1

00:00:00,354 --> 00:00:01,185

[SOUND]

2

00:00:10,729 --> 00:00:15,310

In this lecture, we continue

the discussion of Vector Space Model.

3

00:00:15,310 --> 00:00:18,810

In particular, we are going to

talk about the TF transformation.

4

00:00:18,810 --> 00:00:20,638

In the previous lecture,

5

00:00:20,638 --> 00:00:25,880

we have derived a TF-IDF weighting

formula using the vector space model.

6

00:00:27,100 --> 00:00:32,617

And we have shown that this model

actually works pretty well for

7

00:00:32,617 --> 00:00:37,302

these examples as shown on

this slide except for d5,

8

00:00:37,302 --> 00:00:41,330

which has received a very high score.

9

00:00:41,330 --> 00:00:46,258

Indeed, it has received the highest

score among all these documents.

10

00:00:46,258 --> 00:00:53,084

But this document is intuitively

non-relevant, so this is not desirable.

11

00:00:53,084 --> 00:00:57,343

In this lecture, we're going to

talk about how would you use TF

12

00:00:57,343 --> 00:01:00,190

transformation to solve this problem.

13

00:01:00,190 --> 00:01:05,237

Before we discuss the details,

let's take a look at the formula for

14

00:01:05,237 --> 00:01:09,128

this symbol here for

IDF weighting ranking function and

15

00:01:09,128 --> 00:01:13,520

see why this document has

received such a high score.

16

00:01:13,520 --> 00:01:17,500

So this is the formula, and

if you look at the formula carefully,

17

00:01:17,500 --> 00:01:23,810

then you will see it involves a sum

over all the matched query terms.

18

00:01:23,810 --> 00:01:28,140

And inside the sum, each matched

query sum has a particular weight.

19

00:01:28,140 --> 00:01:30,259

And this weight is TF-IDF weighting.

20

00:01:31,580 --> 00:01:36,655

So it has an IDF component

where we see 2 variables.

21

00:01:36,655 --> 00:01:41,450

One is the total number of documents

in the collection, and that is m.

22

00:01:41,450 --> 00:01:45,880

The other is the documentive frequency.

23

00:01:45,880 --> 00:01:52,250

This is the number of documents

that contain this word w.

24

00:01:52,250 --> 00:01:54,700

The other variables in,

involving the formula,

25

00:01:54,700 --> 00:01:58,340

include the count of the query term.

26

00:02:01,440 --> 00:02:06,100

W in the query, and

the count of the word in the document.

27

00:02:07,650 --> 00:02:11,450

If you look at this document again,

now it's not hard to

28

00:02:11,450 --> 00:02:16,710

realize that the reason why it has

received a high score is because

29

00:02:16,710 --> 00:02:21,016

it has a very high count of campaign.

30

00:02:21,016 --> 00:02:27,150

So the count of campaign in this document

is a four, which is much higher than

31

00:02:27,150 --> 00:02:31,572

the other documents, and has contributed

to the high score of this document.

32

00:02:31,572 --> 00:02:37,240

So intriguingly, in order to lower

the score for this document, we need

33

00:02:37,240 --> 00:02:43,279

to somehow restrict the contribution of,

the matching of this term in the document.

34

00:02:44,620 --> 00:02:48,290

And if you think about the matching of

terms in the document carefully you

35

00:02:48,290 --> 00:02:53,059

actually would realize we

probably shouldn't reward

36

00:02:53,059 --> 00:02:58,360

multiple occurrences so generously.

37

00:02:58,360 --> 00:03:04,525

And by that I mean the first occurrence

of a term says a lot about the,

38

00:03:04,525 --> 00:03:09,720

the matching of this term,

because it goes from zero count

39

00:03:09,720 --> 00:03:15,360

to a count of one, and

that increase means a lot.

40

00:03:17,160 --> 00:03:19,090

Once we see a word in the document,

41

00:03:19,090 --> 00:03:21,690

it's very likely that the document

is talking about this word.

42

00:03:23,370 --> 00:03:26,800

If we see an extra occurrence

on top of the first occurrence,

43

00:03:28,150 --> 00:03:34,940

that is to go from one to two,

then we also can say that well, the second

44

00:03:34,940 --> 00:03:38,950

occurrence kind of confirmed that it's

not a accidental mention of the word.

45

00:03:39,970 --> 00:03:44,270

Now, we are more sure that this

document is talking about this word.

46

00:03:44,270 --> 00:03:50,420

But imagine we have seen, let's say,

50 times of the word in the document.

47

00:03:50,420 --> 00:03:54,820

Then, adding one extra occurrence

is not going to test more about

48

00:03:54,820 --> 00:03:59,580

evidence because we are already sure

that this document is about this word.

49

00:04:01,160 --> 00:04:04,330

So if you're thinking

this way it seems that

50

00:04:04,330 --> 00:04:08,525

we should restrict the contributing

of a high account of term.

51

00:04:09,655 --> 00:04:12,785

And that is the idea of TF Transformation.

52

00:04:12,785 --> 00:04:17,980

So this transformation function is

going to turn the raw count of word

53

00:04:17,980 --> 00:04:22,990

into a Term Frequency Weight,

for the word in the document.

54

00:04:22,990 --> 00:04:27,171

So here I show in x-axis, that raw count,

55

00:04:27,171 --> 00:04:31,470

and in y-axis I show

the Term Frequency Weight.

56

00:04:33,360 --> 00:04:37,981

So, in the previous ranking functions

we actually have increasingly,

57

00:04:37,981 --> 00:04:40,410

used some kind of transformation.

58

00:04:40,410 --> 00:04:44,764

So for example in the zero-one bit

vector retentation we actually use

59

00:04:44,764 --> 00:04:49,070

the Suchier transformation

function as shown here.

60

00:04:49,070 --> 00:04:53,420

Basically if the count is

zero then it has zero weight.

61

00:04:53,420 --> 00:04:55,810

Otherwise it would have a weight of one.

62

00:04:57,570 --> 00:04:58,070

It's flat.

63

00:04:59,550 --> 00:05:04,870

Now what about using

Term Count as a TF weight.

64

00:05:04,870 --> 00:05:06,410

Well that's a linear function, right?

65

00:05:06,410 --> 00:05:10,515

So it has just exactly

the same weight as the count.

66

00:05:11,575 --> 00:05:16,765

Now we have just seen that

this is not desirable.

67

00:05:18,405 --> 00:05:20,795

So what we want is something like this.

68

00:05:20,795 --> 00:05:22,836

So for example with a logarithm function,

69

00:05:22,836 --> 00:05:26,620

we can have a sub-linear

transformation that looks like this.

70

00:05:26,620 --> 00:05:30,740

And this will control the influence of

really high weight because it's going to

71

00:05:30,740 --> 00:05:35,035

lower its inference, yet it will

retain the inference of small count.

72

00:05:36,110 --> 00:05:41,550

Or we might want to even bend the curve

more by applying logarithm twice.

73

00:05:42,730 --> 00:05:46,990

Now people have tried all these methods

and they are indeed working better than

74

00:05:46,990 --> 00:05:52,000

the linear form of the transformation,

but so far what works the best

75

00:05:52,000 --> 00:05:56,620

seems to be this special transformation

called a BM25 transformation.

76

00:05:58,070 --> 00:05:59,480

BM stands for best matching.

77

00:06:01,210 --> 00:06:05,010

Now in this transformation,

you can see there's a parameter k here.

78

00:06:06,460 --> 00:06:10,890

And this k controls the upper

bound of this function.

79

00:06:10,890 --> 00:06:15,710

It's easy to see this function has

a upper bound because if you look at

80

00:06:15,710 --> 00:06:22,180

the x divided by x plus k where

k is not an active number,

81

00:06:22,180 --> 00:06:28,060

then the numerator will never be able

to exceed the denominator, right?

82

00:06:28,060 --> 00:06:29,830

So, it's upper bounded by k plus 1.

83

00:06:29,830 --> 00:06:34,500

Now, this is also difference between

this transformation function and

84

00:06:34,500 --> 00:06:35,660

the logarithm transformation.

85

00:06:37,010 --> 00:06:39,442

Which it doesn't have upperbound.

86

00:06:39,442 --> 00:06:44,490

Now furthermore, one interesting property

of this function is that as we vary K,

87

00:06:45,610 --> 00:06:50,310

we can actually simulate different

transformation functions,

88

00:06:50,310 --> 00:06:52,910

including the two extremes

that are shown here.

89

00:06:52,910 --> 00:06:57,480

That is a zero one bit transformation,

and the unit transformation.

90

00:06:57,480 --> 00:07:01,930

So for example, if we set k to zero,

now you can see

91

00:07:03,630 --> 00:07:05,950

the function value would be one.

92

00:07:07,090 --> 00:07:13,250

So we precisely,

recover the zero one bit transformation.

93

00:07:15,630 --> 00:07:19,030

If you set k to a very large number,

on the other hand,

94

00:07:19,030 --> 00:07:22,919

other hand, it's going to look more

like the linear transformation function.

95

00:07:24,980 --> 00:07:29,400

So in this sense,

this transformation is very flexible,

96

00:07:29,400 --> 00:07:34,600

it allows us to control

the shape of the transformation.

97

00:07:34,600 --> 00:07:37,763

It also has a nice property

of the upper bound.

98

00:07:37,763 --> 00:07:43,637

And this upper bound is useful to control

the inference of a particular term.

99

00:07:43,637 --> 00:07:49,702

And so that we can prevent a, a spammer

from just increasing the count of

100

00:07:49,702 --> 00:07:54,770

1 term to spam all queries

that might match this term.

101

00:07:57,350 --> 00:08:02,010

In other words this upper bound

might also ensure that all terms

102

00:08:02,010 --> 00:08:06,680

will be counted when we aggregate the,

the weights, to compute a score.

103

00:08:06,680 --> 00:08:10,630

As I said, this transformation

function has worked well, so far.

104

00:08:12,300 --> 00:08:14,308

So to summarise this lecture,

105

00:08:14,308 --> 00:08:19,910

the main point is that we need to do

some sub linearity of TF Transformation.

106

00:08:19,910 --> 00:08:24,340

And this is needed to capture

the intuition of diminishing return from

107

00:08:24,340 --> 00:08:25,550

high Term Counts.

108

00:08:26,620 --> 00:08:31,020

It's also to avoid a dominance by

one single term over all others.

109

00:08:31,020 --> 00:08:37,230

This BM25 Transformation, Transformation

that we talked about is very interesting.

110

00:08:37,230 --> 00:08:43,130

It's so far one of the best performing

TF Transforming formation formulas.

111

00:08:43,130 --> 00:08:47,414

It has upper bound, and

it's also robust and effective.

112

00:08:47,414 --> 00:08:52,612

Now, if we're plug in this

function into our TF-IDF weighting

113

00:08:52,612 --> 00:08:57,425

vector space model then we would

end up having the following

114

00:08:57,425 --> 00:09:01,770

ranking function,

which has a BM25 TF component.

115

00:09:01,770 --> 00:09:06,784

Now this is already very close to a state

116

00:09:06,784 --> 00:09:11,940

of the art ranking function called a BM25.

117

00:09:11,940 --> 00:09:19,602

And we will discuss how we can further

improve this formula in the next lecture.

118

00:09:19,602 --> 00:09:29,602

[MUSIC]