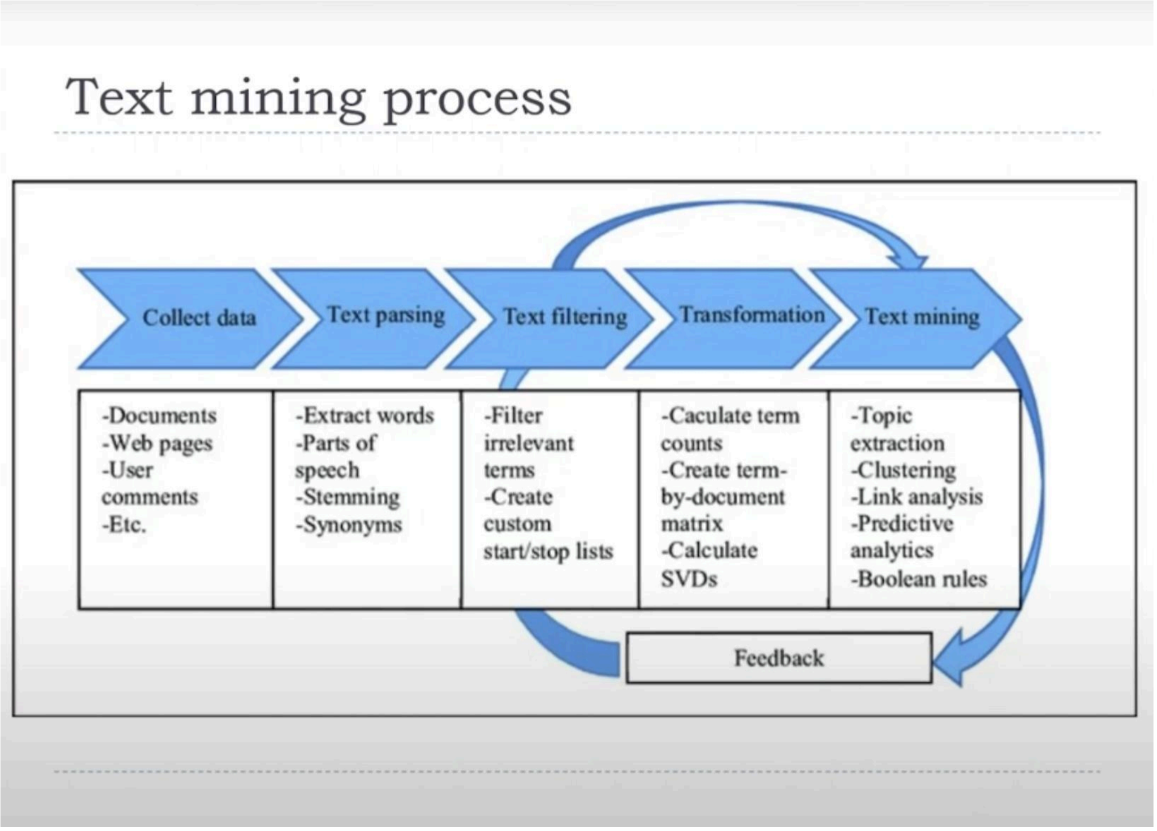
Course Name:Business Intelligence and Analytics
Professor Name:Prof. Saji.K.Mathew
Department Name:Department of Management Studies
Institute Name:Indian Institute of Technology Madras
Week:12
Lecture:45

TEXT MINING PROCESS | BI&A | Prof. Saji K Mathew



Text mining process

Collect data	Text parsing	Text filtering	Transformation	Text mining
-Documents -Web pages -User comments -Etc.	-Extract words -Parts of speech -Stemming -Synonyms	-Filter irrelevant terms -Create custom start/stop lists	-Caculate term counts -Create term-by-document matrix -Calculate SVDs	-Topic extraction -Clustering -Link analysis -Predictive analytics -Boolean rules

Feedback

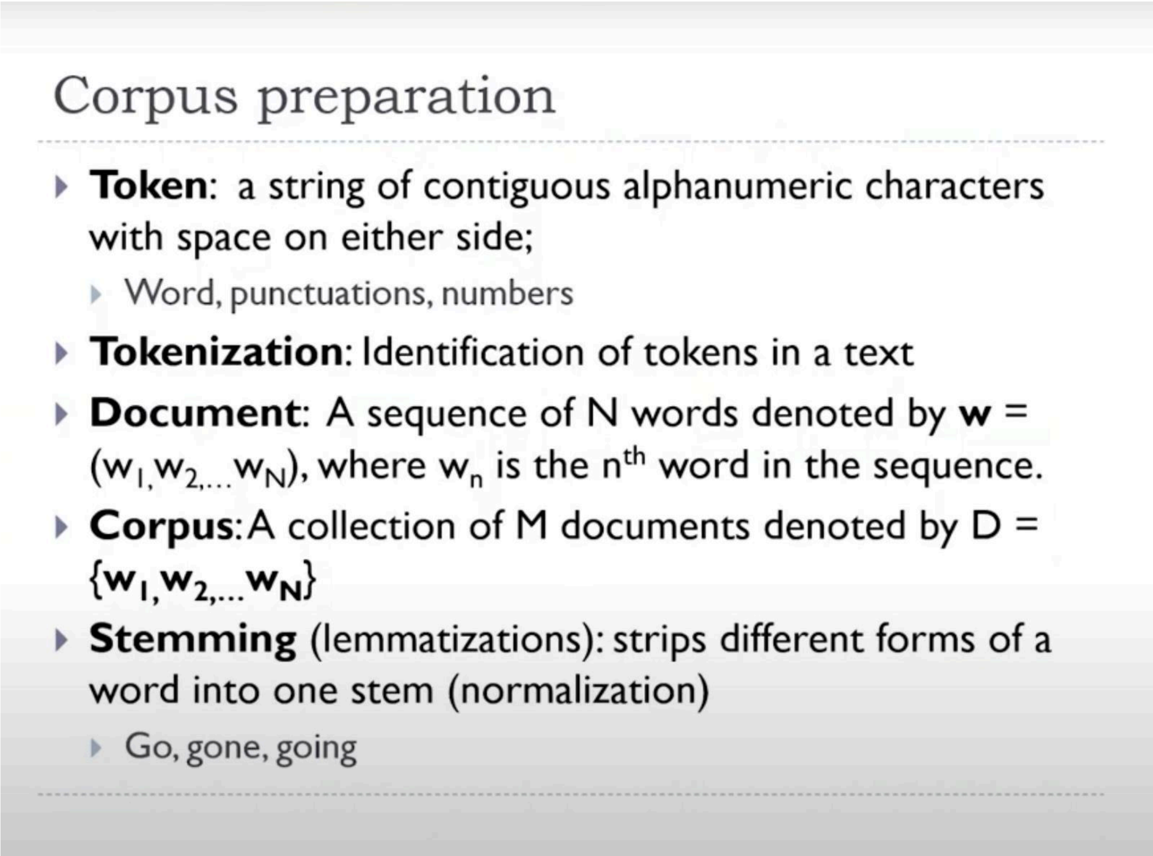
So since text mining deals with textual data as I said, it has to follow a different process, because you do not have numeric data to apply any algorithm. So, how do we do this? So, you can see in text mining process as suggested here, it has four five steps and it starts with data collection and ends with the model, the final model. Now the sources of data for text mining could be the web, a dominant source of data is the web, the world wide web where you have the web pages and using crawlers you can actually collect a lot of data online today. And then it could be books, it could be any other source of data in different formats, it could be word documents, pdf documents, html and so on.

So, you have raw data from multiple sources in multiple formats, so that is your data. Now before, from this I would call as raw data. When you collect a raw data between parsing the text or between preparing the text and collecting the data, there is a steps called corpora or corpus. Building a corpus, a corpus building is the first step towards text mining. What do you mean by corpus? It is a standard term used in text mining parlance, corpus must be a Greek word, corporation comes from corpus, what is corpora, corpus mean? Not more than 30 seconds because Google will tell you, corpus means body. So it is a fake, meaning that is not a fake, we are faking that into a corporation, giving an impression that a corporation is an organic being etc.

To make people feel, that well you are working with collection of people, collection of bodies. So corpus is a body of literature or a body of text which is used to format or create a format of text in such a way that it can be processed further. So, corpus has a particular structure which we will see as we go. So, then there is a step called text parsing where you can identify the parts of speech or POS of the text, what is parts of speech? Well, that is a very elementary question. In school you have learnt parts of speech of a sentence, there is a subject, there is a object and there are lot of words that make the connection between words like I am, I am teaching. So the am is a connector between what I do and who does it. So, the parts of speech could be many, but there are standard parts defined for text mining purpose to parse or break down, break a text into its parts.

So, there are other techniques like stemming, identifying synonyms etc, we will come to that in the next slide. So, parsing the text, so structuring the text, in textual format itself, not giving any numeric value here, understanding the structure of a text. Then filter it, this is all data preparation, again all this I would call as data preparation up till this point, because text is text, so you need to do a lot of hard work to prepare it for modeling. Remove irrelevant words, there is something called stop words in text mining. So you can stop certain commonly occurring words from getting into analysis like a, the, an etc. That is called defining a set of stop words. And at this point of transformation, you are ready to convert the text to numbers and the key construct there is something known as term by document matrix or TDM, term by document matrix.

By matrix itself you mean, there are numbers here, there are numbers, you convert a collection of text into a matrix format known as term by document. One you have TDM, it is like a database of the terms in a text and then you can actually apply some of the techniques that you learn, because you have a database, text database in numeric format. So, this is where that magic happens, where you have transformed text into numbers, but I would say in a very basic level, you are not here understanding the meaning of a text in its context, you are only playing with words and number of words and number of correlated words etc etc. Let us move on, because this is a broad description, but we will go step by step and I will try to give you better understanding of each of the steps in the process of text mining.



Corpus preparation

- **Token**: a string of contiguous alphanumeric characters with space on either side;
  - Word, punctuations, numbers
- **Tokenization**: Identification of tokens in a text
- **Document**: A sequence of N words denoted by **w** = $(w_{1,}w_{2,\dots}w_N)$, where w_n is the n^{th} word in the sequence.
- **Corpus**: A collection of M documents denoted by D = $\{\mathbf{w}_{1,}\mathbf{w}_{2,\dots}\mathbf{w}_N\}$
- **Stemming** (lemmatizations): strips different forms of a word into one stem (normalization)
  - Go, gone, going

So as I said, the first step in text mining process is the corpus preparation, preparing the corpus. So corpus is a technical term in text mining, which shows text is in a format that can be analyzed or prepared for text mining purpose. So, there are certain standard terms or there is a vocabulary of text mining, which you must be very familiar. Otherwise you will get confused when you do text mining. The first or most elementary terminology is a token. What is a token? If a text is given, how do you identify tokens? A text consists of tokens.

What is a token? Token could be a word, could be punctuations, it could be numbers, it could be hypertext, any unit that is part of a text, basic unit that is part of a text. It is not the letters, but it is it starts from words, punctuations, numbers etc. A letter is not a token, unless it is a word in itself. Like A is a token, because it is a word in itself. But an L is not a token. So if I look at this much of a text, A is a token, string is a token, of is a token, is a token, token, token, token, token, that is also a token, punctuation is a token.

So, tokenization is the starting of text mining process or data preparation for text mining, tokens.

Tokenization is identification of tokens in a text, that is what I did here. So of course, you use functions or algorithms for doing this. And now, the next level of aggregation or the next aggregate level is a document. A document consists of tokens and a document is a sequence of n words denoted by w = w 1, w 2 to w n. So, a document consists of tokens, that is fine, but it has some more meaning than what is described here.

What do you consider as a document in a text mining, is a choice of the analyst. For example, your text book has sixteen chapters, assume there are sixteen chapters. What is the document in that book? Is the entire book a document? We can think it that way. You can consider it that way, if you are analyzing a lot of books. Each book can be a document in itself. But if your aim is to analyze different topics within a book, then a document can be a particular chapter.

If your aim is to analyze a particular chapter, then different sections can be a document. So, a document is like a unit. Look at discussion forums. Suppose, I am going to use discussion forum data in the exercise. So, there is a discussion forum to discuss the most popular movie of the time.

And suppose there are thousands of people who give their opinion about the movie. So, who post about the movie? So, I say my opinion, you say your opinion. I reply, you reply, then somebody else comes in. Each is positive. A discussion forum consists of a number of posts, n number of post.

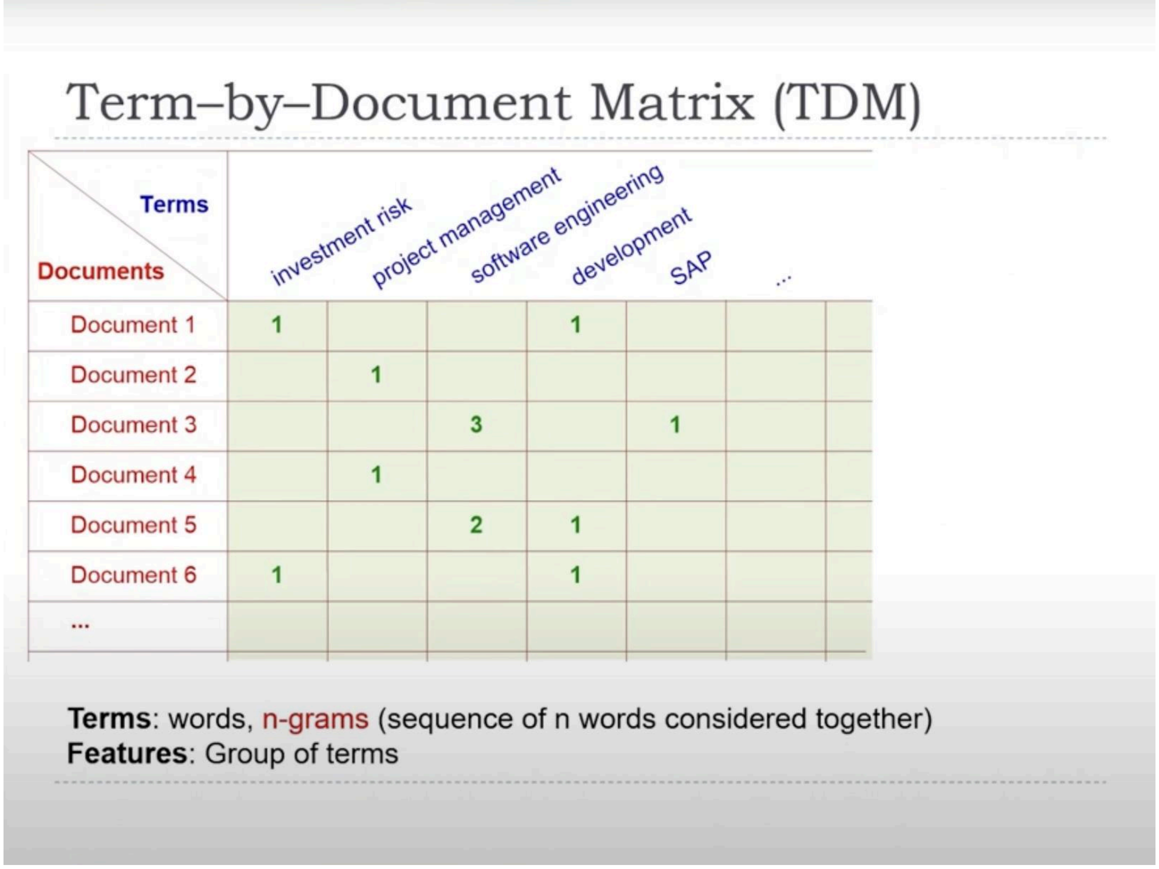
What is a document here? Each post is a document. Your analysis is at the post level. Each email can be a post. Each tweet, twitter data, so twitter itself is designed for text mining. Each tweet is a document.

Retweet, the same tweet is appearing again. It has to be filtered out, because it is the same information, but a document is a basic unit of interest for analysis. What the researcher is interested to analyze? You are interested to analyze tweets, you are interested to analyze discussion forum post, you are interested to analyze chapters of a book, those are your documents. So, please have clear understanding of what is a document. It is not a word document.

So we are so much used to that, you know, .docx as document, but in text mining a document means, what is the fundamental unit you want to actually compare or analyze. Corpus, I talked to you. Corpus is a collection of documents. Corpus is a collection of documents. A document is a collection of tokens.

Now, there is something called stemming. Stemming is particularly useful in lexical analysis. For example, you write, I love, not McDonald's, I love x, I love x, I am loving x. Suppose, someone's interest is to count the sentiment of individuals for a particular product or a person or whatever or a movie or a cine star or whatever or for your boyfriend or girlfriend or whatever you want to analyze. So, there is a sentiment which is a strong sentiment, love, but you look at the different forms of that word love.

There is loving, present tense, present continuous tense, there is loved which is past tense, there is love which is present tense. Now, if you only count the word love, how many times it has occurred? Only one, but you want to see the sentiment wherever it is expressed. So, if you have to include love, you have to cut it here and loved, may be cut it here. You do not miss any form of love.



Term–by–Document Matrix (TDM)

Terms Documents	investment risk	project management	software engineering	development	SAP	...
Document 1	1			1		
Document 2		1				
Document 3			3		1	
Document 4		1				
Document 5			2	1		
Document 6	1			1		
...						

Terms: words, n-grams (sequence of n words considered together)
Features: Group of terms

And you want to count it. How many times the word love has occurred? All forms of it you want to count and therefore, you do something known as stemming or lemmatization. Lemmatization is a technique to strip parts of a word such that the word count is maximized. You understood this example. So sometimes it is called lemmatization, sometimes it is called stemming. We will use a function for stemming and then you will see how this particular method is useful.

And this is a depiction of the term by document matrix, TDM. TDM is a matrix which has two dimensions. One dimension is the dimension of the document, document as a unit. So, you take different documents. Suppose, you have 200 documents, each document is the member of the y axis and on the x axis or the horizontal axis, you have the terms.

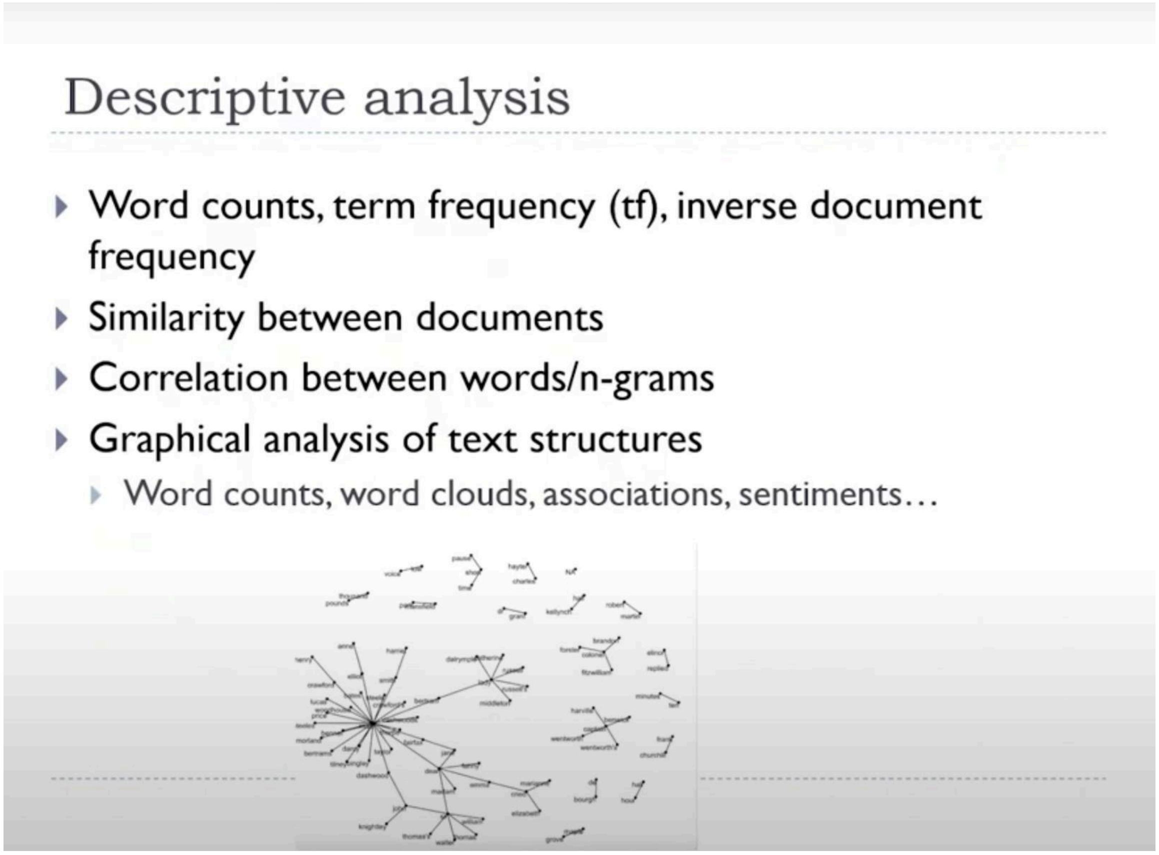
We do not call them words, but we call term because a term can be a word or a phrase or sometimes known as n gram. What is an n gram? There is a concept of n grams in text mining. So, n gram actually is a term. See, project is a word, management is a word, but project management is what? Two words together form a phrase or a term and they together carry some meaning also. That is a term. They go together. It together carries a meaning. So, this is a bigram. This is called a bigram. A sequence of two words, a sequence of n words is known as a n gram. Software project management is an n gram with n= 3, software project management.

You understand? So, n grams are also useful in analyzing. How many times, say project management, investment risk? How many times is, they are occurring together? So, it is a sequence, two words in sequence. Now, you can see that there are terms and there are documents. So, what are these counts about? Investment risk is occurring once in the document 1 and in document 6, but these are all voids.

It does not occur in this documents. Project management, once here in document 2, once in document 4. Software engineering, document 3 and document 5 has it and many times. You see what meaning is conveyed? It has the distribution of words in different documents. And this text mining parlance, these terms would be features. They are actually the features of the text.

The features of the text are nothing but the terms and their occurrences in the documents are the counts. You can actually create a TDM with the actual number of counts or you can also have a TDM, whether they occur or not. So, that could be either 0 or 1. Yeah, it could be binary or it could be based on the actual counts.

Now, you can also see that a lot of sparse, lot of sparsity in this matrix because many documents will not have many terms. So, that is the area which requires separate attention in terms of optimizing a TDM.



Descriptive analysis

- Word counts, term frequency (tf), inverse document frequency
- Similarity between documents
- Correlation between words/n-grams
- Graphical analysis of text structures
  - Word counts, word clouds, associations, sentiments…

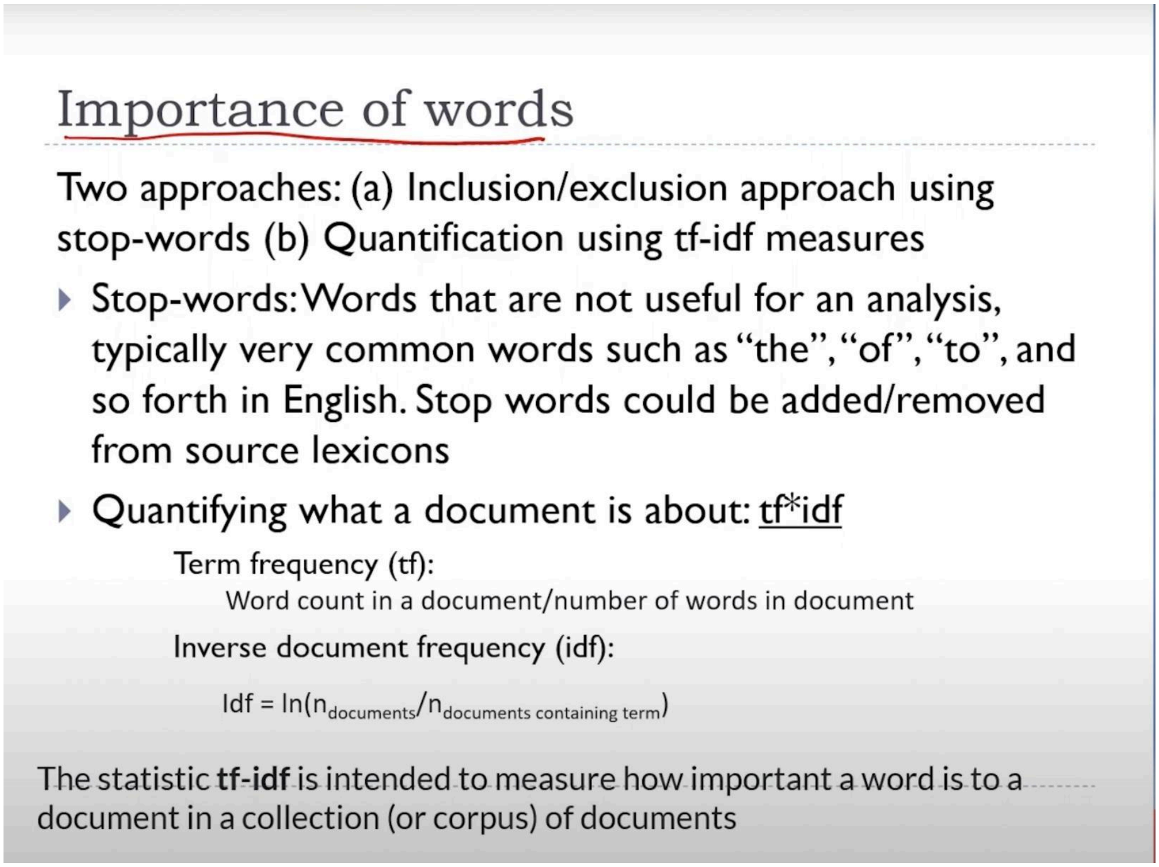
Now, as I said we are not entering into LLM, large language models, but we are entering into lexical analysis. How to make some basic sense of text? How to describe text in terms of what is contained in the text algorithmically? So, some of the basic measures for describing text analytically are word counts. For example, how many times the word love is occurring? How many times the word hate is occurring? How many times the word useless is occurring? So, whatever words are of interest to you, you can actually look at the collection of text or the corpus and make sense.

Term frequency. Term frequency is defined differently, it is not just the count. In the next slide, I will show you the formulae for term frequency, inverse document frequency or IDF. TF and IDF together gives a different understanding of the importance of words. So, these are some basic measures that are used to describe the text. So, keep in mind in analytics, we talked about descriptive, explanatory, predictive etc.

In lexical analysis, we are restricting to a very descriptive analysis of the text. We are not entering into prediction. So, the other interesting measures could be similarity between documents. Can you think of a application of it? We are coming to it, what are some measures available, but similarity. One way to look at a text is word counts, term frequency, how many times particular words are occurring or relatively occurring etc

Other is compare two documents. Did you get that? Plagiarism detection. Similarity which is very important requirement in academic circles. Have you heard of Turnitin? When you are going to submit your project reports, I upload that into Turnitin and see document A, document B, how similar, there is a similarity index we generate and 10 percent similarity, 20 percent similarity, 80 percent similarity and so on. So you get a sense of how similar two documents are, practically very useful.

Correlation between words and n grams. Whenever a word occurs, which is the other word that occurs? Graphical analysis, then you can plot this. So in my exercise, I will try to take you through some of these insights from text description using these measures. Now, one of the important aspects of descriptive analysis of the text is to understand importance of words, to understand what are some of the important words that is being talked about or what are the unimportant words.



Importance of words

Two approaches: (a) Inclusion/exclusion approach using stop-words (b) Quantification using tf-idf measures

- Stop-words: Words that are not useful for an analysis, typically very common words such as "the", "of", "to", and so forth in English. Stop words could be added/removed from source lexicons
- Quantifying what a document is about: tf*idf

Term frequency (tf):

Word count in a document/number of words in document

Inverse document frequency (idf):

$\text{Idf} = \ln(n_{\text{documents}}/n_{\text{documents containing term}})$

The statistic **tf-idf** is intended to measure how important a word is to a document in a collection (or corpus) of documents

So, some words are not important. So in unimportant aspect, there are something called stop words. Stop words in lexical analysis or bag of words analysis, I am using this synonymously. These words make no sense. For example, how many times the is coming, how many times of is coming, how many times to is coming, does it make any sense? It does not. That is not a count you are interested in. So, you want to stop those words from occurring in the analysis.

So, you can stop them. So, stop words are for each language are available in certain dictionaries, in certain lexicons. So, you can use those lexicons and then sort of anti-join, remove those words from your corpus, so that they do not get into your analysis. So, this is related to importance, because they are unimportant. The other is, quantifying the importance of a word in a document. Please see that this is a document level importance articulation for which a measure called td, tf × idf, term frequency into inverse document frequency, idf is used.

It has a certain meaning and you should get what does it mean. Term frequency is nothing but word count in a document divided by number of words in the document. For example, if you are getting this power point, how many times the word text is used divided by number of words in the, in the PPT. You may see that the word text is occurring many times in the PPT. So, this ratio is the term frequency, number of times a particular word is occurring divided by number of words in the document.

That is intuitive and clearly you can understand. If you have a list of all the words and it is t f, you get to and if you sort it in a particular order, you understand which word is most important and which is least important. This is fine. But what is the intuition of inverse document frequency? Inverse document frequency is defined as logarithm to the, logarithm of number of documents divided by number of documents containing the term. And then the final measure is tf × idf. What is this whole sense of tf × idf, which is a widely used measure for importance of words.

What is the sense you get from this? When you multiply, t f is very directly intuitive, how important is a word in a document. But why i d f and why multiply by i d f? i d f goes between documents. It is restricted to one document alone. So, i d f is number of documents divided by number of documents containing the term. To understand this, to illustrate this, let me actually talk about my course.

Suppose this is spread on 14 lectures, 14 PPTs, this is text. Now 14 documents, you have 14 documents. Take a word like analytics. What do you think about the i d f of a word like analytics in the 14 power points? This will be 14 divided by number of documents containing the term. Since it is a course related to analytics, chances are that every PPT has analytics.

So, this may become even 14. It is very difficult to find a PPT without the word analytics. So, it is like a common word.

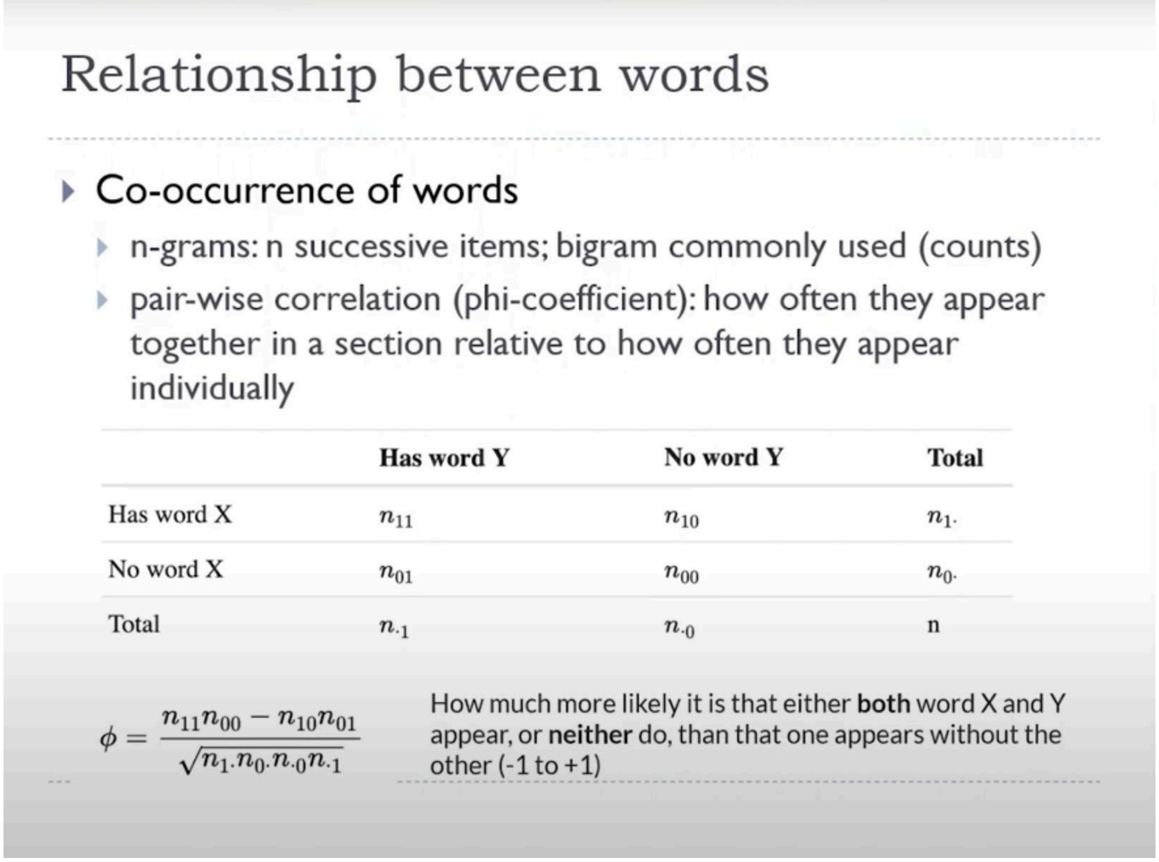
Now take a word like text, text mining. So, I am using or take them together. Take a word like text mining. You have 14 PPTs. In how many PPTs it may be there? May be the starting introduction and the last. Let us take it as i d f is 2 by, sorry it is, sorry, 14 divided by, 14 by 2. Number of documents on the top and 2 documents has this. Now for analytics, the multiplication factor is 1. For text mining, the multiplication factor is 14 by 2 which becomes 7.

Now do you get a sense of why you use the i d f? Analytics is a repeating common word. It is there everywhere, but text is a very niche or unique word and it is available only in one document and t f will give you how important is that word within that document, tf. If that word is very important in text mining, then it will be repeating, t f will have a high value and 7 is a high value compared to 1 and that is getting a lot more weight. It is getting a lot more weight, meaning a word which is specific to a document and it is very dominant in a particular document gets a higher weight and gets signified in t f, d f, in t f into i d f as compared to a word which is commonly occurring.

It does not have any niche or unique importance. So t f ,i d f is a measure of importance of a word in a particular document relative to other documents or in comparison to other documents. Do you follow the meaning of t f, i d f? It captures the importance of a word in a particular document relative to other documents. If it is unique to that particular document and repeating in that particular document, t f, i d f gets a higher value as compared to a common word which is spread across all documents. Now when you interpret t f, i d f, you should have this in mind. Well, this particular word is very important relatively in a particular document.

Decision trees, it is repeating in 1, but it is repeating many times in that 1 and it is unique. Therefore, it will have a high t f i d f. Now, we again have, any descriptive analysis like correlations. How do you understand correlation among words? Co-occurrence of words.

So co-occurrence of words can be by frequency. How many times two words are occurring in sequence? N gram count is an indication of how many times that particular word has occurred alone and that word has occurred together etc. By simple counts, you can actually assess it or measure it. And there is also a very unique measure of correlation like Pearson coefficient of correlation for continuous data that is used in text mining known as phi coefficient. Phi coefficient is equivalent to correlation coefficient in text mining and in the sense that it again varies from - 1 to + 1.



Relationship between words

- Co-occurrence of words
  - n-grams: n successive items; bigram commonly used (counts)
  - pair-wise correlation (phi-coefficient): how often they appear together in a section relative to how often they appear individually

	Has word Y	No word Y	Total
Has word X	n_{11}	n_{10}	$n_{1\cdot}$
No word X	n_{01}	n_{00}	$n_{0\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 0}$	n

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\cdot}n_{0\cdot}n_{\cdot 0}n_{\cdot 1}}}$$

How much more likely it is that either **both** word X and Y appear, or **neither** do, than that one appears without the other (-1 to +1)

Range is the same. The formula for calculating phi coefficient is given here and the way to understand these terms is explained in the table. Suppose, there are two words x and y. x and y are two words. This could be project management, project and management can occur together in sequence.

Project and management can occur separately also. What are these counts? Has word x, has word y, the count is is n 11. Has word x, but does not have word y is n 10. In a document, there is x, but y is not there. That count is n 10. In a document, there is both x and y that is n 11. n 01, x is not there, but y is there and both x and y are not there. Number of counts is n 00. These are individual counts that needs to be generated across documents and then you have column wise totals and row wise totals.

For example, n .1 is the sum of n 11 and n 01. n. 0 is the sum of n 10 and n 00. n 1. is the sum of n 11 and n 10 and n 0. is the sum of n 00 and n 01. Now, n is the total count of words or terms. Now, phi coefficient is given by (n 11× n 00 - n 10 × n 01)/ √ (n 1. × n 0. × n. 0 × n.1.)

Let us look at boundaries. That is a way of easily understanding. When do you get - 1? When do you get +1? And then you can apply, extend that logic. Suppose, there is a document or there is a corpus, a collection of documents where whenever x, then there is y meaning project and management, they always occur together. There is no occurrence of project without management and management without project. Whenever the two words occur, they occur always together.

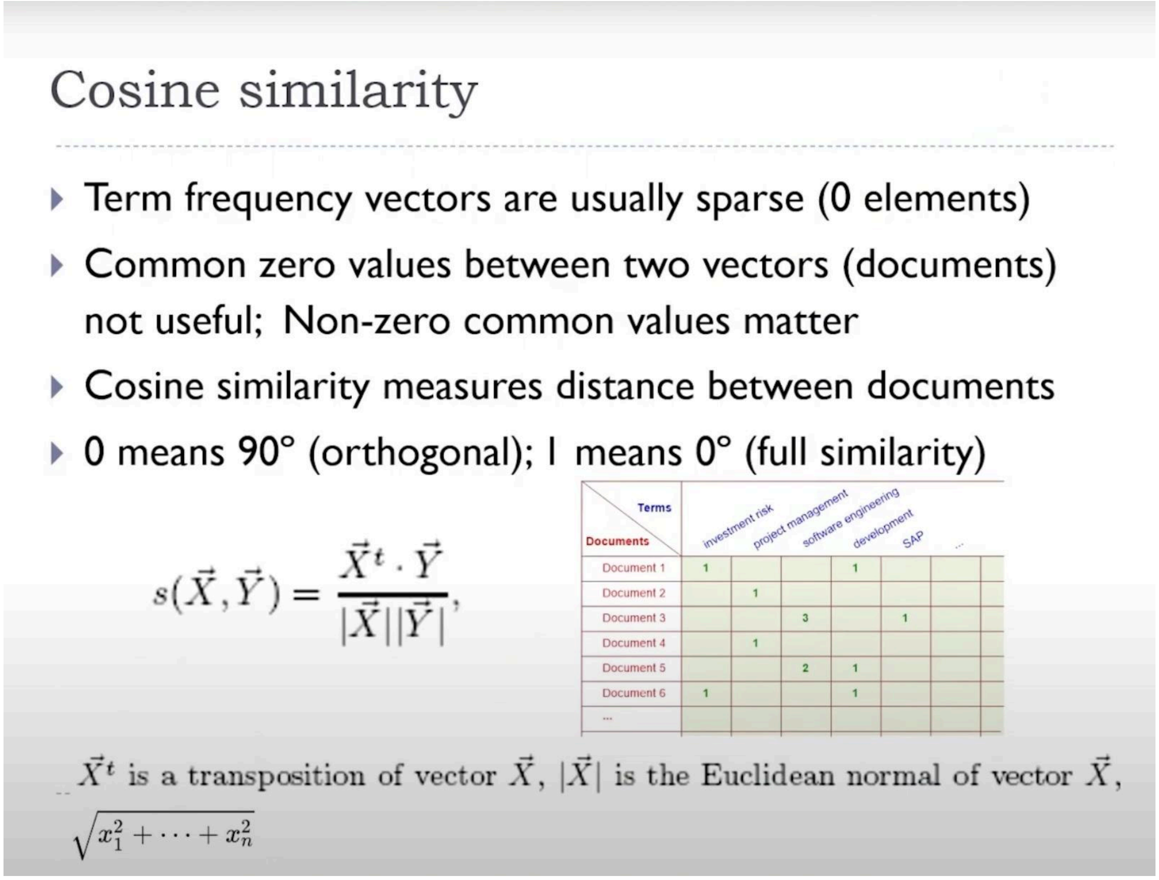
Either they occur together or they do not occur at all. Therefore, what do you think about n 11? That is the count how many times they are occurring together. n 00, number of times they are not occurring together. But what do you think about n.1, n.0, n 10, n 0. etc? What will be this value? There is no 1 without 0.

This will be 0. This will be 0. This will be 0. Sorry, not this 0. Sorry, this is 0. This is 0. Therefore, this one is equal to n 11. This is equal to n 00. This is equal to n 11. This is equal to n 00. Correct? And therefore, this is 0. This is numerator and denominator is also the same. You get when both the words always occur together, that is the highest correlation. They do not occur without the other. That is the +1 meaning that you have the highest confidence, that if one word is occurring, the other is also there.

The - 1 is the other. These two words never occur together. These two words never occur together and that is when this term becomes 0. There is no occurrence of the two words together. Assuming that there are no documents where they are not present, now this becomes a negative value because they do not occur together. And the denominator also then becomes same as the numerator, but numerator has a minus value and you get -1. So, phi coefficient is a measure of correlation between a pair of words and it varies from - 1 to + 1.

+ 1 indicating they always occur together, - 1 indicating they never occur together. And python libraries give you functions for phi coefficient and all of them. So, we will, python and R library. Today we are going to use R.

Now, we talked about similarity between documents. Or I want to see how similar is document 1 and document 2. Document 1 is a collection of sentences or words. Here it is bag of words and document 2 is also a collection of words. If document 1 and document 2 are the same, somebody exactly copied one's assignment and submitted.



Cosine similarity

- Term frequency vectors are usually sparse (0 elements)
- Common zero values between two vectors (documents) not useful; Non-zero common values matter
- Cosine similarity measures distance between documents
- 0 means 90° (orthogonal); I means 0° (full similarity)

$$s(\vec{X},\vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{|\vec{X}||\vec{Y}|},$$

Documents \ Terms	investment risk	project management	software engineering	development	SAP	...	
Document 1	1			1			
Document 2		1					
Document 3			3		1		
Document 4		1					
Document 5			2	1			
Document 6	1			1			
...							

$\vec{X}^t$ is a transposition of vector $\vec{X}$, $|\vec{X}|$ is the Euclidean normal of vector $\vec{X}$, $\sqrt{x_1^2 + \cdots + x_n^2}$

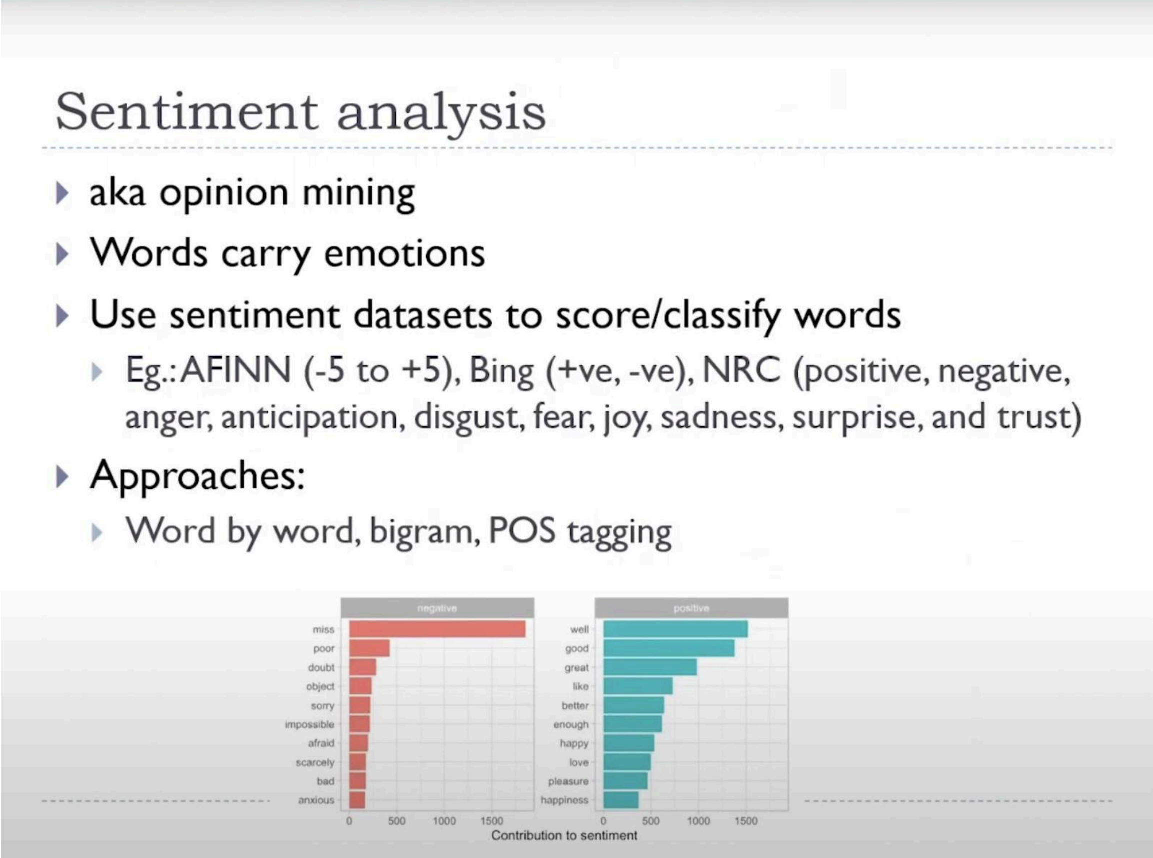
So, two assignments and not even a dot or a comma change. Then it is 100 percent similarity. So, 100 percent similarity technically should mean they are actually in the same plane, they are horizontal. Horizontal means the angle between them should be 0.

The cosine similarity actually means, sorry where am I going. When two documents are similar, they are the same, they are horizontal, that is 1 or cos of 1, sorry cos (0) = 1, right. Am I putting it in the opposite way? We will come back. Let us come back. Let us look at this ratio. This is the ratio that is used to calculate cosine similarity. There is, there is Y vector. For document 1 that is X vector, document 2 it is Y vector. So, the numerator is transpose of X vector× Y. So, that you can you find transpose. So, that you multiply, one vector by, these are two vectors. X transpose × Y divided by the Euclidean normal of X. What is Euclidean normal? Sum of squares of the elements of it × Euclidean normal of Y.

Suppose both x and y are the same. So, this numerator and denominator is the same. Numerator is equal to denominator. It is same as since X and Y are the same, same multiplying X × X or X × Y. So, this ratio becomes 1, meaning what is the angle between them? Angle between them is 0.

They are horizontal. Suppose they are totally different. There is no similarity between X and Y. So, therefore, when document 1, one cell is 1, the other is 0. When this is 0, that is 1. They are completely dissimilar.In which case, there we say they are orthogonal, 90 degree, orthogonal to each other.

So, therefore, it becomes 0. The numerator becomes 0 and therefore, the cosine of 90 is 0. Cos of 90 is 0. So, therefore, this becomes 0. So, essentially meaning when two documents are orthogonal to each other, they do not have any similarity and when they are fully similar, they are horizontal or they are the same. That is a sort of metaphorical meaning. I think we can move on and almost done with the lecture.



Sentiment analysis

- aka opinion mining
- Words carry emotions
- Use sentiment datasets to score/classify words
  - Eg.:AFINN (-5 to +5), Bing (+ve, -ve), NRC (positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust)
- Approaches:
  - Word by word, bigram, POS tagging

N

We will try work on a small problem subsequently and understand this basic concepts in lexical analysis. So, sentiment analysis also known as opinion mining is familiar with most of us today because there are sentiment analysis tools easily available and many of us do that using tools. Also known as opinion mining, is based on the assumption that words have polarity and that polarity can be measured or you can convert polarity of words into a scale. Polarity can be scaled or measured. Think of love and hate, two words. Looks like it is expressing some sentiment, some feeling and that word is, these two words are expressions of a sentiment, but they are opposite.

Love means positive, hate means negative. So, you can put this as a sentiment which is in opposite poles. So, love has a very positive sentiment or you may, if it is a - 1 to + 1 scale or a - 5 to + 5 scale, obviously you have to assume that this polarity should be obtained from some sentiment dictionary. They do exist in each languages. There are projects through which these dictionaries have been created, lexicons have been created and there are different such sentiment data sets which are available. One of them is AFINN which gives a score to each word, English word in a scale of - 5 to + 5 and there is Bing which classifies each word as either positive or negative. This is binary classification.

There is also NRC which actually has how many classes? Positive, negative, anger, 4, 5, 6, 7, 8, 9, 10, 10 classes. Words are classified into 10 classes based on the emotion they express from positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. These are different sentiments. So, each word gets classified into that and then if you analyze a whole corpus, you can actually understand what is the sentiment that is coming out of this corpus and then you can visualize it using different types of graphs provided by different libraries or softwares. And that is the activity in sentiment analysis, very useful in understanding what is going on in a discussion or in a particular text. Thank you.