how we can identify that what all words are related to this topic. Okay. So if you see here, we got the data that each topic and the beta score of each 1671 words.

Okay. But we want to know only the top terms. Okay. Where the beta score is the highest. Okay. So what we will do this kind of little data manipulation, we can achieve by using the supplier package. We will be using our, this topics. So it has the whole topic term and beta score.

So this is our data set is I'll be grouping it by topic. So we want to group it by topic. And this particular operations is from the package. Okay. So this is my, the new object where I will be taking up the JCP terms where the beta score is the highest for each of the topic. And then what exactly this one son is this one son is about where the top 10 value where beta score is the highest.

So if I do like this, so this is the error because of we haven't called the library. So if you haven't installed the package, the player, so you have to install first and then you have to call this library. So, okay. Now if I run this, okay, so we'll see the, our terms. For all the five topics, we have different words. So in the earlier example, we have discussed about that if we have 1,671 terms are there and we have the five topics.

So we found that our entries were 8,355, but in, in this case, if you see, we have 52 entries. So why it is so, because our only condition was here is that top 10, those beta score based on that, we want to have the words. So here, if you see, yeah, you see, we have these three terms, which are the equal beta score. And that is why it is like our condition. This has taken up 11 terms instead of 10.

Our initial goal was to take only top 10, but this is like our 9, 10 and 11 value. They have the same beta score. So 11 values is here. Then for topic two, we have, we have 10 only, 10 terms. For topic three, we have, yes, for topic three also, we have two terms, which are having the same, this beta score, okay.

That one is this beta score of end and another term is term, which is having the beta score of this. So these two terms have the same beta score. That is why this has also taken this extra term for this particular topic. So we have two extra term in this particular data set. So what our goal was, our goal was to take only top 10 for each five topics.

So we want to get only 50 terms, but here we get the 52 terms. So now we'll like, we have the topics and we have the beta score. So we'll plot each of the beta score corresponding to that topic. What we will do because our data is like just height.

So we'll visualize the height. So we will use the function, this bar plot. And if I run this, so here, if you see, I won't give the one is to 10, okay. Because my values for topic one are from one to 11.

Okay. But if you're like, I have the values of one to 10, then I have to use here one to 10 only. Okay. Because I have already seen the data. So I'll just run this. So what exactly, these are the height, what the beta score of each of the words, then this exactly argument is saying that what would be the name, name are the terms.

So these will be the names and this will be the height. Then I'm calling. So this is, I'm calling for having those vertical labeling. Okay. And color is red. Okay. So if I run this, so this is what our plot is for a beta score of each of the words corresponding to topic one in similar way, we can go for topic two.

So here I'm taking up the range from 12 to 21 because my topic two has only 10 terms. So changing to green, then for topic three.

So in topic three, we have 11. So this is topic three. Then we have topic four. And finally we have topic five. Okay. So now you can export these each of the plot and then you can analyze it. That is a very straightforward way that you export each of the plot and then you analyze it by seeing all the five plots one by one. So can we have a such kind of a plot where we can have all these five plots in a single plot? So how we will do that? We will use the function part and where we will define that, that all the five plots should be in one row and it should be a five columns.

Okay. So one row, this is first plot. This is second, this is third, this is four, this is five. So if I call this, this is okay. Now if I call all these, oh error. So what this exactly error is, this error is figure margins too large. So it basically because whenever you are plotting it, it is visualizing on this part.

So you have to like if you do like this margin was too. So now if I, if you do call it, so these are my, all the plots in one single frame. Okay. Now I can zoom it. Okay. So now you can easily see that what all words are related to like this is topic one, this is topic two, topic three, topic four, topic five.

We can also label this here by this is topic one, topic two, topic three, topic four, topic five. Okay. So this is your practice that how we will be having a here topic one, topic two, topic three, topic four, topic five. If you face any of the issue, like you see first question mark, how we can do that and still you can't able to solve. So please let us know in the discussion forum, we will explain maybe in the live session or through discussion forum that how you can have here these title.

 Okay. You can export this by either save as image or PDF. So whenever you are saving it, you just remember the, give the aspect ratio so that the full plot come on in your frame. So now by setting this to one by one, what we are saying that we won't only one row and one column of that plot. Okay. So we have set to our earlier format.

 So this is like, if you have a big data set, okay, maybe 50,000 documents or maybe 60,000 documents, how like giving range. And if you have like 20 topics that could be too complex thing. Okay. It is okay for a small number of topics. So for that, we have this package called ggplot2, which makes our task easier and like using combination of packages like a diplier and ggplot2, we can make a single plot of these all five topics.

 I'm leaving up to you to give it right. Do let us know your experience, whether you will, you are able to do it from using this kind of a code, or you want to go with this step-by-step for a large number of topics. Okay. After this, like we have the topics and terms. Now we want to classify the documents that what exactly the topic is associated with this particular document.

 So for that we have this is score called gamma. So gamma is a posterior topic distribution for each documents. Okay. So like above we have done for matrix beta.

 Now we'll call it as a matrix gamma. And if I do like this. So here we have 250 rows are there. Okay. So what exactly these 250 rows are there. So we have 50 documents and we have the five topics.

 So 50 multiplied by five 250. So these 250 rows are there have gamma score for each of the topic and for each of the document. Okay. So like for document one, we have the gamma score for all the five topics. Okay. So for document one, we have gamma score for topic one, topic two, topic three, topic four, topic five.

 Okay. So we'll call it and we'll assign to this object JCP gamma. And now if I see. So if we see here, we have this document one and this topic one and corresponding gamma score document to topic one gamma score.

 So like first it is displaying like where this topic one can come in all those 50 documents. Okay. So it has total 250 entries. Okay. So now we want to know the gamma score for our this document one. So how we will do that? We will just extract only the gamma score for document one. So for that we will be using this function called subset. So what subset do we can check with the question mark.

So what subset do here is that subset basically extract a small portion of the data based on particular condition. Okay. So here our condition is that that we have the 250 entries inside those 250 entries. We want only that where document one is there.

Okay. So if we see here document one. So for document one, we have these five topics and their corresponding gamma score.

And if we see that topic one gamma score is 0.09, 0.21, then 0.09, then for topic four 0.49 and then five is 0.10. Okay. So now we'll take it like wherever the gamma score is highest. Okay. We will be taking up that particular topic to that document.

Okay. So here if we take that kind of a condition, so we'll assign this particular kind of topic four to document one because it has like 0.49 and other are quite less gamma score. Okay. Now we will set the criteria.

So what our criteria is that so in all the five topics, wherever you will find the maximum gamma score assign that topic to that document. Okay. So what exactly we have done for this example here, we have assigned the topic four to document one based on the high gamma score of this topic four. So again, we'll use the same diplier package and this operation that this is a our object where we have the gamma score, where we have document topic and gamma.

Then I'm saying that for this, you group it by document, then you slice it by maximum. Okay. So you this whole data set is there. You group by document and slide by maximum where the maximum gamma score is there. Okay. And then in the finally you ungroup this particular part where the maximum gamma score is there you ungroup it and store it as abstract JCP class.

Okay. Here we have the object abstract JCP class, which takes the value of highest gamma score of each of the topic for all the 50 documents. Okay. So if I run this now, if I see the abstract JCP class. So if you see here, we have the 50 documents and our topic is five, but we have assigned only single topic. So based on this condition, we should have only 50 entries, but again here we got the 51 entries. So what could be the possible reason? The possible reason is that we may have the case where we have the gamma score for two topics is same.

And that is why that both the topics had been assigned to this particular document. So let us see where we are getting that to equal gamma score. Okay. So there is a document three where we have the this gamma score for topic two is 0.26 and the same way 0.26 equal gamma score.

Okay. So here we can take our decision and we can make an extra condition that if equality is there, then we'll go ahead with this kind of a topic. Okay. So that's based on case to case. So, but see this kind of thing can happen. Okay. If you come with the idea that only you want the maximum value.

So if we see the gamma score for document three for all the five topics, if we see, so we see that there is like topic two has gamma score of 0.26. Then topic four is 0.26 and we have topic three also, which is close to these two topics.

Okay. So this is, you need to like analyze like what topic you have to assign to this particular document. Okay. Then we can save it all these objects as a external file. So how to save it? So we'll call this like this is our object and we'll call the file name.

Okay. So whatever the file name we will define it will be saved there. Okay. So I'll just save this. Now our all 50 documents are mapped with those five topics. So now we want to see that what topics has been mostly published in JCP in this 50 documents. Okay. How we will do that? So we will be using the table function. So using the table function, we can see that, okay, what topic has occurred more.

So if we use this table function, so this is our map data and under which I want to see the occurrence of topic. Okay. So if I do like this, what exactly it is showing? It is showing that topic one is occurred 10 times, topic two, nine times, topic three, six times, topic four, 12 times, topic five, 14 times. From here we can conclude that like on topic five, we have the 14 documents which have been published in JCP.

Okay. And the least number of documents are published in on topic three, which are just six. Okay. So this is how we can just do the topic modeling and then we can assess that, okay, what exactly the key topics have been published on those documents. Okay. So this is all about the topic modeling analysis of our 50 documents. So what our 50 documents are basically those 50 documents are title and abstract of 50 publications of journal of cleaner production, but we can do this analysis on our big data set also.

So do try using this code for your analysis on another data set. So in this week we have discussed about the basic of text mining is what text mining is. And after that we have discussed about the regular expression and how regular expression can help us in doing the pre-processing of the text data, which is very important step before doing any analysis on the text data. After that, we have discussed about the topic modeling using the LD algorithm. We have used different packages for doing the LD analysis. So this is all about this week. See you in the next week. Thank you.