

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

Lecture - 15
Co-occurrence matrix, n-grams

(Refer Slide Time: 00:15)

CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example: Consider the following short documents:
1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0
hates	0	0	0	1	0	1	0	0	0
Maths	0	0	0	0	1	0	0	0	0
She	0	0	0	0	0	0	0	1	0
loves	0	0	0	0	0	0	1	0	1
Biology	0	0	0	0	0	0	0	1	0

11 / 25
NPTEL

Let us see how this could be formed ok. Again instead of just saying those in words, now we have to convert them into respective matrix forms so that we can utilize the similarities and the co-occurrences patterns that exist in a given corpus. in this case, I have considered three documents, one the first document is I love physics, the second one is he hates maths, the third one is she loves biology.

So, here we are going to be looking at the co-occurrences of the word for each of the words that we are looking at. let me start with the first one ok. what I have done is, I have listed all the words along the column right that the column header now. And then if you look at the left side, you still have the same words on the first column right here ok.

So, now what I am going to do is I am going to be looking at the first word I and then see what is the next word to I. So if you want to fill up this particular matrix, let me go back to the first one. I have here. I and I do not co-occur correctly. what you have is 0 here ok. I am using only a binary representation ok.

So, now I love physics. we have loved following I. we have a 1 here ok. And then I and physics right. they do not occur together. we have a 0 here. in this way, I have to keep filling all the elements in the given matrix. he and I do not occur together. He and hates it should be let us that is it the second one, I am sorry that is the second document I have not yet finished the first one.

So, let us take the second word, love ok. And then I and love occur together, love and love do not occur together. love the next word for love is physics. there is a 1 here. And then love and he, 0; hates, maths, these are all 0, because they do not co-occur, love and those words do not co-occur. in the same fashion, physics, you, love, love and physics you have, and then physics and physics do not occur together, and the rest of them are all 0 because they do not co-occur right. And then the next document he. he and I, he and love, he and physics, they do not co-occur. you may ask this question. you can use this right in terms of either.

(Refer Slide Time: 04:00)


CO-OCCURRENCE MATRIX

A co-occurrence is a combination of terms that are likely to be used in the same context. A co-occurrence matrix stores co-occurrences of words. The count of a pair of words that appears in a context window is represented as an element of a matrix.

Example: Consider the following short documents:
 1. I love Physics 2. He hates Maths 3. She loves Biology

	I	love	Physics	He	hates	Maths	She	loves	Biology
I	0	1	0	0	0	0	0	0	0
love	1	0	1	0	0	0	0	0	0
Physics	0	1	0	0	0	0	0	0	0
He	0	0	0	0	1	0	0	0	0

love
loves



So, if you noticed I have loved here and then love. I am not doing any stemming or lemmatization, I am just using them as one word. this is different, and this is different ok. going back to the second document, so there is nothing that occurs together between him and the first document, he and the 0. He and hates, they occur together. Maths no, again third document. again the second word in the second document, they were co-occurrence

word is this another one is maths; the rest of them are all 0 because they do not occur together ok and the third one, so we have maths and hate right.

So, in the same way, you keep filling them ok, and you get a matrix of this. this is called a co-occurrence matrix. it is very similar to the binary incident matrix that you saw. The name is the same, but in this case instead of the document on this column earlier we have the words. the words and the words, and then the co-occurrences of those words are plotted or picked up, and the values are filled in as 1, they co-occur; otherwise the element is 0. I hope you understand this right.

(Refer Slide Time: 05:43)

The slide features a black background with white text. At the top, the title '(UNIGRAM) BIGRAMS, TRIGRAMS' is displayed, with 'UNIGRAM' circled in yellow. Below the title, a list of three bullet points defines n-grams: a sequence of two words is a bigram, a three-word sequence is a trigram, and an n-gram is a sequence of words of length n. To the right of the text, handwritten yellow notes list 'n-gram', '1-gram', '2-gram', '3-gram', and '4-gram'. In the bottom right corner, there is a small red circular logo and the text '12 / 25' and 'NPTEL'. A small inset image of a man in a light blue shirt is visible in the bottom left corner of the slide area.

So, now let us look at some interesting definition, the earlier you know I just mentioning about two words or three words and so on, but there is a separate name for each of those. Unigram is something which is related to one word. if you are using one-word extraction, then it is a unigram extraction. we have been looking at unigram all throughout before this lecture. Now, we are going to be looking at two words together, three words together or n-words together, and we call them grams ok. If you use two words together, then it is called a bigram ok.

If you word three-words in a sequence, then it is called a trigram, or in general you can name this as n-gram. we can call it as 1-gram is your unigram; 2-gram, a 3-gram, 4-gram etcetera. we will be using this term n-gram throughout this lecture ok.

(Refer Slide Time: 06:57)

N-GRAMS

Consider the tongue twister as four documents:
1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers, 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
< s >	< s >Peter	< s1 >< s2 >Peter

Diagram illustrating a sliding window of size 2 over the sentence "Peter Piper picked a peck of pickled peppers". The window is shown as a yellow box containing two words at a time, moving from left to right. Handwritten labels <S> and <E> are present above the diagram.

13 / 25
NPTEL

So, how do you really create the input for us? the reason why we are doing this is we are going to be using those n-grams as input for various NLP applications. we should be able to create those n-grams as input, and then remove whatever we do not need to really keep, and then use the rest of them for or as an input to either a neural network or any other NLP application ok.

So, now, let us look at the force document, but I will be using only this particular document now to construct our n-grams. in this case, I have three columns. The first column will be talking about or showing you the unigram; the second one will show you the bigram, and third will show you the trigrams. For every sentence there is a start symbol. you will use that start symbol as our starting a word ok

The starting word is not the first word that you see, but it is a starting symbol S, and then the ending symbol also we use this symbol ok. let us look at these n-grams one after the other. when you look at the 2-grams, so what you are going to look at looking at is, so we will take a window of size 2 ok. Consider this window and you are going to take this window, and then place it on top of the sentence that means this particular window is flexible enough, so that it only shows you two words at a time ok.

So, when you place it on this let us say you are going to place it here ok, or we going to place it here, or we may place it like this, this is our window. And slide this window along the sentence one have one word at a time. I just move Peter Piper first, keep it on

top of Peter Piper, and then move it, so it becomes Piper and picked like this. it is flexible enough to fit only just two words when I use diagrams.

In trigrams what is going to happen is I am going to use three like this it is a sliding window. when I place this window on top of this sentence, three words will appear within that window. And then I move to the next word that means the next two-three words will be seen and so on. every time when I move, I pick up my input words ok. Let us see how it is done. in this case of unigram, let me remove all this to remove the confusion.

(Refer Slide Time: 10:22)

N-GRAMS

Consider the tongue twister as four documents:
 1. Peter Piper picked a peck of pickled peppers 2. A peck of pickled peppers Peter Piper picked. 3. If Peter Piper picked a peck of pickled peppers. 4. Where's the peck of pickled peppers Peter Piper picked?

Unigrams	Bigrams	Trigrams
< s >	< s >Peter	< s1 >< s2 >Peter
Peter	Peter Piper	< s2 >Peter Piper
Piper	Piper picked	Peter Piper picked
picked	picked a	Piper picked a
a	a peck	picked a peck
peck	peck of	a peck of
of	of pickled	peck of pickled
pickled	pickled peppers	of pickled peppers

13 / 25
 NPTEL

So, we use a starting symbol as I mentioned. The first one let us look at only the unigram part which is very well known to us. You go through that you see picking up only one word at a time, and then it ends at the peppers right. We have seen how unigram words are constructed, and those could be used as input for some applications. Now, let us see how bigram words could be constructed.

So, as I mentioned earlier bigram will have a sliding window, and it will show you just two words when you move that along the sentence. in the first case, we have a starting symbol. it will start with the starting symbol and peter. And then we move the window to the next one so that I fall on top of or it slides on top of Peter Piper. we get Peter Piper, and then you move on to the next one piper and picked as the next two words or bigram, and then picked as the next bigram and so on ok.

So, in this way, you can complete the bigram operations. it is very bigram is very similar again. in this case, since we do not have you know when you want to place this, you want to include every word right, so I do not want to start like this. What is the most commonly used to starting word of a particular corpus you know you may want to get this particular starting symbol used so that it is easy for you to get that particular word as the starting word?

So, in this case I have two different starting symbols, and then I start with right. And now slide these three windows along with the ok. you keep getting those, I am sorry it was the previous ones let me remove that ok. Peter Piper picked my window is nowhere. now when you move the window to the next one, so you have right. you can slide that particular window in all the sentences so that you capture all the trigrams of the sentence or trigrams of the document or trigrams of all the corpus.

So, why do we need this? as I mentioned earlier the co-occurrences of words are important for you to really understand the meaning of the word. in this case, you can capture a bigram, trigram or 4-grams depending on what would be a convenient size ok. As you move along the numbers, the more number that you traverse with respect to the window size, your accuracy would be better, but the complexity is going to be increasing ok.