Learning Analytics Tools

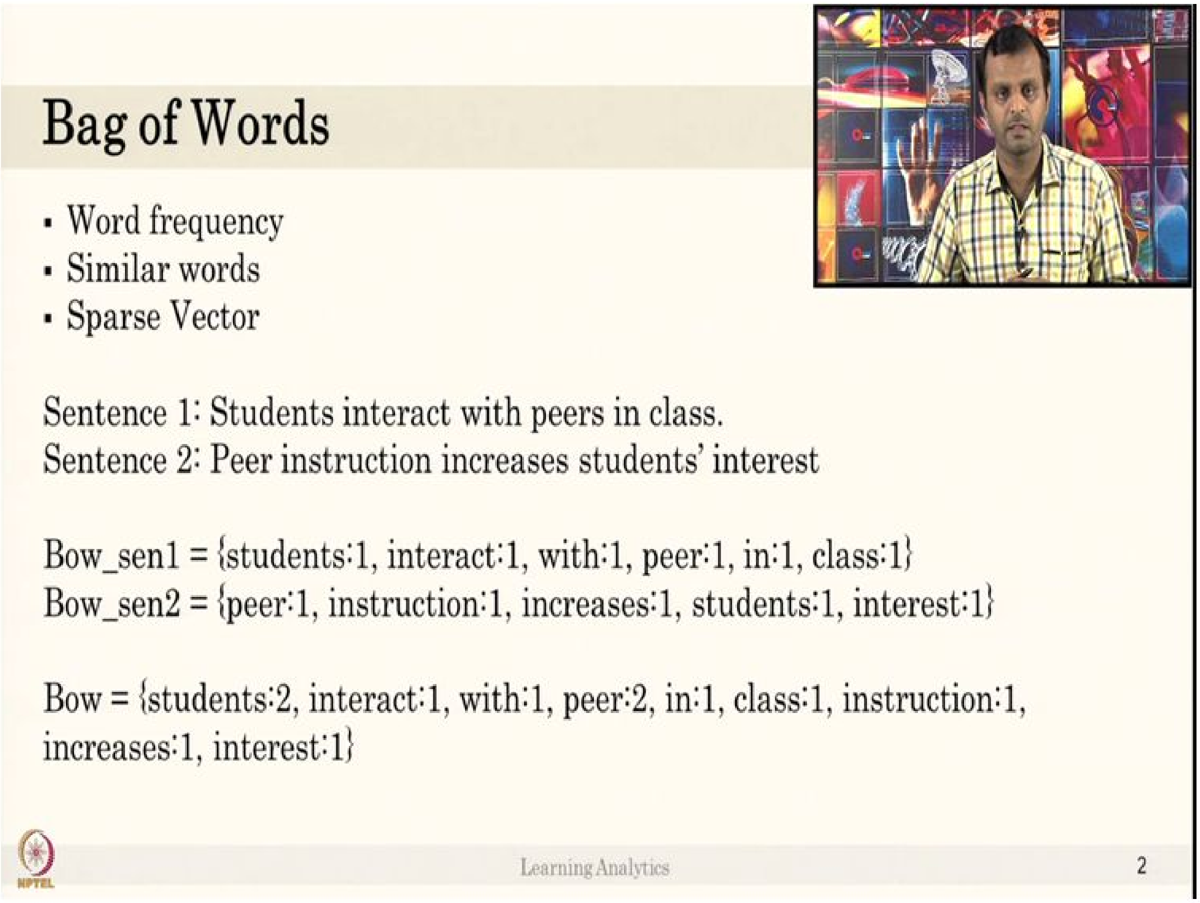
Professor Ramkumar Rajendran

Department of Education Technology

Indian Institute of Technology, Bombay

Lecture 10.3: NLP - II

(Refer Slide Time: 0:29)



Bag of Words

- Word frequency
- Similar words
- Sparse Vector

Sentence 1: Students interact with peers in class.
Sentence 2: Peer instruction increases students' interest

Bow_sen1 = {students:1, interact:1, with:1, peer:1, in:1, class:1}
Bow_sen2 = {peer:1, instruction:1, increases:1, students:1, interest:1}

Bow = {students:2, interact:1, with:1, peer:2, in:1, class:1, instruction:1, increases:1, interest:1}

Learning Analytics 2

In this video, we will talk about a bag of words and how we can use that to create an automatic grading of SA. So, what is a bag of words, it is simple. It picks each word how many times it occurs, so we can put that in the bag of words. And this also can be used to find similar words as we saw How to find similar words?

I will talk about what is sparse vectors in detail. So, firstly remove the running letters, remove - in, an, articles, prepositions, conjunction words. It is up to you how much you want to remove it and correct the sentences, spelling using the similarity words. That is very important. And you have to create a dictionary, that dictionary should have all the words with the frequency of words occurring in a dictionary.

Let us look at what is a bag of words in detail in this slide. Sentence one says

"students interact with peers in class, students are interacting with peers in the class. Peers instruction increases students interest."

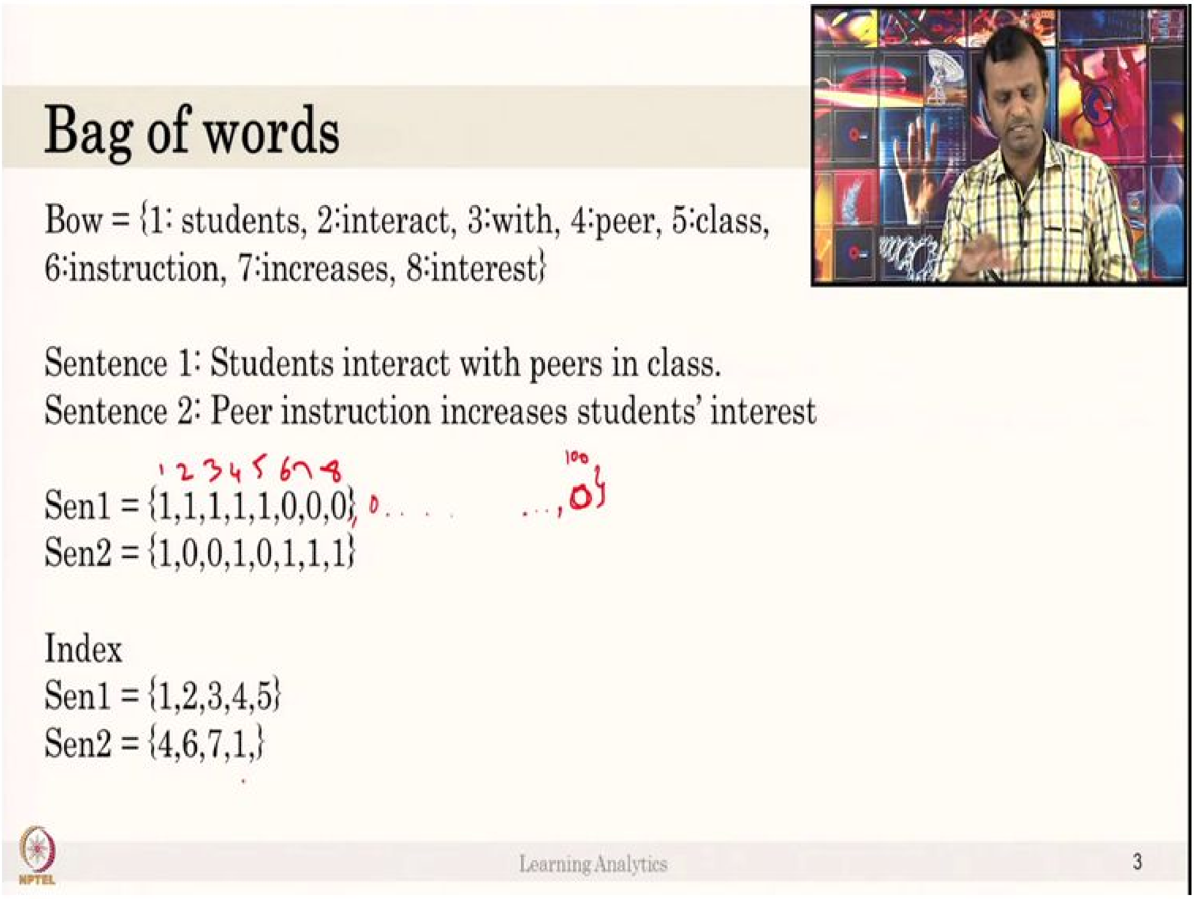
Something like that. Let us say the bag of words of sentence one. The word - student occurred once, Interact occurred once, with occurred once, peer occurred once, in occurred once, class occurred once. That is a bag of word of sentence one.

In the Second sentence, peers occurred once, instruction occurred once, increase occurred once, students occurred once, interest occurred once. See, I just wrote a word in a form given in the sentence. What you can do is you can use lemma or stem from of it. So, the root form or stem form of it that reduces this, the length of this vector, I will talk about how big the vector is going to be. It is not easy.

So, that also possible. Let us look at that. So, if I have two-sentence, I want to create the one bag of word for these two sentences student occurred twice, to one time here, one time here, interact occurred once, with occurred once, peers occurred twice. "in, class, one" - all of them have just occurred once. So hope you understand this particular set I created from these two sentences.

That is very very basic. That is, that is all about bag of words, you have to understand this, how this set has been created from these sentences. I just counted the frequency of each word, all the individual words, the unique words in the sentence, I just counted how many times it occurred that is it, frequency of words occurring in the thing.

(Refer Slide Time: 3:16)



Bag of words

Bow = {1: students, 2:interact, 3:with, 4:peer, 5:class, 6:instruction, 7:increases, 8:interest}

Sentence 1: Students interact with peers in class.
Sentence 2: Peer instruction increases students' interest

1 2 3 4 5 6 7 8 100
Sen1 = {1,1,1,1,1,0,0,0}, 0 . . . . , 0}
Sen2 = {1,0,0,1,0,1,1,1}

Index
Sen1 = {1,2,3,4,5}
Sen2 = {4,6,7,1,}

Learning Analytics 3

So, I have a bag of words, for the two sentences. Now, I have the same two-sentence consider this is my dictionary or this is a complete set I have, the set has eight words. The first word is student second word is interact. I removed "I", I removed "the", what you say, I removed the word "in". What I did, I gave a position for each word. Instead of taking the word and putting its frequency, I have a dictionary my dictionary has only eight words, just eight words.

Students, interact, with, peer, class, instruction increases, interest. That is it, just eight words I have in my dictionary. The sentences coming out of this dictionary. I need to find I need to create the numerical form of the sentences. If I have this dictionary, this, I can say "students" is at the first position in this dictionary. So the second word is interact. It exists in sentence one. Yes, it exists. If it exists, mark it as one, if it does not exist mark it as zero.

With exist in sentence one. Fourth one peer, is peer exist in this sentence. Class exis. Instruction, do not exist. The sixth word is zero. Increases, not exist. So my vector of sentence one is

[1,1,1,1,1,0,0,0]

So, if someone asked me, can you form a sentence. The sentence has students, interact, with, peer, class. That is it.

That is all the words this particular sentence has. The order is not important. The student occurred first, I will tell you why - that is very important. But what are the words from the dictionary exist in this? Sentence one has five words from this dictionary, the position of the word is also given. So, it is easy for machine learning to classify because of numerical values like one or zero. Sentence two have "student" in it.

So, the student is here, (it is not the first position) that is what I am saying, the order or position is not important in a bag of words. Student is one, interest is not there, hence zero at this position, with is not there, peer is there. And class is not there and "instructions, increases, interest" are there. So, it is simple. What I am trying to explain is very simple, so one. I am just putting the order. One, two, three, four, five, six, seven, eight.

Hope you understand what I am talking. So, it is basically which words are occurring in the sentence. Consider we have these kinds of sentences say five-sentence. So, if you combine the word, you might get the bag of word dictionary of thousand, or sometimes hundred thousands or sometimes it goes to even million because if you consider all the forms of the words, the dictionary will go big. The dictionary goes to all the existing words in English, plus different forms they occur. It is very, very huge.

So, this can go to a big vector. But let us take a thousand words. If it is a thousand words dictionary and your students write a sentence like this. So, consider there are six words in this sentence and a lot of zeros up to a thousand. This vector is called a sparse vector. Out of thousand, only five words are there, so it is a sparse vector.

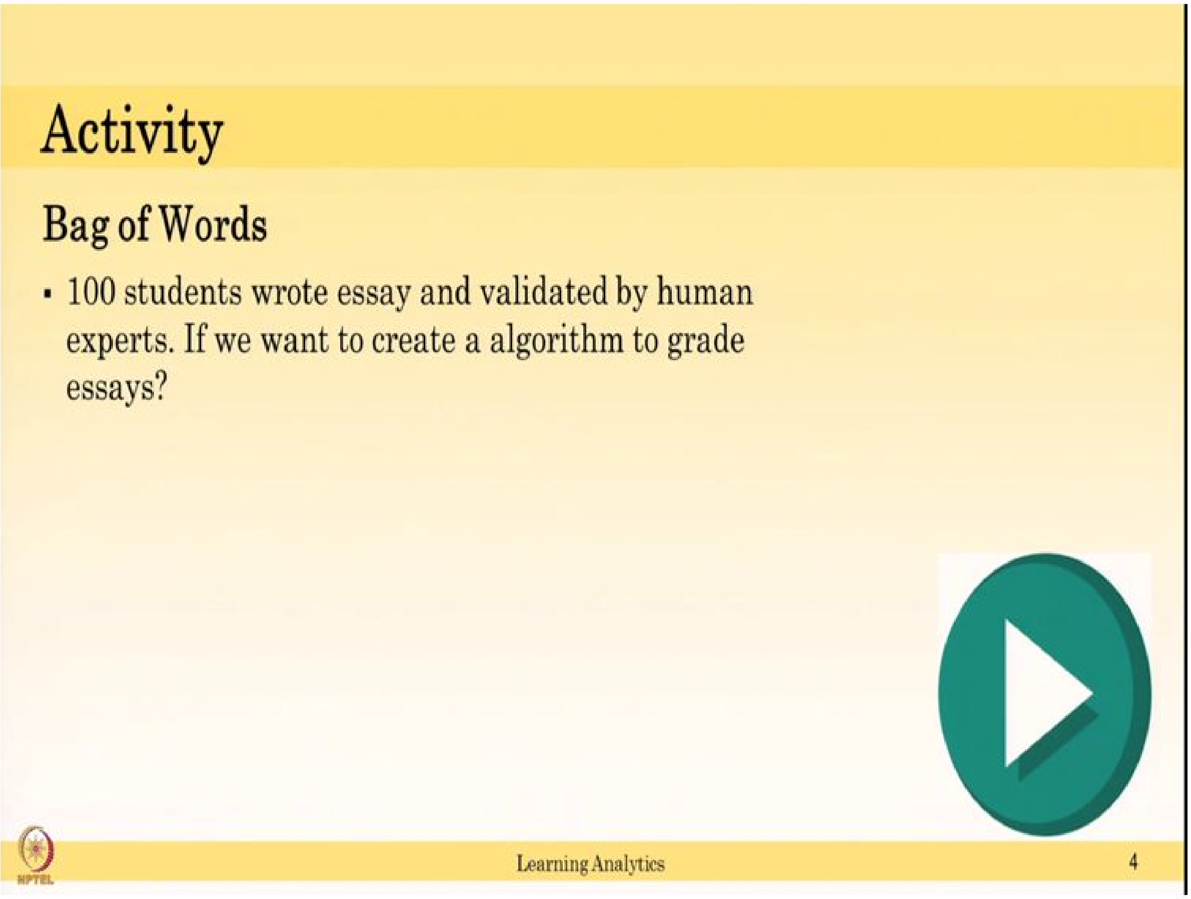
So, in order to avoid the sparseness in the vector or the sentence formation, we can also use the index instead of the particular position you can mark the index. For example, student occurred in the index number one, interact occurred in index number two, with occurred in index number three, peer index number four, class so one.

So, instead of writing each and every vector in a, in a complete dictionary form like this, you can only say how many words that particular sentence has, this sentence has only five words. It is simple to put that, you know, it is very easy because the sparseness has gone. But this also has complexity on its own. Let us see, what is the problem there.

So, which means you need to write a program. The first dictionary with index is already present. Then hashing then you can take sentence one and sentence two and index it with the help of a dictionary. That is enough for a representation. Why we are doing this sentence presented as numbers. Because when you give the words to a machine learning classifier like Naïve Bayes or, or decision tree, they would not be able to understand this word.

So, the decision tree will not work here. Naïve Bayes or some other classifiers, SVM or something. They will not take these words and instead you have to convert the words into particular numerical form. That numerical form can be converted using this kind of bag of words approach. Let us look at this, where it can be applied. Hope you understand there are two sentences and we can convert into words. Let us look at this.

(Refer Slide Time: 8:58)



Activity

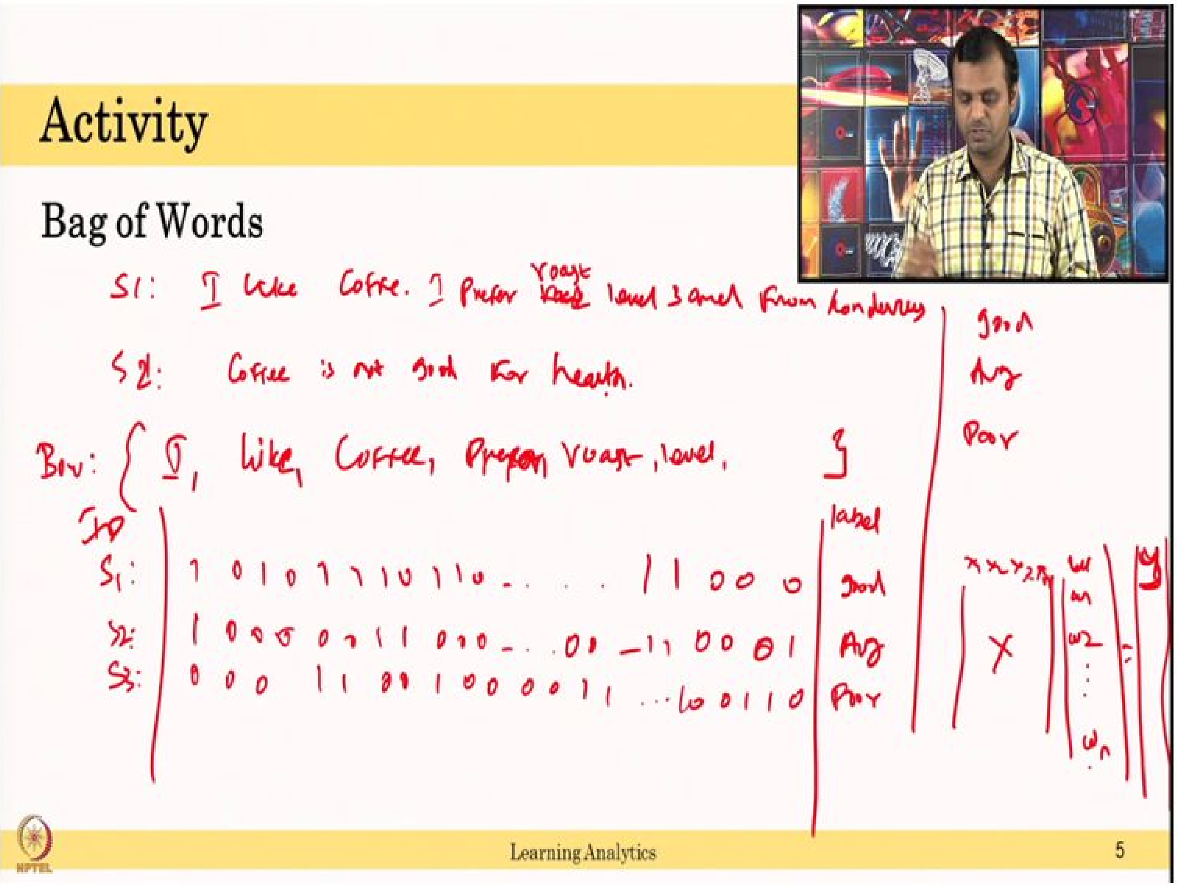
Bag of Words

- 100 students wrote essay and validated by human experts. If we want to create a algorithm to grade essays?

Learning Analytics 4

If we have a hundred students writing an essay and you have validated these essay by the human experts like two, three teachers, validated the exam and classified them as very good, excellent, or average, poor something like that. You want to automatically grade them. If you want to create an automatic algorithm to grade these essays what you have to do? So, consider what you learned in a previous slide. Can you use that knowledge to create an automated essay grading system? That is only that bag of words approach and similarity approaches. Can you think of it? Take a moment think about it. Write your steps and resume to continue.

(Refer Slide Time: 9:47)



Activity

Bag of Words

S1: I like Coffee. I prefer ~~Roast~~ roast level 3 and from londurry

S2: Coffee is not good for health.

BW: { I, like, Coffee, Prefer, roast, level, }

ID

S1: 1 0 1 0 1 1 1 0 1 1 0 . . . . . 1 1 0 0 0 | good

S2: 1 0 0 0 0 1 1 1 0 1 0 . . . 0 0 – 1 1 0 0 0 1 | Avg

S3: 0 0 0 1 1 0 0 1 0 0 0 0 1 1 . . . 0 0 1 1 0 | Poor

label

good

Avg

Poor

X w_1 w_2 $\vdots$ w_n y

Learning Analytics 5

I want to explain that, let us see how it goes, okay. Student one wrote some sentence. I liked the coffee. Something like that, suppose the essay is about coffee. So, I like coffee. I prefer roast level three or medium four something like that. Student says that coffee is not good for health.

Maybe someone does not like coffee, or someone says coffee is very important. Good for heart, all these things coming. So, there are like hundred students wrote this kind of an essay about coffee, and you are a teacher grading that and say that this is a good, average, good, average, poor, only three grades. Let us say there are three grades good, average, poor, you are creating it. Now, the idea is you have to create a bag of words.

From all this hundred sentences, you have to find all the unique words and put that in one dictionary. So, the bag of word for the equals - "I like", "prefer roast", "I just", "prefer roast". So, now you might see what I am talking about, I am talking about a bi-gram, so if you put all the words together, you just actually go in bi-gram and the frequency can be used.

But instead of using frequency, let us say this exact one set, I am creating one set of dictionary from all the existing words you are creating. Once you have that, I want to do student one, (it's ID) as, so some vector [1,0,1,1,0,0,1,1,0] some sparse vector with the label- "good". Student two - "average". Something like that. It is poor. So, the idea is now what you have is sentences in students, essays have been converted into a matrix with a label- good, average, poor there are three things you are having.

So, it is not a binary classifier, so three (class) multi-classifier. You have a student ID and you have the features. The dictionary can go to five thousand or ten thousand. This is a basic form of automatically grading your essays.

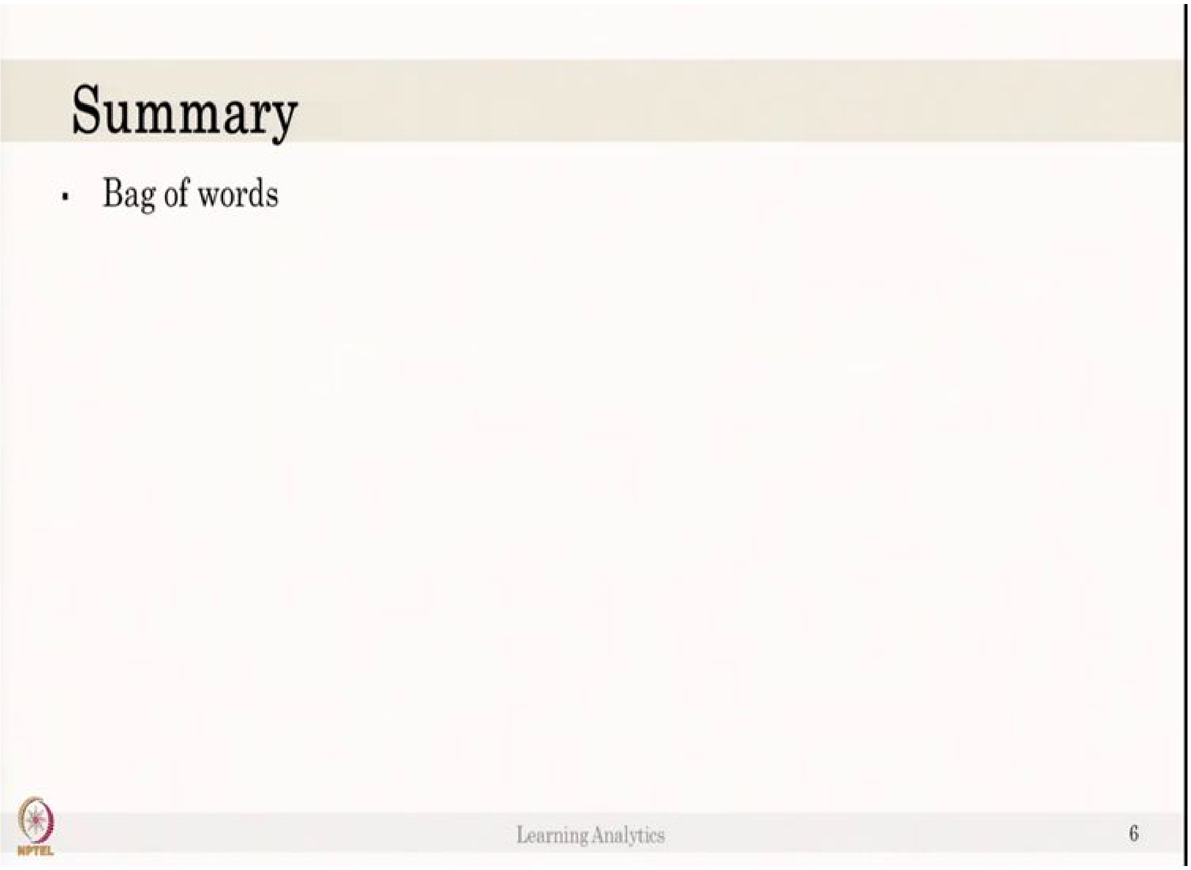
What you have to do, you have a hundred students data and you have done all this grading, you have to take 10 fold cross-validation, train the system and test it on the tenth part for each of ten iterations. You create some accurate classifier, which takes any essay, it will automatically give you the label - average good or poor. So, no need to do it from the next time. It is a very simple, a basic form I was talking about, but it will not yield a good result. But you can check out the latest papers and try to understand what they do. That is very interesting if you can do that.

And I just wanted to tell one more thing. Any form of machine learning classifier we are doing in a supervised classification is this is the form that is a matrix of the features x1, x2, x3. And there is a weight we want to find out, the weight associated with x1 till x_n , equal to the label, label. The label can be multiclass, like a binary class or multiclass. So, that is exactly what all be. It is y. It is just a y label.

So, what I am trying to say is, it is exactly the form of machine learning class, this is exactly what we did in the matrix, in our schooling time. So that is exactly what are the basic for machine learning. If you can imagine any problem in this particular form and able to apply, you can understand that what is an algorithm, how it works, everything is easy to understand. So, if you want to go in detail about that from. Not every matrix have the perfect solution. So, if you do not have the perfect solution, you have to find the nearest solution.

That is exactly what is happening in machine learning. So, not every problem have perfect solution to class. We put the weights to exactly match the words there. So, we have to identify the nearest possible solution that is global minimal solution with least possible error. Some classifier is able to do it. Some classifier is not able to do it. So, the classifiers vary based on the way they approach.

(Refer Slide Time: 15:43)



Summary

- Bag of words

Learning Analytics

6

Hope you understand what is bag of words and how to use a bag of words for automatically grade essay. If we have students done assignments and you have graded them automatically, or if you graded them using your teachers or your friends graded them the assignments or the marks, take that as the input and convert that into a bag of words like a big dictionary, and if you have understood that its very good. So, just simple scripts can help you to convert them.

And you can check it out, whether you can create automated grading system or not. It is very easy to do. If you are from this course start if you are trying to understand programming, this is a very good, simple example to learn and try it out. Because identifying the set of dictionary or the index is not tough at all, it is very easy. So, it is very possible in Panda's library. So, try it out and check it out, whether it works or not. Thank you.