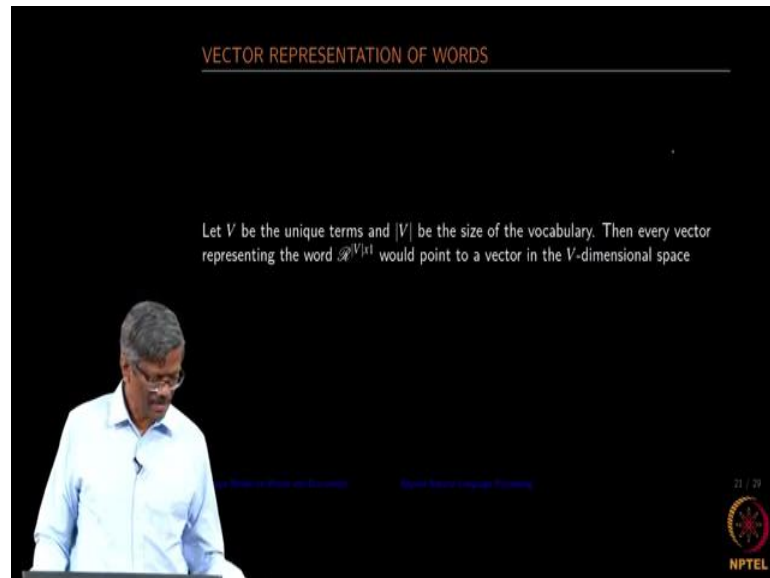


Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Visiting Professor at Chennai Mathematical Institute
Indian Institute Technology, Madras

Lecture - 13
Vector Representation of word

(Refer Slide Time: 00:15)



So, we have been only talking about the word with respect to the frequency with respect to TFIDF and so on, but how do I really represent a word ok. So, if I have a million words, I need to represent it in some way. So, can I use some kind of an indexing mechanism to represent it, or I use some other way? So, how do I represent that as a vector, is what you are going to be seeing right now?

(Refer Slide Time: 00:41)

ONE-HOT VECTOR - 1

Consider all the ≈ 39000 words (estimated tokens in English is $\approx 13M$) in the Oxford Learner's pocket dictionary. We can represent each word as an independent vector quantity as follows in the real space $\mathbb{R}^{39000 \times 1}$

$$v^a = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \quad v^{\text{back}} = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \quad \dots \quad v^{\text{zoom}} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 1 \\ 0 \end{pmatrix} \quad v^{\text{zucchini}} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 1 \end{pmatrix}$$

This is a very simple codification scheme to represent words independently in the vector space. This is known as *one-hot vector*.

NPTEL

So, what I have done is I have just taken a small concise dictionary where there are about 39000 words, it is not 13 million it's about 3 million in all. If you at Google's news corpus, there are about 3 million words. In the dictionary that I am considering there are only about 39000 words. I want to represent each one of the words given in the dictionary in the vector form. Is shown below

So, one simplest way to represent that is, there are 39000 elements in a vector-only one element of the vector is going to be one rest of them are going to be 0; that means, if you look at the first word in the dictionary a, that will be the first one and then the first element of that vector is going to be 1 and rest of the vectors are going to be 0. And then if you look at a back that is the second word in the dictionary. So, number 1 occupies the second spot in that.

And then if you look at zoom it is the last, but one. So, the vector is represented. All the other elements are 0's except this one and then if you look at the word zucchini that is the last word in the dictionary and it is represented by element 1 at the end. So, its a very very simple codification scheme to represent the words in an independent fashion ok. If you look at this you know there is no correlation between a back zoom or zucchini and so on ok. So, they are all independent that means they are orthogonal to each other in the vectors space ok.

(Refer Slide Time: 02:41)

ONE-HOT VECTOR - 2

In one-hot vector, every word is represented independently. The terms, *home*, *house*, *apartments*, *flats* are independently coded. With one-hot vector based model, the dot product

$$(v^{House})^T \cdot v^{Apartment} = 0 \quad (9)$$
$$(v^{Home})^T \cdot v^{House} = 0 \quad (10)$$

With one-Hot vector, there is no notion of similarity or synonyms.

The Goal of Word to Vector

- ▶ Reduce word-vector space into a smaller sub-space
- ▶ Encode the relationship among words

NPTEL

So, what you mean by orthogonal as I mentioned earlier? If you do dot product of two vectors that are orthogonal to each other the result and value would be zero vector ok. So, in this case, the terms, home, house, apartment flats or independently coded even though we know that now they are related to each other, but in this case, since we are using as simplified codification scheme called a one-hot vector, they are coded independently and then there is absolutely no correlation between these two. Even though we know well that they are quite similar to each other, unfortunately when in this case when you do a dot product what you get is a 0 here ok.

When two vectors are similar the value will not be equal to 0, it will be equal to somewhere between 0 and 1 ok. So, it will be greater than 0 and between these two values it will be available. So, if you want to represent these in the vector space you know for example, the 39000 words you are going to having 39000 access. So, it is very hard o represent all of them and then doing computations with respect to that access when you want to find the similarity of different documents it's going to be pretty hard.

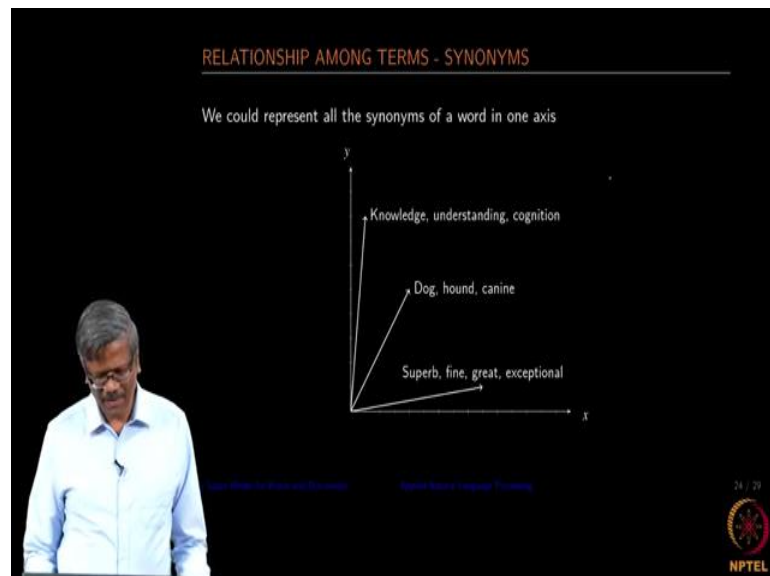
So, is there a way that we can compress it in a smaller fashion is there a way that I can represent those words so, that you know some words which are very similar to each other could be represented in a single access. This is something that we need to think about. Assumed that you know there are a million colors that are named we know only about 10

or 15 or 20 names of colors, but beyond which we do not know those colors, but those colors can be represented in displays in various forms.

Assuming that, we are able to represent those colors with names; that means, just for the color alone this going to be so, many million values or access in that real vector space which is going to be pretty hard right in terms of computation, in term of visualization all that. So, how do I really reduce those similar terms into smaller subspaces? How do I encode those relationships among words? So, these are the question that we need to answer in the natural language processing in order for us to move in the plane of translating a document from one to the other, create a question answering bot or creating a chatbot or anything that acts like a human in the natural language interface.

So, we need to really understand the relationship among words or the machine should be able to bring those words together in some form so, that its ability to do certain operations in the required fashion. So, the main aim of the word to vector is to reduce that a vector space from so, many million access to manageable ones and then also encode the relationship among a word when we do. So that is the dual purpose to the goal of the word to vector.

(Refer Slide Time: 06:30)

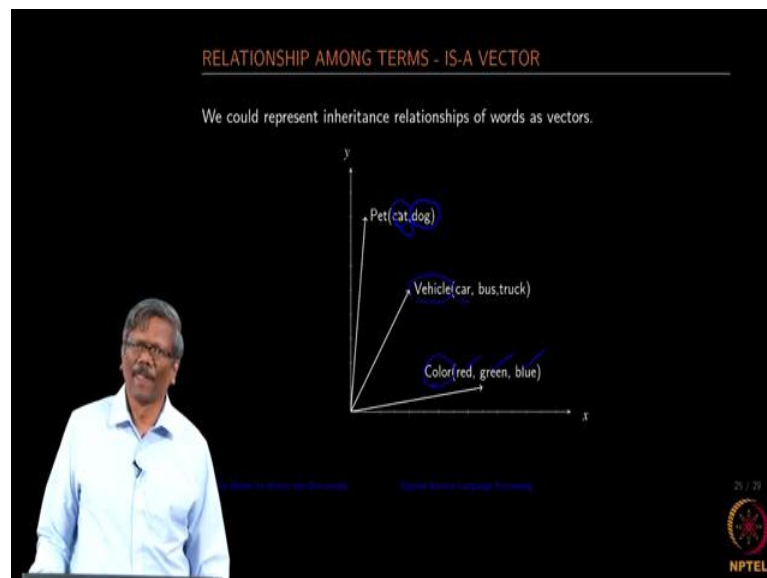


So, how can we represent them differently right? One way to represent them you know you know in an axis x and y is if you want to represent the synonyms in the same axis, you want to represent see for example, this superb fine great exceptional should be in

one axis. So, in this case you consider these as one axis. So, only for reference sake I have given x and y ok. Here these words are somewhat similar, but in one hot vector form they are orthogonal to each other, but we need to bring them as one axis and then we know that dog hound canine they are all the same right.

So, they should be represented on the same axis. And then if you look at knowledge cognition they should be represented in the same axis. So, how do we do that? So, by looking at these synonyms we can identify those synonyms of all those words and then represent them as the axis in the vector space. This is one way of doing, but for this you require thesaurus like wordnet to plot that particular axis. So, you require those kinds of a thesaurus to reduce the dimensionality of this space.

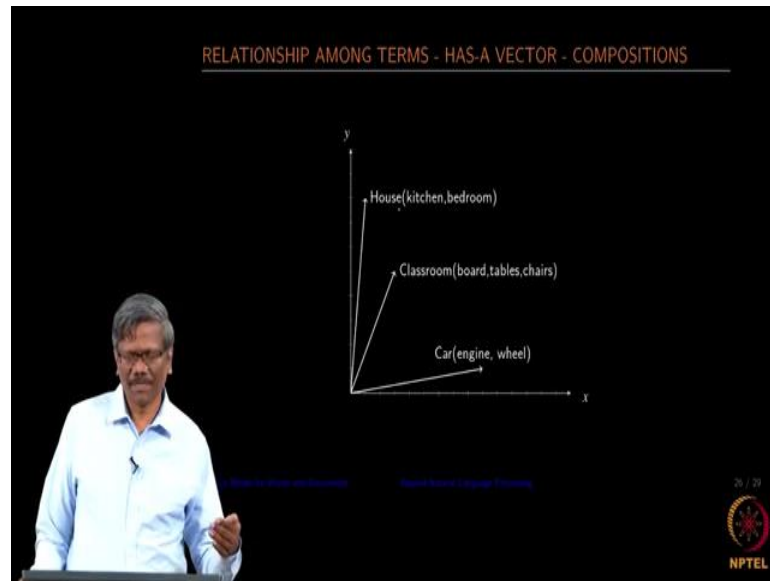
(Refer Slide Time: 08:13)



And then we can also use another relationship called IS-A vector ok. IS-A vector is something that actually tries to find out terms which can be represented by IS-A. For example, if you look at the color red is a color, green is a color, blue is a color. So, I can just use one axis to represent the color and then if you look at the other one car bus truck. So, I can represent them using one axis called vehicle, and then if you look at the cat and dogs you can use a pet cat is a pet dog is a pet.

So, by creating this kind of relationship, it is possible for you to reduce the vector space so, that the number of the axis that you were talking about will be reduced to a larger extent.

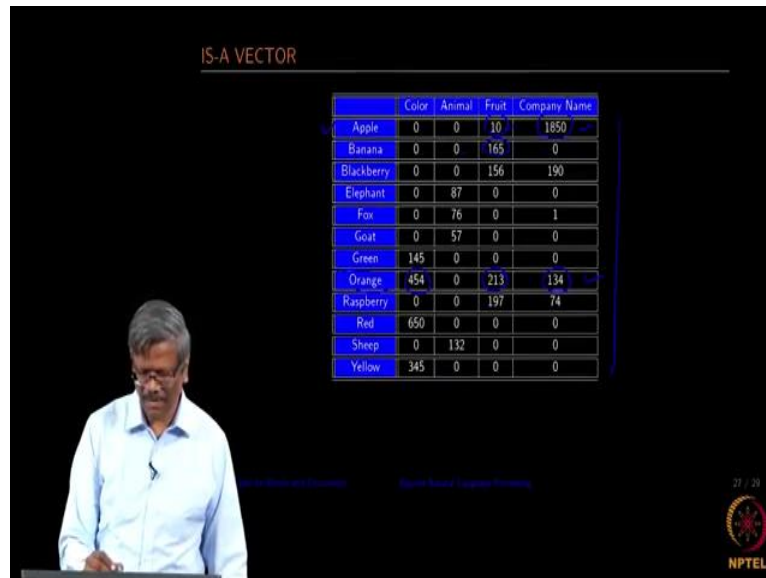
(Refer Slide Time: 09:14)



Another way is to look at it from the compositions ok. So, earlier we saw is a relationship, now we are looking at the as a vector composition. For example, if you consider a classroom you will have a blackboard, you will have a projector, you will have students, tables, chairs and so on. So, if you look at all of those words and then you can definitely say that this may belong to a classroom ok. So, that composition can be used as an axis ok.

So, each one of that right if you look at this synonyms or is a vector or the composition space; each one of them would be useful in doing the certain operations in the information retrieval process. I will just show one example of how this could be used let us take one example from the is a vector matrix ok.

(Refer Slide Time: 10:11)



IS-A VECTOR

	Color	Animal	Fruit	Company Name
Apple	0	0	10	1050
Banana	0	0	165	0
Blackberry	0	0	156	190
Elephant	0	87	0	0
Fox	0	76	0	1
Goat	0	57	0	0
Green	145	0	0	0
Orange	454	0	213	134
Raspberry	0	0	197	74
Red	650	0	0	0
Sheep	0	132	0	0
Yellow	345	0	0	0

So, assuming that I have taken all the documents in the corpus and then searched for words like apple, banana, blackberry, elephant and so on and then I try to represent those words with respect to color, animal, fruit, company name and so on let us take a look at the first one Apple.

Apple is not a color, it is not an animal, its a fruit, its a company name. So, if you go on and search Google for Apple most of the time in the first few pages you get only the details related to Apple the company ok. That is why I have just marked this; I very few documents you find in the first 10 pages of Google you find something related to fruit and then you search for banana It's not a color animal and so on. So, I am just creating a vector with respect to the matrix with respect to the IS-A relationship here ok.

So, in this way I have created my matrix why is this and what is the use of this particular one? Let us see once I have created this if I search using let us say Apple as my keyword. So, I am going to be getting all the documents that contain the name of the company initially and then if you look at the orange right. So, its the color, its a fruit and there is also a company by that name ok. If the company is not very popular the if the color has been used in all the documents many times, you will see the higher frequency of that, and then the probability of retrieving the documents with respect to color is more than the company name ok.

So, this is how I represent using the IS-A relationship with respect to all the documents in the corpus I have.

(Refer Slide Time: 12:50)

INFORMATION EXTRACTION USING IS-A RELATIONSHIP

A simple example of Named Entity Extraction

The Apple Watch has a completely new user interface, different from the iPhone, and the 'crown' on the Apple Watch is a dial called the 'digital crown.' A key quality attribute of apple is its peel or skin color, which affects consumer preferences. Immature fruits are green, and as the fruit ripens the green may fade partially or completely, resulting in very pale cream to green background colors.

The **org;Apple** Watch has a completely new user interface, different from the iPhone, and the 'crown' on the **org;Apple** Watch is a dial called the 'digital crown.' A key quality attribute of **org;apple** is its peel or skin color, which affects consumer preferences. Immature fruits are green, and as the fruit ripens the green may fade partially or completely, resulting in very pale cream to green background colors.

IS-A relationship

NPTEL

So, what can I do with that? Let us take a very small example there is a paragraph on the left side where you have the Apple Watch has a completely new user interface and so on so forth. And I run this application and then mark wherever Apple is present as an organization wherever Apple is present organization ok.

And then if you look at the marking here a key quality attribute of Apple is its peel or skin color. Definitely we know that it is not related to the company Apple, but because the frequency of occurrence of Apple is more in that particular vector, it marks this as Apple. So, this is the wrong classification, but it's fine. So, what is the use for this?

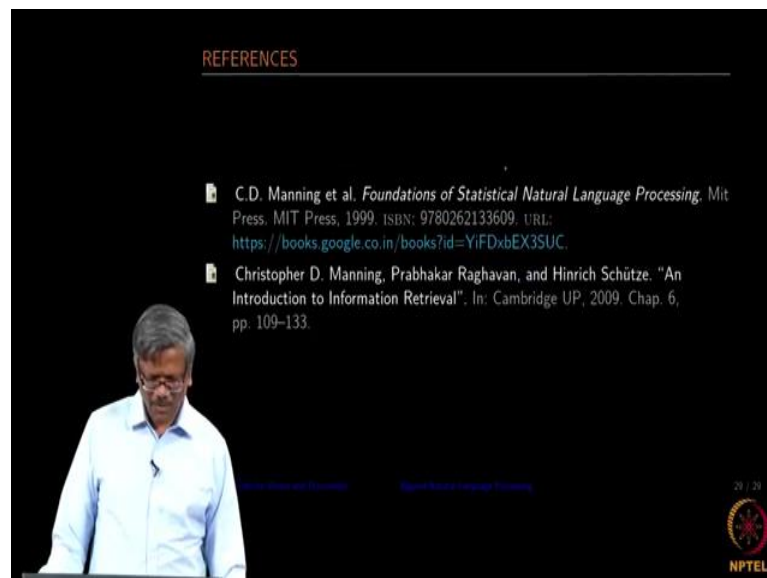
Supposing if I have taken a document set which contains all the terms related to Apple, all the terms related to the company Apple and then I want to extract some more information with respect to the name of the person who had given the presentation with respect new devices and so on. I want to find out what kind of technology Apple had used and when that particular document was released those things were available as dates and years and who actually gave that presentation.

So, if you listen to what I said you know we want to extract the organization names organization value we want to find out some dates, names, year and so on ok. If once I

have run through this particular relationship and marked all my documents in the corpus in this fashion, for all these words organization dates, name and year it is easy for me now to extract the name. So, this is called named entity extraction. So, once I marked, I run the program, get me all the documents related to these and then extract these keywords from those documents and also give me from which document you have extracted these words.

So using this you should be able to extract. So, the purpose of this one is to first mark the documents using this an IS-A relationship. So, I will just mark animal if there are animals found, I just write in front of that animal like this. So, this one of the use cases for this is to extract the named entities from the documents ok.

(Refer Slide Time: 15:50)



So, with this I conclude this presentation on the word to vectors, I have used to these two references for this entire lecture.