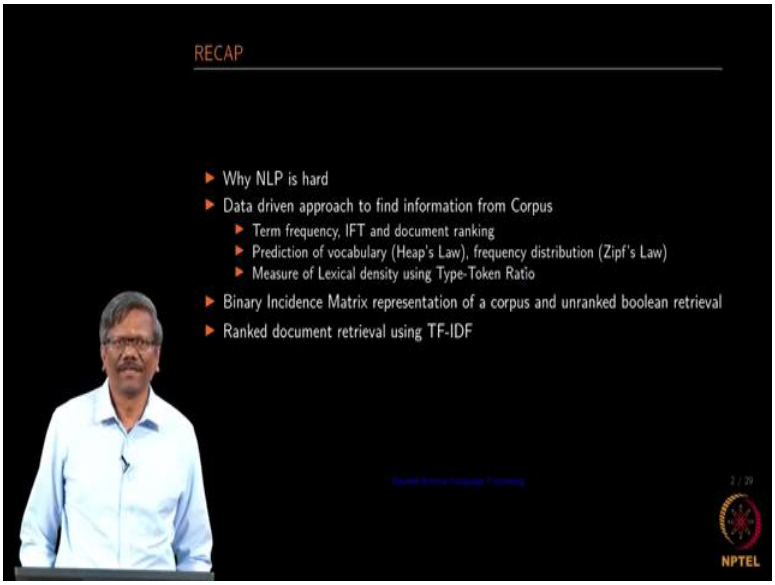
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Visiting Professor at Chennai Mathematical Institute
Indian Institute Technology, Madras

Lecture - 11
Vector Space Models for NLP

Today we going to be talking about converting all the words in a corpus into vectors. This is going to be a very important topic for all the lectures that we going to be hearing from now onwards.

(Refer Slide Time: 00:35)



So, before getting into the details, let me a first recap of what we have done earlier. In the earlier classes we saw how why NLP is hard, we gave some examples in terms of sentences and we wanted those sentences to be converted into a form that can be understood by the machine. So, even for a human need, you know it is going to be very hard to understand and sentences. So, how are you going to make the machine understand some sentences which are even hard for humans?

So, we are mentioned that it was pretty hard, so this NLP is a very hard problem, and then we started looking at the various corpus that was available to us with respect to the terms or the word frequencies. We have actually counted how many words are present in a given document and then we have introduced the term inverse document frequency. And, then we actually introduce some empirical formula or the laws that gave us some prediction terms of

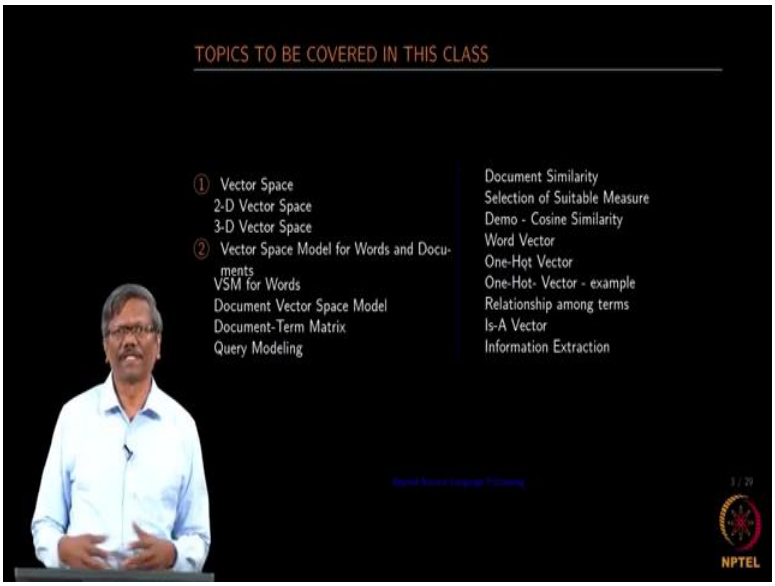
the vocabulary when you are given the corpus using the heap's law. And, then Zipf's law gave us some ideas in terms of the kind of frequency distribution of every word in an empirical way.

And then we also looked at another type-token ratio which measured the lexical density of a given corpus, using which you know you can find out what kind of vocabulary or how rich that particular document or the corpus is with respect to the vocabulary in English. And then we brought in one interesting concept by converting all the documents into a binary format; that means, every word a present in a document is given a binary value either it is present or not present. And, then we constructed one binary incident matrix for the whole corpus, and then we try to apply some binary rules for information retrieval.

And you know well that when you use binary operations it is not possible for you to rank the document within a corpus. So, we brought in the idea of TF IDF and how TF IDF could be used to rank those documents for all. So, you must have understood by now, we are looking at the corpus in terms of the data that is present inside with respect to the frequency of the words. And, then trying to find out how those frequencies could be used to retrieve the documents in the given corpus given a query and so on. So, this is going to be the path that we going to follow throughout this course, where we will be looking at the documents with respect to the data.

That means, the whole approach is going to be data-driven; that means, we are not going to be using a lot of linguistic ideas into this, we will be using a lot of statistics and probability ideas into the document and then see how those things could help us in terms of the understanding document. You also are aware at this point time that the idea of the entire exercise is to find the meaning of the word. So, we so far have not come to that level. So, we are now in the progression towards understanding the meaning of the word from the corpus without really looking at the dictionary and so on.

(Refer Slide Time: 04:30)



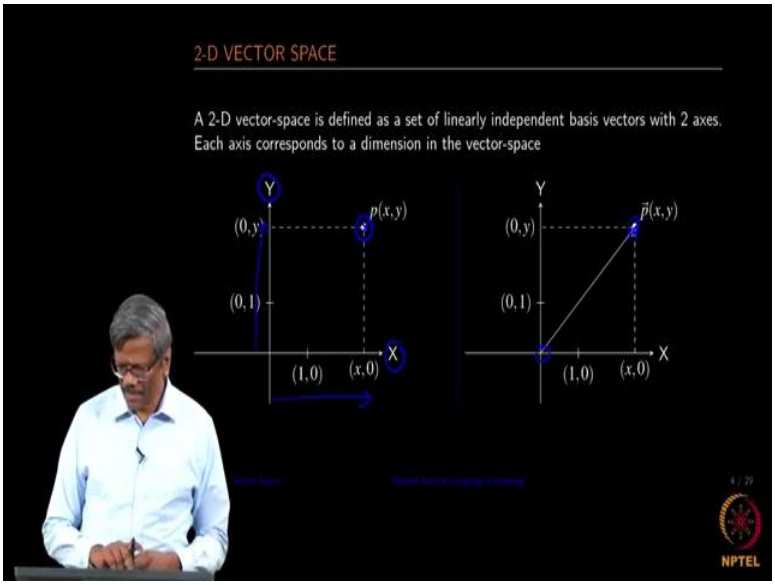
So, we will get to that part a little later, but today we will focus mainly in terms of how do I really convert those words into vectors, and then how do I use are available in this vector algebra to retrieve documents and do certain other operation and so on ok. So, what I will be doing today in this lecture is, I will just introduce the basic concept of the vector space; I am sure some of you are aware of this. For those who are not familiar with this I will just give a few minutes introduction to 2-dimensional vector space and 3-dimensional vector space and what we represent in the spaces.

And, then we try to map those vector space into the words and documents and how we actually map words as access in the vector space, and then how do you represent a document as a point in the vector space which contains a lot of terms and so on ok. Then we will bring in the concept of query modeling and then later we will talk about how do we really find out whether documents are similar. For example, if we have multiple documents in the given corpus, you want to find out the closeness of one document with the other. So, you have some measures that we will introduce in this class. And then we will also mention what could be a suitable measure if you want to find the document similarity.

We will look at one of the demos in the document similarity that would be on cosine similarity, and then we will introduce a what is a word vector really than represent them in terms of the one-hot vector. And, then later we will bring in the concept of how do we represent relationships among the terms with respect to is a relationship for some

composition based operations and so on. And finally, we will show one example of why this could be useful in terms of named entity extraction using the information extraction.

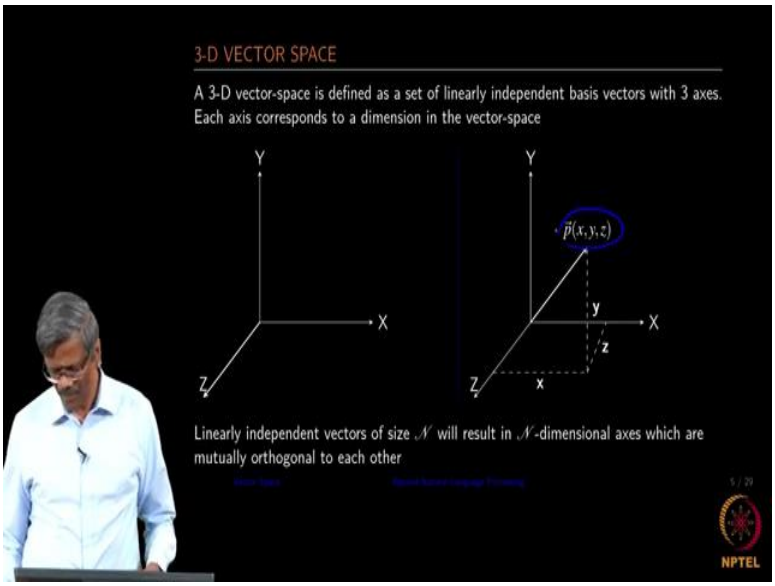
(Refer Slide Time: 06:45)



So, let us move on to the first slide in this exercise which is about a 2-dimensional vector space. A 2-dimensional vector space is defined as a set of linearly independent basis vectors with 2 axes, each ax corresponds to a dimension in the vector space. So, if you look at the diagram here, so we have a linearly independent vectors X and Y ok. So, what that means is, an operation that you perform using this particular linear basis vector we will not result in the value, it will always result in the null value; that means, they are in the vectors spaced in they are orthogonal to each other, they are in this fashion ok.

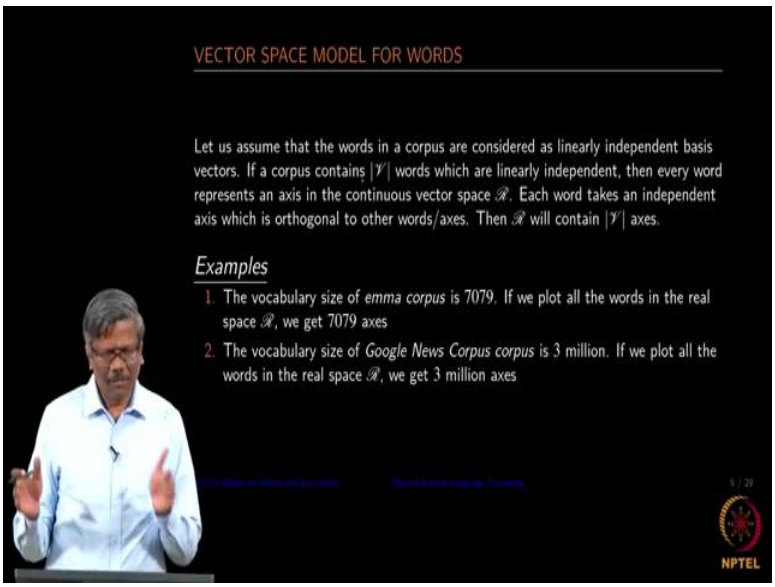
So, if you want to represent a point that corresponds to both these axes we just find out what is the value of that point with respect to the X-axis and Y-axis and then mark that point. So, that let us call that as a point p which measures X from the in the X direction and then Y in the Y direction ok. And then next what you going to do is you can go to define a vector in probably a familiar with, this vector is going to give you the position as well as the direction. So now, if you connect the 0 0 to the point, then now is as a direction attached to it, this is a vector. So now, p becomes a vector in this space; so, extending this to the third dimension.

(Refer Slide Time: 08:21)



We have one more axis is here, and then the same notation that I am using point is represented as three variables here; X, Y and Z, and then the vector in 3 dimensions is mentioned here ok. So; that means if you are standing in front of a corner of the three walls right and then your X Y Z would be here somewhere right. So, again if you look at this, all these basis vectors are independent and any vector operations, for example a dot product of X and Y or X and Z would result in a 0 value. So, we going to mark the same space vector space for words and then see how it looks.

(Refer Slide Time: 09:13)



So, now, let us assume that all the words in the corpus are considered as linearly independent basis vectors; that means, every word is different, they are not connected to any other word. Suppose if you are having about 300 words, all 300 words are independent and they have no relation to each other; that means if I do a dot product of word a and b that mean result would be going to be would be 0. So, again we will be using the notation of v this length of your vocabulary if you consider all of them as linear and then all the vectors related to the words in the vocabulary or linearly independent.

And they do not have a linear relationship with each other and they are represented in the continuous vector spacer. So, as I mentioned earlier again like in the case where I have introduced the vector space, every word now represents axes ok. So, earlier we had X and Y for the name's sake, now we going to be represented in them with respect to words; every word will occupy one ax. So that means if we have the size as v; that means, we are going to have v axes in that particular vector space. I am giving two examples here, I am sure you know that the vocabulary of any language is more than 10,000.

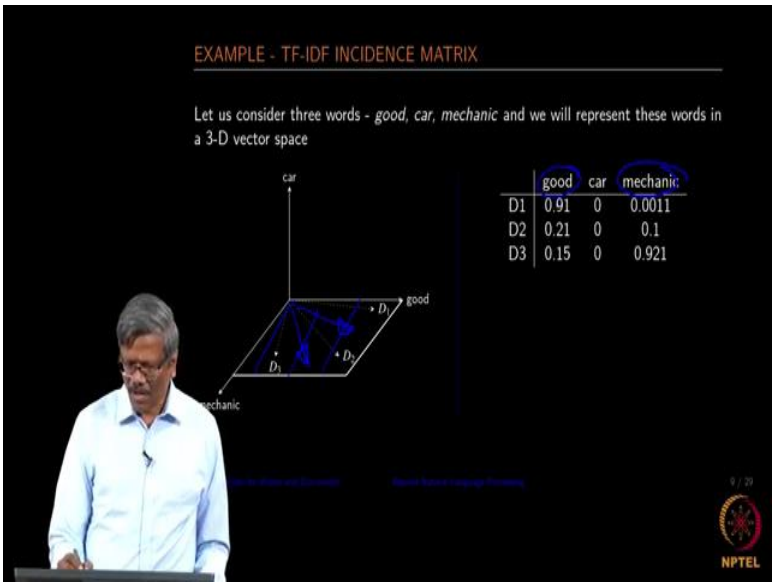
In the case of English it is more than a million or so for example if you take one corpus and then find out what is the vocabulary within that corpus you will have some number. For example, if you take the Emma corpus given from the NLTK platform, it is going to be about 7079. That means, this going to be 7079 axes in that vector space, it is very hot to imagine beyond 3, but this is how it is going to be in the natural language processing, if you want to represent all the words and terms off axes. If you look at the Google News Corpus it contains about 3 million words that mean we are going to have 3 million axes in that real space of R. So, it is very huge, this is very huge and it is very difficult to imagine that kind of axes for many of us ok.

(Refer Slide Time: 11:58)

So, how do I represent the documents? So, now, we know that every word is representing one ax. So, where is the document coming to place? So, you know the document contains a lot of terms right. Here again I am repeating the terms are either words or the combination of words or phrases ok. So, we will be using terms in general, in most of the cases here. So, the document that you are talking about contains lots of terms correct.

So; that means, a document is a point in that vector space occupied by the vectors or words in that particular document. Suppose if the document contains about 100 words; that means, the document is represented by those 100 points in that space ok. So, I mention already that combination of the terms represents document vector in that particular vector space, it is going to be very high dimensional and it is going to be very hard to really look at it from the 2-dimensional display that we have.

(Refer Slide Time: 13:19)



So, what I am going to do is, I am going to do just take a simple example where there are only 3 documents and these 3 documents are going to contain words which are car, good and mechanic ok. That means, good is represented in this axis, car in the usual Y-axis and then mechanic in the usual Z axes or Z axes ok. And, we have 3 documents here; document 1 contains all three words, document 2 contains only good and mechanic document 3 contains car and mechanic.

So if you want to represent the document 1 in the 3-dimensional space it is going to be here right and then if you want to represent good and mechanic, so it is going to be in this, between this 2 axes good and mechanic and then car and mechanics in the space correct. So, now, we actually created a binary incidence matrix for all of those and then represented them in the 3-dimensional space. In general these words are represented using TF IDF ok. So, what we are going to be doing is, we are going to be representing them in terms of these values of these vectors. So, this is your document vector and this is your word vector for all the documents in that space.

So, let me see how we can represent those documents. So, if you look at carefully the right-hand side of the slide we have a document 1, having the word good and mechanic. Document 2 again having good and mechanic, document 3 is again having a good and mechanic that means, I have going to only look at this particular axes for our representation of these documents. So now, document 1 is very close to the good axes right. So, if you look at the

value of document 1, it is pretty high when compared to what you have for a mechanic; that means, there is a good alignment of this document closer to the axes good rather than to the other space, but it is represented in the 2-dimensional space which is marked here ok.

So, I can represent a document 3 again, if you look at document 3 it is pretty close to a mechanic than to the axes good, maybe because there are more number of terms that are representing mechanic then good in this case. So, that is why D 3 is pretty close to the mechanic and then if you look at D 2 which has a fair amount of representation from good as well as from mechanic that is why it is somewhere in the middle. So, this is the actual space that we are looking at and all the documents that contain those two words would be represented in this space and the length of that particular document depends on how those frequencies are distributed in the document.