CSCE 670 - Information Storage and Retrieval

Week 1: Course Overview; Boolean Retrieval

Yu Zhang
yuzhang@tamu.edu

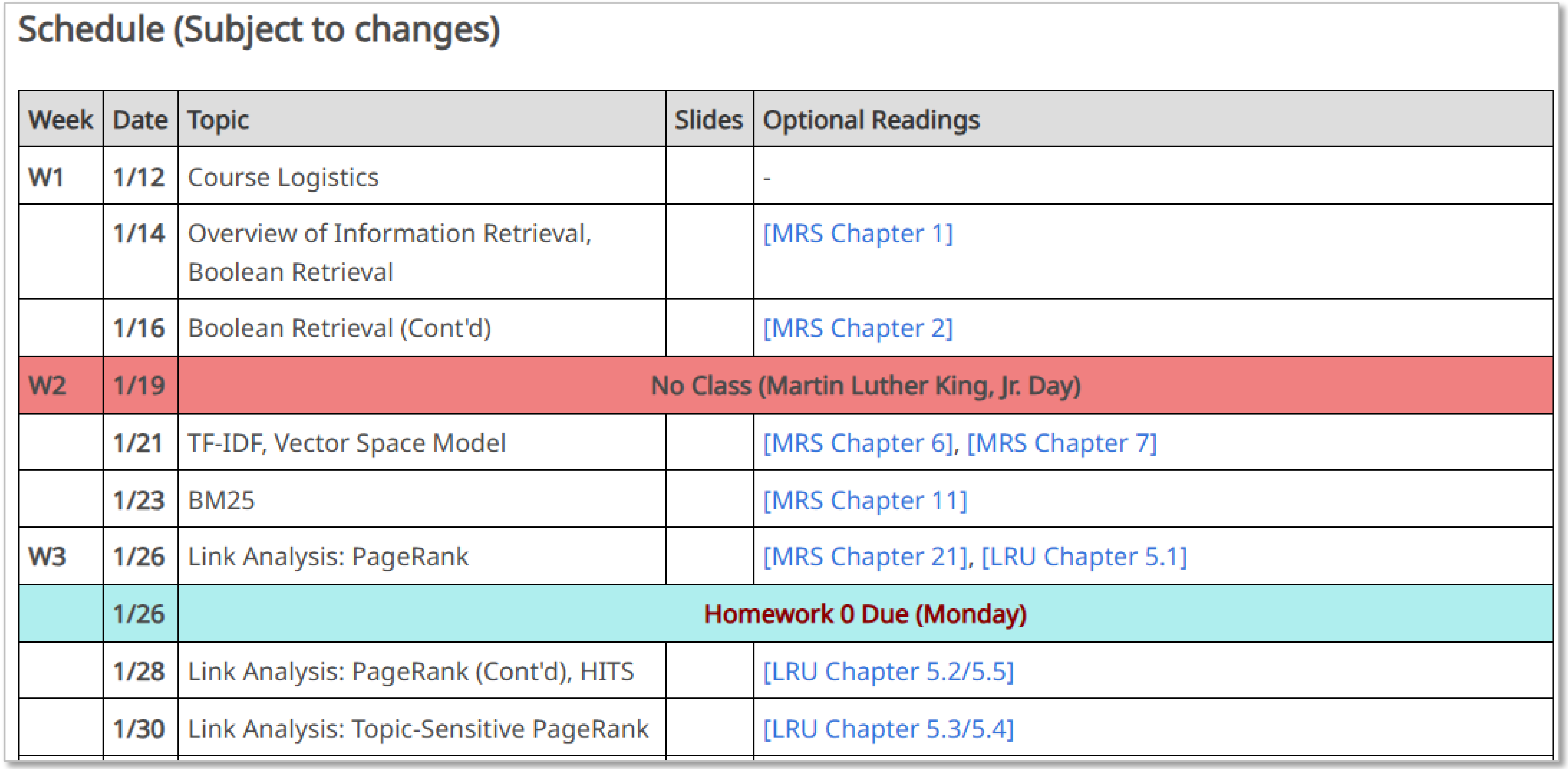
Course Website: https://yuzhang-teaching.github.io/CSCE670-S26.html

Adapted from the slides by Prof. James Caverlee

Course Logistics

Course Website

https://yuzhang-teaching.github.io/CSCE670-S26.html

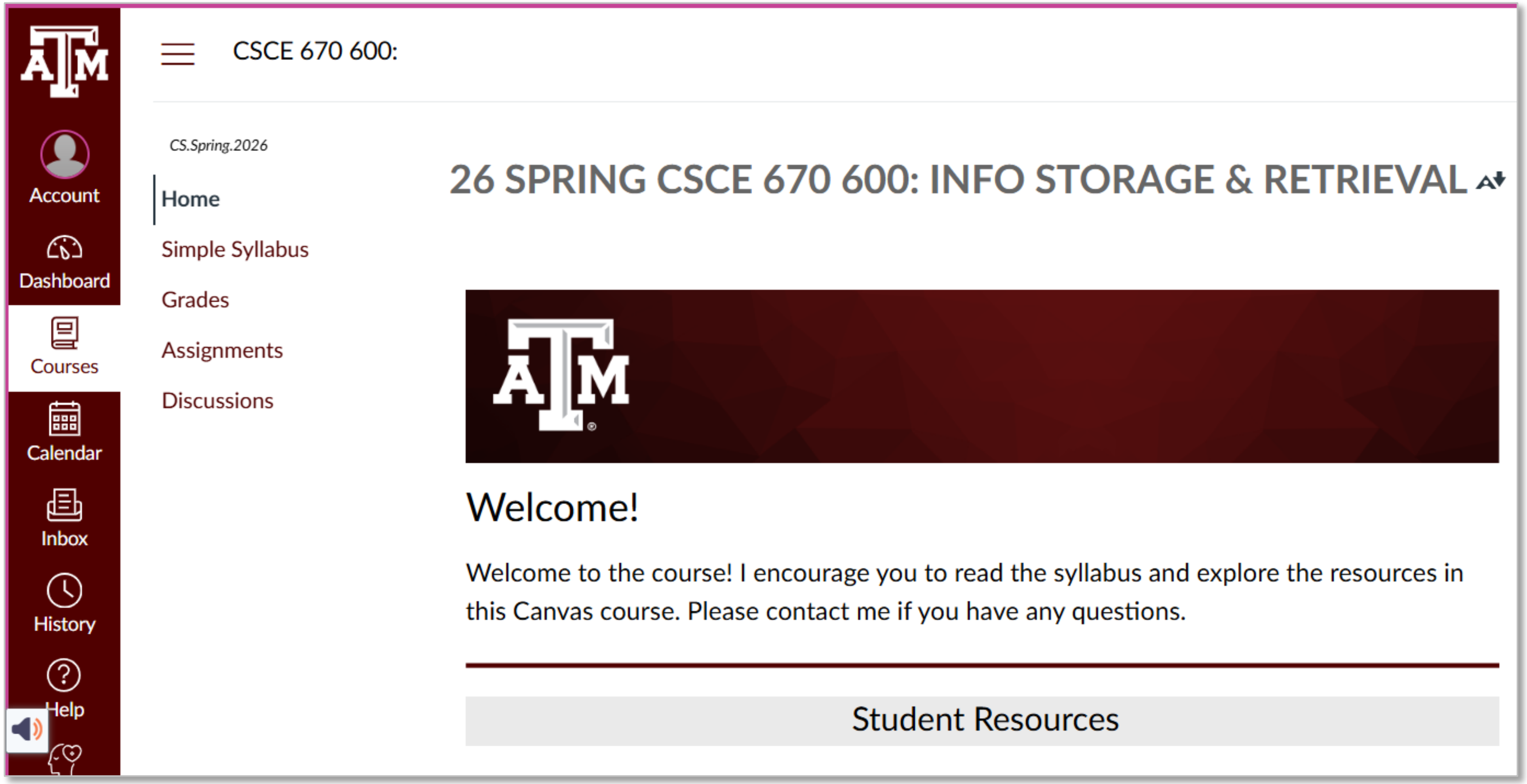


Schedule (Subject to changes)

Week	Date	Topic	Slides	Optional Readings
W1	1/12	Course Logistics		-
	1/14	Overview of Information Retrieval, Boolean Retrieval		[MRS Chapter 1]
	1/16	Boolean Retrieval (Cont'd)		[MRS Chapter 2]
W2	1/19	No Class (Martin Luther King, Jr. Day)		
	1/21	TF-IDF, Vector Space Model		[MRS Chapter 6], [MRS Chapter 7]
	1/23	BM25		[MRS Chapter 11]
W3	1/26	Link Analysis: PageRank		[MRS Chapter 21], [LRU Chapter 5.1]
	1/26	Homework 0 Due (Monday)		
	1/28	Link Analysis: PageRank (Cont'd), HITS		[LRU Chapter 5.2/5.5]
	1/30	Link Analysis: Topic-Sensitive PageRank		[LRU Chapter 5.3/5.4]

Canvas

https://canvas.tamu.edu/courses/426688



CSCE 670 600:

CS.Spring.2026

26 SPRING CSCE 670 600: INFO STORAGE & RETRIEVAL

Welcome!

Welcome to the course! I encourage you to read the syllabus and explore the resources in this Canvas course. Please contact me if you have any questions.

Student Resources

Course Logistics

- Course Website: Syllabus, Schedule, Slides, Optional Readings
- Canvas: Grades, Announcements, Assignments, Discussions

- Two ways to reach me/TA
  - Email us directly (please put [CSCE670] in the subject)
  - Message us through Canvas

Grading (See Syllabus and Course Website for Details)

- Homework: 30%
  - Homework 0: 2% [due Jan 26 (Mon)]
  - Homework 1: 7% [due Feb 9 (Mon)]
  - Homework 2: 7% [due Mar 2 (Mon)]
  - Homework 3: 7% [due Mar 30 (Mon)]
  - Homework 4: 7% [due Apr 20 (Mon)]
- Quizzes: 20%
  - Quiz 1: 5% [in the Feb 6 class (Fri)]
  - Quiz 2: 5% [in the Feb 27 class (Fri)]
  - Quiz 3: 5% [in the Mar 25 class (Wed)]
  - Quiz 4: 5% [in the Apr 15 class (Wed)]
- Group Project: 20%
  - Proposal: 2% [due Mar 7 (Sat)]
  - Presentation: 9% [in the Apr 20, Apr 22, Apr 24, and Apr 27 classes]
  - Report: 9% [due May 2 (Sat)]
- Final: 30% [3:30pm – 5:30pm, May 4, HRBB 113]

Homework (30%)

- We will have 1 "getting-started" homework and then 4 real homework assignments
- Fun opportunity to put concepts into action
- All in Python!

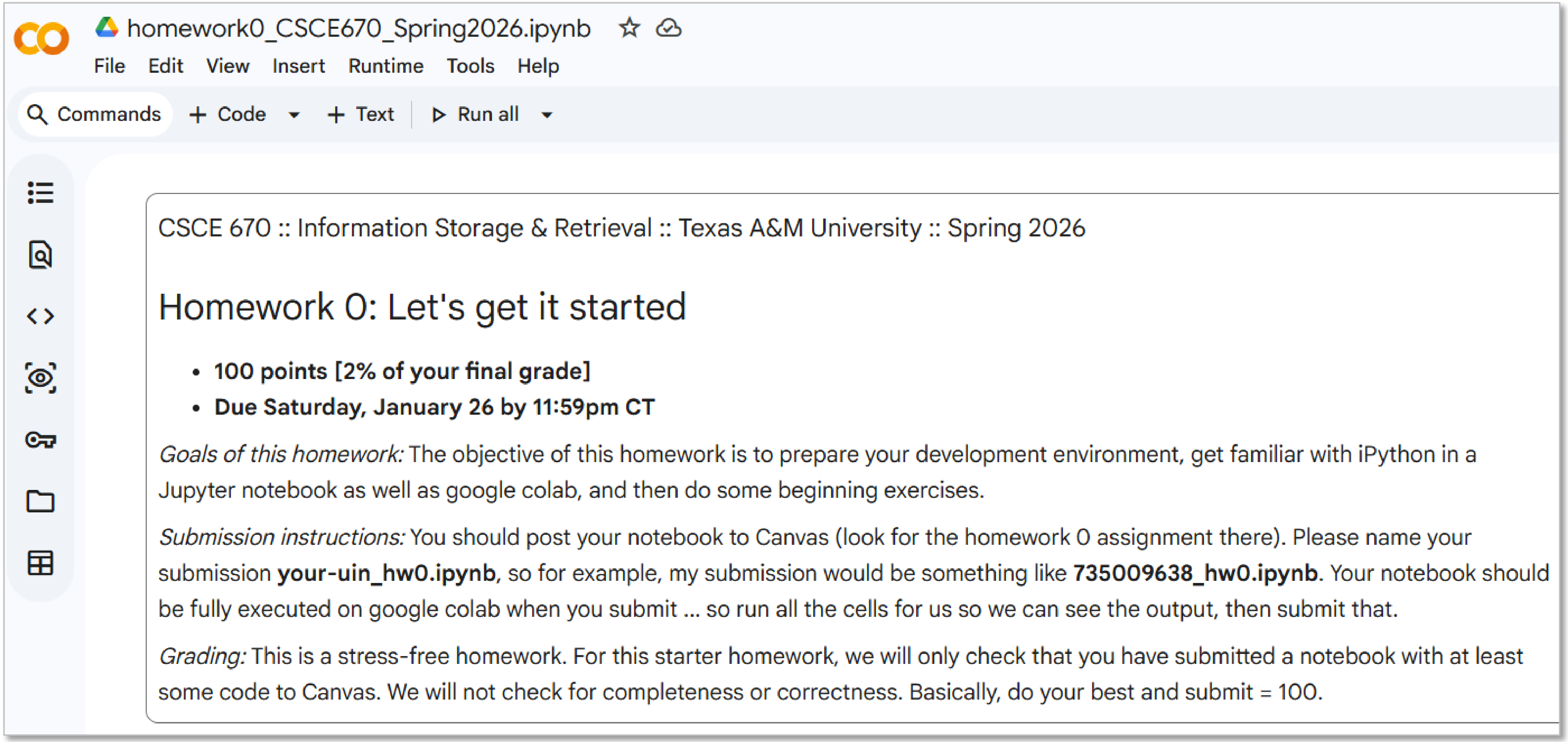
- Individual work, but you may discuss generally with others
  - You should write your own code, by yourself
  - BUT you may talk amongst yourselves about approaches/methods
    - E.g., sit in a group with no laptops, just talking = totally fine
    - E.g., sit next to each other while you code = BAD NEWS
  - You must acknowledge ALL help in your homework
    - Using code comments
    - I will show you an example in 5 minutes

AI Policy

- *"In principle you may submit AI-generated code, or code that is based on or derived from AI-generated code, as long as this use is properly documented in the comments: you need to include the prompt and the significant parts of the response. AI tools may help you avoid syntax errors, but there is no guarantee that the generated code is correct. It is your responsibility to identify errors in program logic through comprehensive, documented testing. Moreover, generated code, even if syntactically correct, may have significant scope for improvement, in particular regarding separation of concerns and avoiding repetitions. The submission itself must meet our standards of attribution and validation."*

- (from Boris Steipe (2023) "Syllabus Resources". The Sentient Syllabus Project: http://sentientsyllabus.org)

Homework 0 (due Jan 26)



CSCE 670 :: Information Storage & Retrieval :: Texas A&M University :: Spring 2026

Homework 0: Let's get it started

- **100 points [2% of your final grade]**
- **Due Saturday, January 26 by 11:59pm CT**

Goals of this homework: The objective of this homework is to prepare your development environment, get familiar with iPython in a Jupyter notebook as well as google colab, and then do some beginning exercises.

Submission instructions: You should post your notebook to Canvas (look for the homework 0 assignment there). Please name your submission **your-uin_hw0.ipynb**, so for example, my submission would be something like **735009638_hw0.ipynb**. Your notebook should be fully executed on google colab when you submit ... so run all the cells for us so we can see the output, then submit that.

Grading: This is a stress-free homework. For this starter homework, we will only check that you have submitted a notebook with at least some code to Canvas. We will not check for completeness or correctness. Basically, do your best and submit = 100.

Homework Late Days

- Due by 11:59pm on the due date
- You get 5 late days total

- Late day = indivisible 24-hour unit
    - E.g., if due date is 11:59pm on Saturday, and you submit at 12:01am on Sunday = one late day
    - No penalty for using a late day; no need to alert me/TA that you are using a late day

- Once you are out of late days, you get 0

Regrade Policy

- Once you receive your graded assignment (e.g., homework and quizzes), you have 7 days to request a regrade in writing (give to me)
- After 7 days = no regrades
- You must give us a written explanation of what the issue is
- We will re-grade the entire assignment

Questions?

4 Quizzes (5% × 4 = 20%)

- In-class
- 45 minutes, but designed to only take 30-35 minutes
- 7 multiple-choice questions
- Answering 5 questions correctly will earn you full credit (5%)

# correct answers	0	1	2	3	4	5	6	7
credit	0%	1%	2%	3%	4%	5%	5%	5%

- Closed book
  - Laptops, books, and notes are NOT allowed.
- Calculators are NOT required, and the questions will NOT involve calculations (such as square roots or logarithms) that cannot be done easily by hand.

Absence Policy

- Please refer to Student Rule 7 (https://student-rules.tamu.edu/rule07/) about excused absences, including definitions, and related documentation and timelines.
  - For students who miss a quiz due to an excused absence, your quiz score will be counted as part of the final exam.
  - Specifically, your final exam weight will increase by 5% for each quiz missed with an excused absence (i.e., 30% + 5% × number of excused quiz absences).

Final (30%)

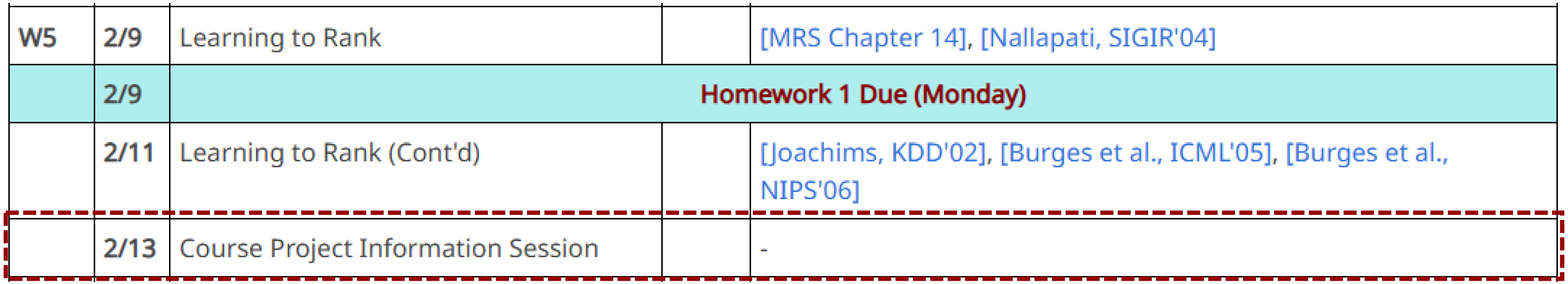
- In our regular classroom
- 3:30pm – 5:30pm on May 4 (Monday)
- 120 minutes; Comprehensive
- You can bring one cheatsheet
  - Cheatsheet = 8.5” x 11” standard sheet of paper with anything on it, front and back

Group Project (20%)

- Teams of 3 or 4 (any deviation from this size requires prior approval from the instructor)
  - Option 1: A prototype (search engine or recommender system)
  - Option 2: A research project (e.g., reasoning-search interleaved LLMs)
  - Option 3: A survey (e.g., recent studies on search-enhanced LLMs)
- Topic-wise: your choice, as long as it is related to information retrieval!
- Project presentations during our last four classes
- Super-fun opportunity for you to explore some compelling aspect of IR

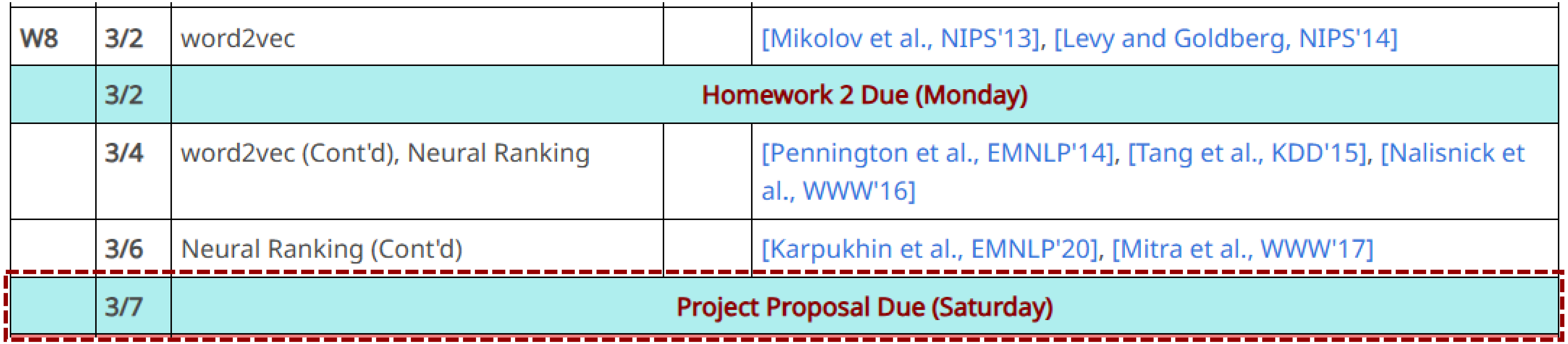
Group Project (20%)

- More details to be discussed in the Feb 13 class



W5	**2/9**	Learning to Rank		[MRS Chapter 14], [Nallapati, SIGIR'04]
	2/9	**Homework 1 Due (Monday)**		
	2/11	Learning to Rank (Cont'd)		[Joachims, KDD'02], [Burges et al., ICML'05], [Burges et al., NIPS'06]
	2/13	Course Project Information Session		-

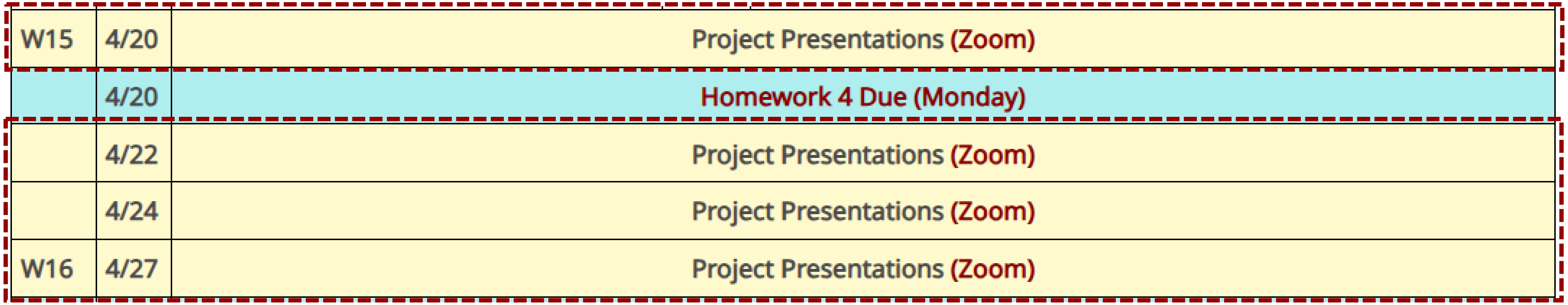
- Project proposal due on Mar 7 (so you still have plenty of time)



W8	**3/2**	word2vec		[Mikolov et al., NIPS'13], [Levy and Goldberg, NIPS'14]
	3/2	**Homework 2 Due (Monday)**		
	3/4	word2vec (Cont'd), Neural Ranking		[Pennington et al., EMNLP'14], [Tang et al., KDD'15], [Nalisnick et al., WWW'16]
	3/6	Neural Ranking (Cont'd)		[Karpukhin et al., EMNLP'20], [Mitra et al., WWW'17]
	3/7	**Project Proposal Due (Saturday)**		

Group Project (20%)

- Project presentations during our last four classes

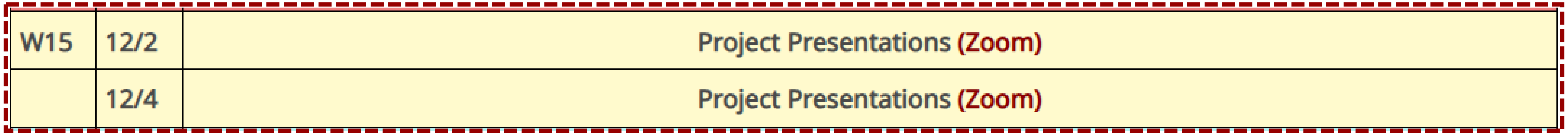


W15	4/20	Project Presentations (Zoom)
	4/20	Homework 4 Due (Monday)
	4/22	Project Presentations (Zoom)
	4/24	Project Presentations (Zoom)
W16	4/27	Project Presentations (Zoom)

- Held on Zoom
  - ❌ To start our summer break early
  - ✔ There will be 16-18 groups presenting in these lectures. Zoom allows us to quickly switch between shared screens, reducing transition time between groups and giving each group more time to present.

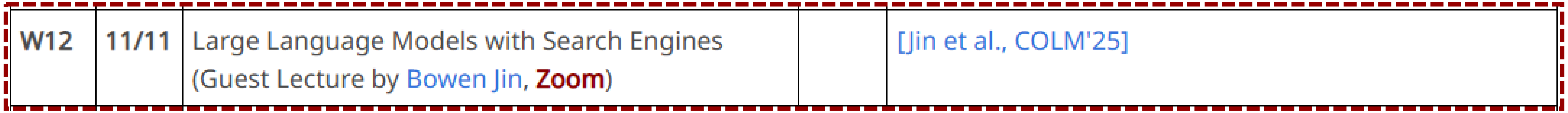
Zoom

- The following 3 classes will be held on Zoom.



W15	12/2	Project Presentations **(Zoom)**
	12/4	Project Presentations **(Zoom)**

  - ✗ To give you/myself a longer Thanksgiving break
  - ✓ There will be 14-18 groups presenting in these 2 lectures. Zoom allows us to quickly switch between shared screens, reducing transition time between groups and giving each group more time to present.

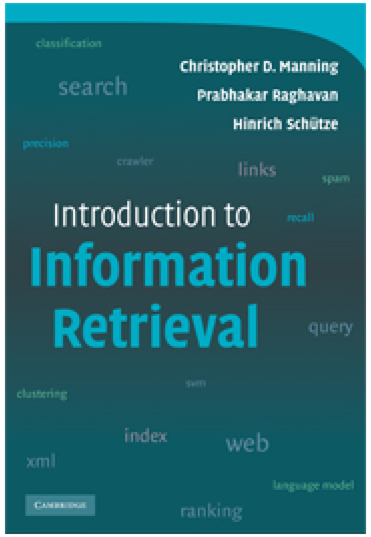


W12	**11/11**	Large Language Models with Search Engines (Guest Lecture by Bowen Jin, **Zoom**)		[Jin et al., COLM'25]

Overview of Information Retrieval (IR)

Information Retrieval is …

- "… *finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*"



Christopher D. Manning
Prabhakar Raghavan
Hinrich Schütze

Introduction to
Information
Retrieval

(according to Manning, Raghavan, Schutze 2008)

Information Retrieval is …

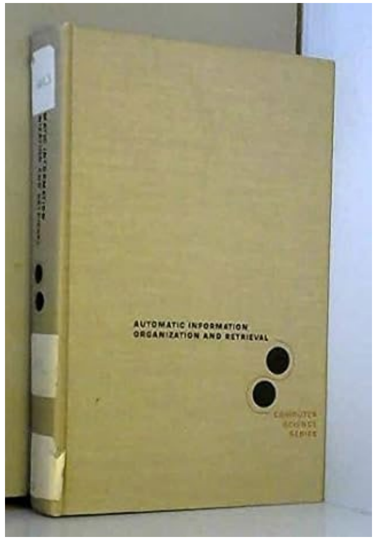
- “… *the process of obtaining information system resources that are relevant to an information need from a collection of those resources.*”

WIKIPEDIA
The Free Encyclopedia

(according to Wikipedia)

Information Retrieval is …

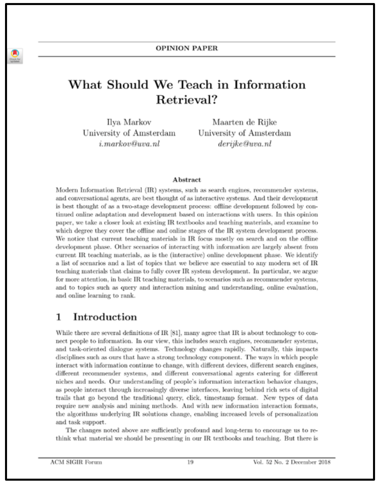
- *"… a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."*



(according to Gerard Salton "Father of IR" 1968)

Information Retrieval is …

- "… *about technology to connect people to information.*"



OPINION PAPER

What Should We Teach in Information Retrieval?

Ilya Markov
University of Amsterdam
i.markov@uva.nl

Maarten de Rijke
University of Amsterdam
derijke@uva.nl

Abstract

Modern Information Retrieval (IR) systems, such as search engines, recommender systems, and conversational agents, are best thought of as interactive systems. And their development is best thought of as a two-stage development process: offline development followed by continued online adaptation and development based on interactions with users. In this opinion paper, we take a closer look at existing IR textbooks and teaching materials, and examine to which degree they cover the offline and online stages of the IR system development process. We notice that current teaching materials in IR focus mostly on search and on the offline development phase. Other scenarios of interacting with information are largely absent from current IR teaching materials, as is the (interactive) online development phase. We identify a list of scenarios and a list of topics that we believe are essential to any modern set of IR teaching materials that claims to fully cover IR system development. In particular, we argue for more attention, in basic IR teaching materials, to scenarios such as recommender systems, and to topics such as query and interaction mining and understanding, online evaluation, and online learning to rank.

1 Introduction

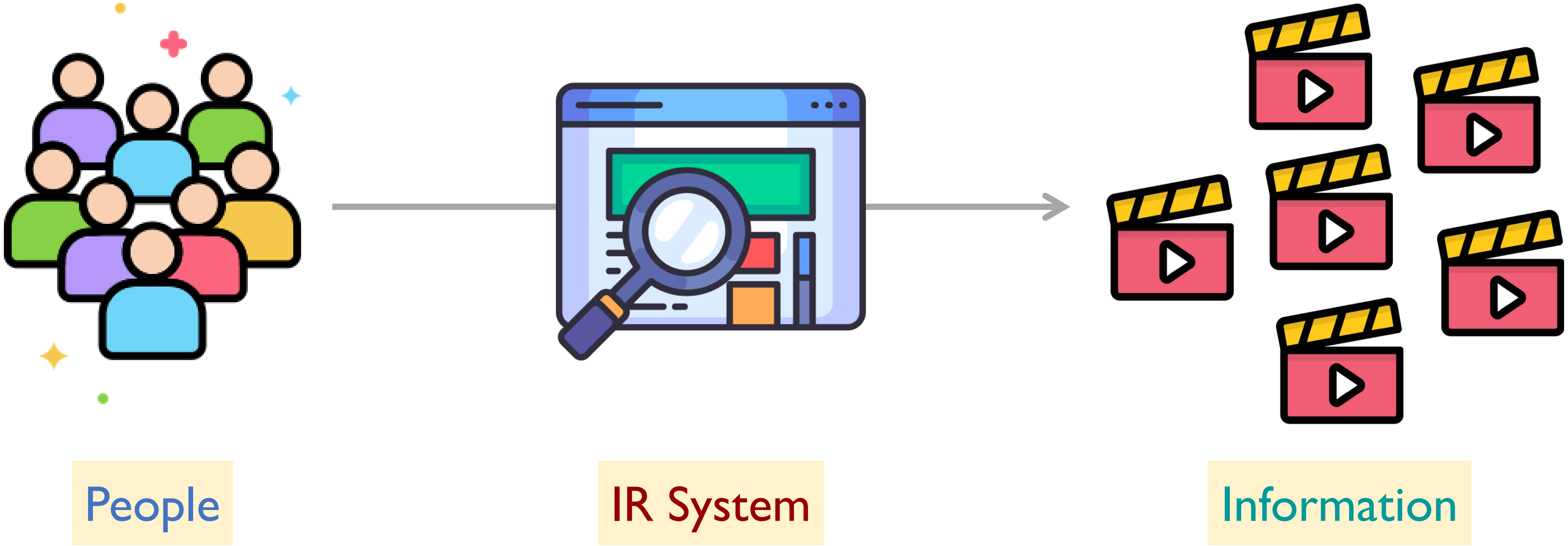
While there are several definitions of IR [81], many agree that IR is about technology to connect people to information. In our view, this includes search engines, recommender systems, and task-oriented dialogue systems. Technology changes rapidly. Naturally, this impacts disciplines such as ours that have a strong technology component. The ways in which people interact with information continue to change, with different devices, different search engines, different recommender systems, and different conversational agents catering for different niches and needs. Our understanding of people's information interaction behavior changes, as people interact through increasingly diverse interfaces, leaving behind rich sets of digital trails that go beyond the traditional query, click, timestamp format. New types of data require new analysis and mining methods. And with new information interaction formats, the algorithms underlying IR solutions change, enabling increased levels of personalization and task support.

The changes noted above are sufficiently profound and long-term to encourage us to rethink what material we should be presenting in our IR textbooks and teaching. But there is

ACM SIGIR Forum 19 Vol. 52 No. 2 December 2018

(according to Markov and de Rijke 2018)

IR connects people to information



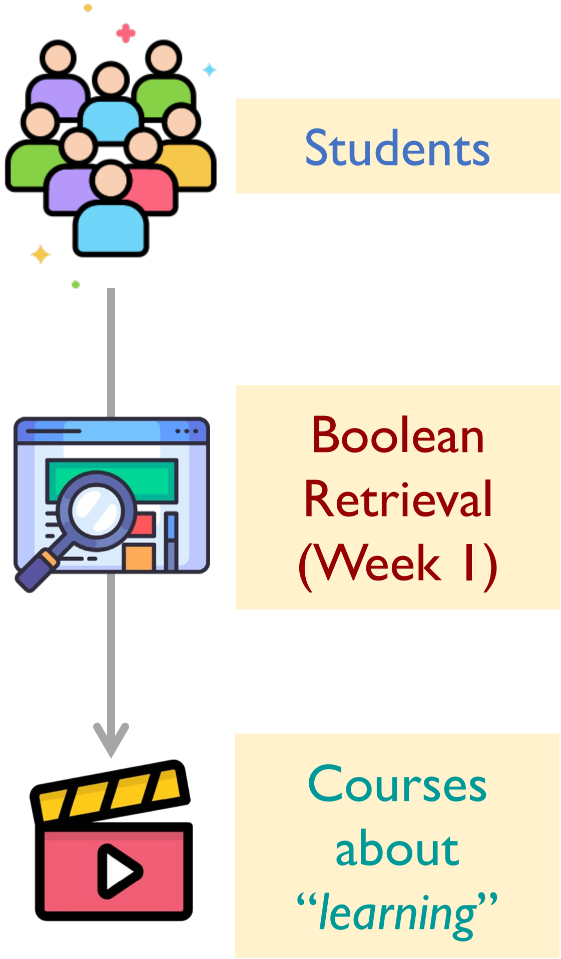
People

IR System

Information

- Examples?

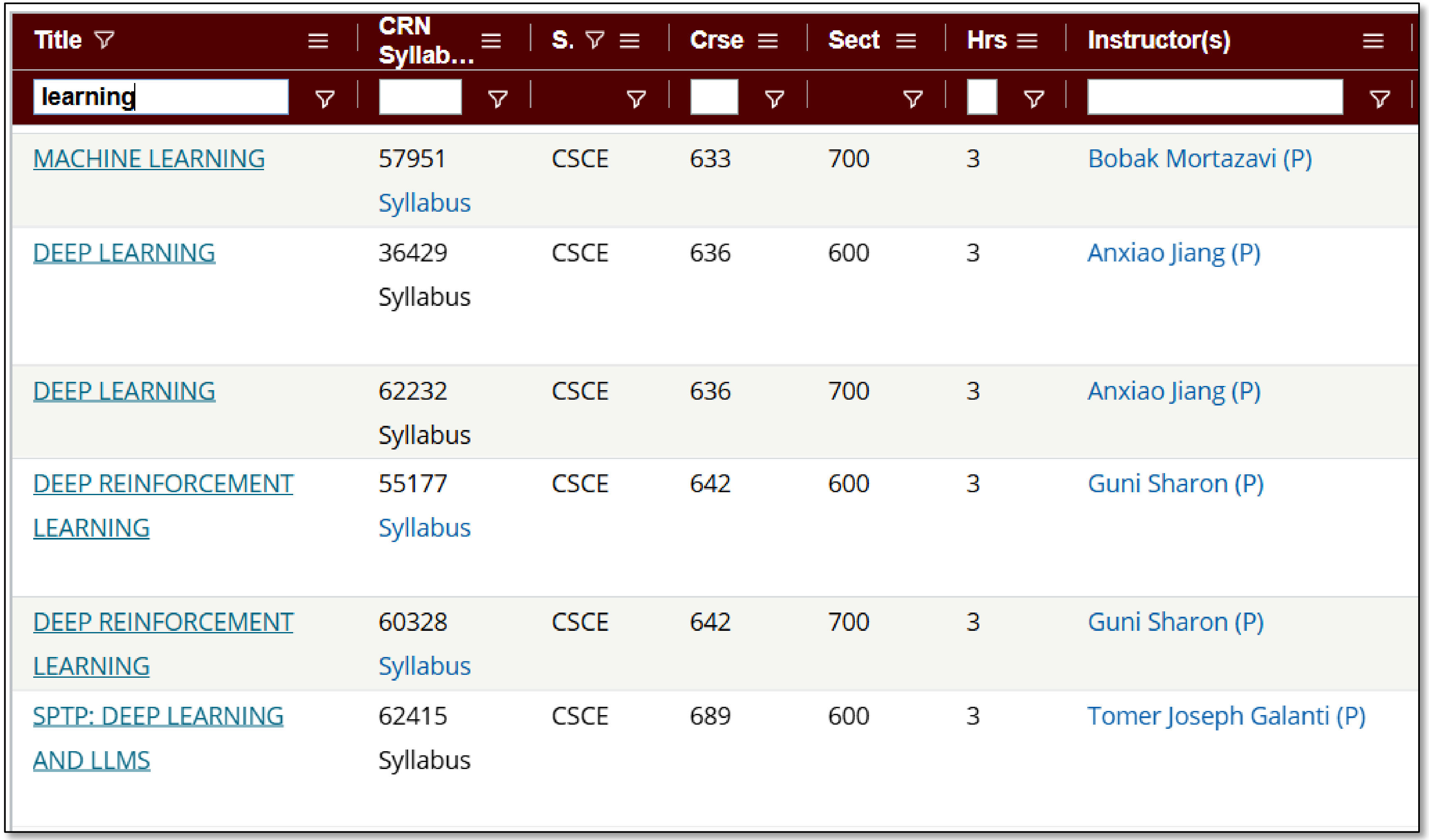
Example: Course Explorer



Students

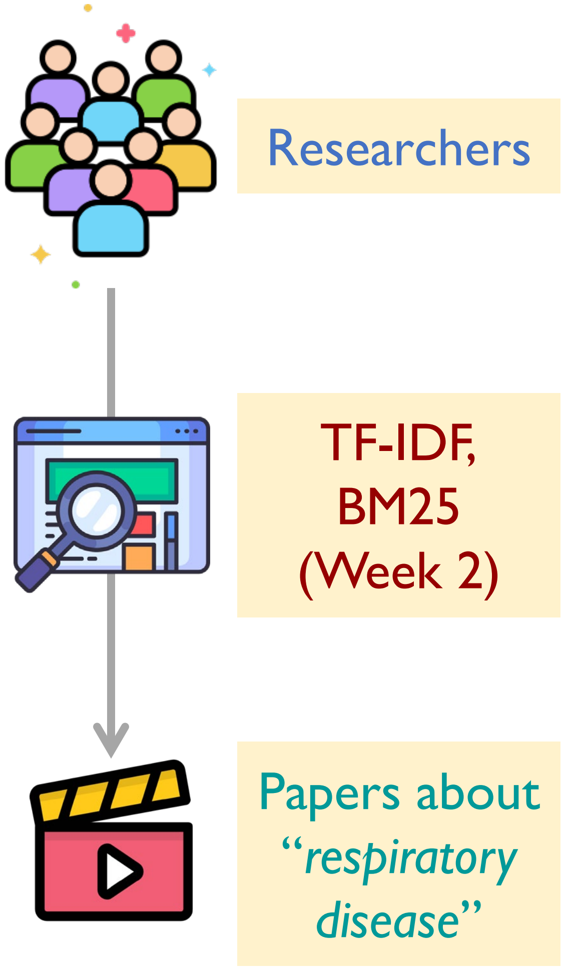
Boolean Retrieval (Week 1)

Courses about *"learning"*



Title	CRN Syllab...	S.	Crse	Sect	Hrs	Instructor(s)
learning						
MACHINE LEARNING	57951 Syllabus	CSCE	633	700	3	Bobak Mortazavi (P)
DEEP LEARNING	36429 Syllabus	CSCE	636	600	3	Anxiao Jiang (P)
DEEP LEARNING	62232 Syllabus	CSCE	636	700	3	Anxiao Jiang (P)
DEEP REINFORCEMENT LEARNING	55177 Syllabus	CSCE	642	600	3	Guni Sharon (P)
DEEP REINFORCEMENT LEARNING	60328 Syllabus	CSCE	642	700	3	Guni Sharon (P)
SPTP: DEEP LEARNING AND LLMS	62415 Syllabus	CSCE	689	600	3	Tomer Joseph Galanti (P)

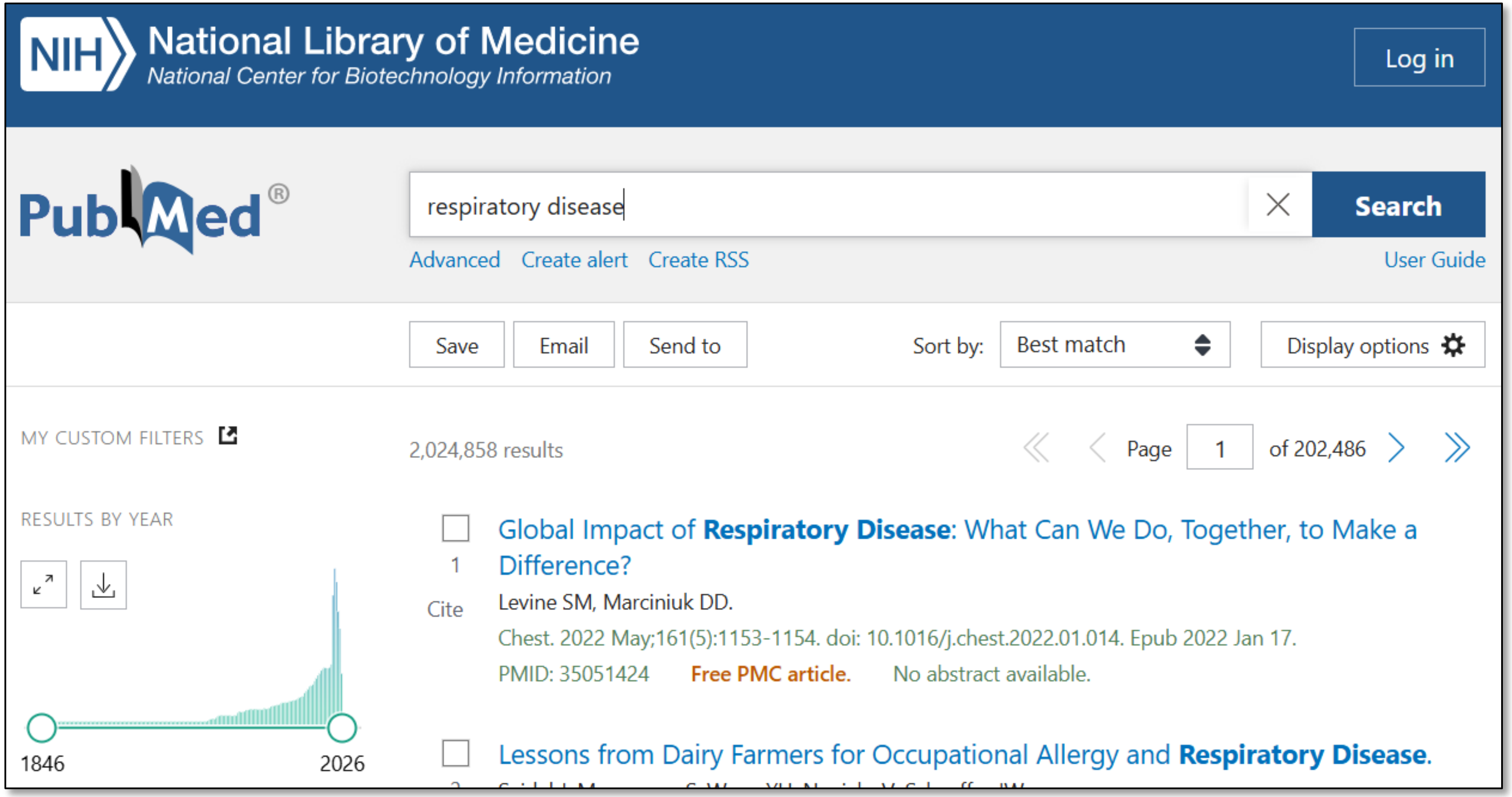
Example: PubMed, Semantic Scholar, Google Scholar



Researchers

TF-IDF, BM25 (Week 2)

Papers about *“respiratory disease”*



NIH National Library of Medicine

National Center for Biotechnology Information

Log in

PubMed®

respiratory disease

Search

Advanced Create alert Create RSS User Guide

Save Email Send to Sort by: Best match Display options

MY CUSTOM FILTERS

RESULTS BY YEAR

1846 2026

2,024,858 results

Page 1 of 202,486

☐ 1 Cite

Global Impact of **Respiratory Disease**: What Can We Do, Together, to Make a Difference?

Levine SM, Marciniuk DD.

Chest. 2022 May;161(5):1153-1154. doi: 10.1016/j.chest.2022.01.014. Epub 2022 Jan 17.

PMID: 35051424 **Free PMC article.** No abstract available.

☐

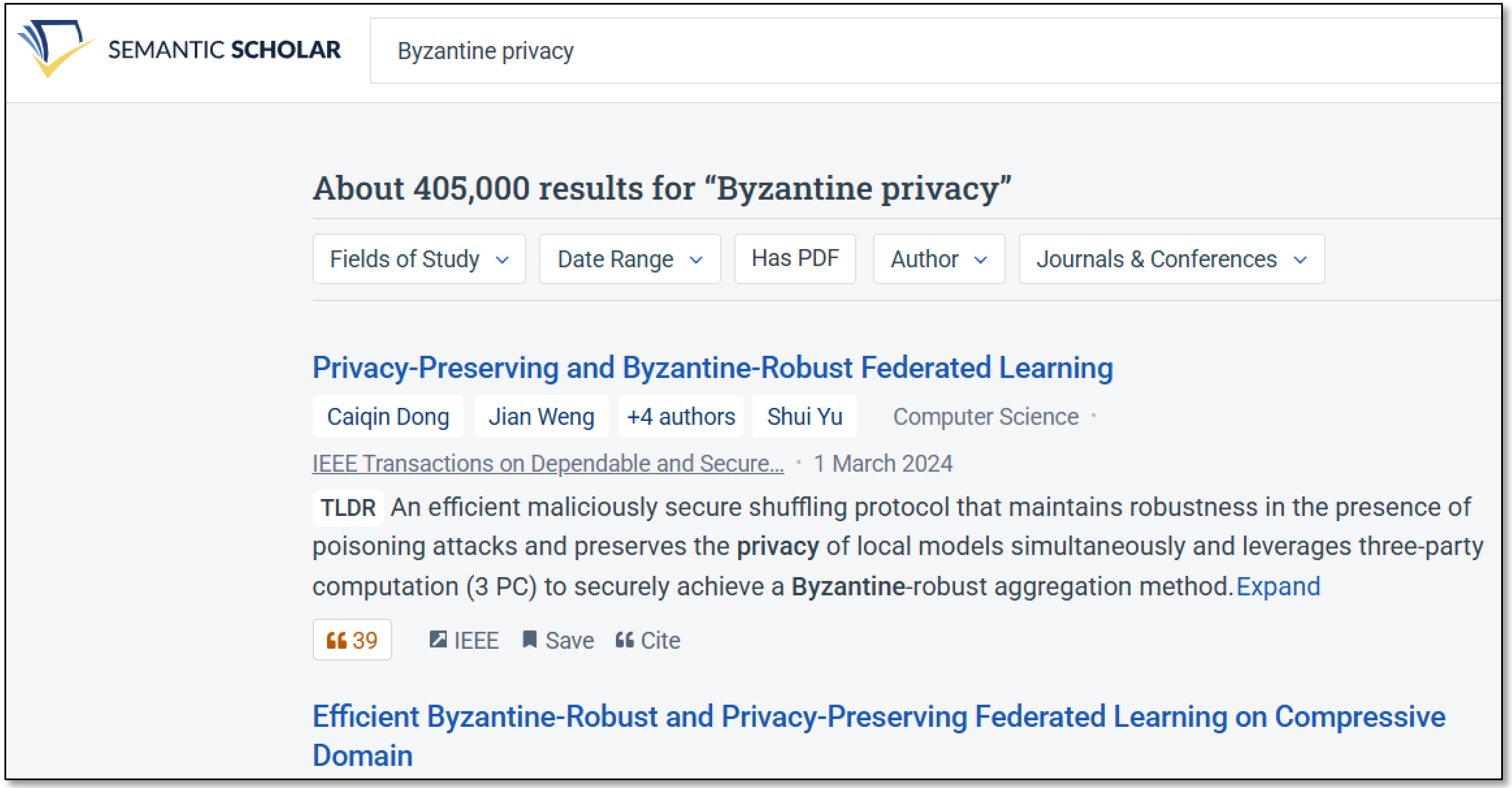
Lessons from Dairy Farmers for Occupational Allergy and **Respiratory Disease**.

Example: PubMed, Semantic Scholar, Google Scholar

Researchers

word2vec, BERT (Weeks 8 and 10)

Papers about *“Byzantine privacy”*



About 405,000 results for “Byzantine privacy”

Privacy-Preserving and Byzantine-Robust Federated Learning

Caiqin Dong · Jian Weng · +4 authors · Shui Yu · Computer Science ·

IEEE Transactions on Dependable and Secure... · 1 March 2024

TLDR An efficient maliciously secure shuffling protocol that maintains robustness in the presence of poisoning attacks and preserves the **privacy** of local models simultaneously and leverages three-party computation (3 PC) to securely achieve a **Byzantine**-robust aggregation method. Expand

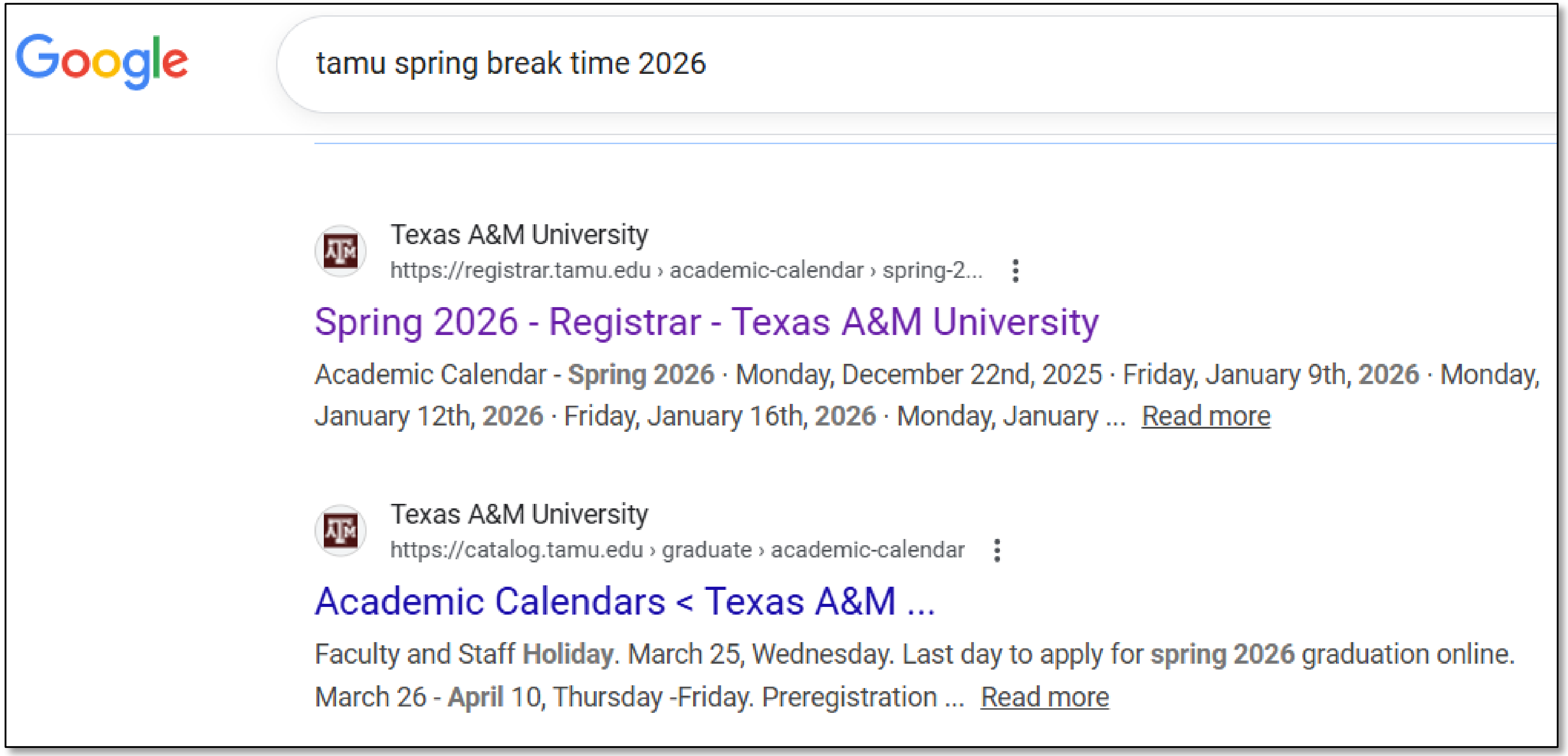
Efficient Byzantine-Robust and Privacy-Preserving Federated Learning on Compressive Domain

Example: Google

Aggies

PageRank (Week 3)

Webpages about "*tamu spring break time 2026*"



tamu spring break time 2026

Texas A&M University
https://registrar.tamu.edu › academic-calendar › spring-2...

Spring 2026 - Registrar - Texas A&M University

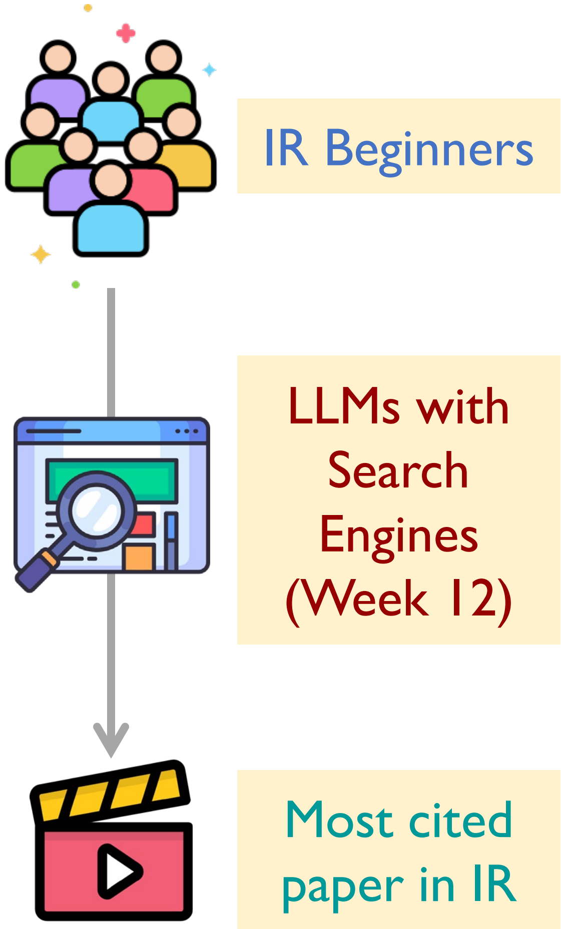
Academic Calendar - **Spring 2026** · Monday, December 22nd, 2025 · Friday, January 9th, **2026** · Monday, January 12th, **2026** · Friday, January 16th, **2026** · Monday, January ... Read more

Texas A&M University
https://catalog.tamu.edu › graduate › academic-calendar

Academic Calendars < Texas A&M ...

Faculty and Staff **Holiday**. March 25, Wednesday. Last day to apply for **spring 2026** graduation online. March 26 - **April** 10, Thursday -Friday. Preregistration ... Read more

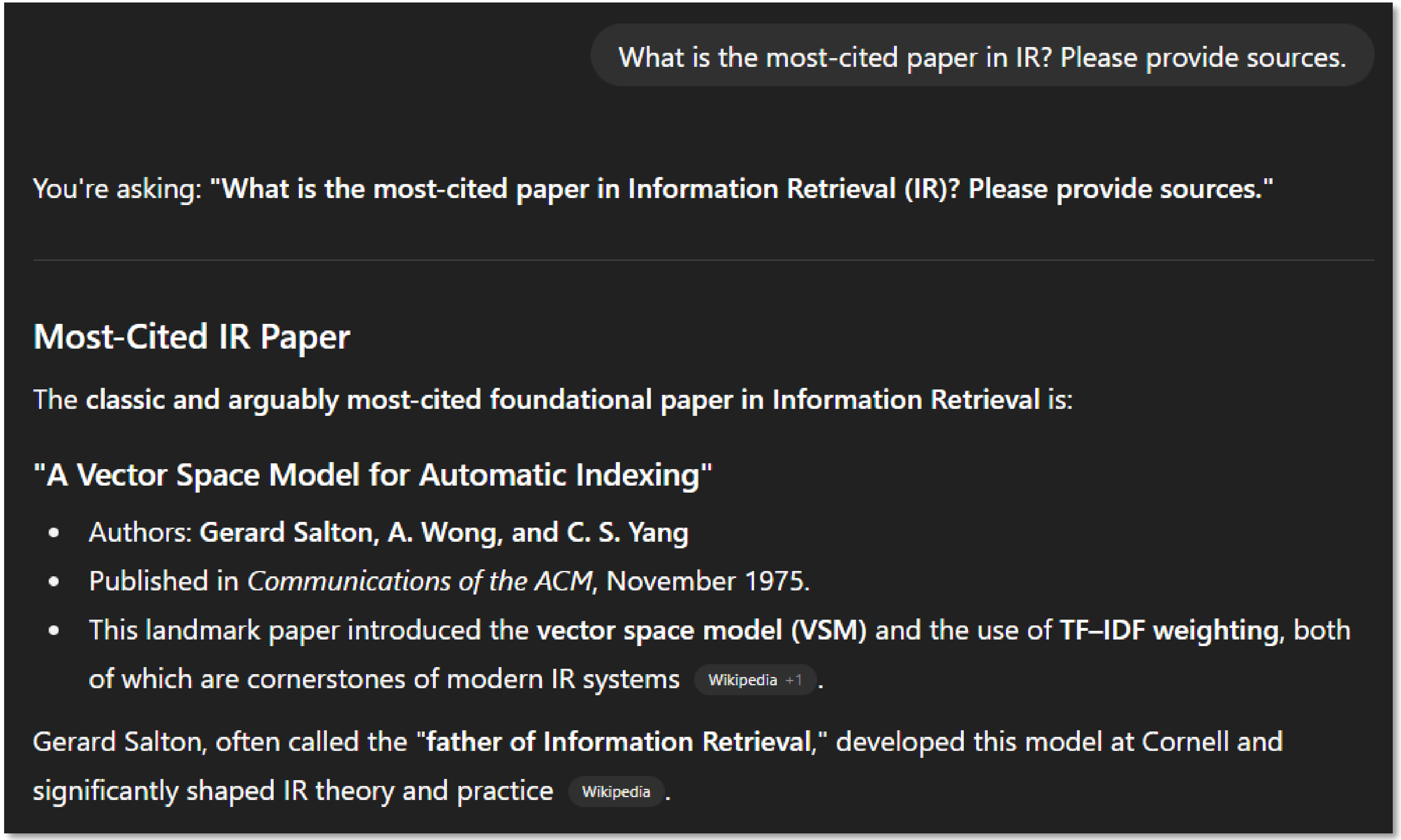
Example: ChatGPT



IR Beginners

LLMs with Search Engines (Week 12)

Most cited paper in IR



What is the most-cited paper in IR? Please provide sources.

You're asking: **"What is the most-cited paper in Information Retrieval (IR)? Please provide sources."**

Most-Cited IR Paper

The **classic and arguably most-cited foundational paper in Information Retrieval** is:

"A Vector Space Model for Automatic Indexing"

- Authors: **Gerard Salton, A. Wong, and C. S. Yang**
- Published in *Communications of the ACM*, November 1975.
- This landmark paper introduced the **vector space model (VSM)** and the use of **TF–IDF weighting**, both of which are cornerstones of modern IR systems.

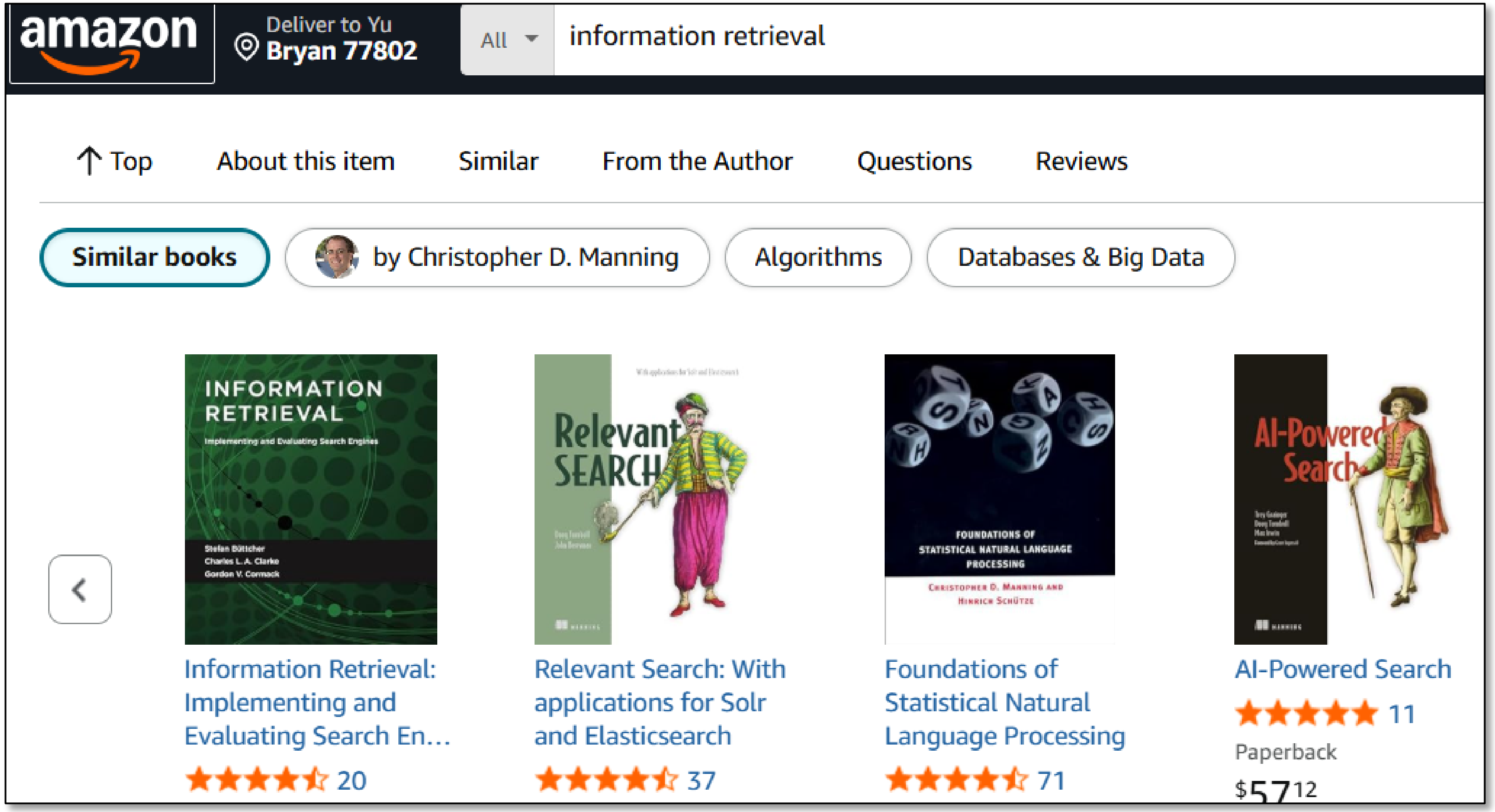
Gerard Salton, often called the **"father of Information Retrieval,"** developed this model at Cornell and significantly shaped IR theory and practice.

Example: Amazon

Customers who bought an IR book

Recommender Systems (Weeks 6-7)

Other books the customers may also like



information retrieval

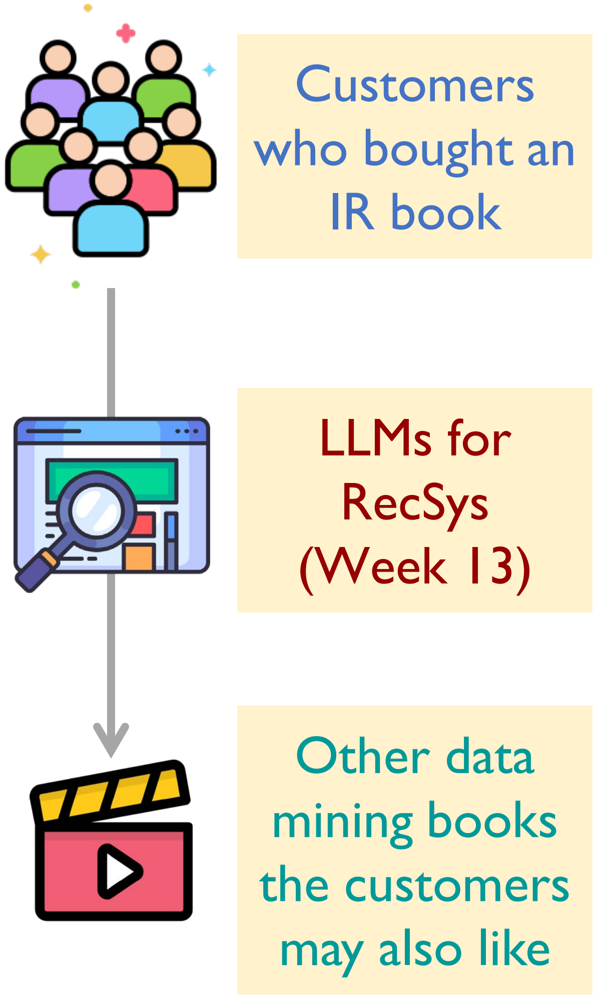
Information Retrieval: Implementing and Evaluating Search En… ★★★★☆ 20

Relevant Search: With applications for Solr and Elasticsearch ★★★★☆ 37

Foundations of Statistical Natural Language Processing ★★★★☆ 71

AI-Powered Search ★★★★★ 11
Paperback
$57.12

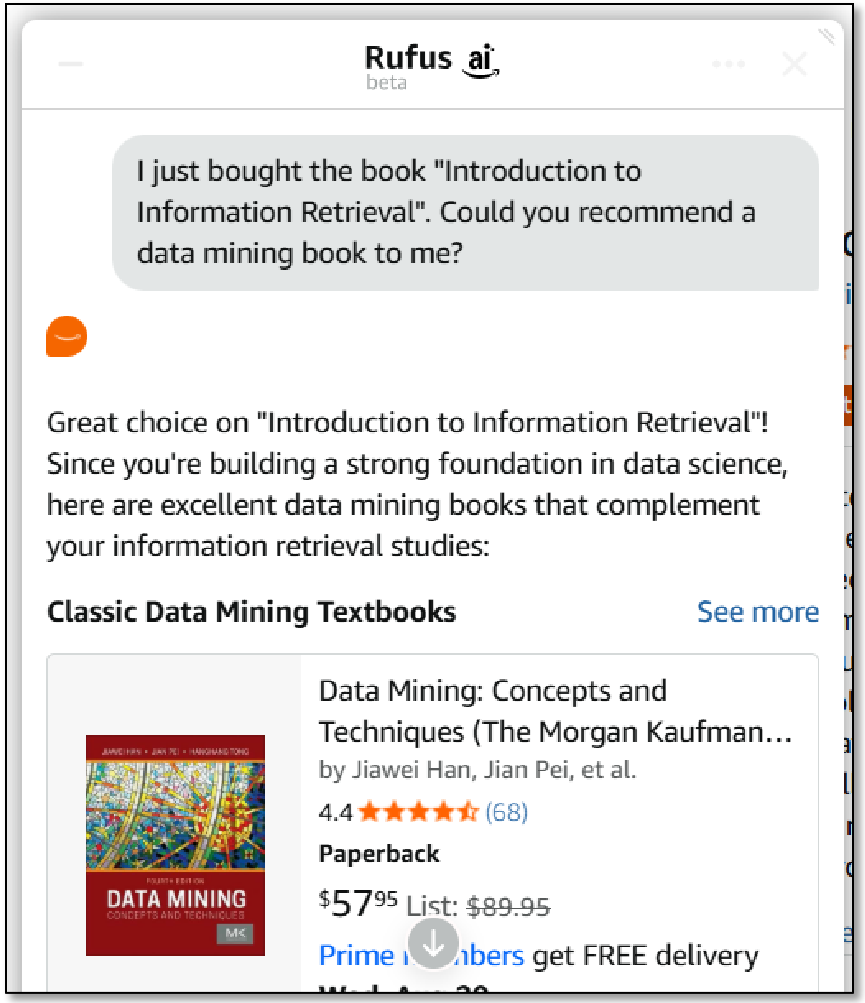
Example: Amazon



Customers who bought an IR book

LLMs for RecSys (Week 13)

Other data mining books the customers may also like

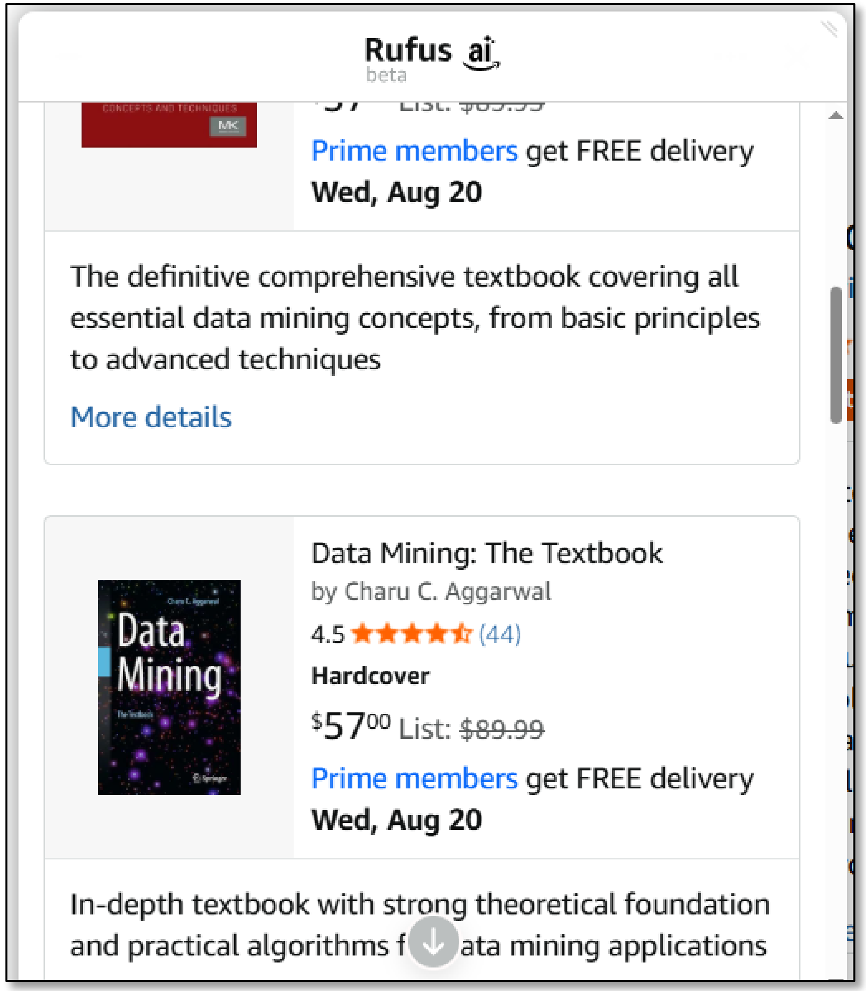


I just bought the book "Introduction to Information Retrieval". Could you recommend a data mining book to me?

Great choice on "Introduction to Information Retrieval"! Since you're building a strong foundation in data science, here are excellent data mining books that complement your information retrieval studies:

Classic Data Mining Textbooks See more

Data Mining: Concepts and Techniques (The Morgan Kaufman…
by Jiawei Han, Jian Pei, et al.
4.4 ★★★★☆ (68)
Paperback
$57.95 List: ~~$89.95~~
Prime [illegible]bers get FREE delivery



Prime members get FREE delivery
Wed, Aug 20

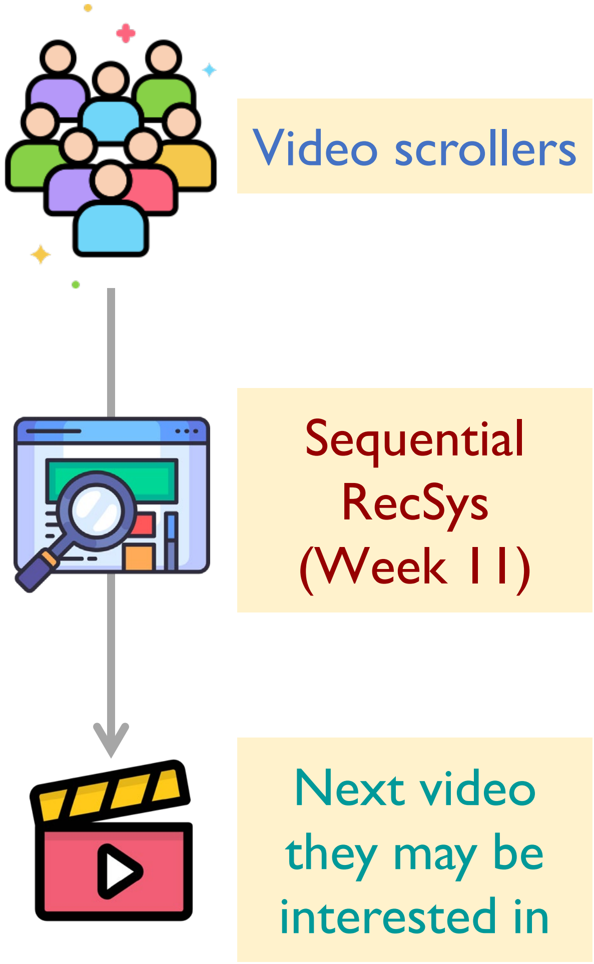
The definitive comprehensive textbook covering all essential data mining concepts, from basic principles to advanced techniques

More details

Data Mining: The Textbook
by Charu C. Aggarwal
4.5 ★★★★☆ (44)
Hardcover
$57.00 List: ~~$89.99~~
Prime members get FREE delivery
Wed, Aug 20

In-depth textbook with strong theoretical foundation and practical algorithms f[illegible]ata mining applications

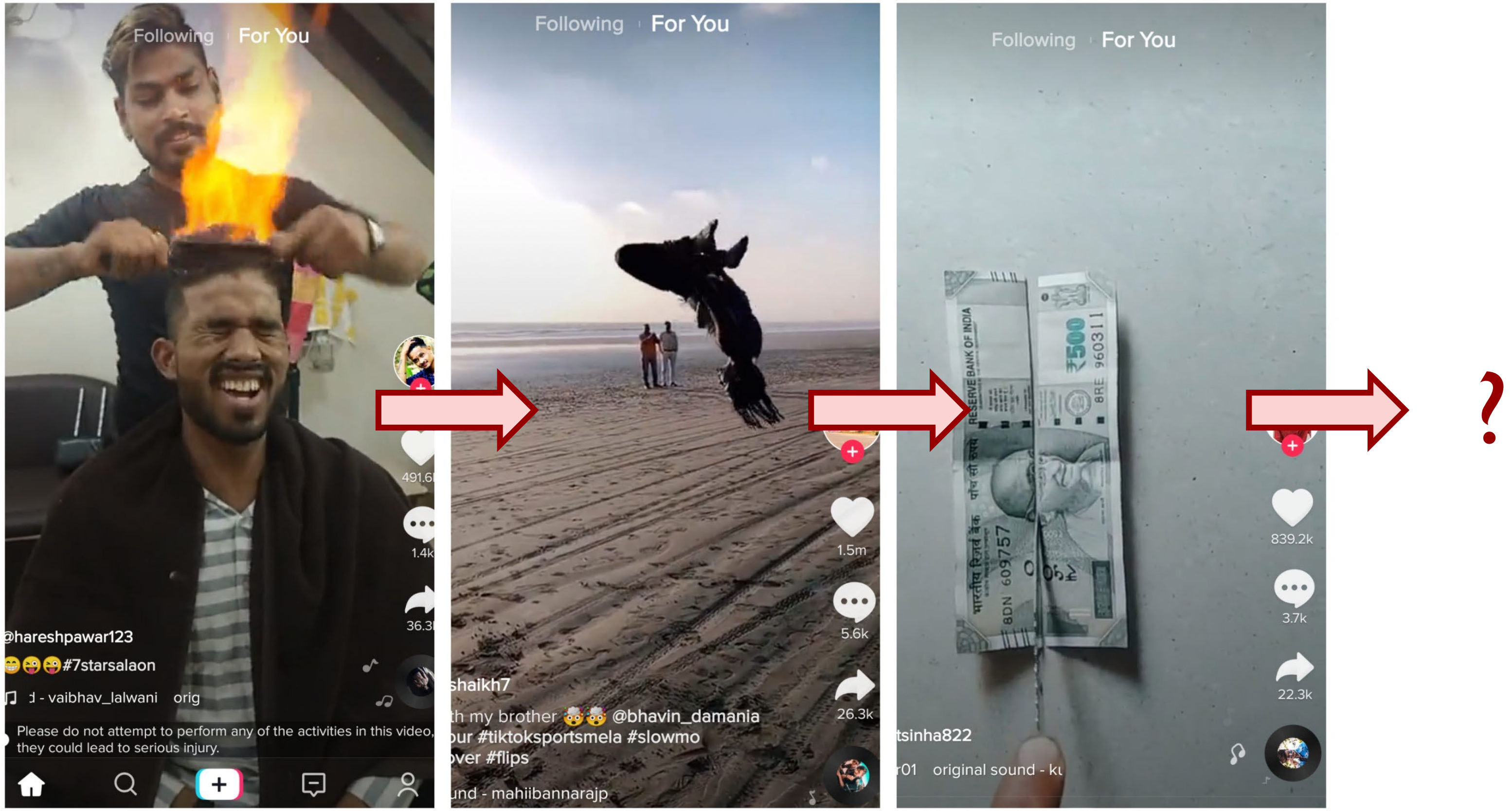
Example: TikTok



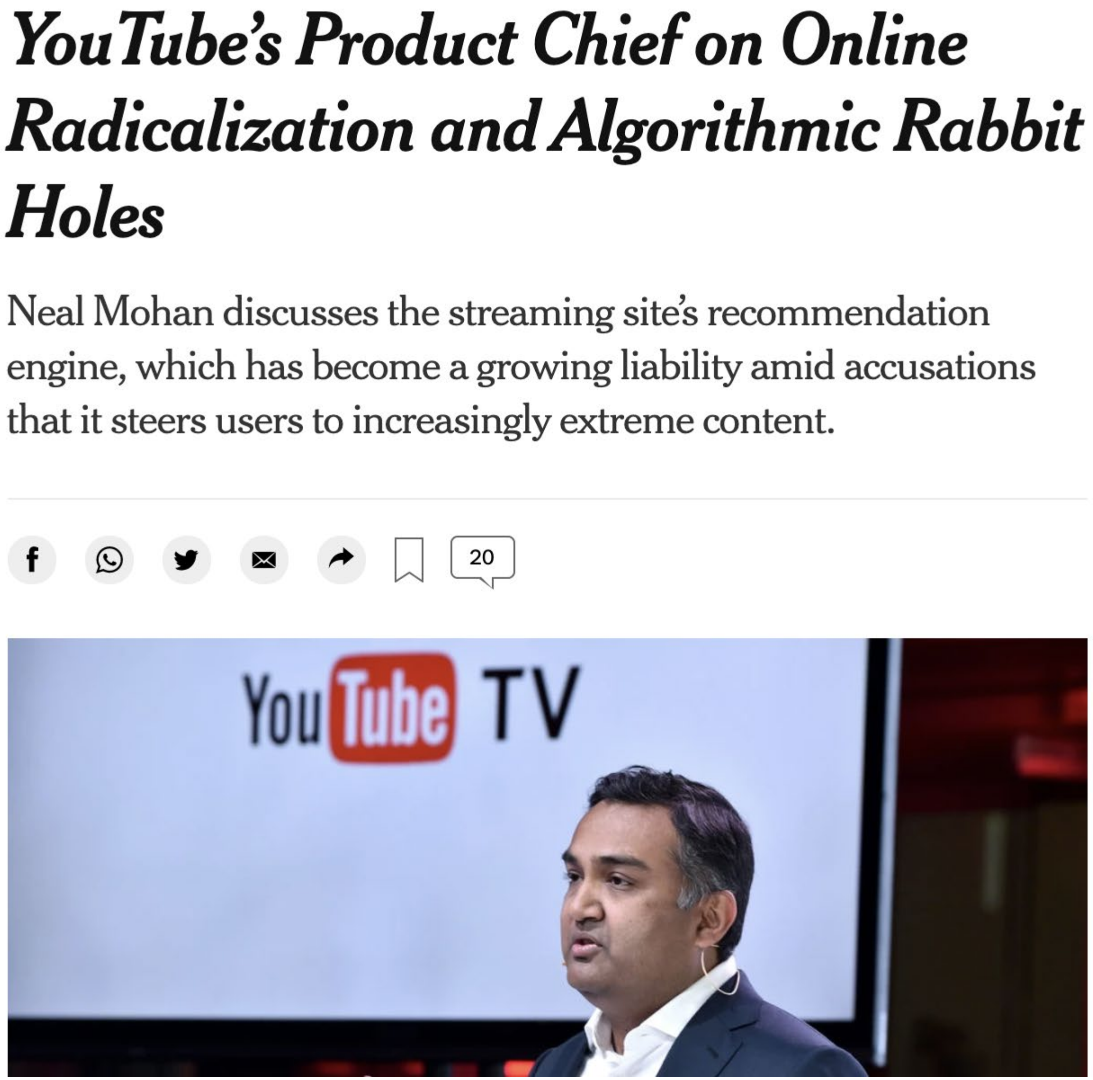
Video scrollers

Sequential RecSys (Week 11)

Next video they may be interested in



IR algorithmically mediates what items a user encounters



YouTube's Product Chief on Online Radicalization and Algorithmic Rabbit Holes

Neal Mohan discusses the streaming site's recommendation engine, which has become a growing liability amid accusations that it steers users to increasingly extreme content.

We as computer scientists need to …

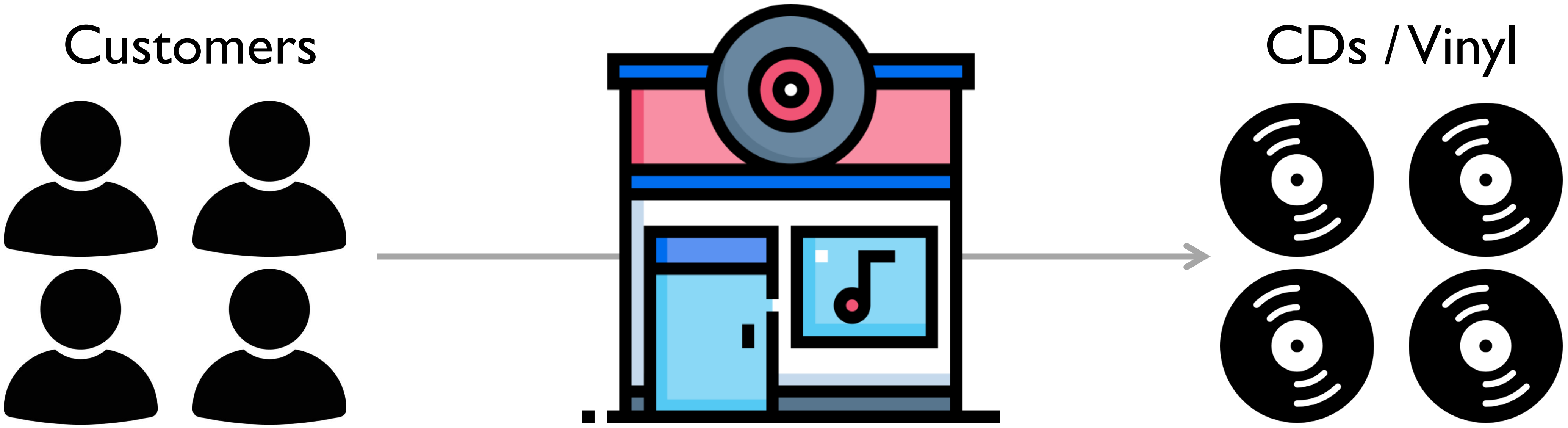
- Understand these algorithms
  - How can we build a search engine or a recommender system? What algorithms can we use? What "features" are important? How to evaluate if it is doing a good job?
- Design new approaches
  - Can we keep pace with rapid developments in industry and in academia? Adopt new ML/DL approaches? Anticipate the "next" big thing?
- Be mindful of the power we wield! Important issues around fairness, bias, misinformation, and other negative outcomes.

This Course

- Phase 1: Search Engines
  - Boolean and ranked retrieval, link analysis, evaluation, learning to rank (ML + ranking), …
- Phase 2: Recommender Systems
  - content-based recommendation, collaborative filtering, matrix factorization, recommendation with implicit feedback, …
- Phase 3: From Foundations to Modern Methods
  - word embedding learning, Transformer, "small" language models, … (for search and recommendation)
- Phase 4: Large Language Models (!!)

Boolean Retrieval

We are opening a record store!

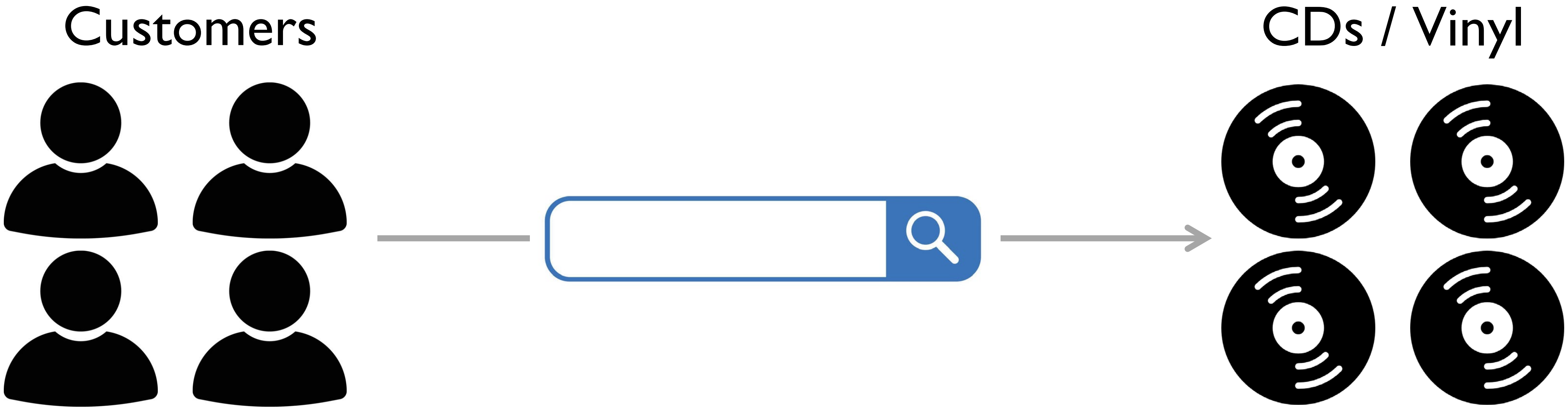


Customers

CDs / Vinyl

- Over the course of the semester, we will progressively build up the search and recommendation capabilities of our store.
  - This + next few weeks: Focus on search basics
  - Then: Focus on recommendation basics
  - Later: Revisit both search and recommendation via advanced topics (e.g., LLMs)

This + Next Few Weeks: Help Users Search our Store



Customers

CDs / Vinyl

Basic Concepts

- Information need
  - *“I want Taylor Swift's latest album.”*
- Query
  - *“Taylor Swift's latest album”*
  - *“Taylor Swift album 2025”*
  - …
- Documents
  - A pool of candidates (e.g., CDs)
  - Some candidates may satisfy the information need.
  - Each candidate is associated with some text information.
    - *“Artist: Taylor Swift; Lyrics: Meet me at midnight, Staring at the ceiling with you …”*

Basic Concepts

- Query Representation
  - If we want to design an automated algorithm for search, we need a way to represent the query that a computer can understand.
  - *[0, 1, 1, 0, …]*
  - *[0.255, -1.342, …]*
- Document Representation
  - We need to use a similar way to represent each “document” (i.e., candidate).
- Relevance Function
  - How can we decide which “document” can satisfy the information need?
  - *Relevance(Query, Document1) = 0.8*
  - *Relevance(Query, Document2) = 0.5*
  - …

Key Challenges Motivating Much of this Course

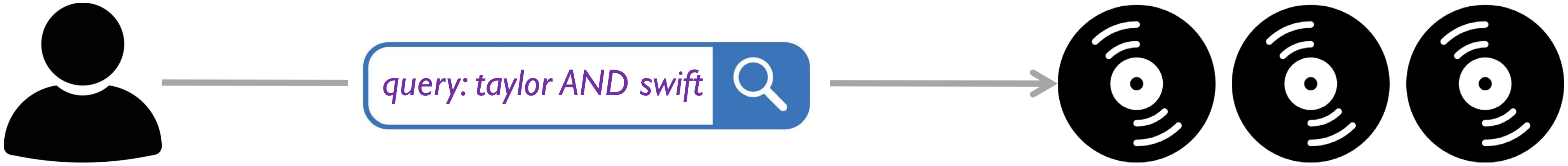
- How do we represent our queries and documents?
    - What is our “representation function”?

- What is our relevance function?

- How do we know if we are doing a good job?

Today: Simplifying Assumptions

- Our store front-page only supports Boolean keyword queries.
  - “*karma*”
  - “*love OR song*”
  - “*taylor AND swift*”
  - “*taylor AND (NOT swift)*”
  - …
- Based on these queries, we return a list of matching albums.



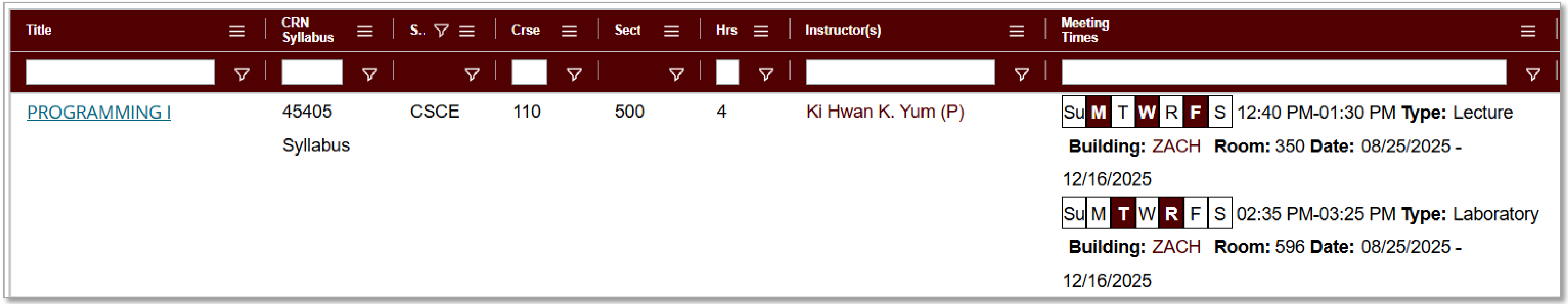
We return a set of matching albums. (No rank order!)

Example Album

- Artist: *Taylor Swift*
- Album Title: *Midnights*
- Year: *2022*
- Track Listing: *Lavender Haze, Maroon, Anti-Hero, …*
- Lyrics: *Meet me at midnight, Staring at the ceiling with you, Oh, you don't ever say too …*

What do our users want to search for?

- Another example where each “document” has multiple fields: Course Explorer



Title	CRN Syllabus	S..	Crse	Sect	Hrs	Instructor(s)	Meeting Times
PROGRAMMING I	45405 Syllabus	CSCE	110	500	4	Ki Hwan K. Yum (P)	Su **M** T **W** R **F** S 12:40 PM-01:30 PM **Type:** Lecture **Building:** ZACH **Room:** 350 **Date:** 08/25/2025 - 12/16/2025 Su M **T** W **R** F S 02:35 PM-03:25 PM **Type:** Laboratory **Building:** ZACH **Room:** 596 **Date:** 08/25/2025 - 12/16/2025

- What parts of an album do we index?
- How to index these parts?
    - Everything in one index?
    - Some parts in one index?
    - Each facet in its own separate index?

One More Simplifying Assumption

- Record store front-page only supports Boolean keyword queries over song lyrics.

- Artist: *Taylor Swift*
- Album Title: *Midnights*
- Year: *2022*
- Track Listing: *Lavender Haze, Maroon, Anti-Hero, …*
- Lyrics: *Meet me at midnight, Staring at the ceiling with you, Oh, you don't ever say too …*

A Simple (but not Good) Boolean Retrieval Algorithm

- Query: *midnight AND staring*
- Algorithm Pseudo Code:

```
results = []
For CD in CDs:
    If "midnight" in CD.lyrics and "staring" in CD.lyrics:
        results.append(CD)
return results
```

A Simple (but not Good) Boolean Retrieval Algorithm

- Query: *believe OR (NOT love)*

- Algorithm Pseudo Code:

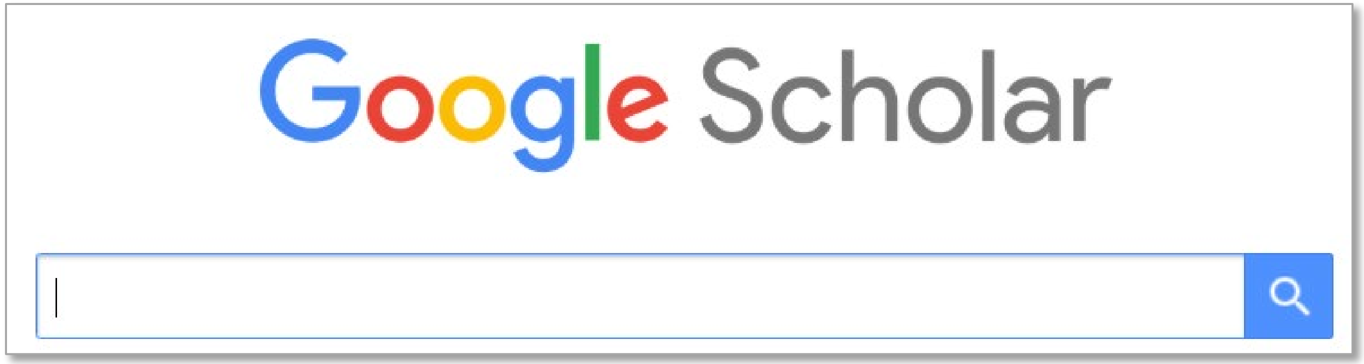
```
results = []
For CD in CDs:
    If “believe” in CD.lyrics or “love” not in CD.lyrics:
        results.append(CD)
return results
```

- What if a new CD arrives in our store?

```
CDs.append(new_CD)
```

Problems?

- For each query, we need to scan all the documents.
  - If there are M documents in total, and each document has N words on average, the time complexity will be $O(MN)$.



- 240,000,000+ papers on the Web by the end of 2019 [1].
- Let's assume that the title and abstract of each paper contain about 200 words.

  - Scanning 48 billion words for each query!
  - If you had to wait several minutes every time to get the paper you are looking for, would you still use this academic search engine?

[1] *Microsoft Academic Graph: When Experts are Not Enough.* Quantitative Science Studies 2020.

Inverted Index: A More Efficient Solution

- Document 1 (d1): "*any choose love*"
- Document 2 (d2): "*zebra any love*"
- …

Vocabulary
any
believe
choose
love
midnight
starring
zebra

Inverted Index: A More Efficient Solution

- Document 1 (d1): “*any choose love*”
- Document 2 (d2): “*zebra any love*”
- …

Vocabulary		
any	→	d1
believe		
choose	→	d1
love	→	d1
midnight		
starring		
zebra		

Inverted Index: A More Efficient Solution

- Document 1 (d1): “*any choose love*”
- Document 2 (d2): “*zebra any love*”
- …

Vocabulary

any → d1 → d2

believe

choose → d1

love → d1 → d2

midnight

starring

zebra → d2

Inverted Index: A More Efficient Solution

Vocabulary

any → d1 → d2 → d5
believe → d4
choose → d1 → d5 → d6 → d10
love → d1 → d2 → d8
midnight → d7 → d9
starring → d7 → d9
zebra → d2 → d8

- Query: “*any*”
  - {d1, d2, d5}
- Query: “*any AND zebra*”
  - {d1, d2, d5} ∩ {d2, d8} = {d2}

Inverted Index: A More Efficient Solution

Vocabulary

any → d1 → d2 → d5
believe → d4
choose → d1 → d5 → d6 → d10
love → d1 → d2 → d8
midnight → d7 → d9
starring → d7 → d9
zebra → d2 → d8

- Query: "*believe OR midnight*"
  - {d4} ∪ {d7, d9} = {d4, d7, d9}
- Query: "*any AND (NOT zebra)*"
  - {d1, d2, d5} – {d2, d8} = {d1, d5}

Questions?

Inverted Index: A More Efficient Solution

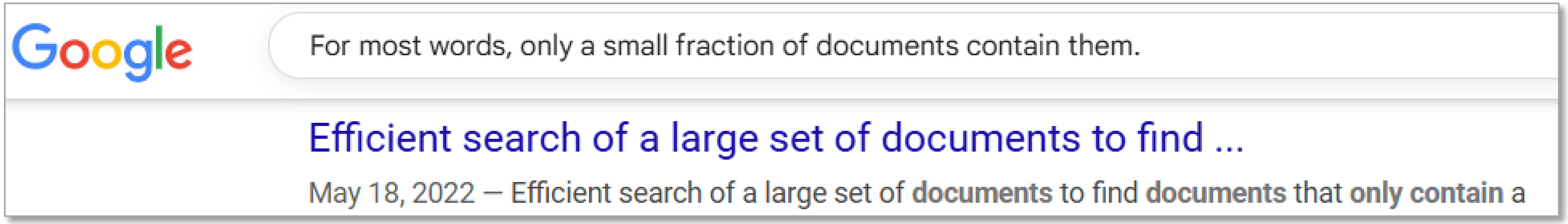
Vocabulary

any → d1 → d2 → d5
believe → d4
choose → d1 → d5 → d6 → d10
love → d1 → d2 → d8
…

- What if a new CD (d1000) arrives in our store?
  - Scan its lyrics and update our inverted index
  - Move the scanning process into index construction, so it does not take up time during the on-the-fly processing of user queries.
  - Moreover, the data only needs to be scanned once, instead of being scanned for every query.

Inverted Index: A More Efficient Solution

- What is the time complexity of the Boolean retrieval process now?
  - Number of words in the query: usually less than 20
  - × Time to access the linked list of a word: $O(1)$
  - + Set operations based on these linked lists: proportional to the total length of these linked lists (i.e., the total number of documents containing these words)
    - For most words, only a small fraction of documents contain them.
    - The remaining words (i.e., stop words) appear in many documents and are typically considered uninformative, so they are usually ignored.



For most words, only a small fraction of documents contain them.

Efficient search of a large set of documents to find ...

May 18, 2022 — Efficient search of a large set of documents to find **documents** that **only contain** a

Summary: Boolean Retrieval

- Advantages
  - Precise, if you know the right strategies (e.g., how to iteratively refine your queries, use of boolean operators, …)
  - Typically, efficient in practice

- Disadvantages
  - Users must understand Boolean logic!
  - Boolean logic does not capture language richness.
  - Feast or famine in results: often get 0 results or 1000s
  - Result sets are unordered.
  - What about partial matches? E.g., a document does not exactly match the query but it is "close"?

Can we improve the index?

- To support phrase queries?
  - *“taylor swift”,“670 homework solution”*
- To support proximity queries?
  - *“taylor NEAR:2 swift”,“670 NEAR:5 solution”*
- To support wildcard queries?
  - *“tayl*”,“*ift”*

Phrase Queries

- “*taylor swift*”, “*670 homework solution*”
- Why might we like to support phrase queries?
    - “*taylor made a swift decision to pivot the project after noticing the early results.*”
    - “*the professor teaches both 670 and 698, but only the 698 homework solution was shared on the course website.*”

Phrase Queries: One Idea

- Bigram index: Index every consecutive pair of terms in the text as a phrase
  - "*taylor made a swift decision …*"

Vocabulary		
taylor	→	…
made	→	…
…		
taylor made	→	…
made a	→	…
a swift	→	…
swift decision	→	…
…		

- What if the query has three words?
  - Trigram index: Index every consecutive span of three terms in the text as a phrase.
- What if the query has four words?
  - …
- Problems with this strategy?

Instead: Positional Index

- Store the position in the index!
- Document 1 (d1):"*any choose love*"
- Document 2 (d2):"*zebra any love any zebra*"

Vocabulary
any
believe
choose
love
midnight
starring
zebra

Instead: Positional Index

- Store the position in the index!
- Document 1 (d1):“*any choose love*”
- Document 2 (d2):“*zebra any love any zebra*”

Vocabulary		
any	→	d1 (0)
believe		
choose	→	d1 (1)
love	→	d1 (2)
midnight		
starring		
zebra		

Instead: Positional Index

- Store the position in the index!
- Document 1 (d1):“*any choose love*”
- Document 2 (d2):“*zebra any love any zebra*”

Vocabulary				
any	→	d1 (0)	→	d2 (1, 3)
believe				
choose	→	d1 (1)		
love	→	d1 (2)	→	d2 (2)
midnight				
starring				
zebra	→	d2 (0, 4)		

Positional Index: Querying

- Query 1: "*any love*" (phrase)

Vocabulary				
any	→	d1 (0)	→	d2 (1, 3)
love	→	d1 (2)	→	d2 (2)

- Constraint 1: "*any*" and "*love*" should appear in the same document.
- Constraint 2: In this document, position("*love*") = position("*any*") + 1

- Query 2: "*love any zebra*" (phrase)

Vocabulary				
any	→	d1 (0)	→	d2 (1, 3)
love	→	d1 (2)	→	d2 (2)
zebra	→	d2 (0, 4)		

Proximity Queries

- "*taylor NEAR:2 swift*", "*670 NEAR:5 solution*"
- "*NEAR:k*" (or "*/k*"): within *k* words of (on either side)
- Why might we like to support proximity queries?
  - "*Taylor Alison Swift (born December 13, 1989) is an American singer-songwriter. Known for her autobiographical songwriting, artistic versatility, and cultural impact …*"

- Positional index still works!
- Query: "*zebra NEAR:2 love*"

Vocabulary				
love	→	d1 (2)	→	d2 (2)
zebra	→	d2 (0, 4)		

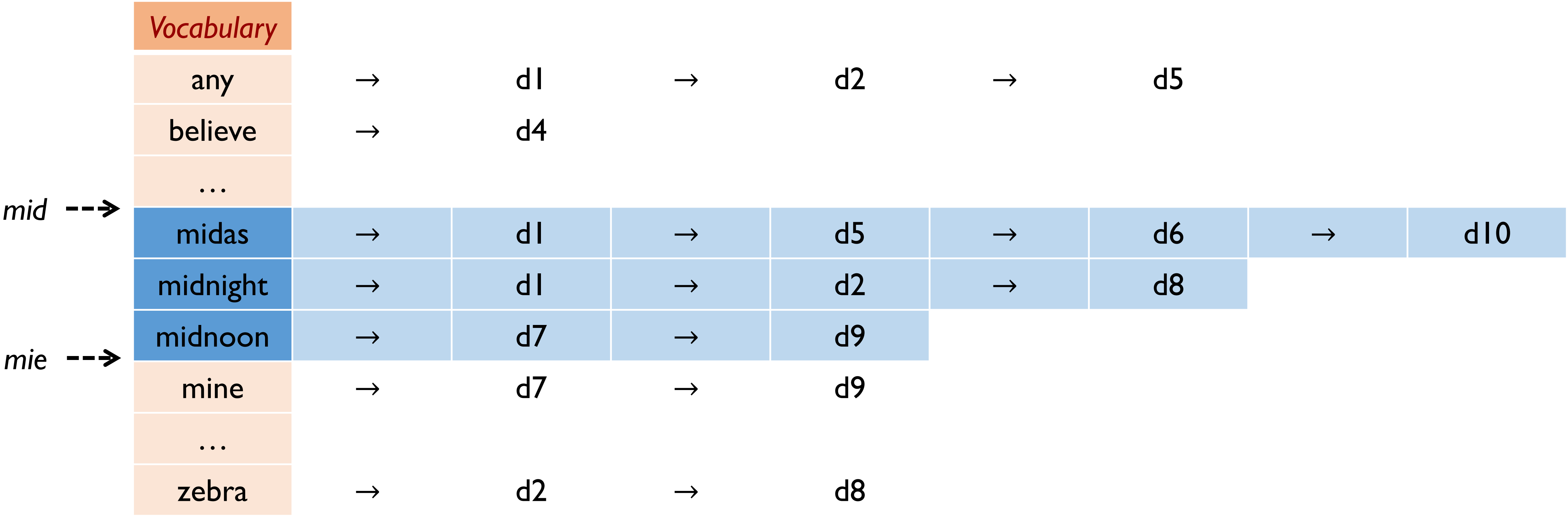
- Constraint 1: "*zebra*" and "*love*" should appear in the same document.
- Constraint 2: In this document, |position("*zebra*") - position("*love*")| ≤ 2.

Wildcard Queries

- “*mid**”
- Find all documents containing any word that begins with “*mid*”
  - “*midnight*”, “*midnights*”, “*midnoon*”, “*midas*”, …

Wildcard Queries: One Idea

- “*mid**”
- Suppose we have a binary search tree over our dictionary
  - Find all words in range: “*mid*” <= words < “*mie*”



Vocabulary								
any	→	d1	→	d2	→	d5		
believe	→	d4						
…								
midas	→	d1	→	d5	→	d6	→	d10
midnight	→	d1	→	d2	→	d8		
midnoon	→	d7	→	d9				
mine	→	d7	→	d9				
…								
zebra	→	d2	→	d8				

But we have harder cases

- What about wildcards at the beginning of a word?
  - "**ift*"
  - Find all documents containing any word that ends with "*ift*"
    - "*swift*", "*lift*", "*rift*", …

- What about wildcards at any point of a word?
  - "*ta*r*": "*taylor*", "*tater*", "*tailor*", …
  - "*m*ight*": "*midnight*", "*moonlight*", …

- Ideas?

Permuterm Index

- Use a special end-of-word token, e.g., “$”
  - “*taylor$*”
- Rotate every term:
  - “*taylor$*”, “*aylor$t*”, “*ylor$ta*”, “*lor$tay*”, “*or$tayl*”, “*r$taylo*”, “*$taylor*”

- If we have the query
  - “*ta*r*”
- Rotate the query, so that the “*” is at the end!
  - “*r$ta**”

- Look up in the rotated dictionary. “*taylor*”, “*tater*”, “*tailor*” should all be near each other:
  - “*r$taylor*”, “*r$tater*”, “*r$tailor*”

Summary

- To support phrase queries?
  - *“taylor swift”, “670 homework solution”*
  - Positional index
- To support proximity queries?
  - *“taylor NEAR:2 swift”, “670 NEAR:5 solution”*
  - Positional index
- To support wildcard queries?
  - *“tayl*”, “*ift”*
  - Permuterm Index

Extended Content
(will not appear in quizzes or the exam)

What should be in the index?

- What are the valid tokens to go in our dictionary?

- Input: a bunch of text
  - E.g., "*Welcome to class 670ers!*"
- Output: valid tokens
  - Approach 1: "*Welcome*", "*to*", "*class*", "*670ers!*"
  - Approach 2: "*welcome*", "*to*", "*class*", "*670ers*", "*!*"
  - Approach 3: "*wel*", "*elc*", "*lco*", "*com*", "*ome*", "*to*", "*cla*", …

- Critical step in determining what our users can search for.
  - If a token is not in our index, then our user cannot search for it!

Typical Issues in Tokenization

- Punctuation
  - “*pre-trained*” → “*pre-trained*” or even “*pretrained*” (better than “*pre*”, “*trained*”)
  - “*U.S.A.*” → “*U.S.A.*” or even “*USA*” (better than “*U*”, “*S*”, “*A*”)
  - “*C.A.T.*” → “*C.A.T.*” (better than “*cat*”)
  - “*A&M*” → “*A&M*” (better than “*AM*” or “*A*”, “*M*”)
- Case
  - “*PageRank*”, “*Pagerank*”, “*PAGERANK*” → “*pagerank*”
  - “*Apple*”, “*Windows*” → ? (depending on the context)
- Domain/Task
  - “*F = ma*” → “*F = ma*” (better than “*F*”, “=”, “*ma*”)
  - “*2%-4%*” → “*2%*”, “-”, “*4%*”
  - “*CaO + CO2 = CaCO3*” → ? (depending on your task: retrieving the reactants vs. retrieving the equation)

Stemming

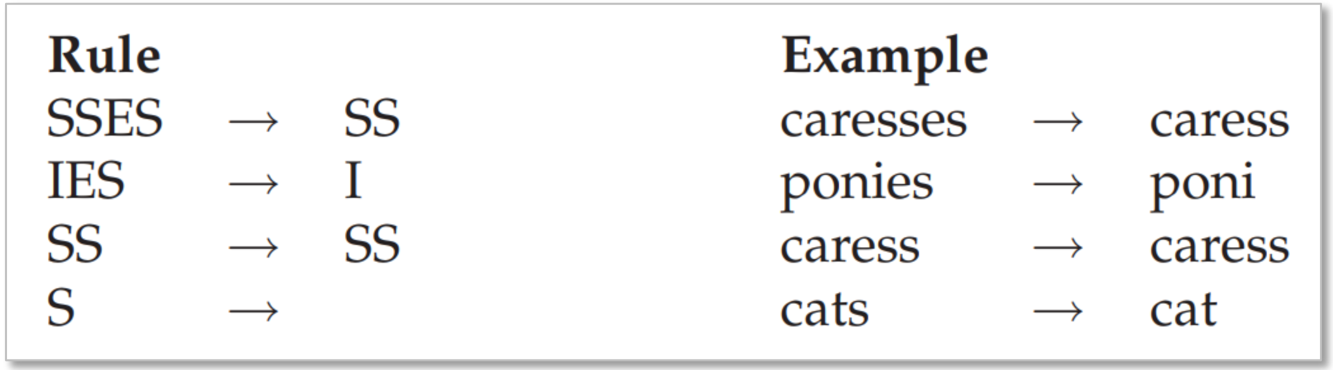
- The same word can be used in different forms.
    - "*organize*", "*organized*", "*organizes*", "*organizing*"
- There are families of derivationally related words with similar meanings.
    - "*democracy*", "*democratic*", "*democratization*"
- When you search one of these words, you may also want documents containing other words in the set.

- Stemming: Reducing inflectional forms and sometimes derivationally related forms of a word to a common base form

Porter’s Algorithm

- Stemming: Reducing inflectional forms and sometimes derivationally related forms of a word to a common base form

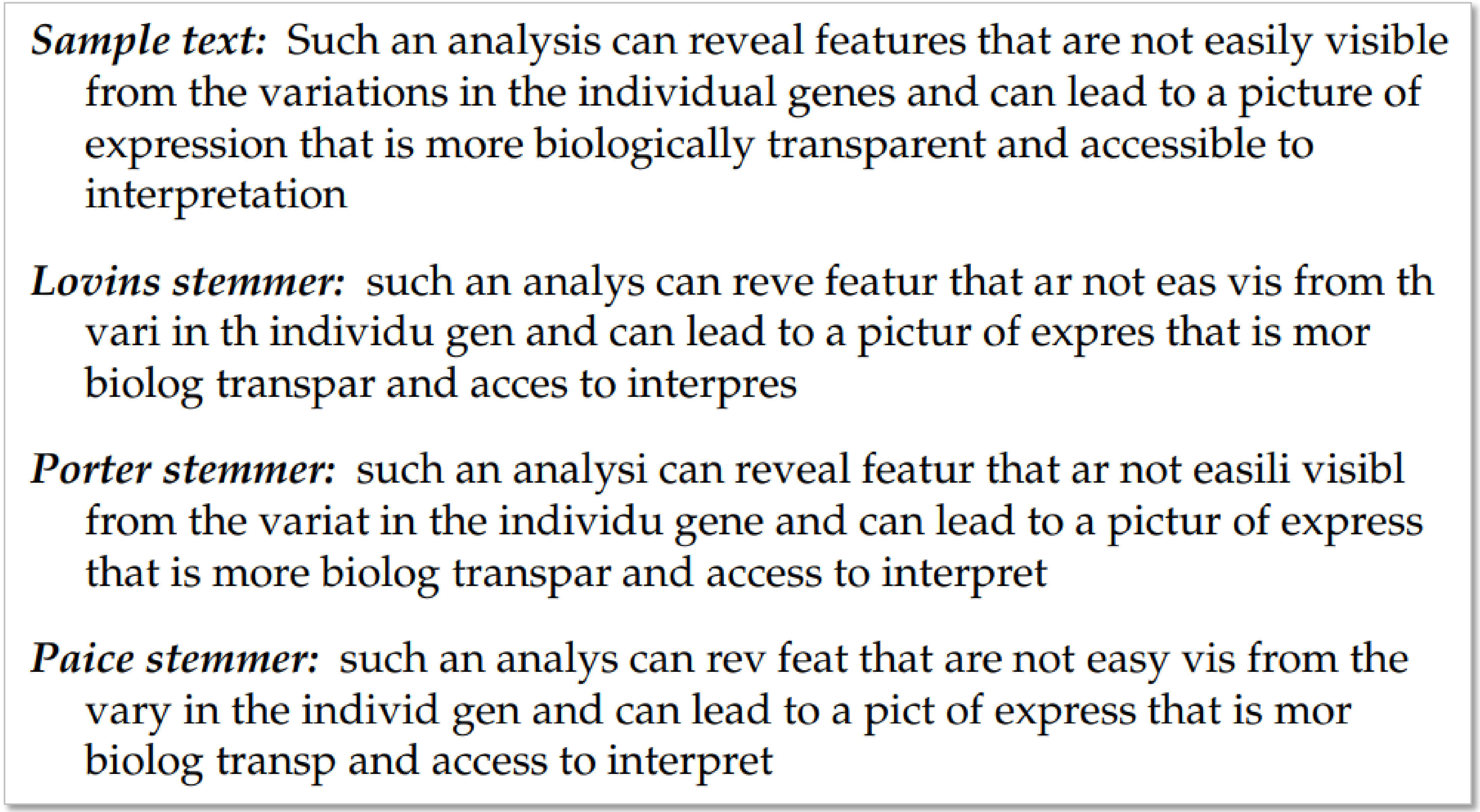
- Porter’s Algorithm: 5 phases of word reduction applied sequentially
    - Phase 1



Rule			Example		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

- For more information: http://www.tartarus.org/martin/PorterStemmer/

Examples of Stemming



Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

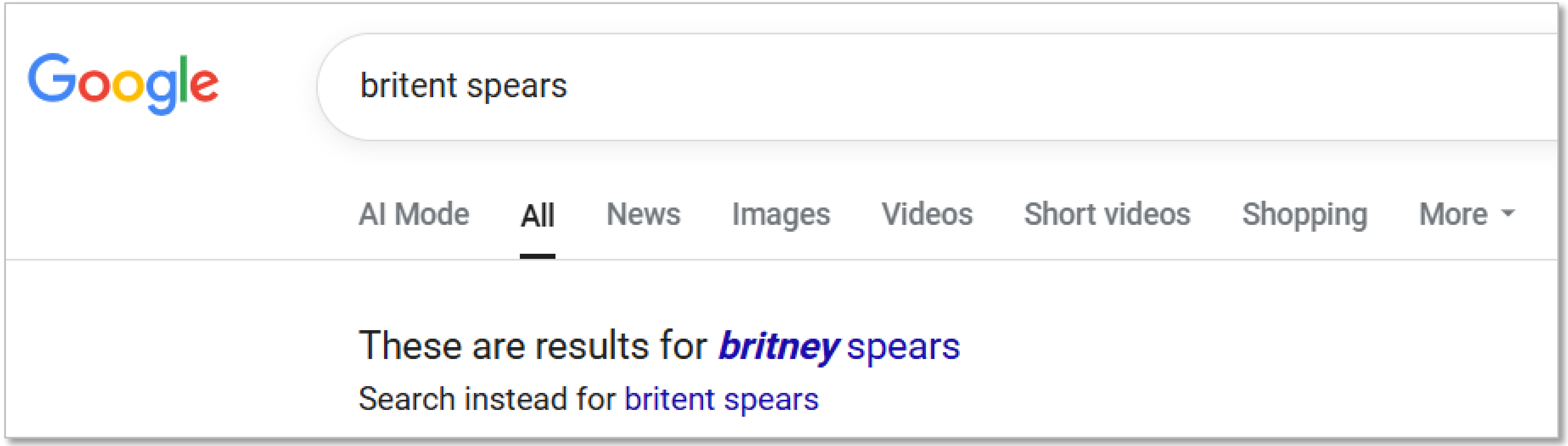
Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Spelling Errors

- Users make spelling mistakes all the time.
- How do we know that?
    - “*britent spears*” → “*britney spears*”



These are results for ***britney*** spears

Search instead for britent spears

Spelling Correction

- Based on edit distance
- Based on everyone's search logs

https://archive.google/jobs/britney.html

```
488941 britney spears      29 britent spears      9 brinttany spears   5 brney spears       3 britiy spears       2 brirreny spears
 40134 brittany spears     29 brittnany spears    9 britanay spears    5 broitney spears    3 britmeny spears     2 brirtany spears
 36315 brittney spears     29 britttany spears    9 britinany spears   5 brotny spears      3 britneeey spears    2 brirttany spears
 24342 britany spears      29 btiney spears       9 britn spears       5 bruteny spears     3 britnehy spears     2 brirttney spears
  7331 britny spears       26 birttney spears     9 britnew spears     5 btiyney spears     3 britnely spears     2 britain spears
  6633 briteny spears      26 breitney spears     9 britneyn spears    5 btrittney spears   3 britnesy spears     2 britane spears
  2696 britteny spears     26 brinity spears      9 britrney spears    5 gritney spears     3 britnetty spears    2 britaneny spears
  1807 briney spears       26 britenay spears     9 brtiny spears      5 spritney spears    3 britnex spears      2 britania spears
  1635 brittny spears      26 britneyt spears     9 brtittney spears   4 bittny spears      3 britneyxxx spears   2 britann spears
  1479 brintey spears      26 brittan spears      9 brtny spears       4 bnritney spears    3 britnity spears     2 britanna spears
  1479 britanny spears     26 brittne spears      9 brytny spears      4 brandy spears      3 britntey spears     2 britannie spears
  1338 britiny spears      26 btittany spears     9 rbitney spears     4 brbritney spears   3 britnyey spears     2 britannt spears
  1211 britnet spears      24 beitney spears      8 birtiny spears     4 breatiny spears    3 britterny spears    2 britannu spears
  1096 britiney spears     24 birteny spears      8 bithney spears     4 breetney spears    3 brittneey spears    2 britanyl spears
   991 britaney spears     24 brightney spears    8 brattany spears    4 bretiney spears    3 brittnney spears    2 britanyt spears
   991 britnay spears      24 brintiny spears     8 breitny spears     4 brfitney spears    3 brittnyey spears    2 briteeny spears
   811 brithney spears     24 britanty spears     8 breteny spears     4 briattany spears   3 brityen spears      2 britenany spears
   811 brtiney spears      24 britenny spears     8 brightny spears    4 brieteny spears    3 briytney spears     2 britenet spears
   664 birtney spears      24 britini spears      8 brintay spears     4 briety spears      3 brltney spears      2 briteniy spears
   664 brintney spears     24 britnwy spears      8 brinttey spears    4 briitny spears     3 broteny spears      2 britenys spears
   664 briteney spears     24 brittni spears      8 briotney spears    4 briittany spears   3 brtaney spears      2 britianey spears
```

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html