

CS 589 Fall 2021

Text Mining and Information Retrieval

**Monday 6:30-9:00
Babbio 122**

All zoom links in Canvas



photo: <https://www.scubedstudios.com/information-retrieval/>



Instructor: Susan Liu

xliu127@stevens.edu

OH: Tuesday 3-5



TA: Guanqun Yang

gyang16@stevens.edu

Friday 3-5

CS589 Lecture 1: Overview

- Logistics
 - COVID requirement
 - Pre-requisite
 - Final grade calculation
 - Q & A
- Lecture 1
 - What is information retrieval?
 - History of IR
 - Earlier IR models
 - TF-IDF, vector space model

Hybrid teaching requirements!

- Mask is required in classroom
- Only come to the classroom if
 - You are fully vaccinated
 - Show a negative testing result within 5 days



Prerequisite

- **Recommended:** A good knowledge on statistics and probability
- **Required:** Being comfortable with Python programming (**pop quiz today**)
- Undergrad: CS115, CS284, MA 232, MA 222
- Contact the instructor if you aren't sure

Grading calculator

- 4 Colab-based assignments - 40%
- Midterm (in class, online) - 30%
- Project (paper review, proposal, presentation, final report) - 30%

Late policy

- Submit within 24 hours of deadline - 90%, within 48 hours - 70%, 0 if code not compile
- Late by over 48 hours are generally not permitted

Plagiarism policy

- **We have a very powerful plagiarism detection system, do not take the risk**
- Cheating case in CS284
 - A student put all his homework on a GitHub public repo
 - In the end, we found 8+ students copied his code



Thu 5/21/2020 1:38 PM

To: Xueqing Liu

Hello Prof. Liu –

I see that [REDACTED] got an F in CS284C for S20. She has done well in all her other classes and found this F to be shocking. I have to reach out to her. It would help me if you can provide some feedback into what went wrong for her. Any feedback you can provide will be helpful.

Thank you,



Final Project

- Week 7** Students choose a topic; for each topic, they pick 2-3 coherent papers, and submit a “peer review” for the paper
- Week 11** Students who share the same interest are categorized into groups; each group propose a research topic motivated by their survey
- Week 15** Deliver a presentation in Week 14
- Week 16** Submit their implementation (code in Python) as well as an 8-page academic paper as their final project.

Final Project

- Example report from last year's course: fake news detection

	CS589 Final Report	
1		59
2		60
3		61
4		62
5		63
6		64
7		65
8		66
9		67
10	ABSTRACT	68
11	While there have been many innovations in fact-checking and	69
12	fake news detection, there has been little innovation in providing	70
13	articles to refute fake news. Our research is meant to incorporate a	71
14	system which solves both the problem of determining whether a	72
15	news article is deemed as fake news, as well as providing reputable	73
16	articles which provide information that disproves the claims made	74
17	in the fake news article. We plan to also do a comparative analysis	75
18	on the varying approaches to distinguish fake news from validated	76
19	news, and provide counter measures for when fake news has been	77
20	detected. This serves as a proof of concept for a course of action	78
21	that may be possible for social media outlets to take to ensure the	79
22	legitimacy of information that gets presented to their users, and	80
23	curtail the spread of fake news.	81
24	KEYWORDS	82
25	datasets, neural networks, attention, CNN, Transformers, Fake	83
26	News Detection, Topic Modelling	84
27		85
28	ACM Reference Format:	86
29	Ian Gomez and Daniel Shapiro. 2020. CS589 Final Report. In <i>Proceedings</i>	87
30	of ACM Conference (Conference'20). ACM, New York, NY, USA, 8 pages.	88
31	https://doi.org/10.1145/nnnnnnnn.nnnnnnnn	89
32		90
33	1 INTRODUCTION	91

Where can I find all the information?

Date	Slides/Readings	Homework/Exams
Week 1 of Aug 30	<p>History of IR, vector space model slides pptx</p> <p>Reading:</p> <ul style="list-style-type: none">o Scoring, term weighting and the vector space model (lecture from Stanford IR course)o The history of information retrieval researcho Pivoted length normalizationo A Formal Study of Information Retrieval Heuristics	
Week 2 of Sept 13	<p>Probabilistic ranking principle, Probabilistic/LM-based retrieval slides ppt</p> <p>Reading:</p> <ul style="list-style-type: none">o RSJ model without relevance judgmento A Study of Retrieval Models for Long Documents and Queries in Information Retrieval (WWW 2016)o A language model approach to information retrievalo Notes on the KL-divergence retrieval formula and Dirichlet prior smoothingo Retrieval Models for Question and Answer Archiveso Two-stage language models for information retrievalo Model-based feedback in the language modeling approach to information retrieval	

Ask ALL questions in Canvas -> Piazza

The screenshot shows the Piazza class dashboard for the course CS 589. The top navigation bar includes links for Apps, ase, papers, useful links, online teaching, undgrd adv, proposal, students, linux, stackoverflow, and 生活. The main menu features Q & A, Resources, Statistics, and Manage Class. On the right, there are buttons for Buy a License, Switch to contribution model, and a user profile for Susan L.

The left sidebar displays a list of pinned posts:

- Search for Teammates! (8/27/21)
- Introduce Piazza to your stu... (Fri)
- Get familiar with Piazza (Fri)
- Tips & Tricks for a successf... (Fri)
- Welcome to Piazza! (Fri)

The central "Class at a Glance" section shows the following statistics:

- 5 unread posts**
- no unanswered questions**
- no unresolved followups**
- license status**: pending instructor license (21 days left)
- 5 total posts**
- 5 total contributions**
- 0 instructors' responses**
- 0 students' responses**
- n/a avg. response time**

The "Student Enrollment" section indicates 6 enrolled students, with a progress bar showing 6 out of 40. There are links to download the app from the App Store and Google Play.

A "Share Your Class" button is located at the bottom right of the dashboard.

Information Retrieval: Real life example



"I need a book on
American history for my
thesis research!"

Information need: to study
about the history of the civil
war

Information Retrieval: Google

cs 589 stevens

All Images Maps News Videos More

About 3,220,000 results (0.58 seconds)

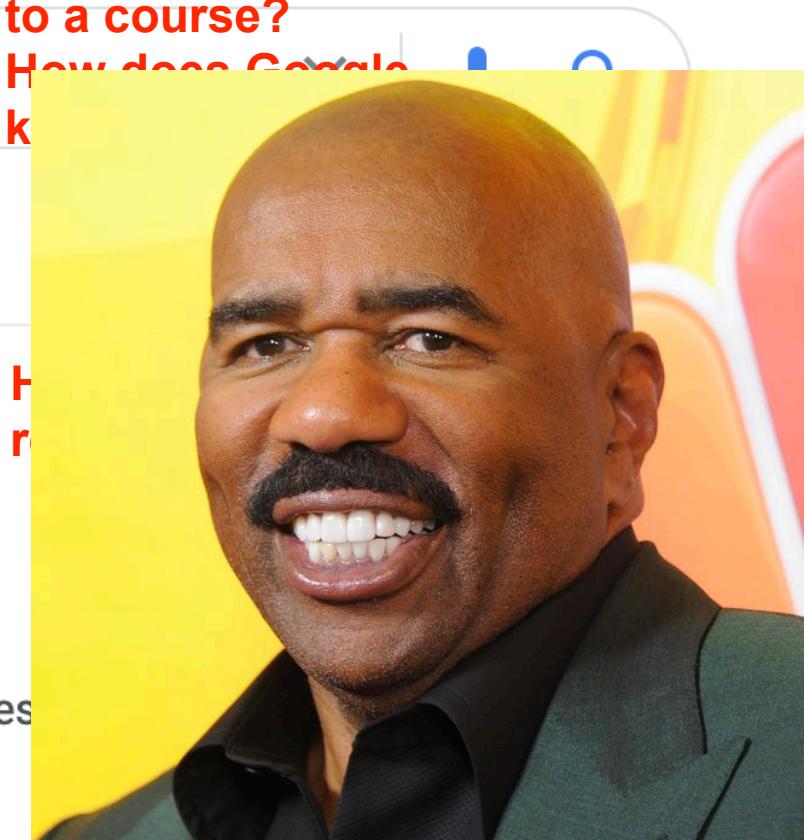
[www.cs.stevens.edu › ~xliu127 › teaching › cs589_20f](http://www.cs.stevens.edu/~xliu127/teaching/cs589_20f/) ▾

cs589

CS 589: Text Mining and Information Retrieval. Home | Canvas | Resources

Stevens' guidelines on the coronavirus emergency (COVID-19), ...

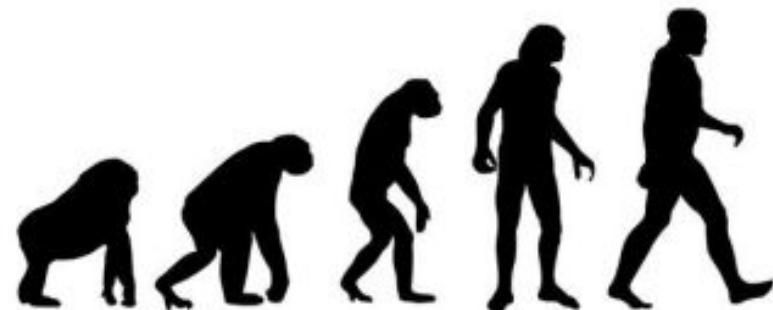
How does Google know cs 589 refers to a course?
How does Google k



Information Retrieval Techniques

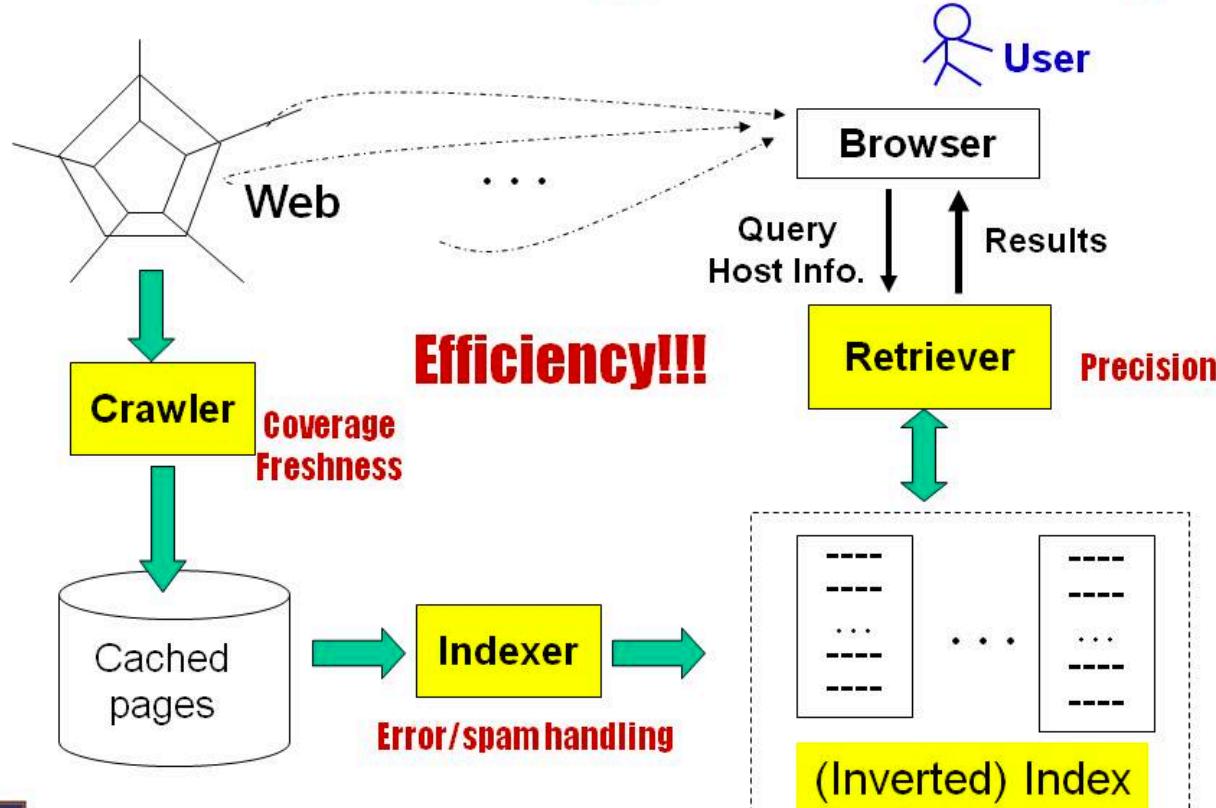
“Because the systems that are accessible today are so easy to use, it is tempting to think the technology behind them is similarly straightforward to build. This review has shown that the route to creating successful IR systems required much innovation and thought over a long period of time. “

— The history of Information Retrieval Research, Mark Sanderson and Bruce Croft



Information Retrieval Techniques

Basic Search Engine Technologies



Query understanding, personalization, results diversification, result page optimization, etc.



Making sure the results are returned to users fast

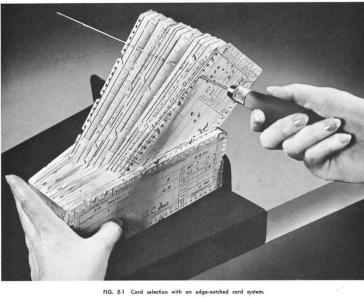
A Brief History of IR

300 BC



Callimachus: the first library catalog

1950s



Punch cards, searching at 600 cards/min

1958

Cranfield evaluation methodology; word-based indexing

1960s

building IR systems on computers; relevance feedback

1970s

TF-IDF; probability ranking principle

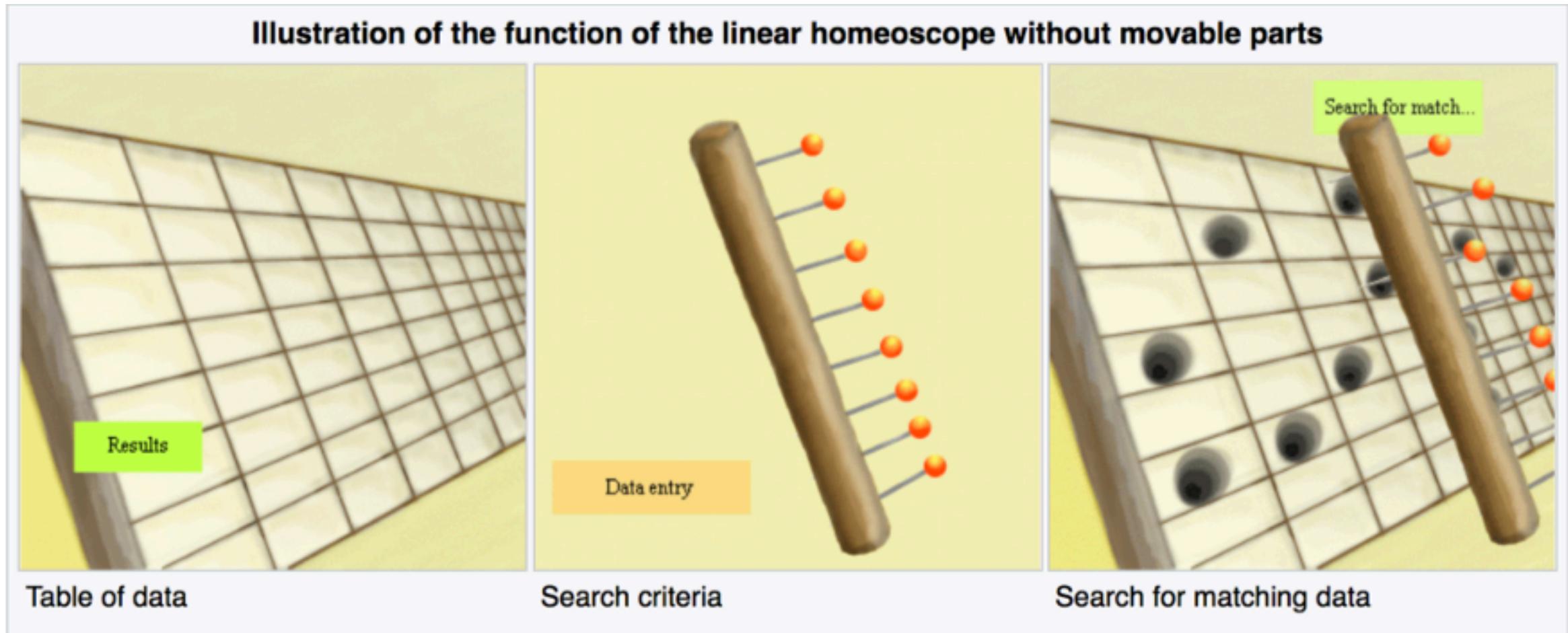
1980s

TREC; learning to rank; latent semantic indexing

1990 - now

web search; supporting natural language queries;

The Boolean retrieval system



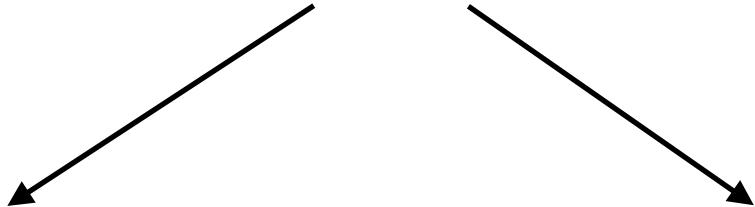
The Boolean retrieval system

- e.g., *SELECT * FROM table_computer WHERE price < \$500 AND brand = "Dell"*
- Primary commercial retrieval system for 3 decades
- Many systems today still use the boolean retrieval system, i.e., eCommerce search, etc.
- **Advantage:** Returns exactly what you want
- **Disadvantage:**
 - can only specify queries based on the pre-defined categories
 - two few / two many queries

The Cranfield experiment (1958)

- Imagine you need to help users search for literatures in a digital library, how would you design such a system?

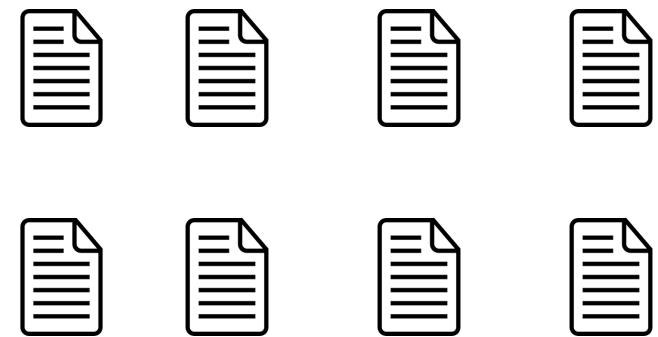
computer science



artificial intelligence

bioinformatics

query = “subject = AI & subject = bioinformatics”



system 1: the Boolean retrieval system

20

The Cranfield experiment (1958)

- Imagine you need to help users search for literatures in a digital library, how would you design such a system?

Document-term matrix

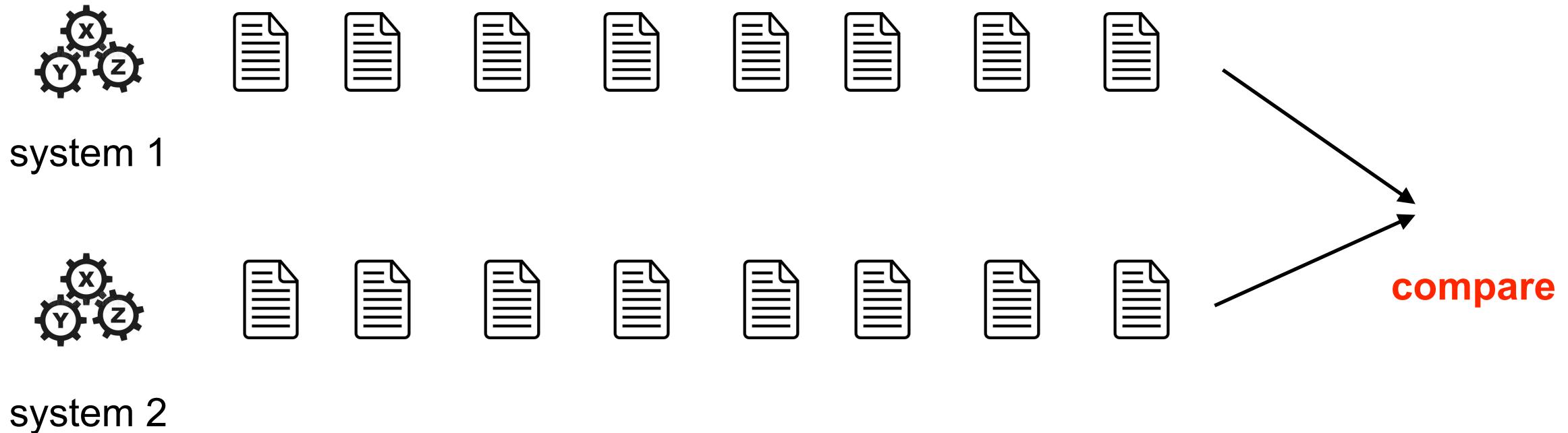
	intelligence	book	the	cat	artificial	dog	business
Doc1	0	1	3	1	0	1	0
Doc2	1	0	0	0	0	0	1
query	1	0	1	0	1	0	0

query = “the artificial intelligence”

bags of words representation

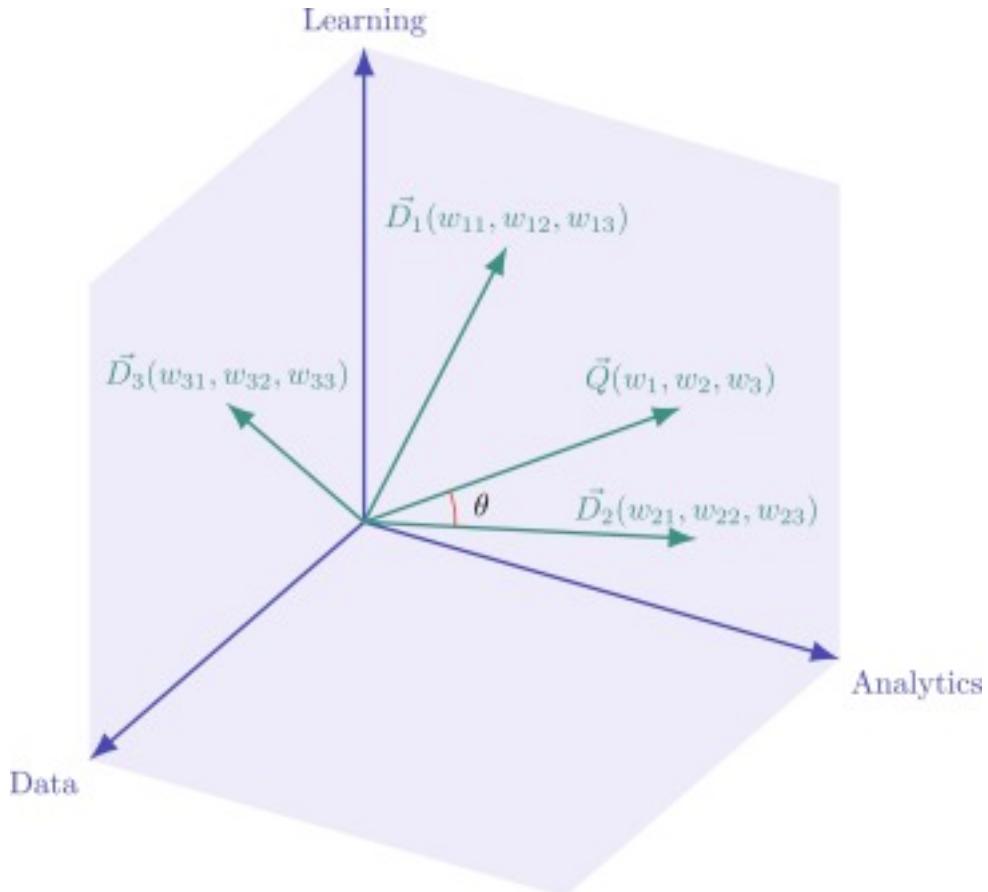
system 2: indexing documents by lists of words

The Cranfield experiment (1958)



Boolean retrieval system < word indexing system

Word indexing: vector-space model



- Represent each document/query as a vector
- The similarity = cosine score between the vectors

Term frequency

Document-term matrix

	intelligence	book	the	cat	artificial	dog	business
Doc1	0	1	3	1	0	1	0
Doc2	1	0	0	0	0	0	1
query	1	0	1	0	1	0	0

$$tf(w, d) = count(w, d)$$

$$d_i = [count(w_1, d_i), \dots, count(w_n, d_i)]$$

- $d_1 = \text{"the cat, the dog, and the book"}$ query = “the artificial intelligence”
[0, 1, 3, 1, 0, 1, 0] • $q = [1, 0, 1, 0, 1, 0, 0]$
- $d_2 = \text{"business intelligence"}$
[1, 0, 0, 0, 0, 0, 1]

Vector space model

Document-term matrix

	intelligence	book	the	cat	artificial	dog	business
Doc1	0	1	3	1	0	1	0
Doc2	1	0	0	0	0	0	1
query	1	1	1	0	1	0	0

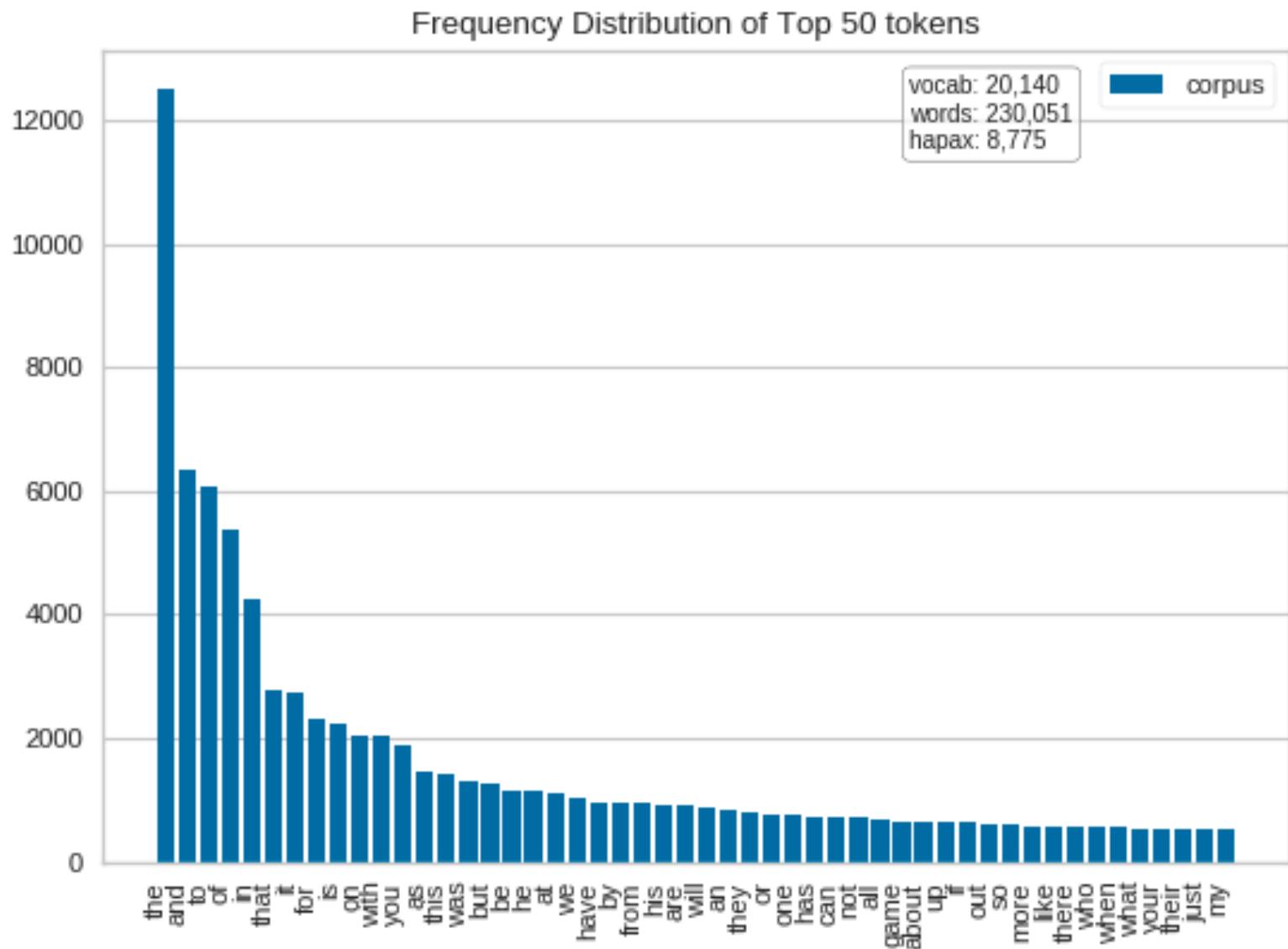
- To answer the query:
 - “the artificial intelligence book”
 - $d_1 = [0, 1, 3, 1, 0, 1, 0]$
 - $d_2 = [1, 0, 0, 0, 0, 0, 1]$
- $q = [1, 1, 1, 0, 1, 0, 0]$

$$score(q, d) = \frac{q \cdot d}{\|q\| \cdot \|d\|}$$

TF-only representations is inaccurate

- Documents are dominated by words such as “the” “a”
- These words do not carry any meanings, nor do they discriminate between documents
 - $q = \text{"the artificial intelligence book"}$ $score(q, d_1) = 0.8164$
 - $d_1 = \text{"the cat, the dog, and the book"}$ $score(q, d_2) = 0.3535$
 - $d_2 = \text{"business intelligence"}$ $\Rightarrow score(q, d_1) > score(q, d_2)$

Zipf's law distribution of words



the and to that for on with you as this was but be at we have by from his will an they or one has can not all game about up if out so more like there who when what your their just my

Stop words

```
> stopwords("english")
[1] "i"          "me"         "my"        "myself"      "we"
[6] "our"        "ours"       "ourselves" "you"        "your"
[11] "yours"      "yourself"   "yourselves" "he"         "him"
[16] "his"        "himself"   "she"       "her"        "hers"
[21] "herself"    "it"         "its"       "itself"     "they"
[26] "them"        "their"     "theirs"    "themselves" "what"
[31] "which"       "who"        "whom"      "this"       "that"
[36] "these"       "those"     "am"        "is"         "are"
[41] "was"         "were"      "be"        "been"      "being"
[46] "have"       "has"        "had"      "having"    "do"
```

Desiderata for a good ranking function

- Rule 1: If a word appears everywhere, it should be penalized
- Rule 2: If a word appears in the same document multiple times, its importance should not grow linearly
- $q = \text{"artificial intelligence"}$
- $d_1 = \text{"Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism"}$
- $d_2 = \text{"Artificial intelligence was founded as an academic discipline in 1955, artificial intelligence"}$

d2 is not twice more relevant than d1

Inverse-document frequency

- **Inverse-document frequency**: penalizing a word's TF based on its document frequency

$$IDF(w) = \log N/df(w)$$

$$q(d, w) = TF(d, w) \times IDF(w)$$

- q = “the **artificial intelligence** book”
- d1 = “the cat, the doc, and the book”
- d2 = “business **intelligence**”

TF-IDF weighting

$$score(q, d_1) = 0.8164 \rightarrow 0.2041$$

$$score(q, d_2) = 0.3535 \rightarrow 0.3535$$

$$\Rightarrow score(q, d_1) < score(q, d_2)$$

Term frequency reweighing

- **Term frequency reweighing**: penalizing a word's TF based on the TF itself
- If a word appears in the same document multiple times, its importance should not grow linearly

Max TF
normalization

$$tf(w, d) = \alpha + (1 - \alpha) \frac{count(w, d)}{\max_v count(v, d)}$$

Log scale
normalization

$$tf(w, d) = \begin{cases} 1 + \log count(w, d) & count(w, d) > 0 \\ 0 & o.w. \end{cases}$$

Term-frequency reweighing

- **Logarithmic normalization**

**Log scale
normalization**

$$tf(w, d) = \begin{cases} 1 + \log count(w, d) & count(w, d) > 0 \\ 0 & o.w. \end{cases}$$

$$score(q, d_1) = 0.8164 \rightarrow 0.7618$$

$$score(q, d_1) = 0.3535 \rightarrow 0.3535$$

- q = “the **artificial intelligence** book”
- d1 = “the cat, the doc, and the book”
- d2 = “business **intelligence**”

Retrieving short documents vs. retrieving long documents

- The difference between retrieving short documents and long documents
 - Longer documents cover more topics, so the query may match a small subset of the vocabulary
 - Longer documents need to be considered differently

$q = \text{"artificial intelligence"}$

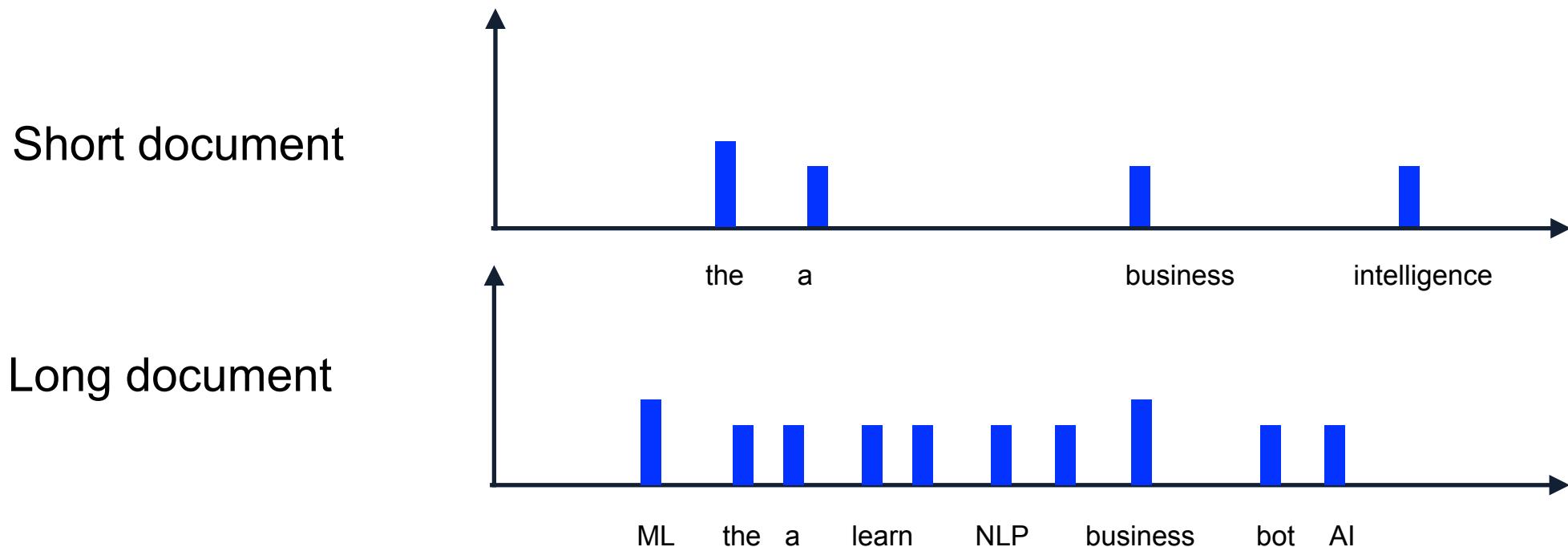
$d_1 = \text{"artificial intelligence book"}$

$d_2 = \text{"Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism"}$

$$score(q, d_1) > score(q, d_2)$$

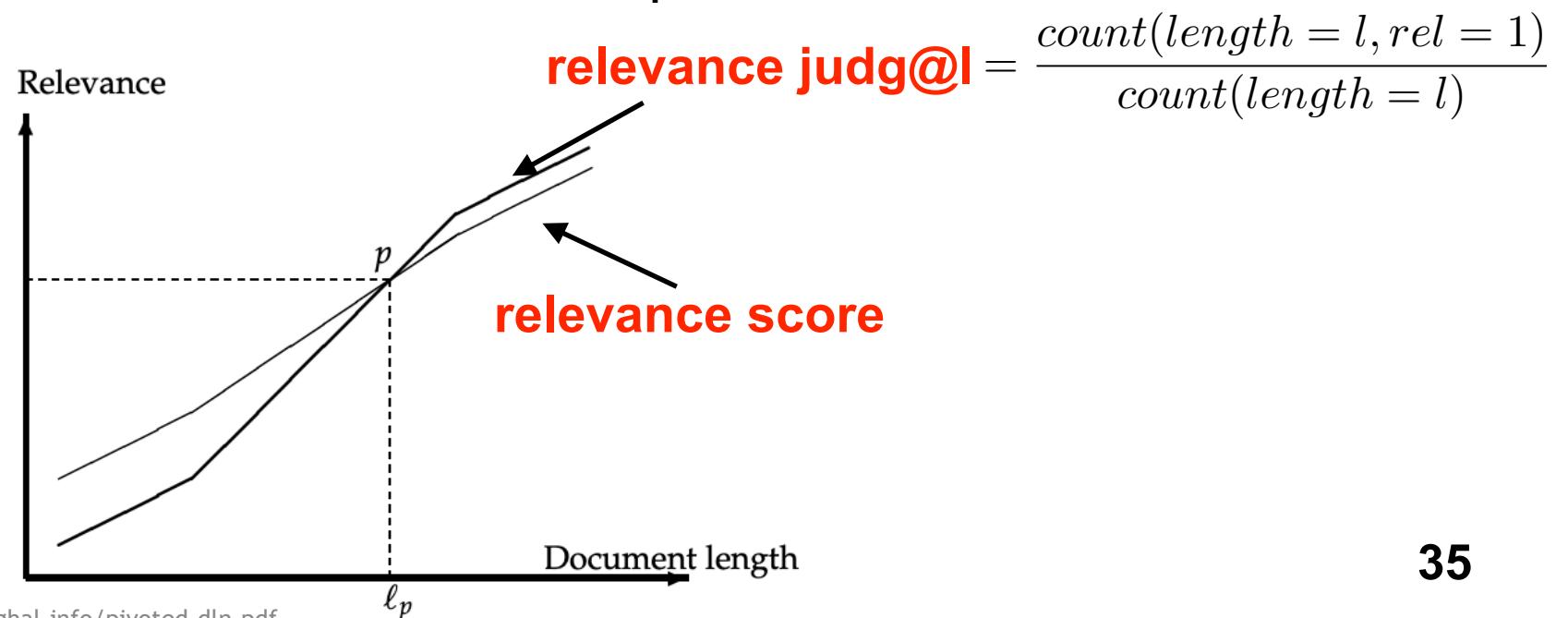
Retrieving short documents vs. retrieving long documents

- The difference between retrieving short documents and long documents
 - Longer documents cover more topics, so the query may match a small subset of the vocabulary
 - Longer documents need to be considered differently



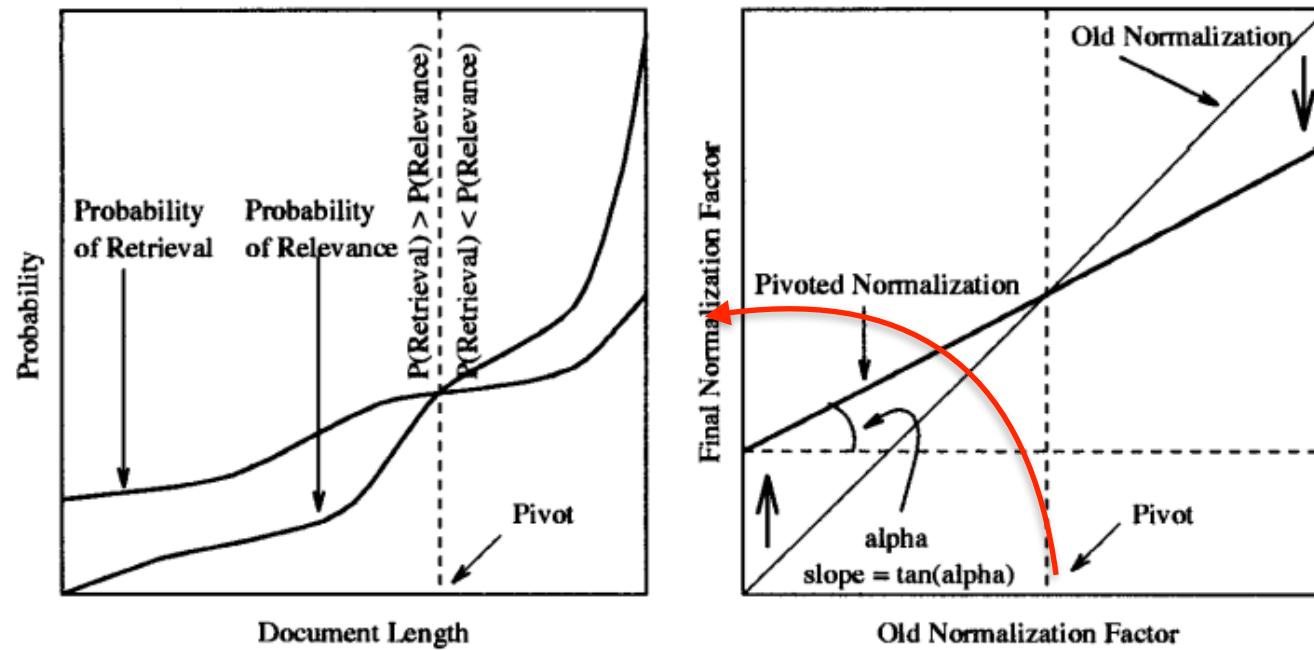
Document length pivoting

- For each query q and each document d , compute their relevance score $\text{score}(q, d)$
- Manually evaluate the relevance between q and d



Document length pivoting

- Rotate the relevance score curve, such that it most closely align with the relevance judgement curve



$$y = x$$

$$\text{pivot} = \text{pivot} \times \text{slope} + \text{intercept}$$

$$\text{pivot}_\text{normalized} = (1.0 - \text{slope}) \times \text{pivot} + \text{slope} \times \text{oldnormalization}$$

Document length pivoting

- Rotate the relevance score curve, such that it most closely align with the relevance judgement curve

$$\frac{tf \cdot idf \text{ weight}}{(1.0 - slope) \times pivot + slope \times old \text{ normalization}}$$

$$\frac{tf \cdot idf \text{ weight}}{1 + \frac{slope}{(1.0 - slope) \times pivot} \times old \text{ normalization}} \quad \leftarrow \text{fixing pivot to avgdl}$$

$$(1.0 - slope) + slope \times \frac{old \text{ normalization}}{\text{average old normalization}} \quad \leftarrow \quad 1 - b + b \frac{|dl|}{|avgdl|}$$

standard formulation for doc length normalization

37