

Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture – 04
Vector Space Models

(Refer Slide Time: 00:16)

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.

If a corpus contains $|V|$ words which are linearly independent, then every word represents an axis in the continuous vector space \mathcal{R} .

Each word takes an independent axis which is orthogonal to other words/axes. Then \mathcal{R} will contain $|V|$ axes.

Examples

1. The vocabulary size of *emma corpus* is 7079. If we plot all the words in the real space \mathcal{R} , we get 7079 axes
2. The vocabulary size of *Google News Corpus* is 3 million. If we plot all the words in the real space \mathcal{R} , we get 3 million axes

NPTEL

The fourth one is to really understand what are the words and what those words convey right. So, that is something that we need to understand in order for us to go to the next level right. So, there we want to find out how can I really do certain mathematical operations on the text. Can I really convert the text into some vector form and can I use some vector algebra to do certain vector algebraic methods to find certain inner meanings or find the document that related to what I am looking for and so on?

To give you one idea let us suppose we have a document collection containing about billion words and about 10000 documents and I have a search mechanism to find what is there in the document. Let us assume that there is a search engine that is running behind and then I just give the word buy right. So, what it should do there is a normal search engine should go and then look for the word buy and then get me all the documents where this word has occurred right and then it will list based on some ranking. Let us not worry about the ranking right now.

Let us see that it listed certain documents in a certain order and then every document will have that word right. So, we do not want just to get to that level, we also want to find out a document that contains words and so on. So, we started with buy right, and then the rudimentary search engine gave me certain results where all the words, all the document that contains the word buy are listed in a certain order, but it would not list anything which are related to bought or if the documents contained these two words right.

So, what is the next step? So, is it possible for me to get these things done in a certain way? So, there is one simple way that we can do is we can use either a stemming or lemmatization let us talk about that, we will talk about the details of that little later. So, where when you give the word bought it will be converted to buying. So, you look at all the verbs in this fashion and convert them into the root, then you are normalizing the entire corpus wherever you have seen this word bought will be replaced by this word buy and buying would be this.

So, in that case it is possible for us to get all the results that contain the word buy. So, that is one way of doing it, but that is not very good, is it not? So, there could be a lot of document that will be incoming you know you cannot expect everybody to keep doing this pre-processing step and that is also not really nice you know it is not the way to learn the language and so on, especially when you are trying to make the machine learn this.

So, what are the ways? So, is it possible for me to use some vector space model to find out the distance between buying and bought and buying? So, if they are close enough bring all the words that are close enough to buy and then get those documents where those words are close enough to right. So, once you find this using some mechanism it is going to be possible for you to list all the documents which do not know list all the document that do not once we do that it is possible to list all the document that not only contains this word by, but it will also list all the documents that contain the words buy and buying correctly.

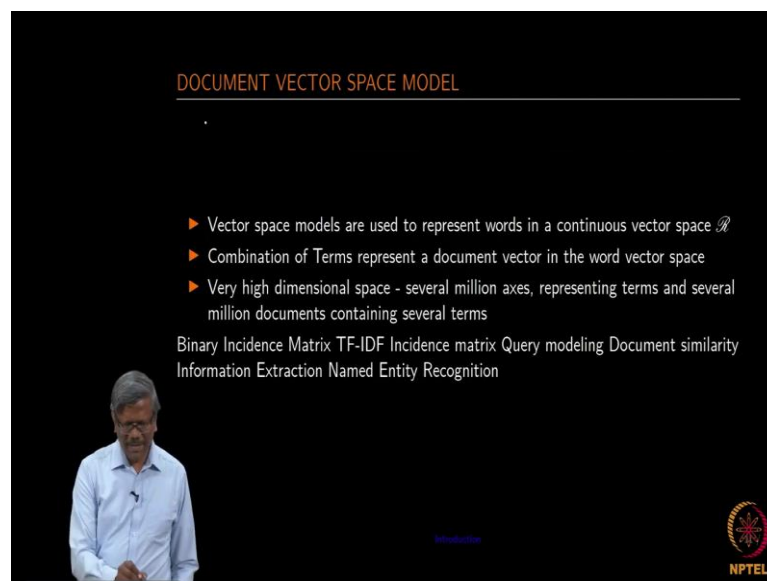
So, this is something that we want to achieve. Is it possible to really get those in a certain fashion? So, can the vector space model for words really help us? This is what we are going to be looking at in the vector space model. So, in the way we want to do is each word would be represented in one axis. For example, if you have about 7079 words as in

vocabulary and if you are representing the words in the vector form there will be 7079 axes because each one would be independent of each other, each word is independent.

So, that is one way of starting the process to convert your words into vector space model, and then later we need to start aligning those vectors in a way that if the vectors are close enough for a given word make them as one axis. So, we will see how that could be done a little later. Then if you have about 3 million words in the corpus you are going to have 3 million axes. That is pretty huge, right? So, it is very difficult to imagine beyond 3 axes for us.

So, we need to be able to do a certain operation in the vector space model to reduce the number of axes, at the same time bring in those words that are close enough into one axis. So, that is the main idea of the vector space model.

(Refer Slide Time: 06:53)



DOCUMENT VECTOR SPACE MODEL

- ▶ Vector space models are used to represent words in a continuous vector space \mathcal{R}
- ▶ Combination of Terms represent a document vector in the word vector space
- ▶ Very high dimensional space - several million axes, representing terms and several million documents containing several terms

Binary Incidence Matrix TF-IDF Incidence matrix Query modeling Document similarity
Information Extraction Named Entity Recognition

NPTEL

So, as I mentioned right we need to be able to plot them in the axis and try to minimize the number of axes so that we have fewer variables to deal with. So, we will talk about this in one of the lectures in the following weeks ok.

(Refer Slide Time: 07:11)

CREATION OF SEMANTICALLY CONNECTED VECTORS

- ▶ Identify a model that enumerates the relationships between terms and documents
- ▶ Identify a model that tries to put similar items closer to each other in some space or structure
- ▶ A model that discovers/uncovers the semantic similarity between words and documents in the latent semantic domain ✓
- ▶ Develop a distributed word vectors or dense vectors that captures the linear combination of word vectors in the transformed domain

NPTEL

So, once we are able to represent these words in a certain way, can we create semantically connected vectors ok? So, where we want to be able to find the relationship between terms and documents right. Like I mentioned earlier buy, buying, bought should be brought as part of one axis not three different axes. So, in that way we are connecting those words together. And, there could be some sort synonyms related to buying that also should be brought as part of that particular axis to which all of these synonyms and words should belong.

So, as I mentioned earlier we need to be able to put similar items closer to each other in some space or structure. The model that discovers or uncovers the semantic similarity between words and documents in the latent semantic domain. So, is it possible for me to translate from the terms and documents from one domain to a different domain so that I will be able to uncover the relationship? So, we look at that as one of the important aspects as we move along.

Develop distributed word vectors or dense vectors that capture the linear combination of word vectors in the transformed domain. For example, when you are transferring from one domain to the other what happens in the transform domain? So, the words are transformed into a different set of vectors, does it really convey the real meaning of the word that we see in the world right now which is in front of us? For example, the

document that you are seeing now contains the words and paragraphs. So, in the transform domain how do they look like right.

So, we will be looking at converting the words into word vectors or dense vectors. So, this is a very important step to do. If we do these kinds of operations we will be able to get into the transformed domain and then see the certain latent relationship between the terms and the documents ok. So, thereby we are learning what the word is all about. The word will not just represent that word alone. So, it will be looking at all the contextual words that are surrounding that particular word or the synonyms that the word has.

So, those are converted into a certain form that will be used for certain NLP applications. So, those we call as a word vectors or dense vector. So, we will go through this in detail, right now it might look very overwhelming; overwhelming right. When we go into the details of that you will understand the importance of each of this and they think you will understand the reason why we are converting them into a different domain and so on.

(Refer Slide Time: 10:56)

The slide is titled "WHY DENSE VECTORS?". It contains the following bullet points:

- ▶ Sparse vectors are too long and not very convenient as features machine learning
- ▶ Abstracts more than just frequency counts
- ▶ It captures neighborhood words that are connected by synonyms
 - ▶ Consider these two documents (1) Automobile association (2) car driver
 - ▶ Connects the neighbor of Automobile and the neighbor of car
 - ▶ "Automobile association" with "car driver" - driver and association could be connected using the similar words *Automobile and car*

The slide also features a speaker in the bottom left corner and the NPTEL logo in the bottom right corner.

So, I am going to skip this part right now. Maybe the last section of this slide I will talk about for example, if you have two documents assuming that only these four words if the first document contains the automobile association and the second one is car driver ok. So, these are the two documents that we are looking at. So, is it possible for me to connect these two documents in a certain fashion, can we connect automobile and car

automatically or can we connect the driver and the automobile association to which he belongs to as one?

So, these are a certain kind of operation that we want to perform on the corpus.

(Refer Slide Time: 11:52)

HUMAN/MACHINE LEARNING

- ▶ How do we solve problems when we lack sufficient knowledge?
- ▶ Finding Examples and using experience gained are useful
- ▶ Examples provide certain underlying patterns
- ▶ Patterns give the ability to predict some outcome or help in constructing an approximate model
- ▶ The model may help resolve some problems, though may not be an ideal one
- ▶ **Learning** is the key to the ambiguous world
- ▶ Linear and non-linear classification
- ▶ Perceptron, perceptron learning, cost function, feed forward neural network, back propagation algorithm

Handwritten annotations in yellow:

- A box around "lack sufficient knowledge?"
- Arrows pointing from "lack sufficient knowledge?" to "Finding Examples..." and "Examples provide..."
- A box around "Learning is the key to the ambiguous world"
- A box around "Perceptron, perceptron learning, cost function, feed forward neural network, back propagation algorithm"
- A separate box containing "Why →", "What →", and "How →" with arrows pointing to the right.

NPTEL logo in the bottom right corner.

So, starting again from the beginning right now we started looking at the counting part, and then we started looking at based on the counter can I do some prediction. So, now, we got a little deeper into understanding the world right in terms of the context in which it is present by creating a dense vector. Dense vector does not just represent that word, but it may represent the contextual word that is surrounding it or it may also represent the synonyms are associated with that ok.

So, once we have that information about the word so, can we make the system learn these associations ok. Do we have enough information to really convert the data into a certain form that machines can use and learn and understand? So, we do not have a lot of information even though we can talk about a huge corpus containing a billion words or one trillion words and so on. The reason being every time we create a sentence we always form a new and innovative one right. So, it need not be part of the corpus that contains about one trillion words or one billion words and so on. So, we are very very innovative in that fashion.

So, we always lack sufficient knowledge in order to feed into the system. It is very difficult to get everything that is possible as knowledge and then feed it as part of the knowledge base into the system right. So, how do we solve problems when we really lack sufficient knowledge right? We start reading more, we start listening to new lectures and so on right. And, then we start looking at examples and we start looking at the experience that we have gained. So, these are certain things that are helpful in terms of learning and then examples really provide certain underlying patterns.

So, based on what is already available in the corpus right so, those are the examples that are already there. So, we can look at some underlying patterns in those and then try to fill in the missing information using those patterns. So, these patterns give us the ability to predict some outcome; the pattern always would give you this ability right. So, we can always build certain models and then claim that this model would be able to solve certain problems, but the model will not be able to solve every problem. So, how do we keep creating new models based on what we have learned ok? So, that is where learning comes into the picture especially in the mission.

So, we keep providing more and more information to the system, and then it starts learning the patterns and probably creates a newer model every time. One example that I can provide here is I am sure most of you would have used the translation from the big companies that are around right. If you had used the translation 3 years back and then if you had used the same translation from the big companies today, you would see a huge difference in terms of the quantity of the translation. The systems are learning as they move along.

So, learning is one of the most important things to have in order to really work in an ambiguous world. So, keep learning from the examples. So, every day is an experience, every day there is new learning that comes into the picture. So, we keep changing the models you know. The model cannot stay static forever. So, the model keeps changing on a regular basis; especially the machine learns models are something that you have to keep changing on a regular basis unless the data does not change.

And, we look at various linear and non-linear classifications. For example, we spoke about the clustering earlier right. So, based on the learning we start refining our classifications in a better way. So, those are the aspects that we keep learning as we

move along, and then we talk about certain lower-level neural net models like perceptrons, its learning and so on. And, then move into the higher levels of feed-forward neural network and backpropagation algorithm.

So, why do we do all this? So, we spoke about the vector models earlier right. So, the neural networks that we are going to be looking at for machine learning would require the data in the numerical form in some way. Mission, the neural net models usually learn the patterns very well. So, some of the patterns it is very difficult for us to learn the system would automatically start predicting based on certain training algorithms that we provide to it. So, it is very important for us to do this step in terms of creating the vector space model for the words.

So, we can start feeding those vectors as input to the neural network, and then during the training process they start learning and then the outcome would be really really good you know if you learned the data and the patterns very well.

(Refer Slide Time: 18:32)

The slide is titled "WORD EMBEDDING" in orange text, underlined in yellow. In the top right corner, "SVD →" is written in yellow. The main content consists of three bullet points in white text:

- ▶ Process each word in a Vocabulary of words to obtain a respective numeric representation of each word in the Vocabulary
- ▶ Reflect semantic similarities, Syntactic similarities, or both, between words they represent
- ▶ Map each of the plurality of words to a respective vector and output a single merged vector that is a combination of the respective vectors

Below the bullet points is a numbered list of models, each with a checkmark:

1. Continuous bag of words (CBOW) Model ✓
2. Skip-gram model ✓
3. Discuss Word2Vec model ✓

Handwritten notes in yellow include "Why" and "What" on the right side, and "FIRTH → A word is known by the company it keeps." at the bottom. The NPTEL logo is in the bottom right corner.

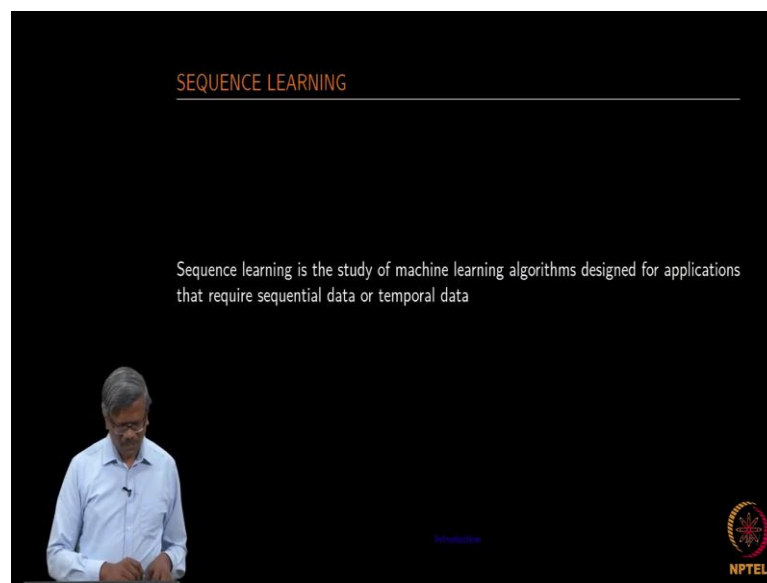
So, we spoke about the vector space models. Again, when you do the neural network-based application word embedding comes into the picture. So, whatever we spoke about as dense vector, again we do the same thing with the machine learning ok. So, we start creating the dense vectors using a neural network as well using certain patterns that we provide as input. So, using which, we would be able to find semantic similarities, syntactic similarities or both.

So, how do we do that is something that we talk. So, you remember we spoke about why. So, why is about why do we do this natural language processing and then I gave certain examples of what we do there right and then what are things that are required for us to do this in order to get to the answer of why and then how do we do that. So, these are all the word parts in terms of what we need especially the dense vectors, the probability that we spoke about they are all from the word part and then how is about is all about how do we really implement that.

So, what are all things that we have to do in order to implement that, and that would answer the question of why do we do. So, look at this as the most important thing you know every time when you do certain things try to answer these three things ok. So, in the entire course we are trying to answer all these three questions ok.

So, we spoke about this right in terms of the word vectors. So, we can use a mechanism in the linear algebraic space using a method called SVD, singular value decomposition. So, that also gives you a dense vector and using neural net also you can do the same thing. So, the neural net provides a better word embedding than this really in many cases. So, how do we do that? So, we have various models using which we can convert the words into dense vectors.

(Refer Slide Time: 21:24)



So, using the dense vectors it is possible for us as I mentioned to understand the semantic similarities, syntactic similarities, or both. For example, also remember this quote. So, it

is the quote from Firth. So, what he said was the word is known by the company it keeps; that means, the word always has certain contextual words surrounding that. And, if the context word is the same, but the central word is a little different then we can say that those two words are somewhat similar in the semantic way. So, that is exactly what he means.

And, the neural net model exactly captures these patterns in a certain fashion and then the word embedding is created or the dense vectors are created which we can use it for later applications.