1

00:00:00,000 --> 00:00:05,140

[SOUND] This lecture is about

2

00:00:05,140 --> 00:00:12,297

the probabilistic retrieval model.

3

00:00:12,297 --> 00:00:18,545

In this lecture, we're going to continue

the discussion of text retrieval methods.

4

00:00:18,545 --> 00:00:23,177

We're going to look at another kind of

very different way to design ranking

5

00:00:23,177 --> 00:00:27,382

functions, then the Vector Space Model

that we discussed before.

6

00:00:32,131 --> 00:00:37,098

In probabilistic models we define

the ranking function based

7

00:00:37,098 --> 00:00:42,170

on the probability that this

document is random to this query.

8

00:00:42,170 --> 00:00:48,175

In other words, we are, we introduced

a binary random variable here.

9

00:00:48,175 --> 00:00:53,420

This is the variable R here.

10

00:00:53,420 --> 00:00:56,069

And we also assume that the query and

11

00:00:56,069 --> 00:01:00,586

the documents are all observations

from random variables.

12

00:01:00,586 --> 00:01:03,462

Note that in the vector space model,

we assume they are vectors.

13

00:01:03,462 --> 00:01:10,796

But here we assumed we assumed they are

the data observed from random variables.

14

00:01:10,796 --> 00:01:14,482

And so the problem, model retrieval

15

00:01:14,482 --> 00:01:19,129

becomes to estimate

the probability of relevance.

16

00:01:19,129 --> 00:01:22,670

In this category of models,

there are different variants.

17

00:01:22,670 --> 00:01:27,662

The classic probabilistic model has

led to the BM25 retrieval function,

18

00:01:27,662 --> 00:01:30,704

which we discussed in

the vector space model,

19

00:01:30,704 --> 00:01:34,920

because it's form is actually

similar to a vector space model.

20

00:01:34,920 --> 00:01:39,991

In this lecture,

we're going to discuss another subclass in

21

00:01:39,991 --> 00:01:45,255

this big class called a language

modeling approaches to retrieval.

22

00:01:45,255 --> 00:01:51,417

In particular, we're going to discuss

the Query Likelihood retrieval model,

23

00:01:51,417 --> 00:01:56,753

which is one of the most effective

models in probabilistic models.

24

00:01:56,753 --> 00:02:02,881

There is also another line called

a divergence-from-randomness model,

25

00:02:02,881 --> 00:02:06,048

which has latitude the PL2 function.

26

00:02:06,048 --> 00:02:10,797

It's also one of the most effective

state of the art attribute functions.

27

00:02:10,797 --> 00:02:16,851

In query likelihood, our assumption

is that this probability readiness

28

00:02:16,851 --> 00:02:23,503

can be approximated by the probability

of query given a document and readiness.

29

00:02:23,503 --> 00:02:29,711

So, intuitively, this probability just

captures the following probability.

30

00:02:29,711 --> 00:02:34,791

And that is if a user likes document d,

how likely would

31

00:02:34,791 --> 00:02:39,878

the user enter query q in

order to retrieve document d.

32

00:02:39,878 --> 00:02:47,336

So we'll assume that the user likes d,

because we have a relevance value here.

33

00:02:47,336 --> 00:02:51,859

And the we ask the question about

how likely we will see this

34

00:02:51,859 --> 00:02:54,545

particular query from this user?

35

00:02:54,545 --> 00:02:56,754

So this is the basic idea.

36

00:02:56,754 --> 00:03:00,756

Now to understand this idea,

let's take a look at the general idea or

37

00:03:00,756 --> 00:03:03,795

the basic idea of probabilistic

retrieval models.

38

00:03:03,795 --> 00:03:10,184

So here, I listed some imagined

relevance status values or

39

00:03:10,184 --> 00:03:14,711

relevance judgments of queries and

documents.

40

00:03:14,711 --> 00:03:20,484

For example, in this slide,

it shows that query one

41

00:03:20,484 --> 00:03:27,701

is a query that the user typed in and

d1 is a document the user has seen and

42

00:03:27,701 --> 00:03:33,379

one means the user thinks

d1 is relevant to to q1.

43

00:03:33,379 --> 00:03:38,973

So this R here can be also approximated

by the clicks little data that the search

44

00:03:38,973 --> 00:03:44,420

engine can collect it by watching how

you interact with the search results.

45

00:03:44,420 --> 00:03:48,052

So, in this case, let's say,

the user clicked on this document, so

46

00:03:48,052 --> 00:03:49,170

there's a one here.

47

00:03:49,170 --> 00:03:56,128

Similarly, the user clicked on d2 also,

so there's a one here.

48

00:03:56,128 --> 00:04:00,420

In other words,

d2 is assumed to relevant at two, q1.

49

00:04:00,420 --> 00:04:07,086

On the other hand, d3 is non relevant,

there's a zero here.

50

00:04:07,086 --> 00:04:13,128

And d4 is non-relevant and then d 5 is

again relevant and so on and so forth.

51

00:04:13,128 --> 00:04:17,378

And this part of maybe,

they are collected from a different user.

52

00:04:17,378 --> 00:04:19,681

Right.

So this user typed in q1 and

53

00:04:19,681 --> 00:04:25,798

then found that d1 is actually not useful,

so d1 is actually non-relevant.

54

00:04:25,798 --> 00:04:30,890

In contrast here we see it's relevant and,

55

00:04:30,890 --> 00:04:38,592

or this could be the same query typing

by the same user at different times,

56

00:04:38,592 --> 00:04:42,586

but d2 is also relevant, et cetera.

57

00:04:42,586 --> 00:04:48,046

And then here, we can see more

data that about other queries.

58

00:04:48,046 --> 00:04:52,545

Now we can imagine,

we have a lot of search data.

59

00:04:52,545 --> 00:04:54,931

Now we can ask the question,

60

00:04:54,931 --> 00:04:59,711

how can we then estimated

the probability of relevance?

61

00:04:59,711 --> 00:05:00,271

Right.

So

62

00:05:00,271 --> 00:05:03,339

how can we compute this

probability of relevance?

63

00:05:03,339 --> 00:05:07,334

Well, intuitively,

that just means if we look at the,

64

00:05:07,334 --> 00:05:12,264

all the entries where we see this

particular d and this particular q,

65

00:05:12,264 --> 00:05:15,670

how likely will we see

a one on the third column?

66

00:05:15,670 --> 00:05:19,420

Basically, that just means

we can correct the counts.

67

00:05:19,420 --> 00:05:24,435

We can first count how many

times where we see q and

68

00:05:24,435 --> 00:05:29,693

d as a pair in this table and

then count how many times

69

00:05:29,693 --> 00:05:35,562

we actually have also seen

one in the third column and

70

00:05:35,562 --> 00:05:39,378

then we just compute the ratio.

71

00:05:39,378 --> 00:05:42,256

So let's take a look at

some specific examples.

72

00:05:42,256 --> 00:05:50,378

Suppose we are trying to computed this

probability for d1, d2 and d3 for q1.

73

00:05:50,378 --> 00:05:53,586

What is the estimated probability?

74

00:05:53,586 --> 00:05:55,920

Now think about that.

75

00:05:55,920 --> 00:05:58,961

You can pause the video if needed.

76

00:05:58,961 --> 00:06:02,246

Try to take a look at the table and

77

00:06:02,246 --> 00:06:07,004

try to give your estimate

of the probability.

78

00:06:07,004 --> 00:06:13,509

Have you seen that if we are interested

in q1 and d1, we've been looking at the,

79

00:06:13,509 --> 00:06:18,601

these two pairs and in both cases or

actually in one of the cases,

80

00:06:18,601 --> 00:06:22,878

the user has said that this is one,

this is relevant.

81

00:06:22,878 --> 00:06:26,503

So R is equal to 1 in only

one of the two cases.

82

00:06:26,503 --> 00:06:28,795

In the other case, this is zero.

83

00:06:28,795 --> 00:06:32,004

So that's one out of two.

84

00:06:32,004 --> 00:06:35,128

What about the d1 and the d2?

85

00:06:35,128 --> 00:06:40,170

Well, they're are here,

you want d2, d1, d2.

86

00:06:40,170 --> 00:06:42,754

In both cases,

in this case R is equal to 1.

87

00:06:42,754 --> 00:06:46,128

So, it's two out of two and

so and so forth.

88

00:06:46,128 --> 00:06:48,606

So you can see with this approach,

89

00:06:48,606 --> 00:06:52,378

we captured it score these documents for

the query.

90

00:06:52,378 --> 00:06:52,878

Right?

91

00:06:52,878 --> 00:06:56,920

We now have a score for d1,

d2 and d3 for this query.

92

00:06:56,920 --> 00:07:00,702

We can simply ranked them based

on these probabilities and so

93

00:07:00,702 --> 00:07:04,420

that's the basic idea of

probabilistic retrieval model.

94

00:07:04,420 --> 00:07:06,045

And you can see, it makes a lot of sense.

95

00:07:06,045 --> 00:07:09,961

In this case, it's going to rank

d2 above all the other documents.

96

00:07:09,961 --> 00:07:16,253

Because in all the cases, when you

have seen q1 and d2, R is equal to 1.

97

00:07:16,253 --> 00:07:19,336

The user clicked on this document.

98

00:07:19,336 --> 00:07:24,930

So this also showed showed that

with a lot of click through data,

99

00:07:24,930 --> 00:07:30,549

a search engine can learn a lot from

the data to improve the search engine.

100

00:07:30,549 --> 00:07:35,622

This is a simple example that shows that

with even a small number of entries here,

101

00:07:35,622 --> 00:07:38,378

we can already estimate

some probabilities.

102

00:07:38,378 --> 00:07:43,321

These probabilities would give us some

sense about which document might be more

103

00:07:43,321 --> 00:07:46,797

read or more useful to a user for

typing this query.

104

00:07:46,797 --> 00:07:51,291

Now, of course, the problem is that

we don't observe all the queries and

105

00:07:51,291 --> 00:07:54,420

all of the documents and

all the relevance values.

106

00:07:54,420 --> 00:07:55,003

Right?

107

00:07:55,003 --> 00:07:57,049

There will be a lot of unseen documents.

108

00:07:57,049 --> 00:08:01,780

In general, we can only collect data from

the document's that we have shown to

109

00:08:01,780 --> 00:08:02,811

the users.

110

00:08:02,811 --> 00:08:05,207

There are even more unseen queries,

111

00:08:05,207 --> 00:08:09,547

because you cannot predict what

queries will be typed in by users.

112

00:08:09,547 --> 00:08:13,473

So, obviously, this approach won't work

113

00:08:13,473 --> 00:08:17,796

if we apply it to unseen queries or

unseen documents.

114

00:08:17,796 --> 00:08:22,543

Nevertheless, this shows the basic idea

of the probabilistic retrieval model and

115

00:08:22,543 --> 00:08:24,255

it makes sense intuitively.

116

00:08:24,255 --> 00:08:28,670

So what do we do in such a case when we

have a lot of unseen documents and, and

117

00:08:28,670 --> 00:08:29,753

unseen queries?

118

00:08:29,753 --> 00:08:32,798

Well, the solutions that we have

to approximate in some way.

119

00:08:32,798 --> 00:08:36,275

Right.

So, in this particular case called

120

00:08:36,275 --> 00:08:41,892

the Query LIkelihood Retrieval Model,

we just approximate this

121

00:08:41,892 --> 00:08:47,420

by another conditional probability,

p q | d, R is equal to 1.

122

00:08:47,420 --> 00:08:52,023

So, in the condition part, we assume

that the user likes the document,

123

00:08:52,023 --> 00:08:55,878

because we have seen that the user

clicked on this document.

124

00:08:55,878 --> 00:08:56,717

And this part,

125

00:08:56,717 --> 00:09:01,378

shows that we're interested in how likely

the user would actually enter this query.

126

00:09:01,378 --> 00:09:04,628

How likely we will see this

query in the same row.

127

00:09:04,628 --> 00:09:08,503

So note that here, we have made

an interesting assumption here.

128

00:09:08,503 --> 00:09:13,490

Basically, we, we're going to assume

that whether the user types in this

129

00:09:13,490 --> 00:09:17,586

query has something to do with

whether user likes the document.

130

00:09:17,586 --> 00:09:22,192

In other words, we actually

make the foreign assumption and

131

00:09:22,192 --> 00:09:27,711

that is a user formula to query based

on an imaginary relevant document.

132

00:09:27,711 --> 00:09:30,471

Well, if you just look at this

as a conditional probability,

133

00:09:30,471 --> 00:09:32,711

it's not obvious we

are making this assumption.

134

00:09:32,711 --> 00:09:37,636

So what I really meant

is that to use this new

135

00:09:37,636 --> 00:09:42,931

conditional probability to help us score

136

00:09:42,931 --> 00:09:47,255

then this new condition of probability.

137

00:09:47,255 --> 00:09:51,464

We have to somehow be able

to estimate this conditional

138

00:09:51,464 --> 00:09:54,924

probability without

relying on this big table.

139

00:09:54,924 --> 00:09:59,128

Otherwise, it would be having

similar problems as before.

140

00:09:59,128 --> 00:10:04,816

And by making this assumption, we have

some way to bypass this big table and

141

00:10:04,816 --> 00:10:08,712

try to just mortar how to

use a formula to the query.

142

00:10:08,712 --> 00:10:11,919

Okay.

So this is how you can simplify the,

143

00:10:11,919 --> 00:10:18,253

the general model so that we can

give either specific function later.

144

00:10:18,253 --> 00:10:21,464

So let's look at how this model works for

our example.

145

00:10:21,464 --> 00:10:22,520

And basically,

146

00:10:22,520 --> 00:10:27,128

what we are going to do in this case

is to ask the following question.

147

00:10:27,128 --> 00:10:31,073

Which of these documents is most likely

the imaginary relevant document in

148

00:10:31,073 --> 00:10:33,961

the user's mind when the user

formulates this query?

149

00:10:33,961 --> 00:10:37,722

And so we ask this question and

we quantify the probability and this

150

00:10:37,722 --> 00:10:42,360

probability is a conditional probability

of observing this query if a particular

151

00:10:42,360 --> 00:10:46,545

document is in fact the imaginary

relevant document in the user's mind.

152

00:10:46,545 --> 00:10:51,439

Here you can see we compute all these

query likelihood probabilities,

153

00:10:51,439 --> 00:10:54,586

the likelihood of queries

given each document.

154

00:10:54,586 --> 00:10:56,374

Once we have these values,

155

00:10:56,374 --> 00:11:00,045

we can then rank these documents

based on these values.

156

00:11:00,045 --> 00:11:05,574

So to summarize, the general idea of

modern relevance in the probability

157

00:11:05,574 --> 00:11:11,295

risk model is to assume that we introduce

a binary random variable, R here.

158

00:11:11,295 --> 00:11:15,250

And then let the scoring function be

defined based on this conditional

159

00:11:15,250 --> 00:11:16,128

probability.

160

00:11:16,128 --> 00:11:22,091

We also talked about a proximate in this

[SOUND] by using the query likelihood.

161

00:11:22,091 --> 00:11:26,030

And in this case,

we have a ranking function that's

162

00:11:26,030 --> 00:11:30,878

basically based on a probability

of a query given the document.

163

00:11:30,878 --> 00:11:35,681

And this probability should be

interpreted as the probability

164

00:11:35,681 --> 00:11:39,587

that a user who likes document

d would pose query q.

165

00:11:39,587 --> 00:11:44,295

Now the question, of course is how do

we compute this additional probability?

166

00:11:44,295 --> 00:11:49,999

At this in general has to do with how

to compute the probability of text,

167

00:11:49,999 --> 00:11:51,629

because q is a text.

168

00:11:51,629 --> 00:11:56,212

And this has to do with a model

called a Language Model.

169

00:11:56,212 --> 00:12:01,878

And this kind of models

are proposed to model text.

170

00:12:01,878 --> 00:12:07,409

So most specifically, we will be

very interested in the following

171

00:12:07,409 --> 00:12:11,756

conditional probability as I show you,

you this here.

172

00:12:11,756 --> 00:12:19,045

If the user like this document, how

likely the user would approve this query?

173

00:12:19,045 --> 00:12:25,660

And in the next lecture, we're going

to give introduction to Language Model,

174

00:12:25,660 --> 00:12:32,295

so that we can see how we can model text

with a probability risk model in general.

175

00:12:32,295 --> 00:12:42,295

[MUSIC]