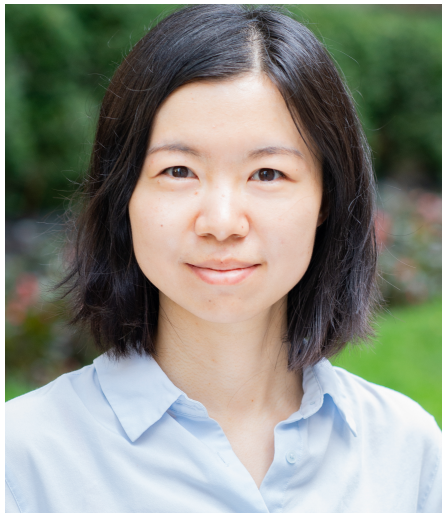
CS 589 Fall 2021

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

TA: Guanqun Yang

gyang16@stevens.edu

OH: Tuesday 3-5

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 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

Ian Gomez*
igomez1@stevens.edu
Stevens Institute of Technology

Daniel Shapiro†
dshapir1@stevens.edu
Stevens Institute of Technology

ABSTRACT

While there have been many innovations in fact-checking and fake news detection, there has been little innovation in providing articles to refute fake news. Our research is meant to incorporate a system which solves both the problem of determining whether a news article is deemed as fake news, as well as providing reputable articles which provide information that disproves the claims made in the fake news article. We plan to also do a comparative analysis on the varying approaches to distinguish fake news from validated news, and provide counter measures for when fake news has been detected. This serves as a proof of concept for a course of action that may be possible for social media outlets to take to ensure the legitimacy of information that gets presented to their users, and curtail the spread of fake news.

KEYWORDS

datasets, neural networks, attention, CNN, Transformers, Fake News Detection, Topic Modelling

ACM Reference Format:
Ian Gomez and Daniel Shapiro. 2020. CS589 Final Report. In *Proceedings of ACM Conference (Conference'20).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

that should have been taken when dealing with the novel Covid-19 epidemic. Both of which had damaging effects on shaping the recent history of our nation.

Previous research has mostly been focused on the detection of fake news but does not address how to prevent the spread of fake news. Therefore, the main purpose of our project is to provide a model that not only identifies fake news but also provides articles to refute the news being spread. To define this we are looking at articles which are fact checked by fact-checking websites, and have information that is contradictory to articles which are deemed as false. An example of a fact checking website would would be Politifact. This approach aims to educate end-users, and further prevent the spread of fake news. We plan to do this through two main methods. Initially we classify whether an article is fake or legitimate news. Secondly, if the classified article is determined to be fake news, we then perform a recommendation, where the goal of this recommendation to find a fact checked article which refutes the false claims made in the original fake news article. The reasoning for this is that we hope by providing informative content to the user, they will be able to see for themselves whether something is real before spreading it. After being confronted with legitimate counter-arguments, spreaders of fake news are more likely to remove their post and also become less likely to spread fake news[1].

We hope that after providing analysis on multiple models to discover which methodology has the highest success rate at detect-

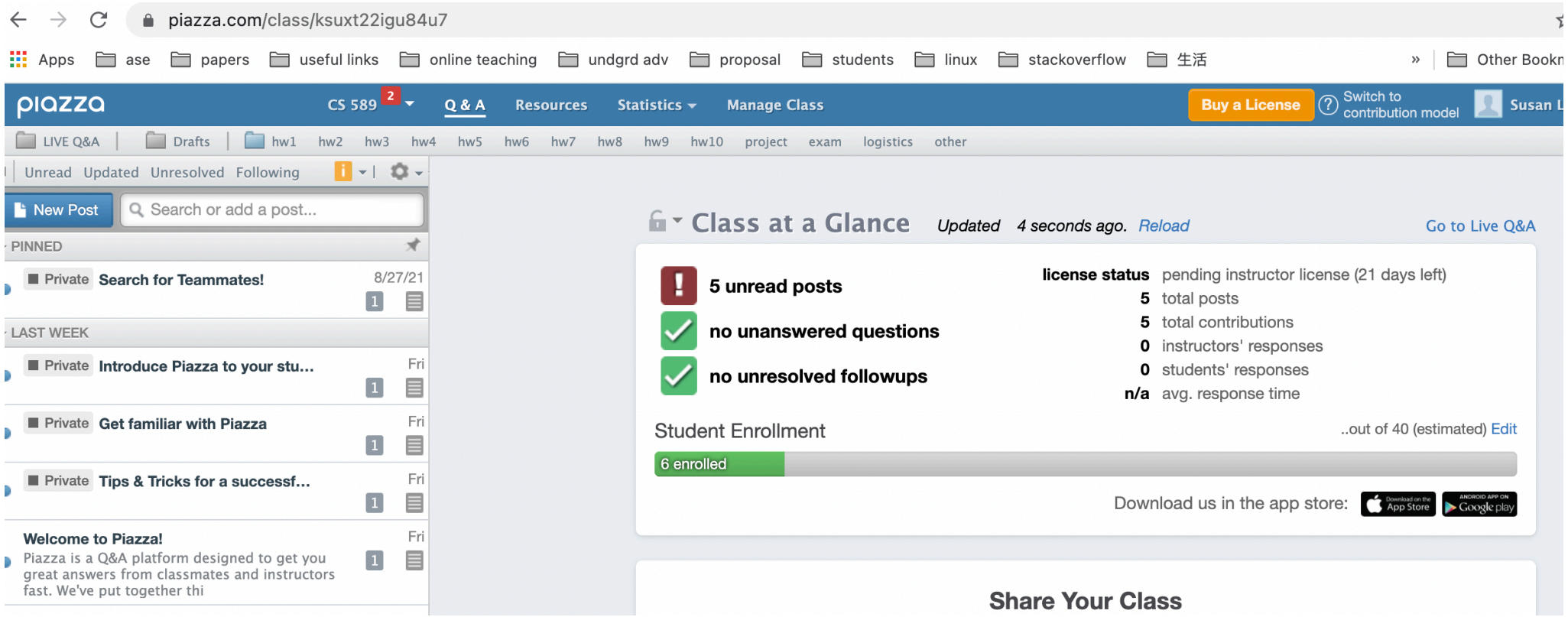
Where can I find all the information?

liususan091219.github.io/teaching/cs589_21f/cs589.html

Apps | ase | papers | useful links | online teaching | undgrd adv | proposal | students | linux | stackoverflow | 生活 | » | Other Bookmarks

Date	Slides/Readings	Homework/Exams
Week 1 of Aug 30	History of IR, vector space model slides pptx Reading: ◦ Scoring, term weighting and the vector space model (lecture from Stanford IR course) ◦ The history of information retrieval research ◦ Pivoted length normalization ◦ A Formal Study of Information Retrieval Heuristics	
Week 2 of Sept 13	Probabilistic ranking principle, Probabilistic/LM-based retrieval slides ppt Reading: ◦ RSJ model without relevance judgment ◦ A Study of Retrieval Models for Long Documents and Queries in Information Retrieval (WWW 2016) ◦ A language model approach to information retrieval ◦ Notes on the KL-divergence retrieval formula and Dirichlet prior smoothing ◦ Retrieval Models for Question and Answer Archives ◦ Two-stage language models for information retrieval ◦ Model-based feedback in the language modeling approach to information retrieval	

Ask ALL questions in ~~Canvas~~ -> Piazza

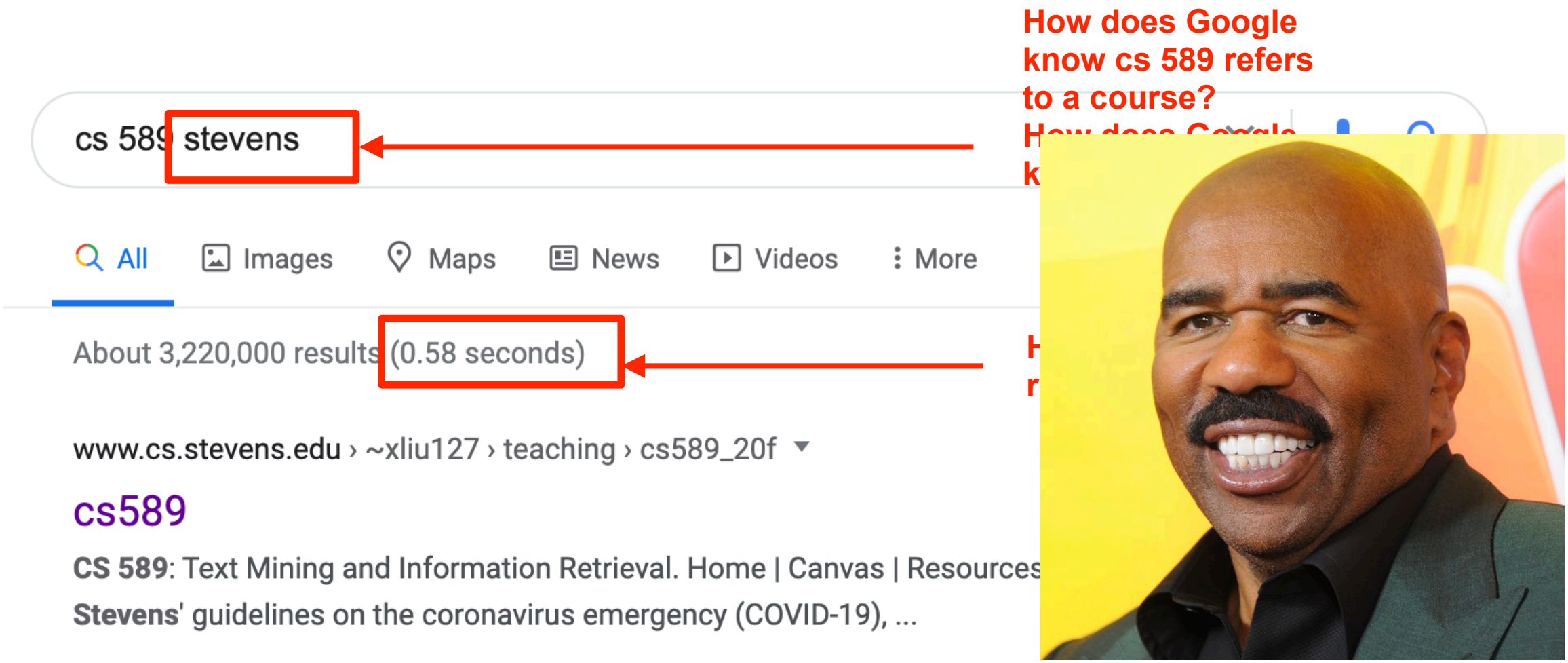


Information Retrieval: Real life example

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

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

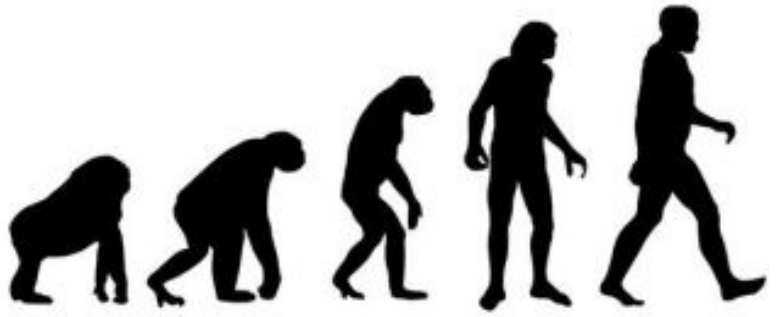
Information Retrieval: Google



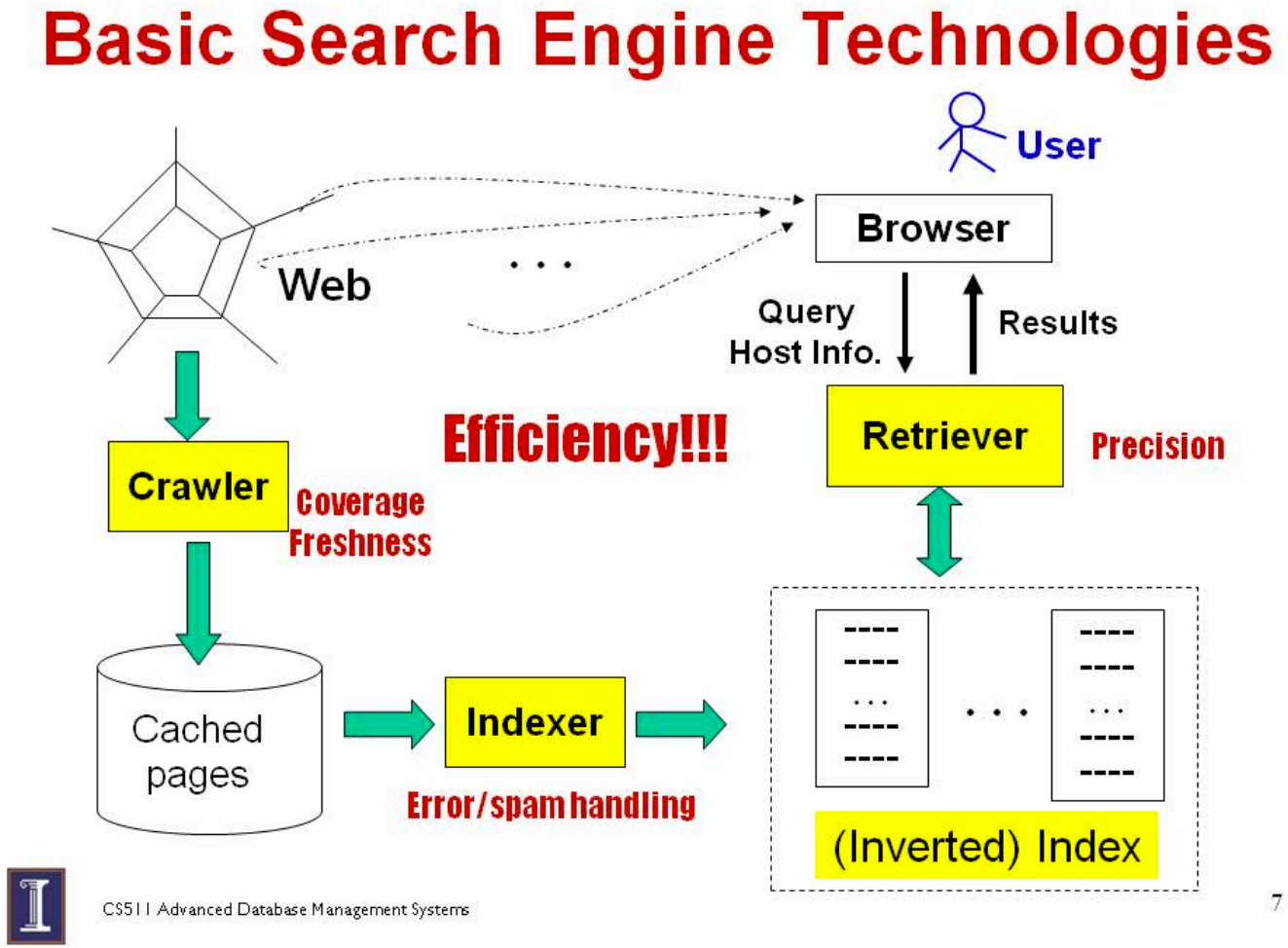
Information Retrieval Techniques

“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



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

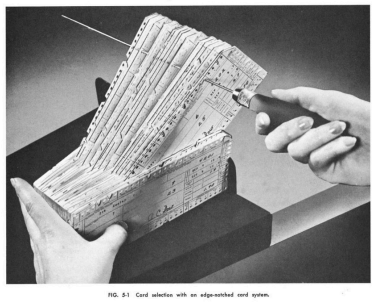
Making sure the results are returned to users fast

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

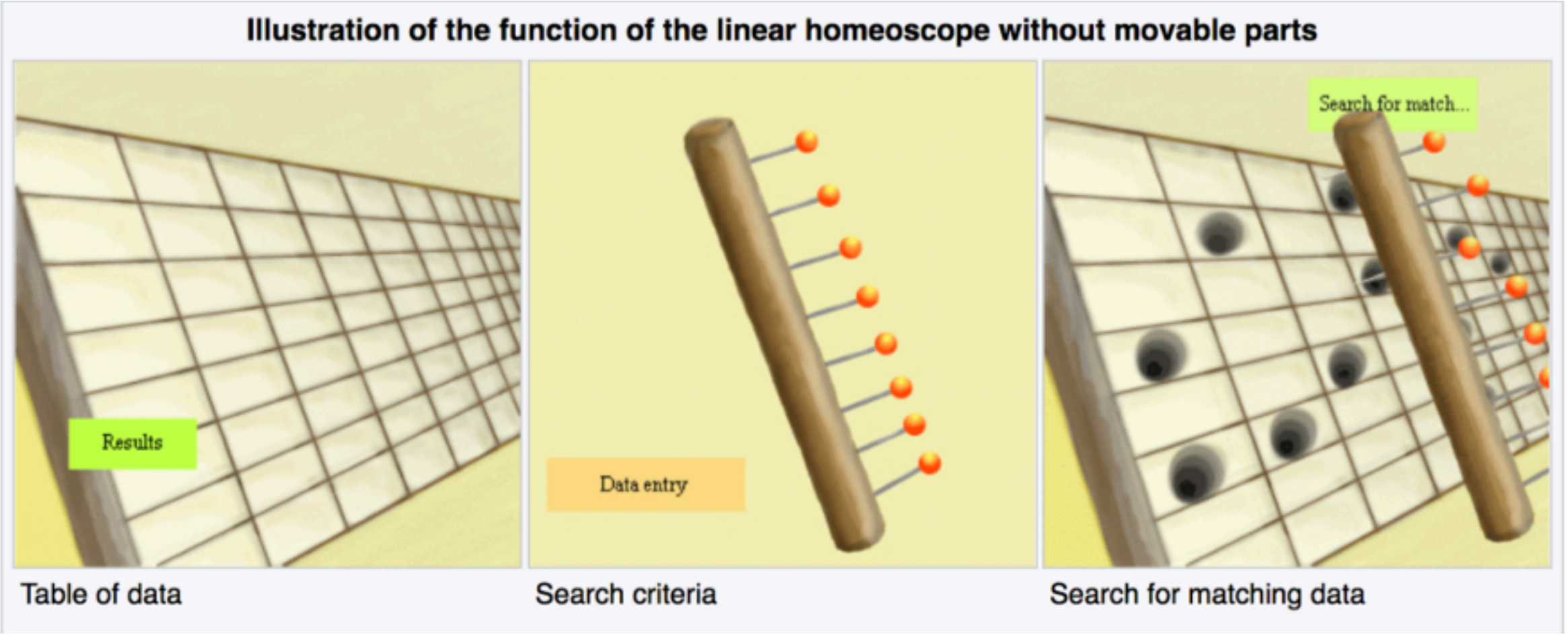
1960s building IR systems on computers; relevance feedback

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

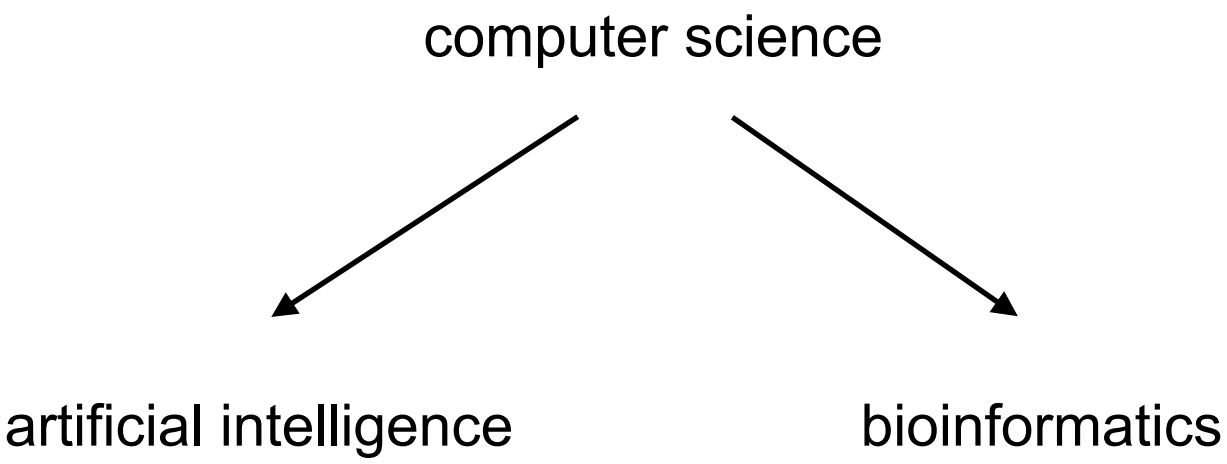
- *e.g., SELECT * FROM table_computer WHERE price < $500 AND brand = "Dell"*
- Primary commercial retrieval system for 3 decades
- Many systems today still use the boolean retrieval system, i.e., eCommerce search, etc.

- **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?



query* = *"subject = AI & subject = bioinformatics"

system 1: the Boolean retrieval system

On the history of IR: https://journals.sagepub.com/doi/abs/10.1177/0165551507086989?journalCode=jisb

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

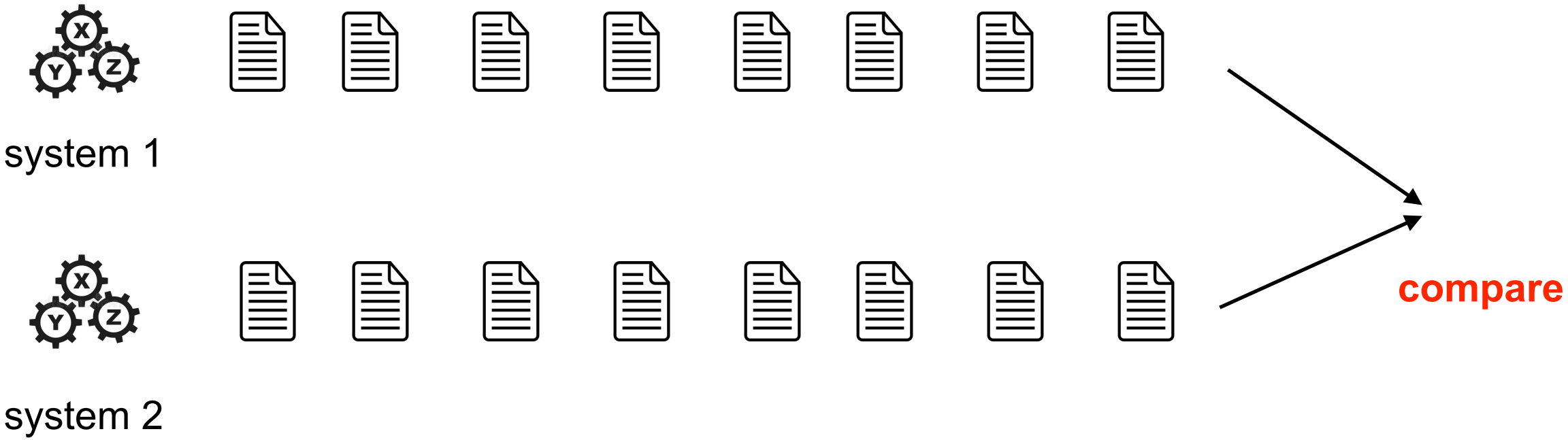
	intelligence	book	the	cat	artificial	dog	business
Doc1	0	1	3	1	0	1	0
Doc2	1	0	0	0	0	0	1
query	1	0	1	0	1	0	0

***query* =** *" the artificial intelligence"*

bags of words representation

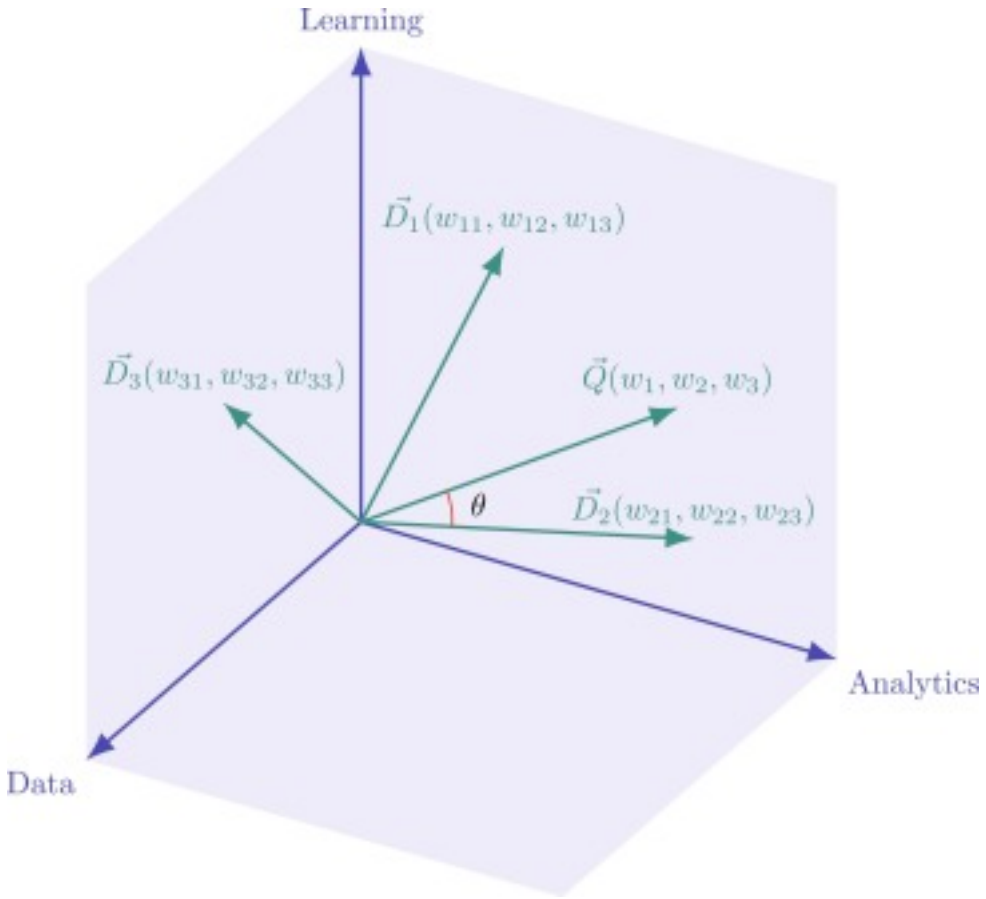
system 2: indexing documents by lists of words

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

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

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

- d1= “the cat, the dog, the book”
  [0, 1, 3, 1, 0, 1, 0]
- d2 = “business intelligence”
  [1, 0, 0, 0, 0, 0, 1]
- query = “the artificial intelligence”
- q = [1, 0, 1, 0, 1, 0, 0]

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”
- q = [1, 1, 1, 0, 1, 0, 0]

- d1 = [0, 1, 3, 1, 0, 1, 0]
- d2 = [1, 0, 0, 0, 0, 0, 1]

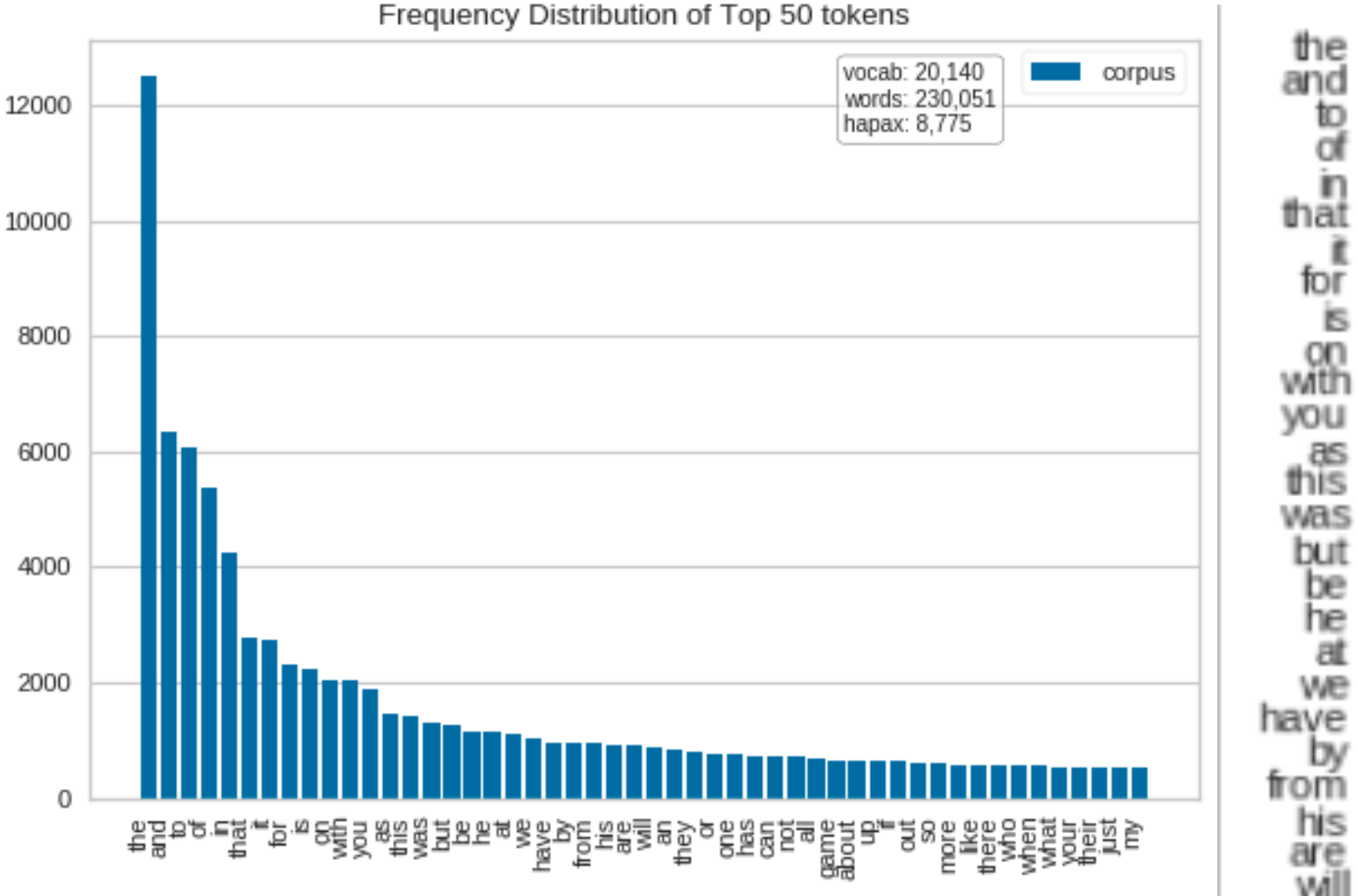
$$score(q, d) = \frac{q \cdot d}{\|q\| \cdot \|d\|}$$

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 = “the **artificial intelligence** book”
  - d1 = “the cat, the dog, the book”
  - d2 = “business **intelligence**”
  - d3= "the artificial world"

$$score(q, d_1) = 0.5773$$
$$score(q, d_2) = 0.3535$$
$$\Rightarrow score(q, d_1) > score(q, d_2)$$

Zipf’s law distribution of words



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, it's importance should not grow linearly
- q = "**artificial intelligence**"
- d1 = ""**Artificial intelligence** was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism"
- d2 = ""**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, the book"
- d2 = "business **intelligence**"
- d3 = "the artificial world"

TF-IDF weighting

$$score(q,d_1) = 0.5773 \rightarrow 0.3869$$
$$score(q,d_2) = 0.3535 \rightarrow 0.3992$$
$$\Rightarrow score(q,d_1) < score(q,d_2)$$

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, it's 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

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

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 = "artificial intelligence"

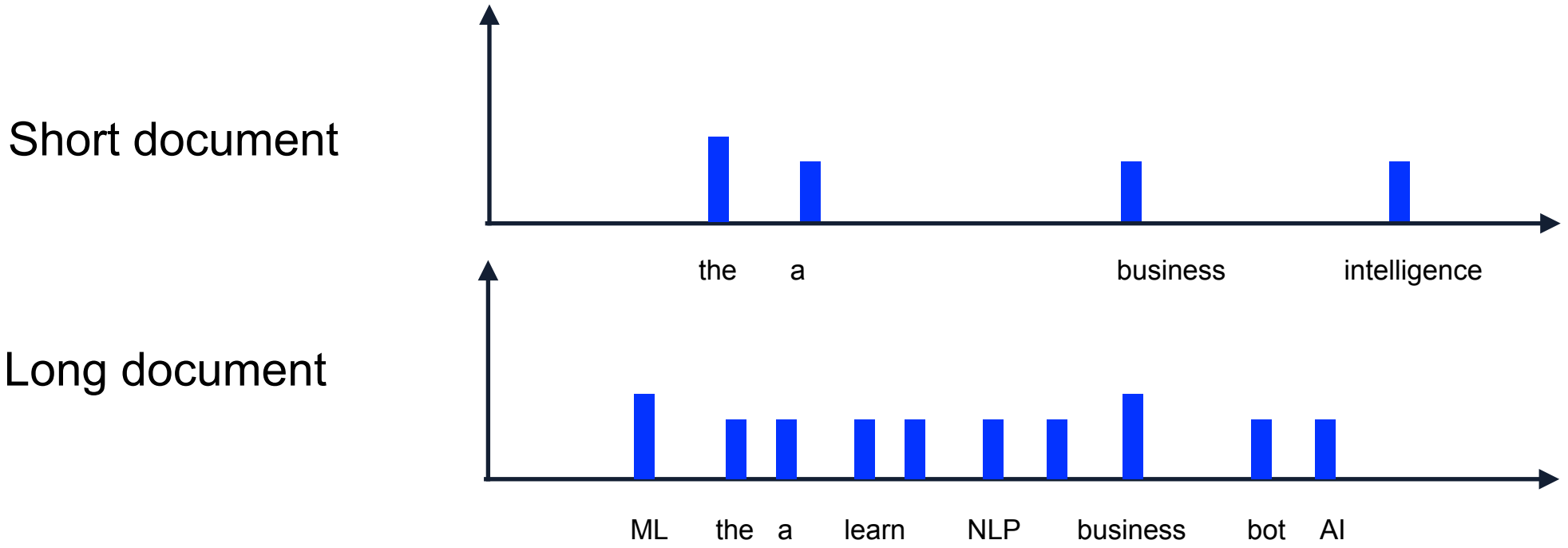
d1 = "**artificial intelligence** book"

d2 = "**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

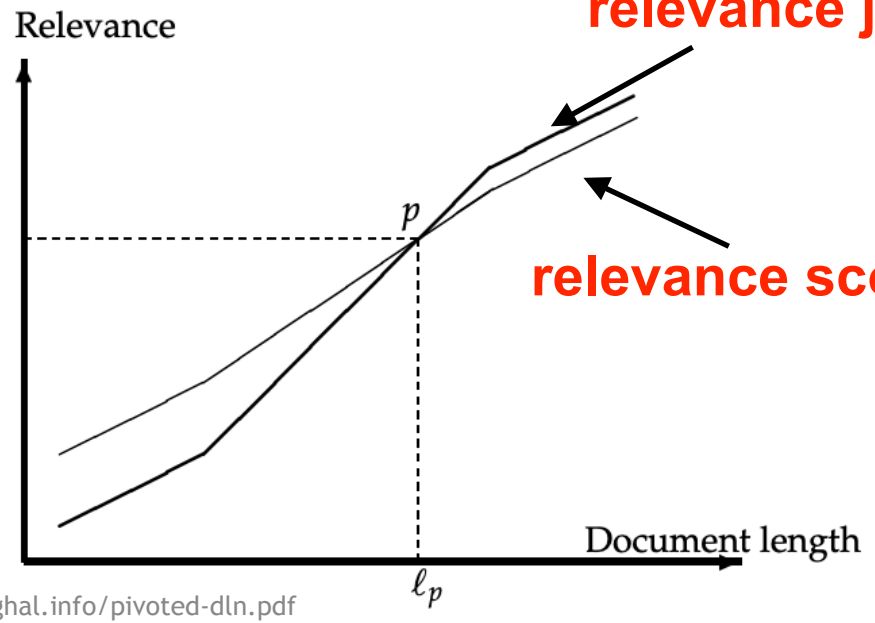


Document length pivoting

- For each query q and each document d, compute their relevance score score(q, d)

- Manually evaluate the relevance between q and d

$$\textbf{relevance judg@l} = \frac{count(length = l, rel = 1)}{count(length = l)}$$

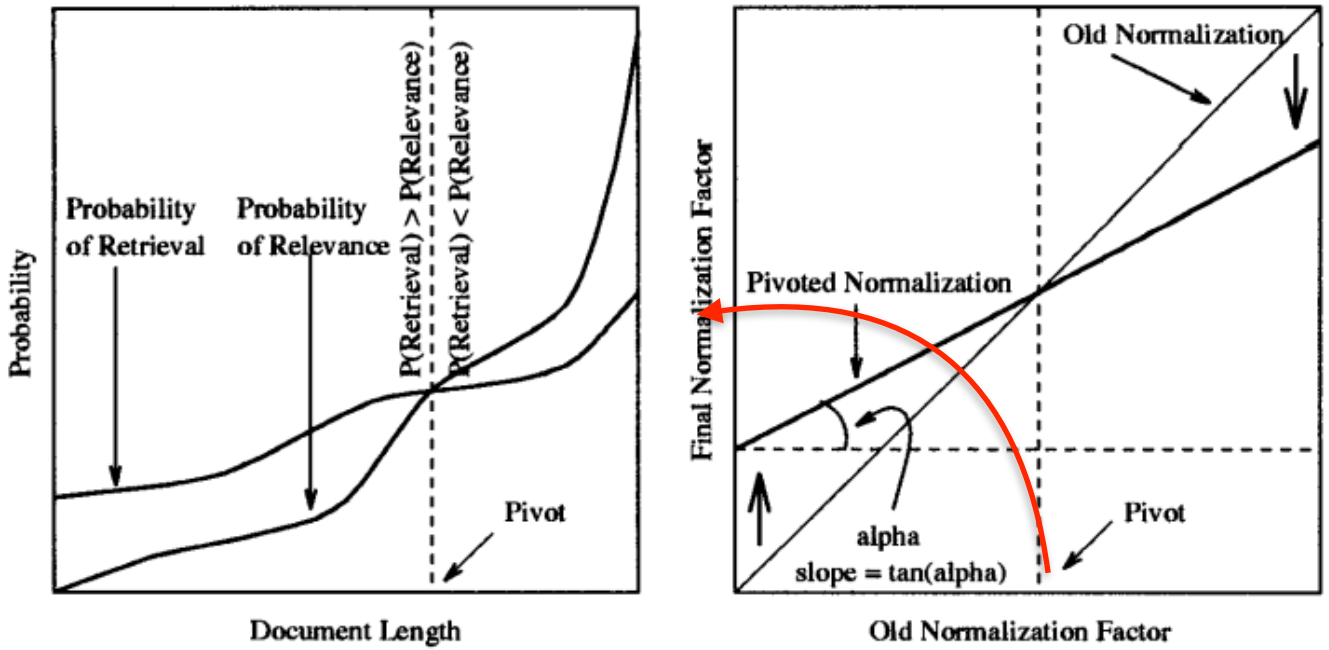


relevance score

Pivoted Document Length Normalization: http://singhal.info/pivoted-dln.pdf

Document length pivoting

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



$$y = x$$

$$pivot = pivot \times slope + intercept$$

$$pivoted\_normalization = (1.0 - slope) \times pivot + slope \times oldnormalization$$

Pivoted Document Length Normalization: http://singhal.info/pivoted-dln.pdf

Document length pivoting

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

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

$$\frac{tf \cdot idf\ \ weight}{1 + \frac{slope}{(1.0 - slope) \times pivot} \times old\ \ normalization}$$

← fixing pivot to avgdl

$$(1.0 - slope) + slope \times \frac{old\ \ normalization}{average\ \ old\ \ normalization}$$

← $1 - b + b\frac{|dl|}{|avgdl|}$

standard formulation for doc length normalization

Pivoted Document Length Normalization: http://singhal.info/pivoted-dln.pdf