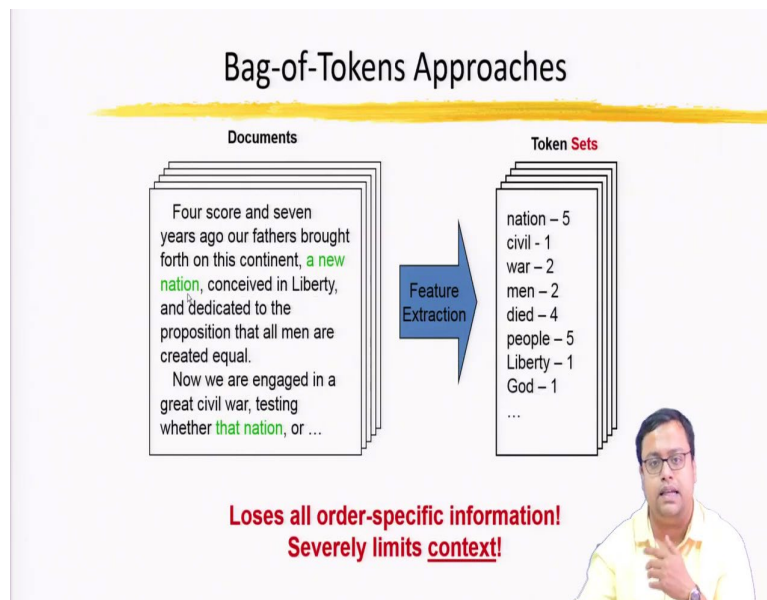


Marketing Analytics
Professor Swagato Chatterjee
Vinod Gupta School of Management
Indian Institute of Technology, Kharagpur
Lecture 52
Text Mining & Sentiment Analytics (Contd.)

Hello everybody, welcome to Marketing Analytics course. This is Dr. Swagato Chatterjee from VGSOM IIT, Kharagpur. We are in week 10 and session 2 and we were discussing text mining natural language processing and basically, tokenization. So, as I told in the last video the last slide for the last video that, tokenization severely limits the context so, that is true, what is tokenization? That means that I first from the text data.

(Refer Slide Time: 00:47)



If you see here, from the text data, we convert the text to token sets. But, before doing that I actually remove various kinds of stop words the punctuations and then after removing all of these and probably sometime stemming also and after removing all of these things, we convert each unique word to a particular a token and now, what we do? We are, you can find out which token is most frequent, which token is least frequent and so on.

(Refer Slide Time: 01:21)


Term frequency

Term frequency tf_{ij} is a measure of the importance of term i in document j

Inverse document frequency (which we see next) is a measure of the *general* importance of the term.

I.e. High term frequency for “apple” means that apple is an important word in a specific document.

But high document frequency (low inverse document frequency) for “apple”, given a particular set of documents, means that apple is not all that important overall, since it is in all of the documents.



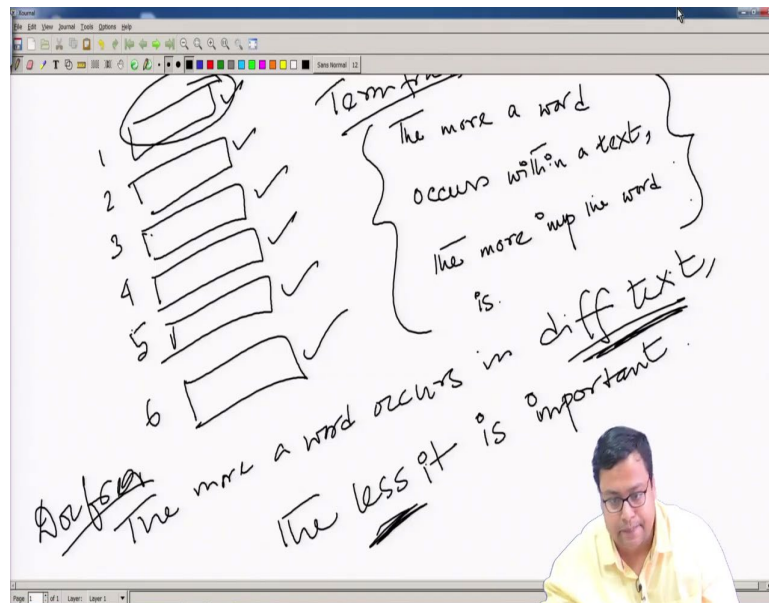
Now, tokens has a term called term frequency and inverted document frequency which tells me that, which token, which word is more important. Let us assume that I am analysing customer reviews on hotels. Can you tell me when I am analysing customer reviews of hotels and they are let us say, 10,000 reviews each row is 1 review I have a excels file but, if you want review 1 review 2 review 3 review 4 that is serial number, then the text of the reviews are there.

And with those texts of the reviews I removed the stopwords remove these that etcetera. After that, can you tell me that which word in a review of hotels will be most frequent which word will be most used? Just think about it.

So, the if you do that, if you sometimes do that and we will do that later probably the word which will be most frequent is hotel, the hotel h, o, t, e, l hotel this word because, we are reviewing hotels. So, everybody will say that this hotel is good this hotel is this hotel is that so, the word hotel will be the most common one.

The question is that, whether this word is meaningful enough or not. How I can say that some word is meaningful in creating information and some word is not meaningful in creating information. So, that is what is TFIDF context comes in, just think about TFIDF.

(Refer Slide Time: 03:04)



So, if there is my review1, review2, review3, review4, review5 and review6 let us say, and some text, some text is there this is 1 text, text 2, text 3, text 4, text 5, text 6. If a word occurs most amount of time the more a word occurs within a text, the more important the word is that makes sense, the more a word occurs the more important that particular word is, let us find it on the other hand.

If the word is actually occurring here also, here also, here also, here also, here also, here also, then that particular word is less important. So, the more a word occurs in different text the less it is important. So this is something that we try to make sure so, for example, when you, when somebody says about a particular term quite a lot in a particular text that means within the text the particular term is important.

But, if I say the term every time I speak, so I remember in our childhood I should not say probably in our childhood there was a teacher who used to say some word, let us say got it, got it, got it this got it he used to say a lot when, he is teaching. So, let us say he is teaching that Akbar did this that, that and etcetera and then probably Ashok did this, this, this got it? And then, this one, this one, this one, xy, xyz, xyz and then, got it? So, this got it term was his problem of speech probably he used to use this particular term quite a lot.

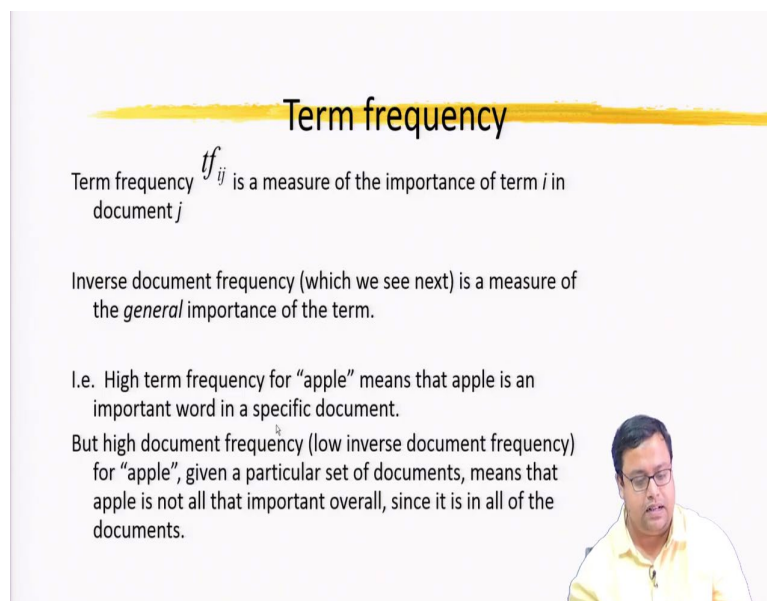
Now, in every term if this got it, got it, got it terms comes in then he is actually not this got it has no meaning, that means in all the documents that he is creating the speech documents that his creating his got it term is common. So, the mode of word occurs in different text the less

is it is important that particular got it does not meaning that is you actually is more concerned about whether you have understood or not, that is his way of teaching and saying got it is something that is very common for him.

On the other hand in we need a text let us he is not commonly says got it but, in a particular type of text he says got it a lot, let us say while he is teaching only then it says got it a lot in other kind of conversation he do not say got it. So, if then that is that particular got it is more concerned about more important in this particular text.

So similarly, the words which are more commonly occurring in one particular document, here document means this is one document. So, one particular document that is most important to the document, so that is term frequency inverse document frequency is the more it occurs in all various kinds of documents. Then, less it is important that is why, it is inverse to document frequency. So this is called sorry, this is called term frequency and this one is called document frequency. So, that is word is also something that we are trying to discuss.

(Refer Slide Time: 07:50)



Term frequency

Term frequency tf_{ij} is a measure of the importance of term i in document j

Inverse document frequency (which we see next) is a measure of the *general* importance of the term.

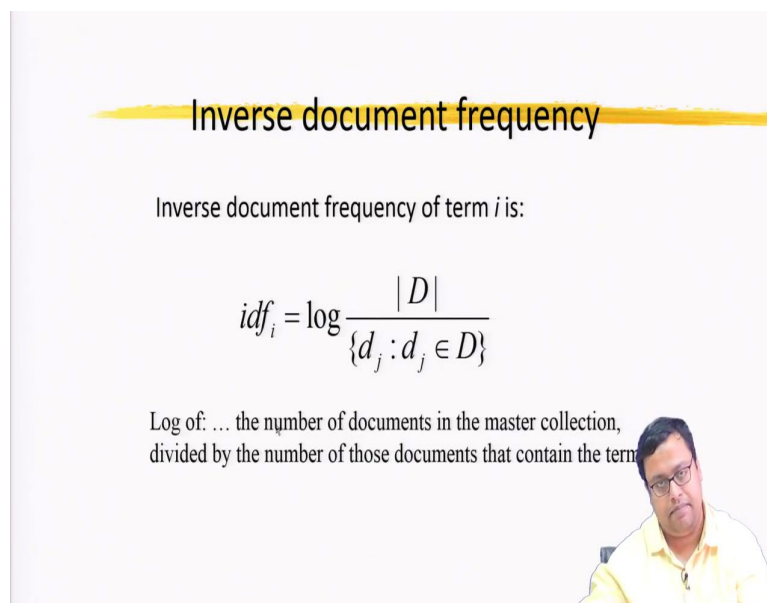
I.e. High term frequency for "apple" means that apple is an important word in a specific document.

But high document frequency (low inverse document frequency) for "apple", given a particular set of documents, means that apple is not all that important overall, since it is in all of the documents.

What is the term frequency? A frequency tf_{ij} is a measure of importance of term i in document j . So that is what a term frequency is, it is a measure of importance of term i in document j . So, inverse document frequency is a measure of the general importance of the term so, that is something that also is meaningful so, what is term frequency?

Term frequency is the measure of importance of term i in document j and then inverse document frequency is the measure of general importance of this particular term. So, high term frequency for apple means that apple is an important word in this specific document, but high document frequency for apple given a particular set of document means that apple is not at all that important is occurring everywhere. So, since it is in all the documents, this is a topic about probably apples only that is why apple is coming up nobody is focusing on apple a lot it is just common. So, that is what term frequency and inverse document frequency is.

(Refer Slide Time: 09:03)



Inverse document frequency

Inverse document frequency of term i is:

$$idf_i = \log \frac{|D|}{\{d_j : d_j \in D\}}$$

Log of: ... the number of documents in the master collection,
divided by the number of those documents that contain the term

(A video feed of a presenter in a yellow shirt is visible in the bottom right corner of the slide.)

So, inverse document frequency is been measured in like this $idf_i = \log |D| / \{d_j : d_j \in D\}$ so, there is log of count how many documents are there, that comes in the numerator and the denominator is the number of documents in the master collection / the number of those documents that contains the term, fair enough.


(Refer Slide Time: 09:27)

TFIDF encoding of a document

So, given:

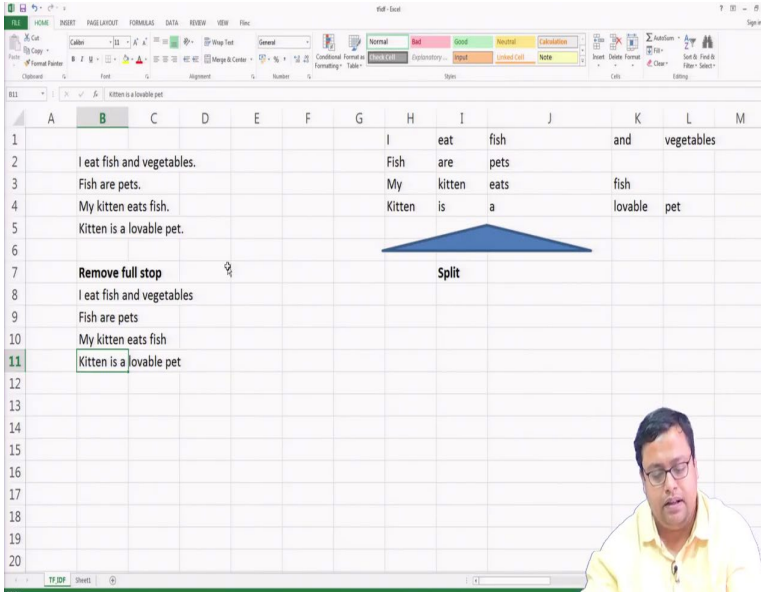
- a background collection of documents
(e.g. 100,000 random web pages,
all the articles we can find about cancer
100 student essays submitted as coursework
...)
- a specific ordered list (possibly large) of terms

We can encode any document as a vector of TFIDF numbers, where the i th entry in the vector for document j is:


$$tf_{ij} \times idf_i$$


So, that is and TF and IDF is nothing but $IDF = tf_{ij} \times idf_i$ is the importance of that particular term within that document. So, a background collection of document has to be given and the specific order list of term is given we can encode any document as a vector of TFIDF numbers. But, i th entry is the vector for document j is. So that is something that we can do. So, let me just show you an example what I am talking about, let it open, what this TFIDF is? Let me just make it bigger.

(Refer Slide Time: 10:27)



	A	B	C	D	E	F	G	H	I	J	K	L	M
1							I	eat	fish		and	vegetables	
2							Fish	are	pets				
3							My	kitten	eats		fish		
4							Kitten	is	a		lovable	pet	
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													



So, let us say I have this four terms I eat fish and vegetables. Fish are pets, my kitten eats fish kitten is a lovable pet this four lines are there. So, each of this line is one document so, first job is removing full stop you can remove other things also, so I eat fish and vegetables fish are pets my kitten eats fish and kitten is a lovable pet this is something. The next job is to split them tokenize them.

So, I eat fish and vegetables is something that I create and fish are pets, my kitten eats fish are separate words and kitten is our lovable pet all these things I am converting them to separate words that is the second step. Then, what I am doing? I am picking up each of the, this thing here so, we are will be calculating TFIDF.

(Refer Slide Time: 11:37)

Entire Document				Doc1				
List of all words	TF	IDF	TF IDF	List of all cCount	Key	List of all cCount	TF	
I	0.2	2	0.4	1	Doc1I	1	0.2	
eat	0.2	2	0.4	1	Doc1eat	eat	1	0.2
fish	0.2	0.41504	0.08301	3	Doc1fish	fish	1	0.2
and	0.2	2	0.4	1	Doc1and	and	1	0.2
vegetables	0.2	2	0.4	1	Doc1veget	vegetables	1	0.2
Fish	0.33333	0.41504	0.13835	1	Doc1are	are	0	0
are	0.33333	2	0.66667	1	Doc1pets	pets	0	0
pets	0.33333	2	0.66667	1	Doc1My	My	0	0
My	0.25	2	0.5	2	Doc1kitter	kitten	0	0
kitten	0.25	1	0.25	1	Doc1eats	eats	0	0
eats	0.25	2	0.5	1	Doc1is	is	0	0
fish	0.25	0.41504	0.10376	1	Doc1a	a	0	0
Kitten	0.2	1	0.2	1	Doc1lovab	lovable	0	0
is	0.2	2	0.4	1	Doc1pet	pet	0	0
a	0.2	2	0.4		Doc1Total	Total word		
lovable	0.2	2	0.4					
pet	0.2	2	0.4					
Total word				17				

So, the entire document has, like this I eat fish and vegetables. Theek hai? Then, fish are love, fish are pets, so fish is occurring here, so fish has already been taken into account, then that is why are pets this is there, then my kitten eats and then, fish is here again then, kitten is a lovable pet so, kitten is again occurring so, that is why to is a lovable pet.

Now here, you see that there is pets and pet, so I should have before I have done anything I should have lemmatize it because, that otherwise, it is considering pets and pet has to be two different word. So if I consider pets and pet to be two different word then corresponding term frequency corresponding inverse document frequency will be different.

Because, if in the same text if I am saying pet and pets both then a term frequency should go up but, I am not considering that if, there is two different document word in one document

pet is occurring another document pets are occurring, if that is happening and I am not considering them to be same word, then the document frequency is coming down, because when I am calculating pet I am not considering pets, when I am calculating pets and not considering pet. So two different document frequency I am getting so, which is lower than the combined document frequency.

So, that kind of problem comes up if you do not lemmatize I should have lemmatized it is a I should have removed them because, these are stopwords, but I am not doing it because, I am just showing you a small example, fair enough. So now, what I do is I find out for each document what is the document frequency?

So, you see that in document one I this is my key that means document one and I, so doc this and this basically, comes up to be this you see AA5 which is Doc1 and the word numb I A7 this is occurring once, this is occurring, this is occurring, this is occurring, this is occurring, but these guys are not occurring so, in Doc1 this particular words are not there. So, that is why counts are 1, 1, 1, 1, 1 total count is 5 that term frequency for each word is 0.2, fair enough.

(Refer Slide Time: 14:28)

Doc1				Doc2			
Key	List of all cCount	TF		Key	List of all cCount	TF	Key
Doc1l	l	1	0.2	Doc2l	l	0	Doc
Doc1eat	eat	1	0.2	Doc2eat	eat	0	Doc
Doc1fish	fish	1	0.2	Doc2fish	fish	1	Doc
Doc1and	and	1	0.2	Doc2and	and	0	Doc
Doc1veget	vegetables	1	0.2	Doc2veget	vegetables	0	Doc
Doc1are	are	0	0	Doc2are	are	1	Doc
Doc1pets	pets	0	0	Doc2pets	pets	1	Doc
Doc1My	My	0	0	Doc2My	My	0	Doc
Doc1kitter	kitten	0	0	Doc2kitter	kitten	0	Doc
Doc1eats	eats	0	0	Doc2eats	eats	0	Doc
Doc1is	is	0	0	Doc2is	is	0	Doc
Doc1a	a	0	0	Doc2a	a	0	Doc
Doc1lovab	lovable	0	0	Doc2lovab	lovable	0	Doc
Doc1pet	pet	0	0	Doc2pet	pet	0	Doc
Doc1Total	Total word	5		Doc2Total	Total word	3	Doc

The same thing I have done for the next word. So, in document 2 fish are pets, so fish is occurring so, document 2 fish is there document 2 are is there document 2 pets are there the total frequency is 3 term frequencies 0.3, 0.33, 0.33 makes sense.

(Refer Slide Time: 14:49)

Doc3				Doc4				Document name	List of all words
Key	List of all cCount	TF		Key	List of all cCount	TF			
Doc3l	l	0	0	Doc4l	l	0	0	Doc1	l
Doc3eat	eat	0	0	Doc4eat	eat	0	0	Doc1	eat
Doc3fish	fish	1	0.25	Doc4fish	fish	0	0	Doc1	fish
Doc3and	and	0	0	Doc4and	and	0	0	Doc1	and
Doc3veget	vegetables	0	0	Doc4veget	vegetables	0	0	Doc1	vegetables
Doc3are	are	0	0	Doc4are	are	0	0	Doc2	Fish
Doc3pets	pets	0	0	Doc4pets	pets	0	0	Doc2	are
Doc3My	My	1	0.25	Doc4My	My	0	0	Doc2	pets
Doc3kitter	kitten	1	0.25	Doc4kitter	kitten	1	0.2	Doc3	My
Doc3eats	eats	1	0.25	Doc4eats	eats	0	0	Doc3	kitten
Doc3is	is	0	0	Doc4is	is	1	0.2	Doc3	eats
Doc3a	a	0	0	Doc4a	a	1	0.2	Doc3	fish
Doc3lovab	lovable	0	0	Doc4lovab	lovable	1	0.2	Doc4	Kitten
Doc3pet	pet	0	0	Doc4pet	pet	1	0.2	Doc4	is

Similarly, third one and fourth one you just check here, there a four words that is why 0.25 no word is repeating also you have to understand that and here, there are five, four words again. So, this five words this and this five so, I got list of term frequencies.

(Refer Slide Time: 15:07)

Document	Term	TF	IDF	TF-IDF	Entire Document List of all c/Count
Doc1	I	0.2	2	0.4	I 1
Doc1	eat	0.2	2	0.4	eat 1
Doc1	fish	0.2	0.41504	0.08301	fish 3
Doc1	and	0.2	2	0.4	and 1
Doc1	vegetables	0.2	2	0.4	vegetables 1
Doc2	Fish	0.33333	0.41504	0.13835	are 1
Doc2	are	0.33333	2	0.66667	pets 1
Doc2	pets	0.33333	2	0.66667	My 1
Doc3	My	0.25	2	0.5	kitten 2
Doc3	kitten	0.25	1	0.25	eats 1
Doc3	eats	0.25	2	0.5	is 1
Doc3	fish	0.25	0.41504	0.10376	a 1
Doc4	Kitten	0.2	1	0.2	lovable 1
Doc4	is	0.2	2	0.4	pet 1
Doc4	a	0.2	2	0.4	Total word 17
Doc4	lovable	0.2	2	0.4	
Doc4	pet	0.2	2	0.4	

So, correspondingly I have written here, that Doc1 I Doc1 eat Doc1 fish corresponding term frequencies I have written here, fair enough. Now, how will I find out the inverse document frequency? You have to understand in the inverse document frequency is if you remember the formula.

(Refer Slide Time: 15:34)

Inverse document frequency

Inverse document frequency of term / is:

$$idf_i = \log \frac{|D|}{d_i}$$

Log of... the number of documents in the master collection, divided by the number of those documents that contain the term.

IDF	TF	IDF	TF-IDF	Entire Document List of all c/Count	Key
0.2	2	0.4		I	1 Doc
0.2	2	0.4		eat	1 Doc
0.2	0.41504	0.08301		fish	3 Doc
0.2	2	0.4		and	1 Doc
0.2	2	0.4		vegetables	1 Doc
0.33333	0.41504	0.13835		are	1 Doc
0.33333	2	0.66667		pets	1 Doc
0.33333	2	0.66667		My	1 Doc
0.25	2	0.5		kitten	2 Doc
0.25	1	0.25		eats	1 Doc
0.25	2	0.5		is	1 Doc
0.25	0.41504	0.10376		a	1 Doc
0.2	1	0.2		lovable	1 Doc
0.2	2	0.4		pet	1 Doc

The formula is log (total number of documents / documents word a particular word is occurring). So here, the Doc role number of documents is four, so log (4)/by count if you

count whether, the Q7 that means this particular word is occurring in ay7 to a23. So, ay7 to a23 this is my documents. Here, it is how many times it is occurring?

So, the occurrences are there and we are taken our $\log_2 2$ the base has been taken as 2, does not matter actually, but let this particular in this particular case the $\log_2 2$ is something that we are taking and correspondingly I am getting the inverse document frequency is for each of the word just check the formula and TFIDF is nothing but, the multiplication of this 2.

So, this is a score to decide that how important a word is we can put a cut off and then only create a subset, for example, you can clearly see that there can be certain words which are less important. For example, fish we this is talking about fish only I eat fish, kitten eat fish, fish are pets. So, fish is the most common word that is why the IDF of is very low and that is why fish is not something that is important here in this particular text. So, we have to find out which one is important which one is not important using this TFIDF analysis.

(Refer Slide Time: 17:17)

	A	B	C	D	E	F	G	H	I	J	K
1		Sachin is a cricketer			Sachin	is	cricketer		Sachin		
2		Roger plays tennis			Roger	plays	tennis		is		
3		Sachin meets Roger			Sachin	meets	Roger		cricketer		
4									Roger		
5									plays		
6				tf	idf	tf-idf			tennis		
7		doc1	Sachin	0.333333	0.584963	0.194988			meets		
8		doc1	is	0.333333	1.584963	0.528321					
9		doc1	cricketer	0.333333	1.584963	0.528321					
10		doc2	Roger	0.333333	0.584963	0.194988					
11		doc2	plays	0.333333	1.584963	0.528321					
12		doc2	tennis	0.333333	1.584963	0.528321					
13		doc3	Sachin	0.333333	0.584963	0.194988					
14		doc3	meets	0.333333	1.584963	0.528321					
15		doc3	Roger	0.333333	0.584963	0.194988					

Can we do the same thing for this Sachin is a cricketer, Roger plays and Sachin meets Roger. So, let us just do this for this particular word once more, for so if I have to break it Sachin is a cricketer, Roger plays tennis and Sachin meets Roger this is the words. So, then what is the list of the word? Basically, Sachin is cricketer, Roger plays tennis. Now, see here Sachin meets Roger, Sachin has already occurred Roger has already occurred so, only meets will come fair enough. Now, what is the term frequency?

This is occurring so, the term frequency of a word these are the word list, these are the list of words I will just copy it and paste it here these are my list of words. So, then what are the documents?

Document1 has 3 words, document1, document1, document1 then doc2, doc2, doc2 and then doc3, doc3, doc3 if I have these things, these are my word list, these are my document list this my word list so, what is term frequency?

The term frequency for this guy is basically, $1/3$ this is also $1/3$ Roger plays tennis and Sachin meets Roger. So, term frequencies are in all the cases $1/3$, what is inverse document frequency so, how many documents are there?

There are 3 documents $3/\log(3)$, sorry, how many documents Sachin is occurring in how many documents? 2 documents so, 2 then if I just drag it just tell me that is is occurring in how many documents? 1, cricketer is occurring in how many documents? 1, Roger is occurring in 2 documents, plays 1, tennis is 1 Sachin 2, meets 1 and Roger and 2. So, then what is TFIDF? It is nothing but, this multiplied by this with this and if I just drag that I got the TFIDF scores.

So, I can say here the most important one were tennis or plays or cricketer. So, is should have removed cricketer or meets. So, they are these are the words which are most common most important in this particular text. So, that is how we try to find out TFIDF. So now, going ahead.

(Refer Slide Time: 21:30)

From Wikipedia's plot summary
(method: string search)

- ...
- Natasha is convinced that she loves Anatole and writes to Princess Maria, Andrei's sister, breaking off her engagement [with Andrei]. At the last moment, Sonya discovers her plans to elope and foils them. Pierre is initially horrified by Natasha's behavior, but realizes he has fallen in love with her. During the time when the [Great Comet of 1811-2](#) streaks the sky, life appears to begin anew for Pierre.
- Prince Andrei coldly accepts Natasha's breaking of the engagement. He tells Pierre that his pride will not allow him to renew his proposal. Ashamed, Natasha makes a suicide attempt and is left seriously ill.
- ...
- Having lost all will to live, [Andrei] forgives Natasha in a last act before dying.
- Pierre's wife Hélène dies from an overdose of [abortion](#) medication (Tolstoy does not state it explicitly but the euphemism he uses is unambiguous). Pierre is reunited with Natasha, while the victorious Russians rebuild Moscow. Natasha speaks of Prince Andrei's death and Pierre of Karataev's. Both are aware of a growing bond between them in their bereavement. With the help of Princess Maria, Pierre finds love at last and, revealing his love after being released by his former wife's death, marries Natasha.

So, we can also use from Wikipedia's plot summary with a string search we can find out that what is the summary of the text? So, Natasha is convinced that she loves Anatole and etcetera and this kind of takes question is that, what is a summary of the plot we can do that, that is where we can also.

(Refer Slide Time: 21:50)

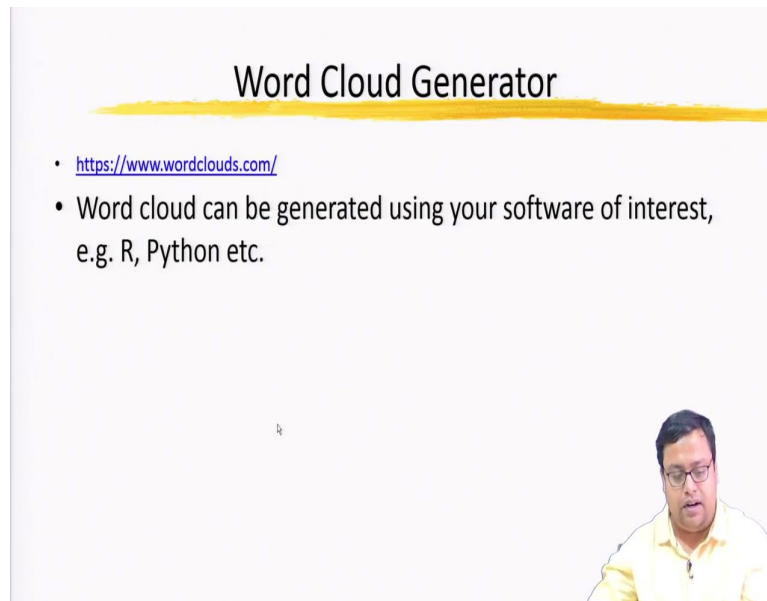
From Wikipedia's plot summary
(method: string search)

- ...
- Natasha is convinced that she loves Anatole and writes to Princess Maria, Andrei's sister, breaking off her engagement [with Andrei]. At the last moment, Sonya discovers her plans to elope and foils them. Pierre is initially horrified by Natasha's behavior, but realizes he has fallen in love with her. During the time when the [Great Comet of 1811-2](#) streaks the sky, life appears to begin anew for Pierre.
- Prince Andrei coldly accepts Natasha's breaking of the engagement. He tells Pierre that his pride will not allow him to renew his proposal. Ashamed, Natasha makes a suicide attempt and is left seriously ill.
- ...
- Having lost all will to live, [Andrei] **Total time: 29 mins since creation of word cloud, 17 mins since creation of Pierre-Natasha-Andrew chart (includes making these slides for you)**
- Pierre's wife Hélène dies from an overdose of [abortion](#) medication (Tolstoy does not state it explicitly but the euphemism he uses is unambiguous). Pierre is reunited with Natasha, while the victorious Russians rebuild Moscow. Natasha speaks of Prince Andrei's death and Pierre of Karataev's. Both are aware of a growing bond between them in their bereavement. With the help of Princess Maria, Pierre finds love at last and, revealing his love after being released by his former wife's death, marries Natasha.

So, total of time is 29 minutes since, creation of word cloud 17 minutes since, creation of Pierre-Natasha-Andrew chart into the making this light for you.

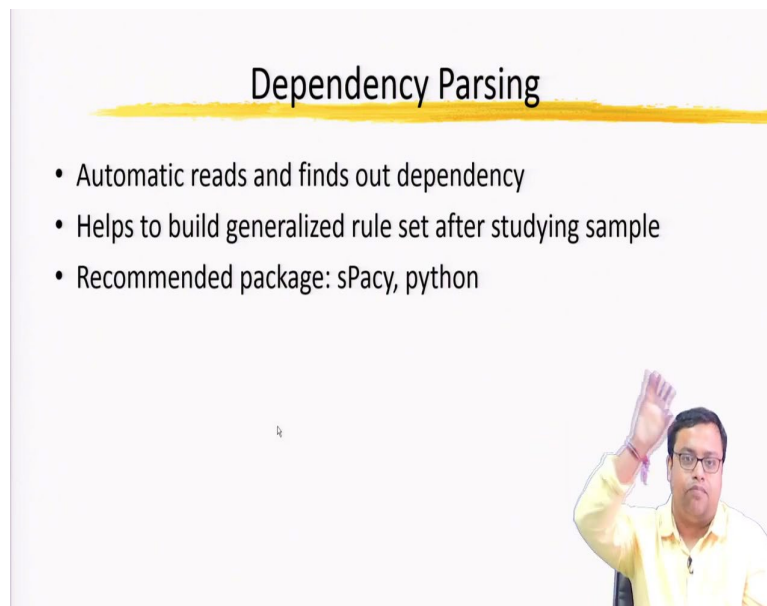
Trending topics, that is also a form of summarization, so we can see that in which part of the world, which kind of topic is most coming common usually come done into twitter. So in twitter there is a geo code which is also kept in the twitter tack. So, if you want to know that in which kind of which part of the world which text this becoming more common, then I can know that whether a rumour is spreading why it is spreading this and that.

(Refer Slide Time: 23:11)



We can also use this kind of word cloud generator word cloud can be generated using your software of interest like our R or python. So, we will be using R other than that, there is something called word[clouds.com](https://www.wordclouds.com/), which you can also use to create word clouds, that is also one option.

(Refer Slide Time: 23:28)



Dependency Parsing

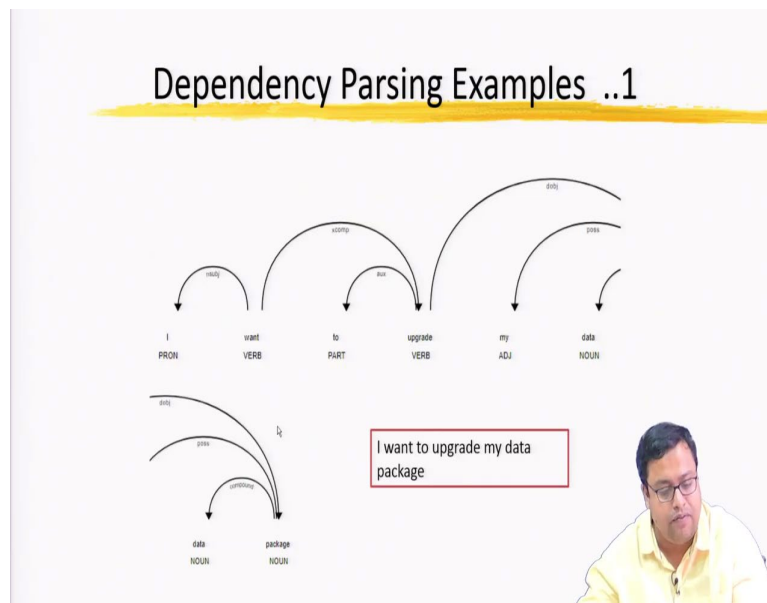
- Automatic reads and finds out dependency
- Helps to build generalized rule set after studying sample
- Recommended package: sPacy, python

Then comes another interesting thing called Dependency Parsing. So, automatic reads Dependency Parsing is what? It is another method of using natural language processing where you read the text and automatically understand the dependency, dependency means what? How one word is dependent on other. So let us say if I say that I am eating breakfast, who is the subject here? I, who were is the object here, breakfast what is the verb here, eating.

So, this kind of relationship who is subject, who is doing the job, what is the jobs name and what he is doing with this job. All of these things the subject-object predicate verb can be found out using this dependency parsing, that helps to understand the text it is not one single text only in a single text context the well overall context is lost that is in dependency parsing the context is still there.

It helps to build generalize rules set after studying sample and it, there can some of the packages in Python, sPacy and python which is used in R also we can do that we will try to do that.

(Refer Slide Time: 24:45)



So, dependency parsing is like let us say, I want to upgrade my data package, so you have to understand that, what is the verb here? So verb here is want or upgrade, so key verb is want one to word, who wants, who is the subject, the subject is I, which is the pronoun. So the first job is to find out verbs and then you have to find out that whether, one verb is a component verb of another verb. So, I want to upgrade this is the verb component total and want is the major verb and the want what?

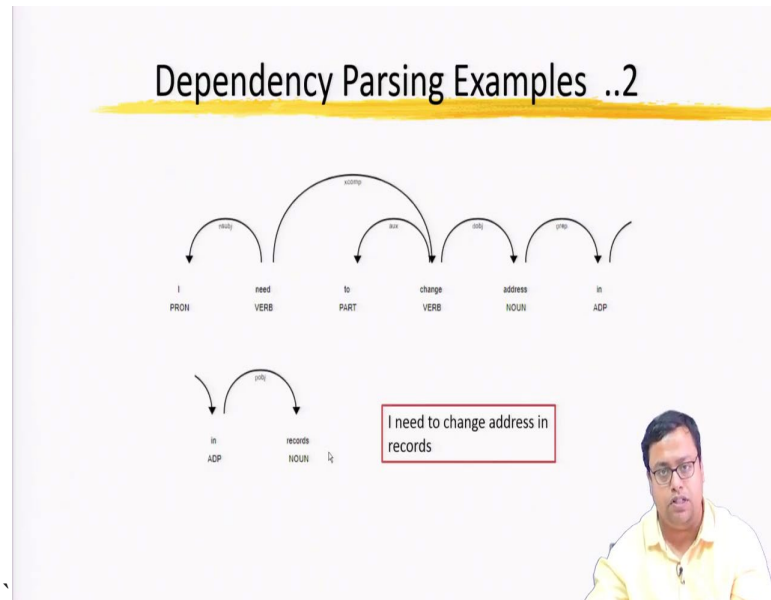
To upgrade the second verb is upgrade which is the component verb of the initial verb and two is just an auxiliary. Then, want to upgrade what? Want to upgrade my data package so, my data package the package is the noun data package is actually compound noun this is what I want to upgrade and my is a basically a adjective, which is, whose data package my data package.

So, this kind of relationship building if, you can read find out this relationship you know that what this guys doing why it is applicable in let us say the text, we what I want to cancel my flight they know that okay want to cancel, want is the major thing cancel is the, the cancel what? Cancel my flight, flight means my ticket whose ticket my ticket.

So, this kind of answers can be done by dependency parsing, who is I? I I if, I write my wife wants to upgrade her seat, then who wants you then he will ask the, the automator will ask that can you tell me the passenger name or can you tell me your wife's name? But, if you say that I want to do this then it will automator says can you tell me your name?

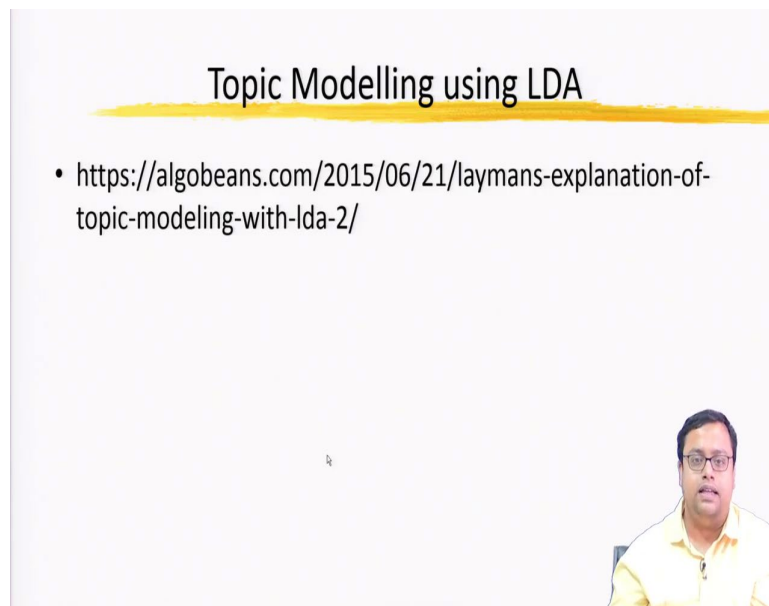
So, this kind of answers can only be given if you understand the dependency parsing properly. So, that is something which also comes under the domain of natural language processing another example.

(Refer Slide Time: 27:03)



I need to change address in records. So, what do you want, what is the major need to change need is the major verb changes the adjoining verb and 2 is just auxiliary, who wants, who needs to change? I so, I is my subject, what? Address, address is my object where? In records. So, in records is basically, the possible the further things further details about this particular object that we want to change. So, this kind of information can be done by dependency parsing.

(Refer Slide Time: 27:41)



Another thing is **Topic Modelling**, we will discuss at the right time Topic Modelling is basically, from a huge text if you can find out, which part of the text is talking about what? What kind of words comes together, when you creating topics that can also be done under the natural language processing.

So, that is all for this particular video, we will do the term frequency inverse document frequency and what clouds in the next video then we will do something on spam detection in the next video in the next week early next week, we will do topic modelling also and dependency parsing also. So let us see how it goes, thank you for being with me I will meet you in the next thanks.