

CS 589 Fall 2020

Text Mining and Information Retrieval

Instructor: Susan Liu

TA: Huihui Liu

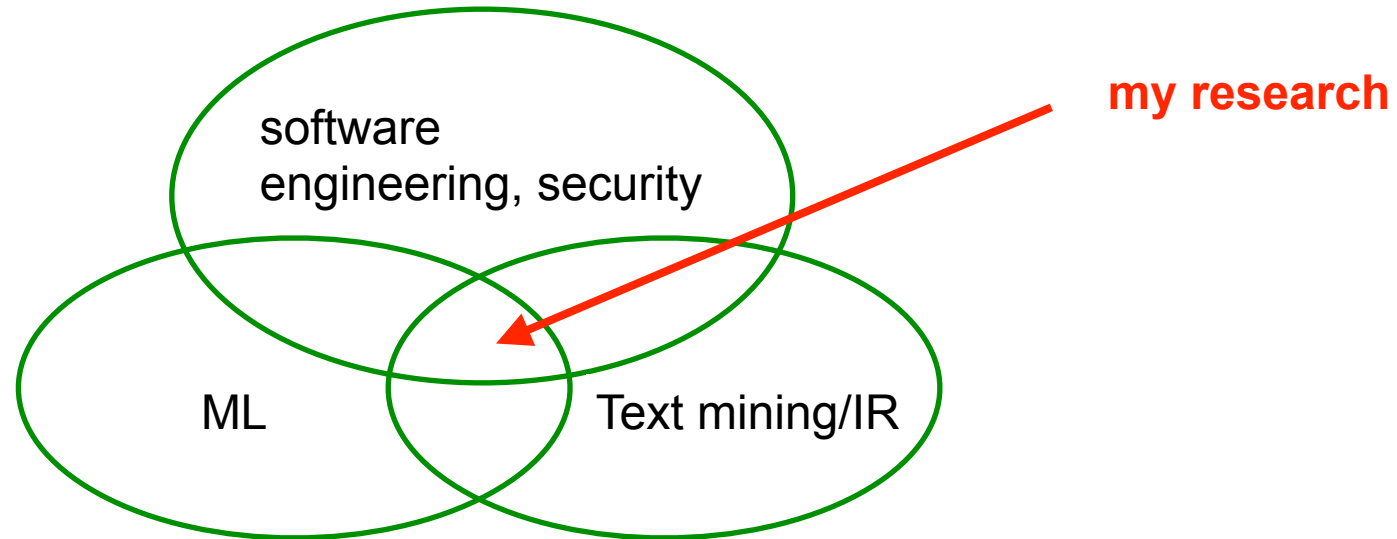
Stevens Institute of Technology

Welcome to CS589

- **Instructor:** Susan (Xueqing) Liu
- **Email:** xliu127@stevens.edu
- **CAs:**
 - Huihui Liu hliu79@stevens.edu

Who am I?

- Assistant professor joined Jan 2020
- PhD@UIUC 2019
- My research:
 - Helping users (especially software developers) to more quickly search for information



What is CS589 about?

- Text Mining
 - The study of extracting high quality information from raw texts
- Information retrieval
 - The study of retrieving relevant information/resources/knowledge to an **information need**

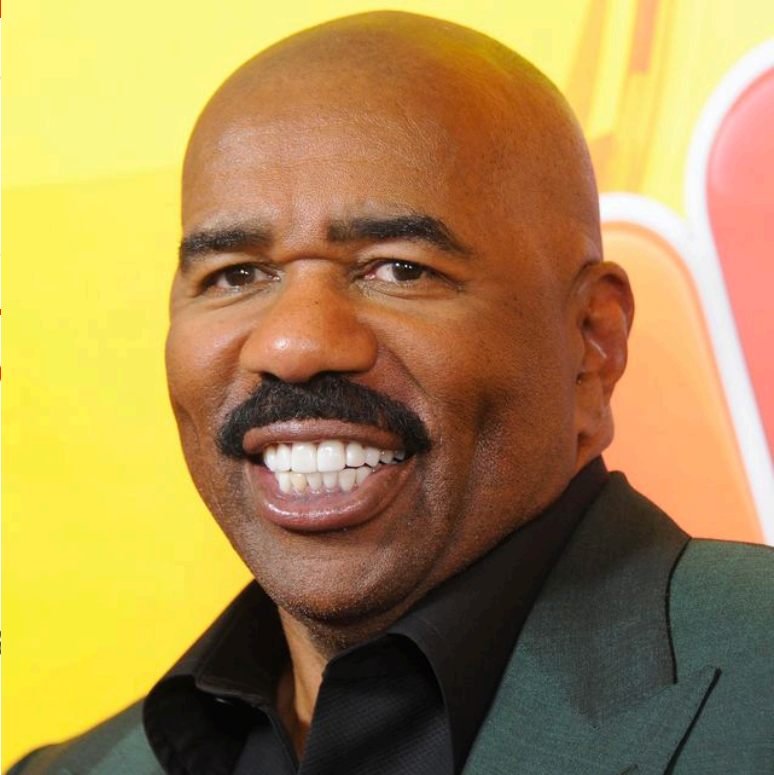
Information Retrieval Techniques

“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

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



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

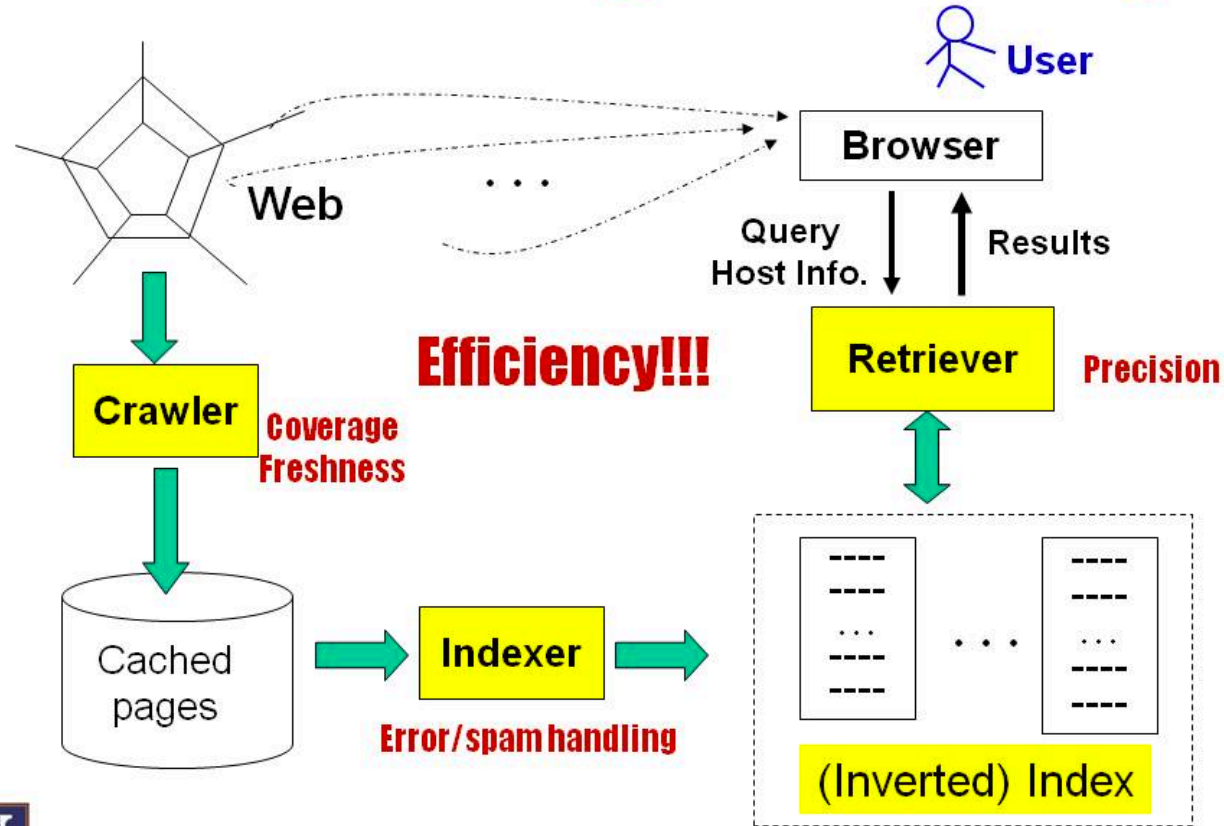
cs589

CS 589: Text Mining and Information Retrieval. Home | Canvas | Resources
Stevens' guidelines on the coronavirus emergency (COVID-19), ...

Information Retrieval Techniques

Basic Search Engine Technologies

Getting enough coverage of users' information need



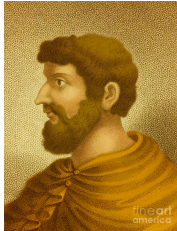
Query understanding, personalization, results diversification, result page optimization, etc.



Making sure the results are returned to users fast

A Brief History of IR

300 BC



Callimachus: the first library catalog

1950s

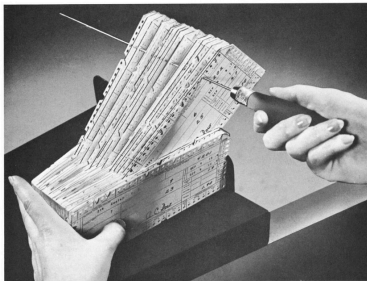


FIG. 51. Card machine with an alphabetical card system.

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;

Information need



information need

“An individual or group's desire to locate and obtain information to satisfy a need”, e.g., question answering, program repair, route planning

query

A (short) natural language representation of users' information need

The Boolean retrieval system

Illustration of the function of the linear homeoscope without movable parts

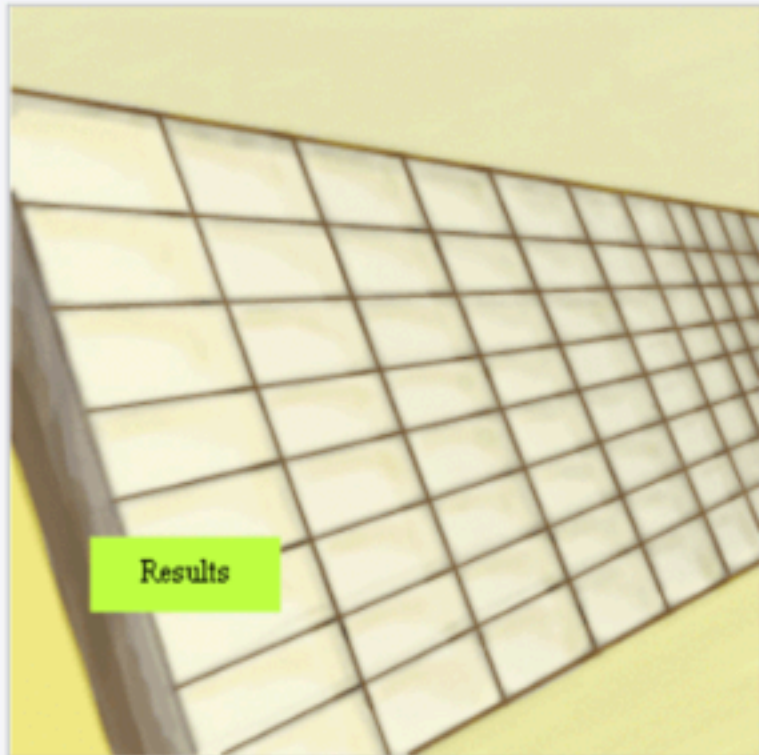
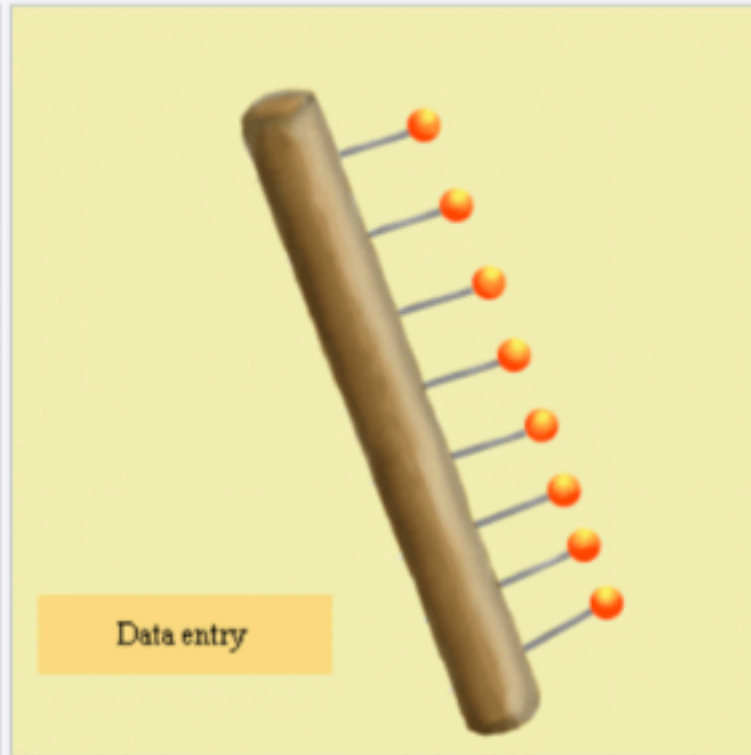
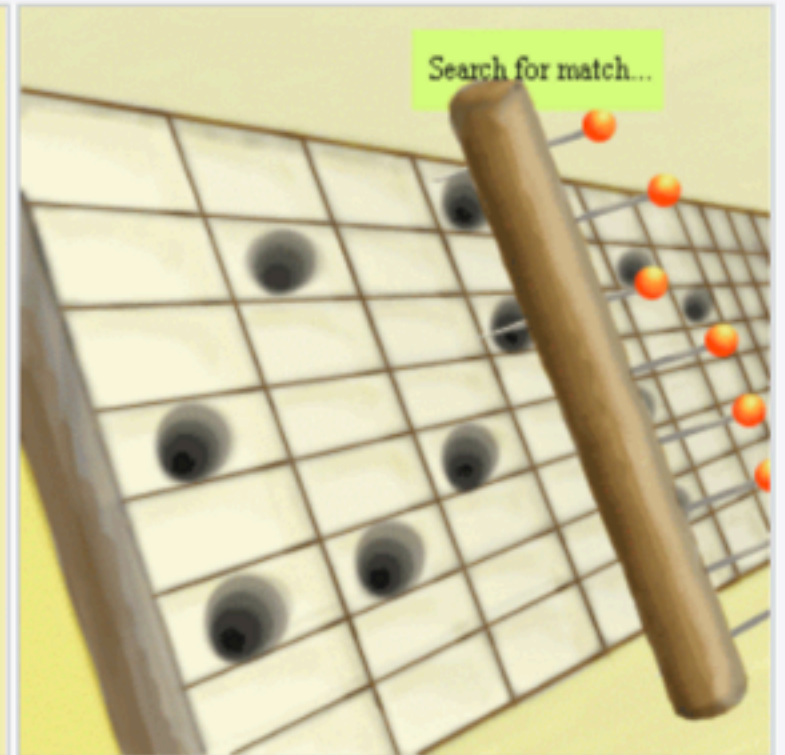


Table of data



Search criteria



Search for matching data

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., faceted search
 - Library catalog, eCommerce search, etc.
- **Advantage:** Returns exactly what you want
- **Disadvantage:**
 - can only specify queries based on the pre-defined categories
 - too few / too many queries

The Boolean retrieval system

The screenshot shows a search results page for "Digital cameras". The page is annotated with several callout boxes:

- Manufacturer is a facet, a way of categorizing the results**: Points to the "Manufacturer" facet.
- Canon, Sony, and Nikon are constraints, or facet values**: Points to the "Canon USA (5)", "Sony (2)", and "Nikon (2)" items under the Manufacturer facet.
- The facet count or constraint count shows how many results match each value**: Points to the counts in parentheses next to the facet values.
- The breadcrumb trail shows what constraints have already been applied and allows for their removal**: Points to the "you selected:" section.
- Regular search results list**: Points to the main list of search results.

The "Refine your results" section includes the following facets:

- Manufacturer**: Canon USA (5), Sony (2), Nikon (2), Olympus (6), Pentax (2)
- Resolution**: 6 megapixels (3), 8 megapixels and up (14)
- Zoom range**: 3X to 4X (11), 8X to 12X (1)
- More**: LCD size, Image stabilizer, Flash memory, Still image format, Maximum ISO, See all >

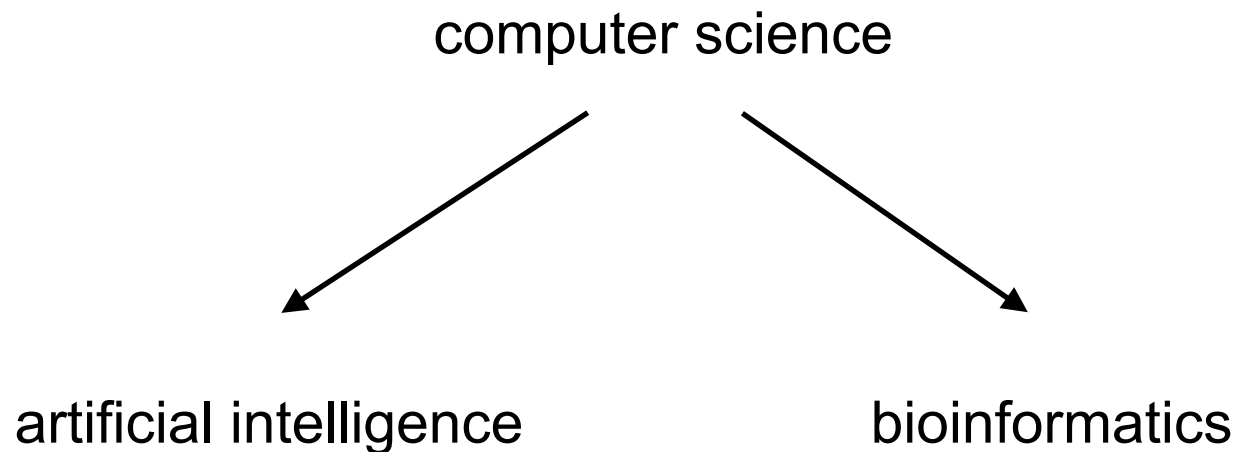
The "you selected:" section shows: \$400 - \$500, SLR, and remove all.

The "Regular search results list" shows 17 results. The first result is: Canon EOS Rebel XS (silver, with 18-55mm) \$459 to \$699.

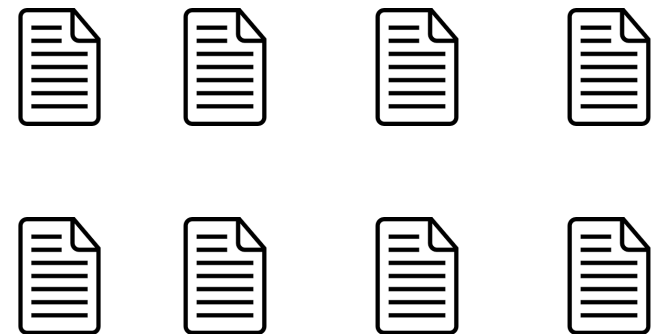
The user may specify a condition that does not exist

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

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

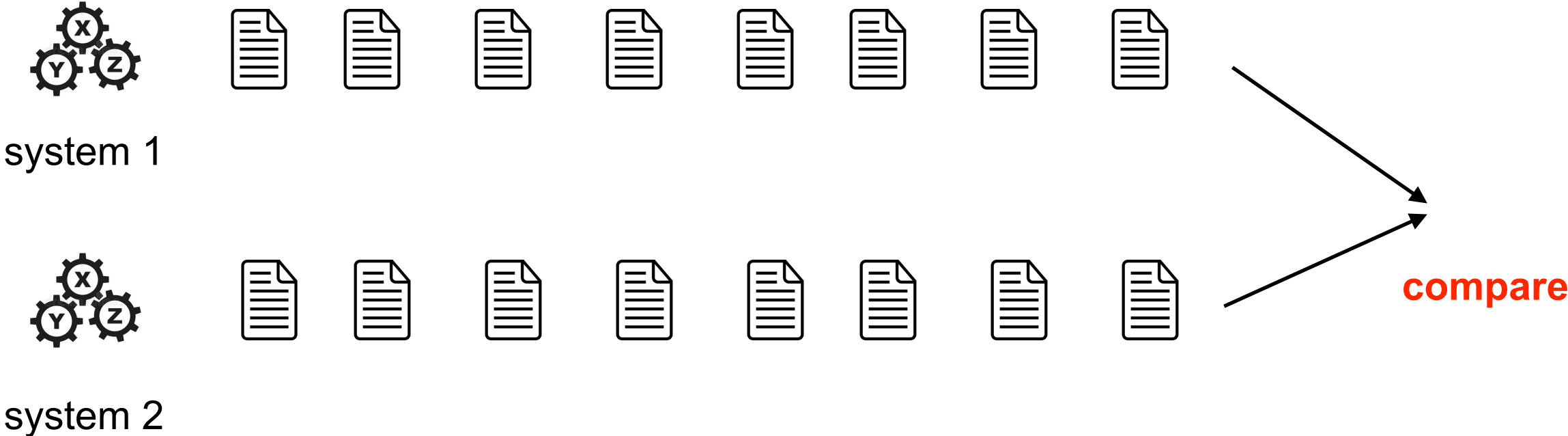
	intelligent	applications	creates	business	processes	bots	are	i	do	artificial
Doc 1	1	1	1	1	1	0	0	0	0	0
Doc 2	1	1	0	0	0	1	1	0	0	0
Doc 3	0	0	0	1	0	0	0	1	1	1

query = "artificial intelligence"

bags of words representation

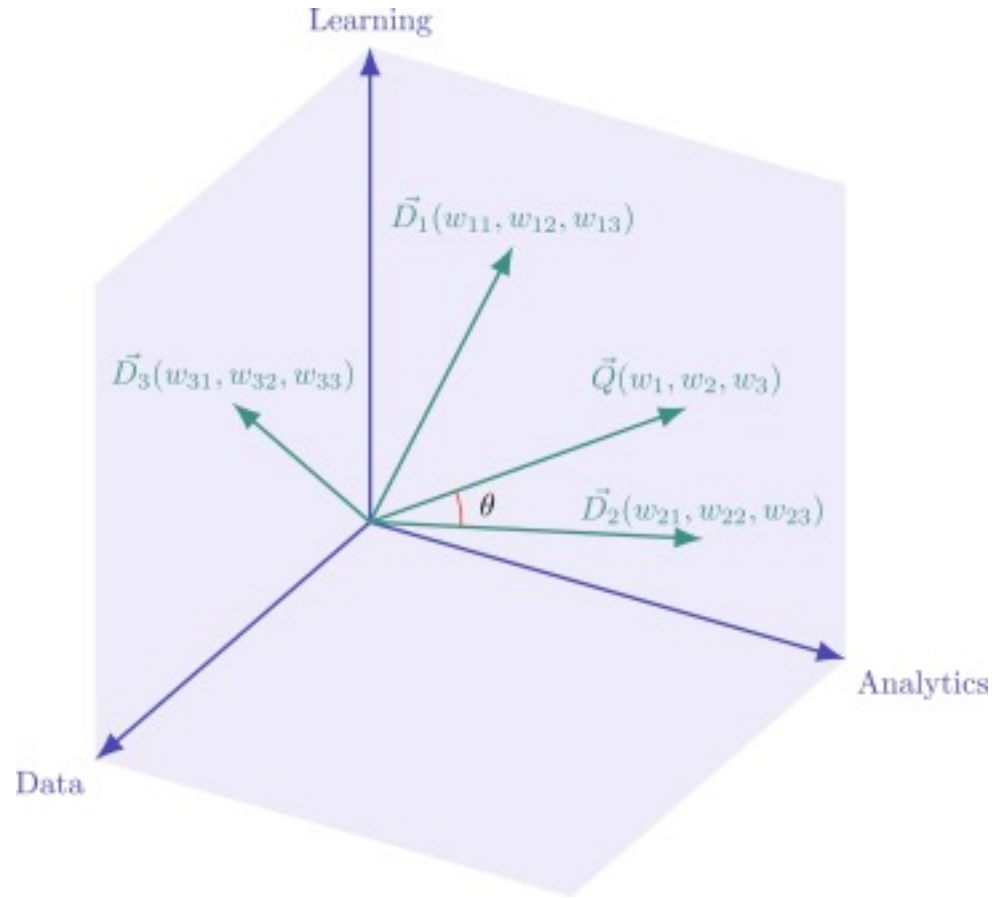
system 2: indexing documents by lists of words

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

	intelligent	applications	creates	business	processes	bots	are	i	do	artificial
Doc 1	2	1	1	1	1	0	0	0	0	0
Doc 2	1	1	0	0	0	1	1	0	0	0
Doc 3	0	0	0	1	0	0	0	1	1	1

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

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

- $d1 = [2, 1, 1, 1, 1, 0, 0, 0, 0, 0]$
- $d2 = [1, 1, 0, 0, 0, 1, 1, 0, 0, 0]$
- $d3 = [0, 0, 0, 1, 0, 0, 0, 1, 1, 1]$
- query = “business intelligence”
- $q = [0, 0, 0, 1, 0, 0, 0, 0, 0, 1]$

Vector space model

Document Term Matrix

	intelligent	applications	creates	business	processes	bots	are	i	do	artificial
Doc 1	2	1	1	1	1	0	0	0	0	0
Doc 2	1	1	0	0	0	1	1	0	0	0
Doc 3	0	0	0	1	0	0	0	1	1	1

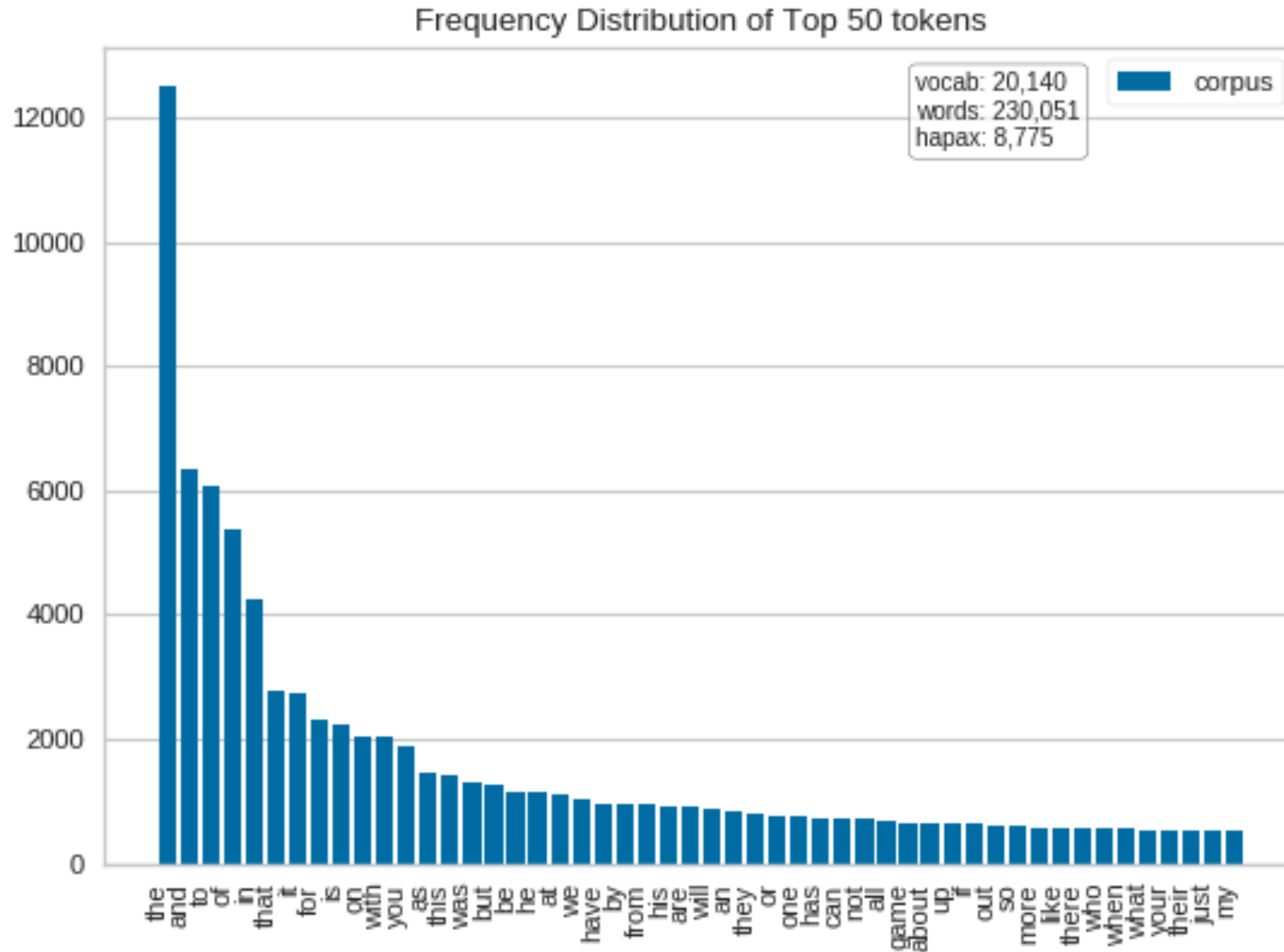
- To answer the query:
 - “business intelligence”
- $q = [0, 0, 0, 1, 0, 0, 0, 0, 0, 1]$
- $d1 = [2, 1, 1, 1, 1, 0, 0, 0, 0, 0]$
- $d2 = [1, 1, 0, 0, 0, 1, 1, 0, 0, 0]$
- $d3 = [0, 0, 0, 1, 0, 0, 0, 1, 1, 1]$

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

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
 - $q = \text{“the } \mathbf{artificial\ intelligence} \text{ book”}$ $score(q, d_1) = 0.8164$
 - $d_1 = \text{“the cat, the doc, and the book”}$ $score(q, d_2) = 0.3535$
 - $d_2 = \text{“business } \mathbf{intelligence} \text{”}$ $\Rightarrow score(q, d_1) > score(q, d_2)$

Zipf's law distribution of words



the
and
to
of
in
that
it
for
is
on
with
you
as
this
was
but
be
he
at
we
have
by
from
his
are
will

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

- If a word appears everywhere, it should be penalized
- 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, 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, 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**

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

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

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

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

Document length pivoting

- Another problem with TF-IDF weighting
 - 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**”

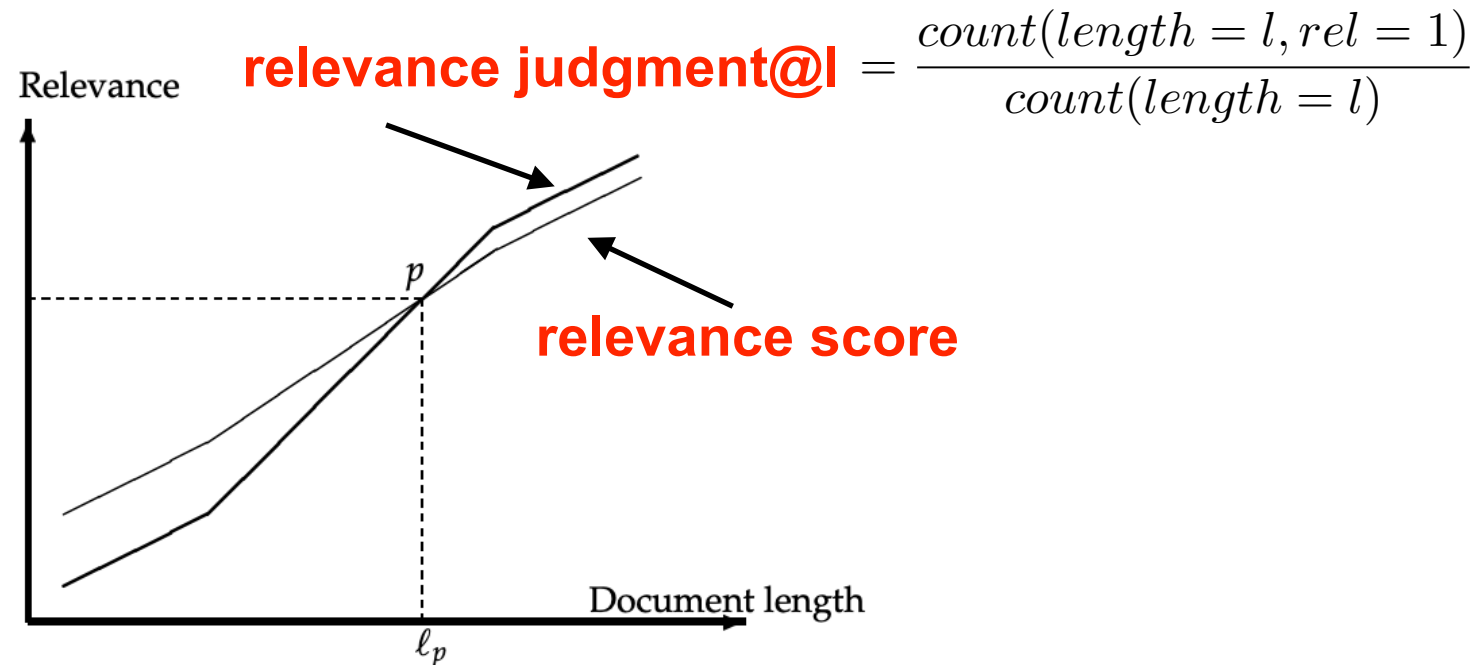
d1 = “**artificial intelligence** book”

d2 = “**Artificial intelligence** was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism “

$$\text{score}(q, d_1) > \text{score}(q, d_2)$$

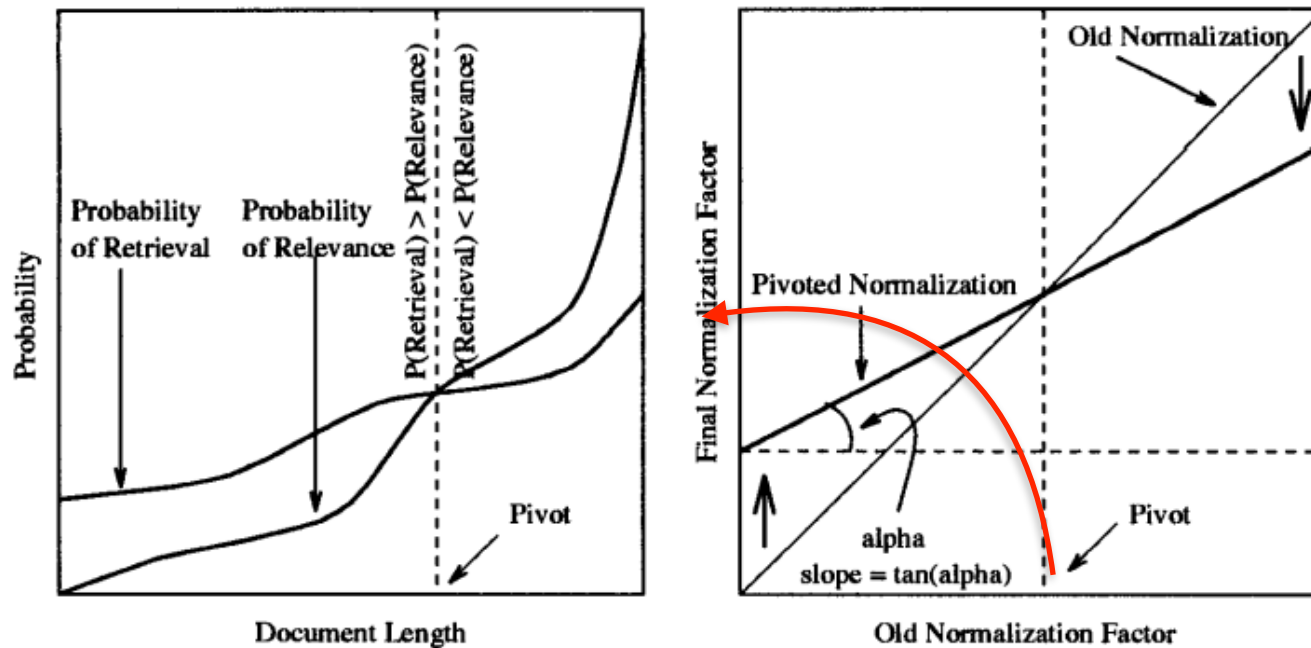
Document length pivoting

- 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



$$y = x$$

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

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

Document length pivoting

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

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

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

the similar formulation will be frequently used later

More on retrieval model design heuristics

- Axiomatic thinking in information retrieval [Fang et al., SIGIR 2004]

Table 1: Summary of intuitions for each formalized constraint

Constraints	Intuitions
TFC1	to favor a document with more occurrence of a query term
TFC2	to favor document matching more distinct query terms
TFC2	to make sure that the change in the score caused by increasing TF from 1 to 2 is larger than that caused by increasing TF from 100 to 101.
TDC	to regulate the impact of TF and IDF
LNC1	to penalize a long document (assuming equal TF)
LNC2, TF-LNC	to avoid over-penalizing a long document
TF-LNC	to regulate the interaction of TF and document length

IR != web search

- The other side of information retrieval techniques
 - Recommender systems (users who bought this also bought...)
 - Online advertising

Frequently repurchased items from popular brands




AmazonBasics 48-Count AA High-Performance Alkaline Batteries, 10-Year Shelf Life, Easy to Open...
★★★★☆ 76,645
\$15.49 ✓prime


Blue Buffalo Life Protection Formula Natural Adult Dry Dog Food
★★★★☆ 8,760
\$35.53 ✓prime


HP 61 | 2 Ink Cartridges | Black, Tri-color | CH561WN, CH562WN
★★★★☆ 12,919
6 offers from \$58.00

AdHawk | Digital Advertising Simplified | tryadhawk.com

[Ad www.tryadhawk.com/](http://www.tryadhawk.com/) ▼

Stress-Free Account Management & Reporting Tools From Google and Facebook Ads Experts!
Data Driven Software. Founded by ex-Googlers. Free Consultations. Services: In Depth Reporting,
Data Driven Optimization, Managed Services, 24/7 Reporting.

[AdHawk Managed Services](#) · [Free Google Ads Audit](#) · [Our Blog](#) · [Franchises & Co-Ops](#)

Put your ads on Pinterest | A different kind of platform

[Ad www.pinterest.com/](http://www.pinterest.com/) ▼

Pinterest ads can help you reach people planning their next purchase. Get tips, tools and insights to help you succeed. Try it today. Increase traffic. Boost sales. Get discovered. Increase video views. Engage your audience. Now available. App downloads.

Web Advertising? | Manage Your Free Yelp Listing

[Ad biz.yelp.com/Web-Advertising](http://biz.yelp.com/Web-Advertising) ▼

Reach More Customers. List Your Business on Yelp - It's Quick & Easy. Start Now! [Continue...](#)

IR != web search

- Reasoning-based question answering systems

q

When was the football club founded in which **Walter Otto Davis** played at centre forward?

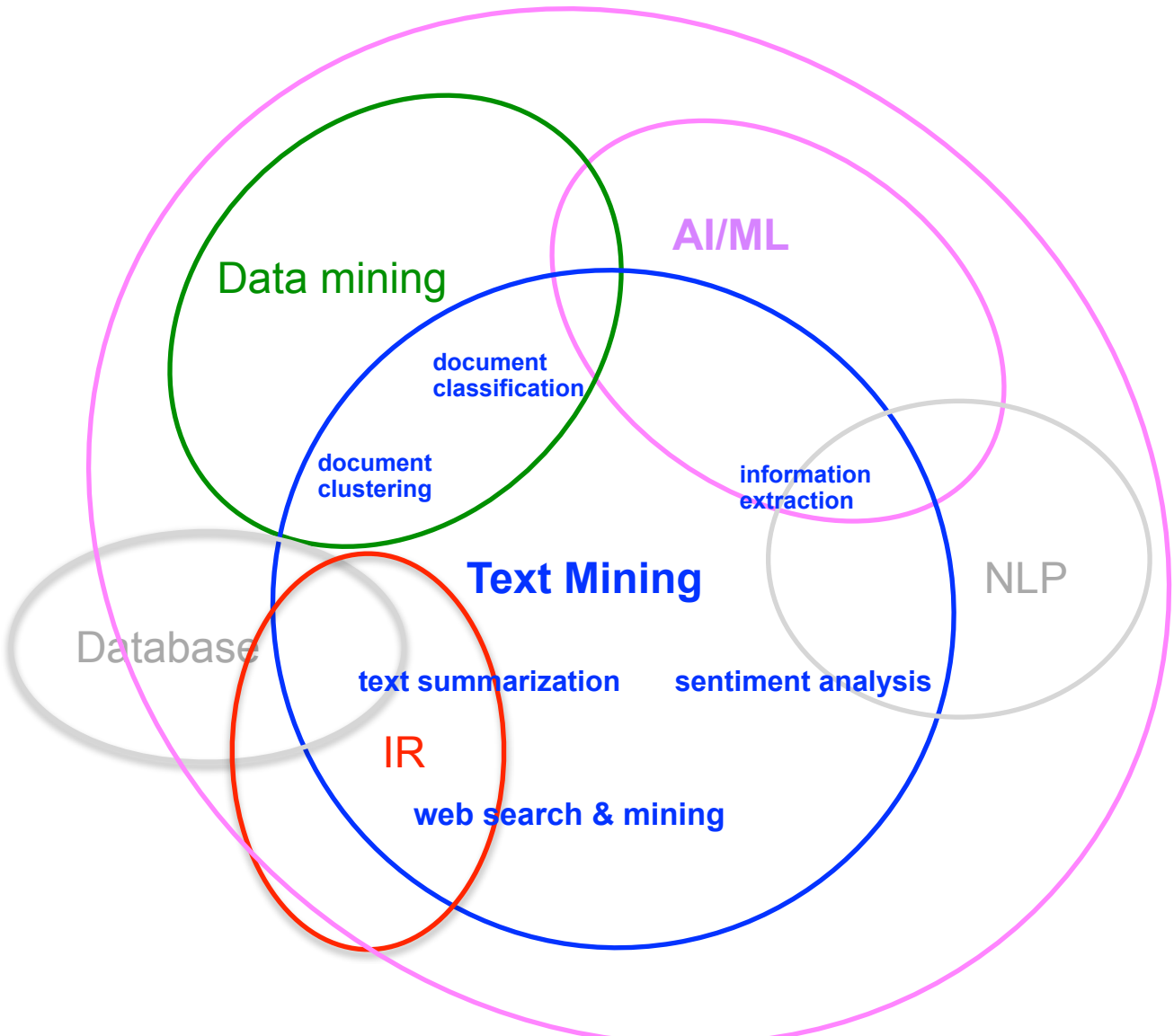
Paragraph 1: [Walter Davis (footballer)]

Walter Otto Davis was a Welsh professional footballer who played at centre forward for **Millwall** for ten years in the 1910s.

Paragraph 2: [Millwall F.C.]

Millwall Football Club is a professional football club in South East London, ... Founded as Millwall Rovers in **1885**.

What about text mining?



Syllabus

- Vector space model, TF-IDF
- Probability ranking principle, BM25
- IR evaluation, query completion
- Inverted index, ES, PageRank, HITS
- Relevance feedback, PRF
- Neural IR
- EM algorithm
- RNN/LSTM
- Transformer/Bert
- Frontier topic: recommender system
- Frontier topic: opinion analysis/mining
- Frontier topic: NMT, program synthesis

Assignment goals

Upon successful completion of this course, students should be able to:

- Evaluate ranking algorithms by using information retrieval evaluation techniques, and implement text retrieval models such as TF-IDF and BM25;
- Use Elastic search to implement a prototypical search engine on Twitter data;
- Derive inference algorithms for the maximum likelihood estimation (MLE), implement the expectation maximization (EM) algorithm;
- Use state-of-the-art tools such as LSTM/Bert for text classification tasks

Prerequisite

- CS116 is required for undergrad, CS225 is recommended (data structure in Java)
- **Fluency in Python is required**
- **A good knowledge on statistics and probability**
- Knowledge of one or more of the following areas is a plus, but not required: Information Retrieval, Machine Learning, Data Mining, Natural Language Processing
- Contact the instructor if you aren't sure

Format

- Meeting: every Monday 8:15-9:45
- 4 programming assignments
 - Submit code + report
- 1 midterm
 - in class
- Final project

Final Project

**Oct 19
- Oct 26**

Students choose a topic; for each topic, they pick 2-3 coherent papers, and write a summary for the paper

**Oct 26 -
Nov 16**

Students who share the same interest are categorized into groups; each group propose a novel research topic motivated by their survey





Dec 14

Deliver a presentation in Week 14

Dec 20

Submit their implementation (code in Python) as well as an 8-page academic paper as their final project.

Grading

- Homework - 40%, Midterm - 30%, Project - 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
 - **Medical conditions** 
 - **A sudden increase in family duty** 
 - **Too much workload from other courses** 
 - **The assignment is too difficult** 

Plagiarism policy

- **We have a very powerful plagiarism detection pipeline, 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,




Question answering

- Please do not ask your questions in Canvas, most questions can be asked on Piazza, otherwise use emails


 **Class at a Glance** Updated 31 seconds ago. [Reload](#) [Go to Live Q&A](#)

 7 unread posts	116 total posts
 5 unanswered questions	220 total contributions
 1 unresolved followups	67 instructors' responses
	4 students' responses
	8 min avg. response time

Student Enrollment

52 enrolled 



Educator Tips during COVID-19:



Sharon Reckinger

Sharing Faculty Experiences

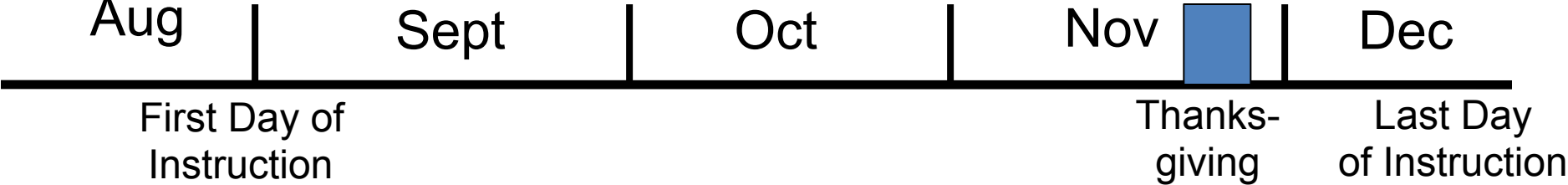
..out of 60 (estimated) [Edit](#)

Download us in the app store:  

Question asking protocol

- Regrading requests: **email TA**, cc myself, titled [CS589 regrading]
- Deadline extension requests: **email** myself, titled [CS589 deadline]
- Dropping: email myself, titled [CS589 drop]
- All technical questions: **Piazza**
 - Homework description clarification
 - Clarification on course materials
- Having trouble with homework: join my **office hour** directly, no need to **email** me
 - If you have a time conflict, **email me** & schedule another time
- Project discussion: join my **office hour**
- **Ask any common questions shared by the class on Piazza**

Your workload



Books

- No text books
- Recommended readings:
 - Zhai, C., & Massung, S. (2016). Text data management and analysis: a practical introduction to information retrieval and text mining. Association for Computing Machinery and Morgan & Claypool
 - Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008