1

00:00:07,440 --> 00:00:09,410

This lecture is about the web search.

2

00:00:11,950 --> 00:00:14,790

In this lecture we

are going to talk about one of

3

00:00:14,790 --> 00:00:19,150

the most important applications of

text retrieval, web search engines.

4

00:00:19,150 --> 00:00:21,510

So let's first look at some

general challenges and

5

00:00:21,510 --> 00:00:23,380

opportunities in web search.

6

00:00:23,380 --> 00:00:27,450

Now, many information retrieval

algorithms had been developed at the,

7

00:00:27,450 --> 00:00:29,010

before the web was born.

8

00:00:29,010 --> 00:00:33,890

So, when the web was born,

it created the best opportunity to apply

9

00:00:33,890 --> 00:00:39,890

those algorithms to major application

problem that everyone would care about.

10

00:00:39,890 --> 00:00:45,780

So naturally, there had to be some

further extensions of the classical

11

00:00:45,780 --> 00:00:52,480

search algorithms to address some new

challenges encountered in web search.

12

00:00:53,480 --> 00:00:56,200

So here are some general challenges.

13

00:00:56,200 --> 00:00:58,510

Firstly, this is a scalability challenge.

14

00:00:58,510 --> 00:01:00,040

How we handle the size of the web,

15

00:01:00,040 --> 00:01:02,760

and ensure completeness of

coverage of all the information.

16

00:01:03,870 --> 00:01:07,790

How to serve many users quickly,

and by answering all their queries.

17

00:01:07,790 --> 00:01:10,350

All right, so, that's one major challenge.

18

00:01:10,350 --> 00:01:15,990

And before the web was born,

the scale of search was relatively small.

19

00:01:15,990 --> 00:01:20,110

The second problem is that there

is low quality information.

20

00:01:20,110 --> 00:01:22,130

And there are often spams.

21

00:01:22,130 --> 00:01:24,350

The third challenge is

dynamics of the web.

22

00:01:24,350 --> 00:01:27,410

The new pages are constantly created and

23

00:01:27,410 --> 00:01:32,390

some pages may be updated,

eve-, very quickly.

24

00:01:32,390 --> 00:01:36,480

So it makes it harder to,

keep the index fresh.

25

00:01:36,480 --> 00:01:38,540

So these are some of

the challenges that the,

26

00:01:38,540 --> 00:01:42,970

we have to solve in order to,

build a high quality web search engine.

27

00:01:44,100 --> 00:01:46,900

On the other hand, there are also some

interesting opportunities that we can

28

00:01:46,900 --> 00:01:49,930

leverage to improve search results.

29

00:01:49,930 --> 00:01:52,730

There are many additional heuristics.

30

00:01:52,730 --> 00:02:00,020

For example you know using links that

we can leverage to improve scoring.

31

00:02:00,020 --> 00:02:03,633

Now the errors that we talked about such

as the vector space model are general

32

00:02:03,633 --> 00:02:04,320

algorithms.

33

00:02:05,610 --> 00:02:11,070

And they can be applied to any search

applications, so that's, the advantage.

34

00:02:11,070 --> 00:02:15,880

On the other hand, they also don't take

advantage of special characteristics

35

00:02:15,880 --> 00:02:21,375

of pages, or documents, in the specific

applications such as web search.

36

00:02:21,375 --> 00:02:23,325

Web pages are linked with each other so

37

00:02:23,325 --> 00:02:28,645

obviously the linking is something

that we can also leverage.

38

00:02:28,645 --> 00:02:33,890

So because of these challenges and

opportunities there are new techniques

39

00:02:33,890 --> 00:02:39,110

that have been developed for web search,

or due to the need of a web search.

40

00:02:39,110 --> 00:02:43,050

One is parallel indexing and searching,

and this is to address the issue of

41

00:02:43,050 --> 00:02:48,070

scalability, in particular

Google's imaging of MapReduce

42

00:02:48,070 --> 00:02:53,570

is very inferential, and

has been very helpful in that aspect.

43

00:02:53,570 --> 00:02:56,720

Second, there are techniques

that are developed for,

44

00:02:56,720 --> 00:02:58,580

addressing the problem of spams.

45

00:02:58,580 --> 00:03:00,490

So, spam detection.

46

00:03:00,490 --> 00:03:03,620

We'll have to prevent those,

spam pages from being ranked high.

47

00:03:04,680 --> 00:03:07,660

And there are also techniques

to achieve robust ranking.

48

00:03:07,660 --> 00:03:10,520

And we're going to use a lot

of signals to rank pages so

49

00:03:10,520 --> 00:03:15,410

that it's not easy to spam the search

engine with particular tricks.

50

00:03:15,410 --> 00:03:20,710

And the third line of

techniques is link analysis.

51

00:03:21,770 --> 00:03:25,610

And these are techniques

that can allow us to

52

00:03:26,660 --> 00:03:30,780

to improve search results by

leveraging extra information.

53

00:03:30,780 --> 00:03:34,220

And in general in web

search we're going to use

54

00:03:34,220 --> 00:03:35,710

multiple features for ranking.

55

00:03:35,710 --> 00:03:41,700

Not just link analysis but

also exploiting all kinds of crawls like

56

00:03:41,700 --> 00:03:47,650

the layout of web pages or anchor text

that describes a link to another page.

57

00:03:47,650 --> 00:03:51,310

So here's a picture showing the basic

search engine technologies.

58

00:03:51,310 --> 00:03:55,700

Basically, this is the web on the left and

then user on the right side.

59

00:03:55,700 --> 00:04:00,540

And we're going to help these, this

user get access to the web information.

60

00:04:00,540 --> 00:04:04,720

And the first component is the crawler

that with the crawl pages and

61

00:04:04,720 --> 00:04:07,050

the second component is indexer.

62

00:04:07,050 --> 00:04:09,810

That will take these pages

create an invert index.

63

00:04:10,840 --> 00:04:14,320

The third component that is a retrieval,

not with the using,

64

00:04:14,320 --> 00:04:19,840

but the index to answer user's query,

by talking to the user's browser.

65

00:04:19,840 --> 00:04:23,275

And then, the search results would be,

given to the user.

66

00:04:23,275 --> 00:04:26,130

And, and then the browser

will show those results and,

67

00:04:26,130 --> 00:04:29,110

to allow the user to

interact with the web.

68

00:04:29,110 --> 00:04:32,500

So we're going to talk about

each of these component.

69

00:04:32,500 --> 00:04:37,730

First we're going to talk about

the crawler also called a spider or

70

00:04:37,730 --> 00:04:43,400

a software robot that would do something

like a crawling pages on the web.

71

00:04:43,400 --> 00:04:46,810

To build a toy crawler is relatively easy

because you just need to start with a set

72

00:04:46,810 --> 00:04:53,820

of seed pages and then fetch pages from

the web and parse these pages new links.

73

00:04:53,820 --> 00:04:56,660

And then add them to the priority of q and

74

00:04:56,660 --> 00:05:01,040

then just explore those additional links,

right.

75

00:05:01,040 --> 00:05:05,500

But to build a real crawler

actually is tricky and

76

00:05:05,500 --> 00:05:09,590

there are some complicated issues

that we have do deal with.

77

00:05:09,590 --> 00:05:13,650

For example robustness,

what if the server doesn't respond.

78

00:05:13,650 --> 00:05:19,480

What if there's a trap that generates

dynamically generated webpages that might,

79

00:05:19,480 --> 00:05:23,680

attract your crawler to keep

crawling the same site and

80

00:05:23,680 --> 00:05:26,690

to fetch dynamically generated pages.

81

00:05:26,690 --> 00:05:28,960

The results of this issue of crawling and

82

00:05:28,960 --> 00:05:32,739

you don't want to overload one particular

server with many crawling requests.

83

00:05:34,320 --> 00:05:39,020

And you have to respect the,

the robot exclusion protocol.

84

00:05:39,020 --> 00:05:42,560

You also need to handle

different types of files.

85

00:05:42,560 --> 00:05:46,230

There are images, PDF files,

all kinds of formats on the web.

86

00:05:47,530 --> 00:05:51,490

And you have to also

consider URL extensions.

87

00:05:51,490 --> 00:05:57,630

So, sometimes those are cgi scripts, and,

you know, internal references, etc., and

88

00:05:57,630 --> 00:06:03,995

sometimes, you have JavaScripts on the

page that, they also create challenges.

89

00:06:03,995 --> 00:06:07,495

And you ideally should also

recognize [INAUDIBLE] the pages

90

00:06:07,495 --> 00:06:11,495

because you don't have to

duplicate to the, those pages.

91

00:06:11,495 --> 00:06:15,365

And finally, you may be interesting

to discover hidden URLs.

92

00:06:15,365 --> 00:06:19,385

Those are URLs that may not be linked,

to any page.

93

00:06:19,385 --> 00:06:22,028

But if you truncate the URL to,

shorter pass,

94

00:06:22,028 --> 00:06:24,499

you might be able to get

some additional pages.

95

00:06:27,170 --> 00:06:29,360

So, what are the major

crawling strategies?

96

00:06:29,360 --> 00:06:32,577

In general, Breadth-First, is most common,

97

00:06:32,577 --> 00:06:36,560

because it naturally balance,

balances server load.

98

00:06:36,560 --> 00:06:42,625

You would not, keep probing

a particular server [INAUDIBLE].

99

00:06:42,625 --> 00:06:46,505

Also parallel crawling is very natural,

because this task is very easy

100

00:06:46,505 --> 00:06:51,135

to parallelise and there are some

variations of the crawling task.

101

00:06:51,135 --> 00:06:54,568

One interesting variation

is called focused crawling.

102

00:06:54,568 --> 00:06:59,860

In this kind we're going to crawl just

some pages about a particular topic.

103

00:06:59,860 --> 00:07:02,180

For example, all pages about automobiles.

104

00:07:04,110 --> 00:07:08,130

And, and, this is typically

going to start with a query,

105

00:07:08,130 --> 00:07:11,440

and then you can use the query

to get some results.

106

00:07:11,440 --> 00:07:12,690

From the major search engine.

107

00:07:12,690 --> 00:07:18,780

And then you can start it with those

results and gradually crawl more.

108

00:07:18,780 --> 00:07:25,400

So one challenge in crawling is to find

the new pages that people have created,

109

00:07:25,400 --> 00:07:30,320

and people probably are creating

new pages all the time, and this is

110

00:07:30,320 --> 00:07:35,930

very challenging if the new pages have

not been actually linked to any old page.

111

00:07:35,930 --> 00:07:40,539

If they are, then you can probably refine

them by recrawling the older page.

112

00:07:41,890 --> 00:07:45,940

So these are also some um,interesting

challenges that have to be solved.

113

00:07:47,120 --> 00:07:52,930

And finally we might face the scenario of

incremental crawling or repeated crawling.

114

00:07:52,930 --> 00:07:53,588

Right?

So your first,

115

00:07:53,588 --> 00:07:55,760

let's say if you want to be

able to web search engine.

116

00:07:55,760 --> 00:07:58,580

And you were the first to crawl

a lot of data from the web.

117

00:07:58,580 --> 00:08:03,900

And then, but then once you

have collected all the data and

118

00:08:03,900 --> 00:08:08,630

in future we just need to crawl the,

the update pages.

119

00:08:08,630 --> 00:08:13,105

You, you, in general you don't have

to re-crawl everything, right?

120

00:08:13,105 --> 00:08:14,960

Or it's not necessary.

121

00:08:16,550 --> 00:08:21,480

So, in this case you,

you go as you minimize a resource overhead

122

00:08:21,480 --> 00:08:26,370

by using minimum resource to,

to just still crawl updated pages.

123

00:08:27,430 --> 00:08:31,948

So this is after a very interesting

research question here.

124

00:08:31,948 --> 00:08:38,592

And [INAUDIBLE] research

question is that there aren't

125

00:08:38,592 --> 00:08:46,810

many standard algorithms [INAUDIBLE] for

doing this, this task.

126

00:08:46,810 --> 00:08:48,850

Right?

But in general, you can imagine,

127

00:08:48,850 --> 00:08:51,080

you can learn from the past experience.

128

00:08:52,680 --> 00:08:53,640

Right.

129

00:08:53,640 --> 00:08:58,330

So the two major factors that

you have to consider are first,

130

00:08:58,330 --> 00:09:00,760

will this page be updated frequently?

131

00:09:00,760 --> 00:09:03,610

And do I have to crawl this page again?

132

00:09:03,610 --> 00:09:07,480

If the page is a static page

that hasn't been changed for

133

00:09:07,480 --> 00:09:11,510

months you probably don't have

to re-crawl it everyday, right?

134

00:09:11,510 --> 00:09:14,600

Because it's unlikely that it

will be changed frequently.

135

00:09:14,600 --> 00:09:16,720

On the other hand if it's you know,

136

00:09:16,720 --> 00:09:19,980

sports score page that gets

updated very frequently and

137

00:09:19,980 --> 00:09:25,780

you may need to re-crawl it maybe

even multiple times, on the same day.

138

00:09:25,780 --> 00:09:31,120

The other factor to consider is,

is this page frequently accessed by users?

139

00:09:31,120 --> 00:09:35,190

If it, if it is,

that means it's a high utility page, and

140

00:09:35,190 --> 00:09:39,849

then thus it's more important to

ensure such a page to be fresh.

141

00:09:40,960 --> 00:09:45,750

Compare it with another page that has

never been fetched by any users for

142

00:09:45,750 --> 00:09:46,620

a year.

143

00:09:46,620 --> 00:09:49,750

Than, even though that page

has been changed a lot, then,

144

00:09:49,750 --> 00:09:55,810

it's probably not necessary to crawl that

page or at least it's not as urgent as,

145

00:09:55,810 --> 00:10:01,780

to maintain the freshness of

frequently accessed page by users.

146

00:10:01,780 --> 00:10:02,810

So to summarize,

147

00:10:02,810 --> 00:10:06,070

web search is one of the most important

applications of text retrieval.

148

00:10:06,070 --> 00:10:08,932

And there are some new challenges

particularly scalability,

149

00:10:08,932 --> 00:10:10,660

efficiency, quality information.

150

00:10:10,660 --> 00:10:15,480

There are also new opportunities

particularly, rich link information and

151

00:10:15,480 --> 00:10:16,780

layout, et cetera.

152

00:10:18,040 --> 00:10:22,140

Crawler is an essential component

of web search applications.

153

00:10:22,140 --> 00:10:24,290

And, in general,

we can classify two scenarios.

154

00:10:24,290 --> 00:10:28,730

Once is initial crawling and

here we want to have complete crawling

155

00:10:30,100 --> 00:10:33,160

of the web if you are doing

a general search engine or

156

00:10:33,160 --> 00:10:37,550

focus crawling if you want to just

target it at a certain type of pages.

157

00:10:38,600 --> 00:10:43,280

And then there is another scenario that's

incremental updating of the crawl data or

158

00:10:43,280 --> 00:10:44,740

incremental crawling.

159

00:10:44,740 --> 00:10:47,830

In this case you need to

optimize the resource.

160

00:10:47,830 --> 00:10:52,170

For to use minimum resource

we get the [INAUDIBLE]

161

00:10:54,381 --> 00:11:04,381

[MUSIC].