1

00:00:00,025 --> 00:00:05,172

[SOUND] This lecture is about

2

00:00:05,172 --> 00:00:12,786

the feedback in the vector space model.

3

00:00:12,786 --> 00:00:18,040

In this lecture, we continue talking

about the feedback and text retrieval.

4

00:00:18,040 --> 00:00:21,210

Particularly we're going to talk about

feedback in the vector space model.

5

00:00:23,930 --> 00:00:29,730

As we have discussed before in

the case of feedback the task of

6

00:00:29,730 --> 00:00:34,910

a text retrieval system is relearned from

examples to improve retrieval accuracy.

7

00:00:34,910 --> 00:00:37,620

We will have positive examples,

8

00:00:37,620 --> 00:00:40,460

those are the documents that

are assumed that will be random or

9

00:00:40,460 --> 00:00:45,160

judged with being random and all

the documents that are viewed by users.

10

00:00:45,160 --> 00:00:48,790

We also have negative examples, those

are documents known to be non-relevant.

11

00:00:48,790 --> 00:00:52,970

They can also be the documents

that are escaped by users.

12

00:00:55,350 --> 00:00:58,804

The general method in

the vector space model for

13

00:00:58,804 --> 00:01:02,690

feedback is to modify our query vector.

14

00:01:03,930 --> 00:01:08,390

Now we want to place the query vector in

a better position to make that accurate

15

00:01:10,120 --> 00:01:11,520

and what does that mean exactly?

16

00:01:11,520 --> 00:01:14,930

Well, if you think about the query vector

that would mean you would have to do

17

00:01:14,930 --> 00:01:17,260

something to vector elements.

18

00:01:17,260 --> 00:01:21,224

And in general that would

mean we might add new terms.

19

00:01:21,224 --> 00:01:27,129

We might adjust weights of old terms or

assign weights to new terms.

20

00:01:28,490 --> 00:01:32,770

And as a result in general

the query will have more terms so

21

00:01:32,770 --> 00:01:35,130

we often call this query expansion.

22

00:01:37,960 --> 00:01:41,780

The most effective method in the vector

space model of feedback is called Rocchio

23

00:01:41,780 --> 00:01:44,900

feedback which was actually

proposed several decades ago.

24

00:01:47,490 --> 00:01:53,060

So, the idea is quite simple we illustrate

this idea by using a two-dimensional

25

00:01:53,060 --> 00:01:58,360

display of all the documents in

the collection and also the query vector.

26

00:01:58,360 --> 00:02:03,300

So, now we can see

the query vector is here in

27

00:02:03,300 --> 00:02:07,850

the center and

these are all of the documents.

28

00:02:07,850 --> 00:02:09,690

So when we use a query vector and

29

00:02:09,690 --> 00:02:13,100

use a similarity function to

find the most similar documents.

30

00:02:13,100 --> 00:02:14,805

We are basically drawing a circle here and

31

00:02:14,805 --> 00:02:18,960

then these documents would be

basically the top-ranked documents.

32

00:02:18,960 --> 00:02:23,000

And this process of relevant documents,

right?

33

00:02:23,000 --> 00:02:27,090

And these are random documents for

example that's relevant, etc.

34

00:02:27,090 --> 00:02:32,360

And then these minuses

are negative documents like this.

35

00:02:34,310 --> 00:02:37,160

So our goal here is trying

36

00:02:37,160 --> 00:02:42,780

to move this query vector to some position

to improve the retrieval accuracy.

37

00:02:42,780 --> 00:02:48,730

By looking at this diagram

what do you think where

38

00:02:48,730 --> 00:02:53,920

should we move the query vector so that

we can improve the retrieval accuracy.

39

00:02:53,920 --> 00:02:57,030

Intuitively, where do you want

to move the query back to?

40

00:02:58,180 --> 00:03:01,320

If you want to think more

you can pause the video.

41

00:03:03,010 --> 00:03:04,550

Now if you think about

42

00:03:05,580 --> 00:03:10,940

this picture you can realize that

in order to work well in this case

43

00:03:10,940 --> 00:03:15,530

you want the query vector to be as close

to the positive vectors as possible.

44

00:03:15,530 --> 00:03:20,630

That means, ideally you want to place

the query vector somewhere here or

45

00:03:20,630 --> 00:03:24,640

we want to move the query

vector closer to this point.

46

00:03:26,500 --> 00:03:29,559

Now, so what exactly at this point?

47

00:03:29,559 --> 00:03:35,720

Well, if you want these relevant

documents to be ranked on the top

48

00:03:35,720 --> 00:03:41,340

you want this to be in the center of

all of these relevant documents, right?

49

00:03:41,340 --> 00:03:44,720

Because then if you draw

a circle around this one

50

00:03:44,720 --> 00:03:47,230

you get all these relevant documents.

51

00:03:47,230 --> 00:03:52,250

So that means we can move the query

back toward the centroid of

52

00:03:52,250 --> 00:03:54,640

all the relevant document vectors.

53

00:03:55,670 --> 00:03:57,790

And this is basically the idea of Rocchio,

54

00:03:59,200 --> 00:04:02,620

of course you then can see that

the centroid of negative documents.

55

00:04:02,620 --> 00:04:06,980

And one move away from

the negative documents.

56

00:04:06,980 --> 00:04:09,550

Now geometrically we're

talking about a moving vector

57

00:04:09,550 --> 00:04:12,430

closer to some other vector and

away from other vectors.

58

00:04:13,820 --> 00:04:17,340

Algebraically it just means

we have this formula.

59

00:04:18,340 --> 00:04:22,980

Here you can see this is

original query vector and

60

00:04:22,980 --> 00:04:29,680

this average basically is the centroid

vector of relevant documents.

61

00:04:29,680 --> 00:04:32,260

When we take the average

over these vectors

62

00:04:32,260 --> 00:04:35,570

then we're computing

the centroid of these vectors.

63

00:04:35,570 --> 00:04:41,150

And similarly this is the average in

that non-relevant document of vectors so

64

00:04:41,150 --> 00:04:46,080

it's essentially of now random, documents.

65

00:04:46,080 --> 00:04:51,740

And we have these three parameters here,

alpha, beta and gamma.

66

00:04:51,740 --> 00:04:55,200

They're controlling

the amount of movement.

67

00:04:55,200 --> 00:05:00,700

When we add these two vectors together

we're moving the query at the closer

68

00:05:00,700 --> 00:05:05,760

to the centroid, alright, so

when we add them, together.

69

00:05:05,760 --> 00:05:12,060

When we subtracted this part we kind

of move the query vector away from that

70

00:05:13,270 --> 00:05:18,420

centroid so

this is the main idea of Rocchio Feedback.

71

00:05:18,420 --> 00:05:23,100

And after we have done this we

will get a new query vector

72

00:05:23,100 --> 00:05:25,640

which can use it to store documents.

73

00:05:25,640 --> 00:05:30,470

This new New query vector will then

74

00:05:30,470 --> 00:05:35,388

reflect the move of this

Original query vector toward

75

00:05:35,388 --> 00:05:40,210

this Relevant centroid vector and

76

00:05:40,210 --> 00:05:45,660

away from the Non-relevant

centroid vector, okay?

77

00:05:45,660 --> 00:05:48,220

So let's take a look at example, right?

78

00:05:48,220 --> 00:05:54,350

This is the example that we have seen

earlier only that I in the, the display

79

00:05:54,350 --> 00:05:59,220

of the actual documents I only showed the

vector representation of these documents.

80

00:05:59,220 --> 00:06:03,230

We have five documents here and we have

81

00:06:04,760 --> 00:06:09,448

true red in the documents here, right?

82

00:06:09,448 --> 00:06:15,030

They are displayed in red and

these are the term vectors.

83

00:06:15,030 --> 00:06:18,190

Now, I just assumed an idea of weights,

84

00:06:18,190 --> 00:06:20,640

a lot of times we have

zero weights of course.

85

00:06:20,640 --> 00:06:26,120

These are negative documents, there

are two here, there is another one here.

86

00:06:26,120 --> 00:06:30,860

Now in this Rocchio method we

first compute the centroid of

87

00:06:30,860 --> 00:06:34,360

each category and so let's see.

88

00:06:36,100 --> 00:06:39,880

Look at the centroid of

the positive document but

89

00:06:39,880 --> 00:06:42,910

we simply just so it's very easy to see.

90

00:06:42,910 --> 00:06:48,920

We just add this with this one

the corresponding element and

91

00:06:48,920 --> 00:06:51,740

that's down here and take the average.

92

00:06:51,740 --> 00:06:54,680

And then we're going to add

the corresponding elements and

93

00:06:54,680 --> 00:06:56,770

then just take the average, right?

94

00:06:56,770 --> 00:06:58,790

So we do this for all these.

95

00:06:58,790 --> 00:07:02,510

In the end, what we have is this one.

96

00:07:02,510 --> 00:07:08,370

This is the average vector of these two so

it's a centroid of these two, right?

97

00:07:10,010 --> 00:07:13,498

Let's also look at the centroid

of the nested documents.

98

00:07:13,498 --> 00:07:18,150

This is basically the same we're going to

take the average of three elements.

99

00:07:18,150 --> 00:07:22,280

And these are the corresponding

elements in these three vectors and

100

00:07:22,280 --> 00:07:22,990

so on and so forth.

101

00:07:22,990 --> 00:07:25,120

So in the end, that we have this one.

102

00:07:26,210 --> 00:07:29,340

Now, in the Rocchio feedback

method we're going to combine all

103

00:07:29,340 --> 00:07:32,920

these with original query vector,

which is this.

104

00:07:32,920 --> 00:07:35,090

So now let's see how we

combine them together.

105

00:07:36,140 --> 00:07:38,880

Well, that's basically this, right?

106

00:07:38,880 --> 00:07:45,210

So we have a parameter outlier controlling

the original query term weight that's 1.

107

00:07:45,210 --> 00:07:50,460

And now I've beta to control

the inference of the positive

108

00:07:50,460 --> 00:07:55,460

centroid Vector weight that's

1.5 that comes from here, right?

109

00:07:55,460 --> 00:08:00,570

So this goes here and

110

00:08:00,570 --> 00:08:04,594

we also have this negative wait here.

111

00:08:04,594 --> 00:08:08,100

Conduit by a gamma here and

112

00:08:08,100 --> 00:08:12,550

this weight has come from of

course the nective centroid here.

113

00:08:14,520 --> 00:08:18,740

And we do exactly the same for

other terms each is for one term.

114

00:08:22,257 --> 00:08:23,850

And this is our new vector.

115

00:08:25,710 --> 00:08:31,530

And we're going to use this new query

vector, this one to run the documents.

116

00:08:31,530 --> 00:08:33,840

You can imagine what would happen, right?

117

00:08:33,840 --> 00:08:35,590

Because of the movement that this one or

118

00:08:35,590 --> 00:08:38,550

the match of these red

documents much better.

119

00:08:38,550 --> 00:08:42,890

Because we move this

vector closer to them and

120

00:08:42,890 --> 00:08:47,290

it's going to penalize these black

documents, these non-relevant documents.

121

00:08:47,290 --> 00:08:49,790

So this is precisely what

we want from feedback.

122

00:08:50,810 --> 00:08:57,230

Now of course, if we apply this method in

practice we will see one potential problem

123

00:08:58,240 --> 00:09:05,794

and that is the original query has

only four times that are not zero.

124

00:09:05,794 --> 00:09:07,790

But after we do queries,

125

00:09:07,790 --> 00:09:13,210

imagine you can imagine we'll have many

terms that would have a number of weights.

126

00:09:13,210 --> 00:09:16,580

So the calculation would

have to involve more terms.

127

00:09:18,080 --> 00:09:21,809

In practice,

we often truncate this vector and

128

00:09:21,809 --> 00:09:25,470

only retain the terms which

is the highest weight.

129

00:09:27,000 --> 00:09:29,460

So let's talk about how we

use this method in practice.

130

00:09:30,640 --> 00:09:35,580

I just mentioned that we often truncate

the vector consider only a small number

131

00:09:35,580 --> 00:09:38,690

of words that have highest

weights in the centroid vector.

132

00:09:38,690 --> 00:09:39,890

This is for efficiency concern.

133

00:09:41,460 --> 00:09:45,520

I also say that here that a negative

examples or non-relevant examples

134

00:09:45,520 --> 00:09:49,540

tend not to be very useful especially

compared with positive examples.

135

00:09:50,860 --> 00:09:52,430

Now you can think about the, why.

136

00:09:52,430 --> 00:09:57,320

One reason is because negative documents

137

00:09:57,320 --> 00:10:02,210

tend to distract the query in

all directions so when you take

138

00:10:02,210 --> 00:10:06,860

the average it doesn't really tell you

where exactly it should be moving to.

139

00:10:06,860 --> 00:10:10,220

Whereas, positive documents tend

to be clustered together and

140

00:10:10,220 --> 00:10:13,350

they respond to you to

consistent the direction.

141

00:10:13,350 --> 00:10:19,250

So that also means that sometimesw we

don't have **to use** those negative examples but

142

00:10:19,250 --> 00:10:19,860

note that in,

143

00:10:19,860 --> 00:10:24,570

in some cases in difficult queries where

most top random results are negative.

144

00:10:24,570 --> 00:10:26,390

Negative feedback

afterwards is very useful.

145

00:10:27,540 --> 00:10:30,550

Another thing is to avoid

over-fitting that means we have to

146

00:10:30,550 --> 00:10:34,000

keep relatively high weight

on the original query terms.

147

00:10:35,150 --> 00:10:35,780

Why?

148

00:10:35,780 --> 00:10:42,250

Because the sample that we see in

feedback is a relatively small sample.

149

00:10:42,250 --> 00:10:46,410

We don't want to overly

trust the small sample and

150

00:10:46,410 --> 00:10:49,390

the original query terms

are still very important.

151

00:10:49,390 --> 00:10:51,910

Those terms are typed in by the user and

152

00:10:51,910 --> 00:10:55,850

the user has decided that those

terms are most important.

153

00:10:55,850 --> 00:11:01,700

So in order to prevent the us

from over-fitting or drifting.

154

00:11:01,700 --> 00:11:07,040

A type of drift prevent type of

drifting due to the bias toward the,

155

00:11:07,040 --> 00:11:08,910

the feedback examples.

156

00:11:08,910 --> 00:11:12,860

We generally would have to keep a pretty

high weight on the original terms so

157

00:11:12,860 --> 00:11:13,980

it is safe to do that.

158

00:11:15,040 --> 00:11:19,660

And this is especially, true for

pseudo awareness feedback.

159

00:11:19,660 --> 00:11:22,180

Now this method can be used for

both relevance feedback and

160

00:11:22,180 --> 00:11:23,240

pseudo relevance feedback.

161

00:11:23,240 --> 00:11:27,370

In the case of pseudo feedback,

the parameter beta should be set to a,

162

00:11:27,370 --> 00:11:32,340

a smaller value because

the random examples are assumed

163

00:11:32,340 --> 00:11:36,940

to be random there not as reliable

as your relevance feedback, right?

164

00:11:36,940 --> 00:11:40,830

In the case of relevance feedback,

we obviously could use a larger value.

165

00:11:40,830 --> 00:11:45,010

So, those parameters

still have to be set and **empirically**.

166

00:11:45,010 --> 00:11:48,500

And the ro, Rocchio method is

usually robust and effective.

167

00:11:48,500 --> 00:11:52,993

It's, it's still a very popular method for

feedback.

168

00:11:52,993 --> 00:12:02,993

[MUSIC]