



CSCE 670 - Information Storage and Retrieval

Lecture 3: TF-IDF, Vector Space Model

Yu Zhang

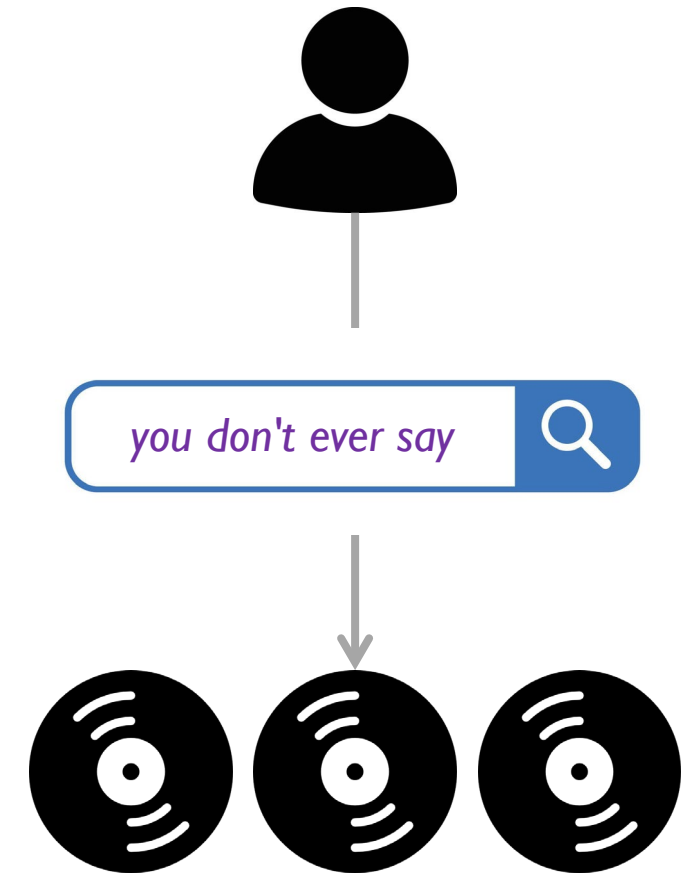
yuzhang@tamu.edu

September 2, 2025

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

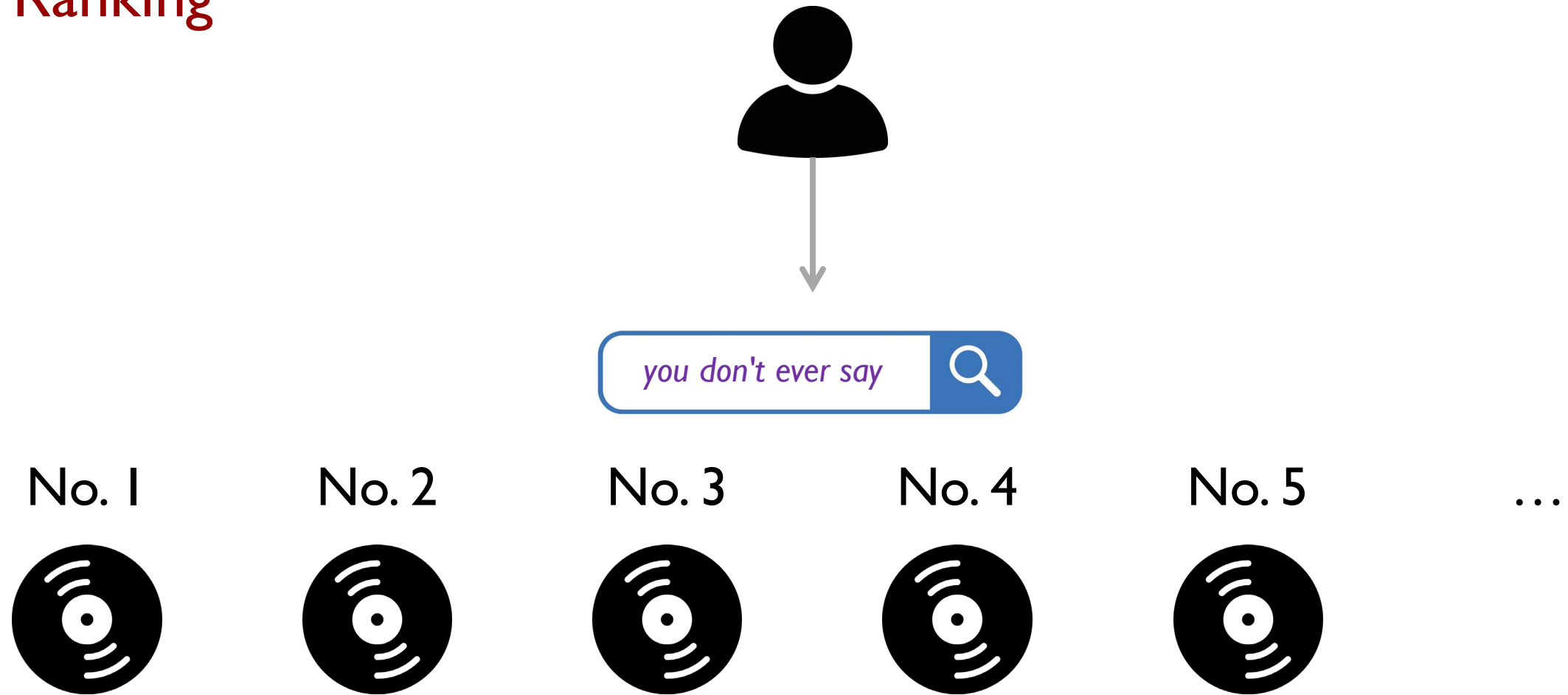
Recap: Boolean Retrieval

- Our capabilities so far:
 - Boolean keyword queries (AND, OR, NOT)
 - Inverted index
 - Phrase queries (“x y”)
 - Positional index
 - Proximity queries (“x NEAR:3 y”)
 - Positional index
 - Wildcard queries (“x*”)
 - Permuterm Index



We return a set of matching albums. (100s or 1000s of matches in some cases!)

Ranking



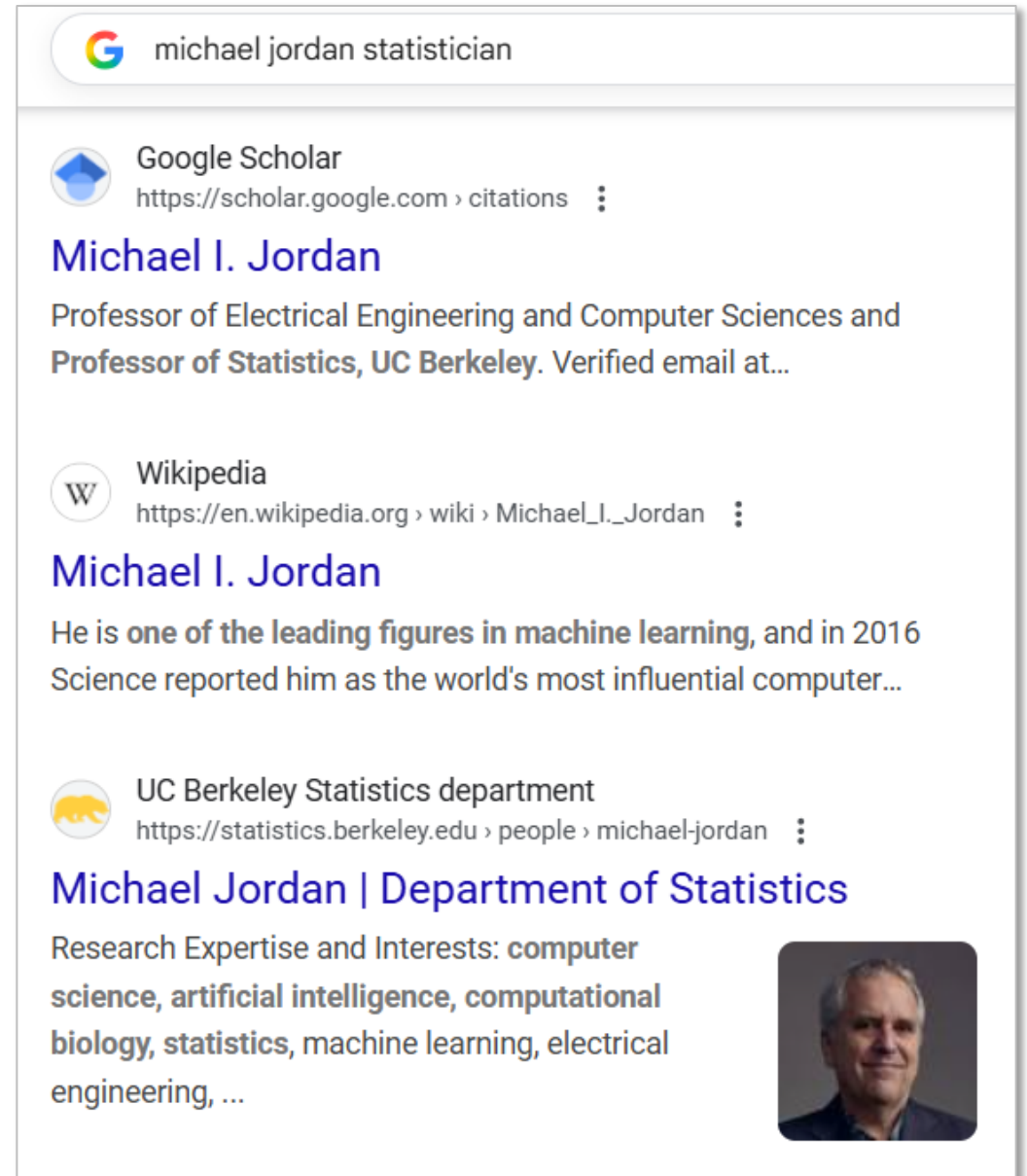
We return a **ranked** list of albums.

Our Plan: Ranking

- Why is ranking important?
- What factors impact ranking?
- Two foundational text-based approaches
 - TF-IDF
 - BM25
- Two foundational link-based approaches
 - PageRank
 - HITS
- Machine-learned ranking (“learning to rank”)

Why is ranking important?

- **User Study:** eye-tracking and relevance
- **Scenario:** Participants were asked to answer 10 questions using Google.
 - E.g., “*Find the homepage of Michael Jordan, the statistician.*”
- **Eye-Tracking:**
 - Record the sequence of eye movements
 - Analyze how users scan the results page of Google



Google search results for "michael jordan statistician".

Google Scholar
https://scholar.google.com › citations


Michael I. Jordan
Professor of Electrical Engineering and Computer Sciences and Professor of Statistics, UC Berkeley. Verified email at...

Wikipedia
https://en.wikipedia.org › wiki › Michael_I._Jordan

Michael I. Jordan
He is **one of the leading figures in machine learning**, and in 2016 Science reported him as the world's most influential computer...

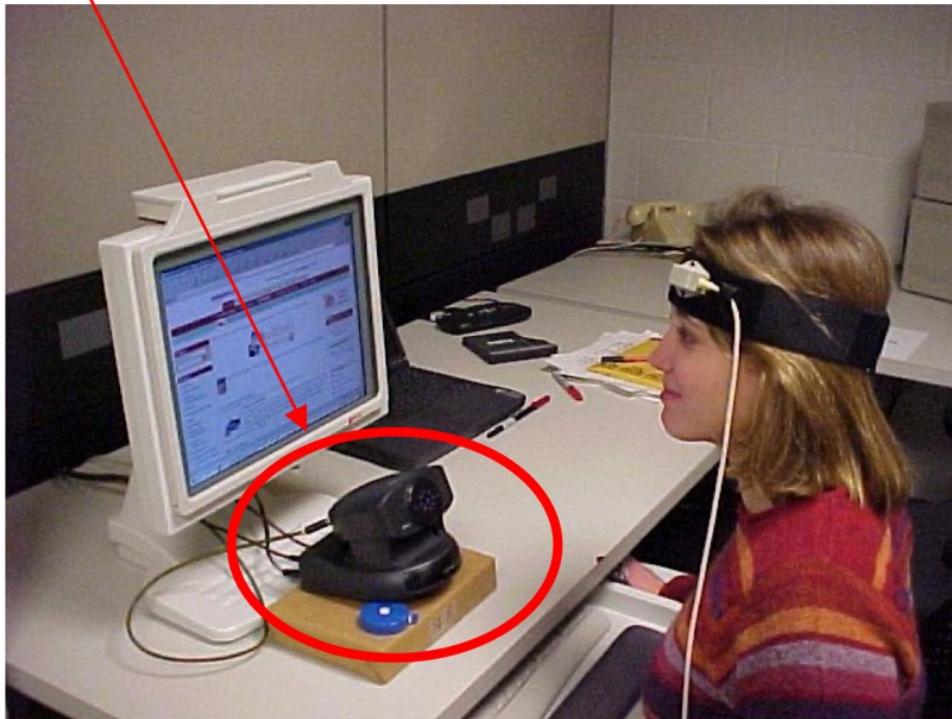
UC Berkeley Statistics department
https://statistics.berkeley.edu › people › michael-jordan

Michael Jordan | Department of Statistics
Research Expertise and Interests: **computer science, artificial intelligence, computational biology, statistics**, machine learning, electrical engineering, ...



What is Eye-Tracking?

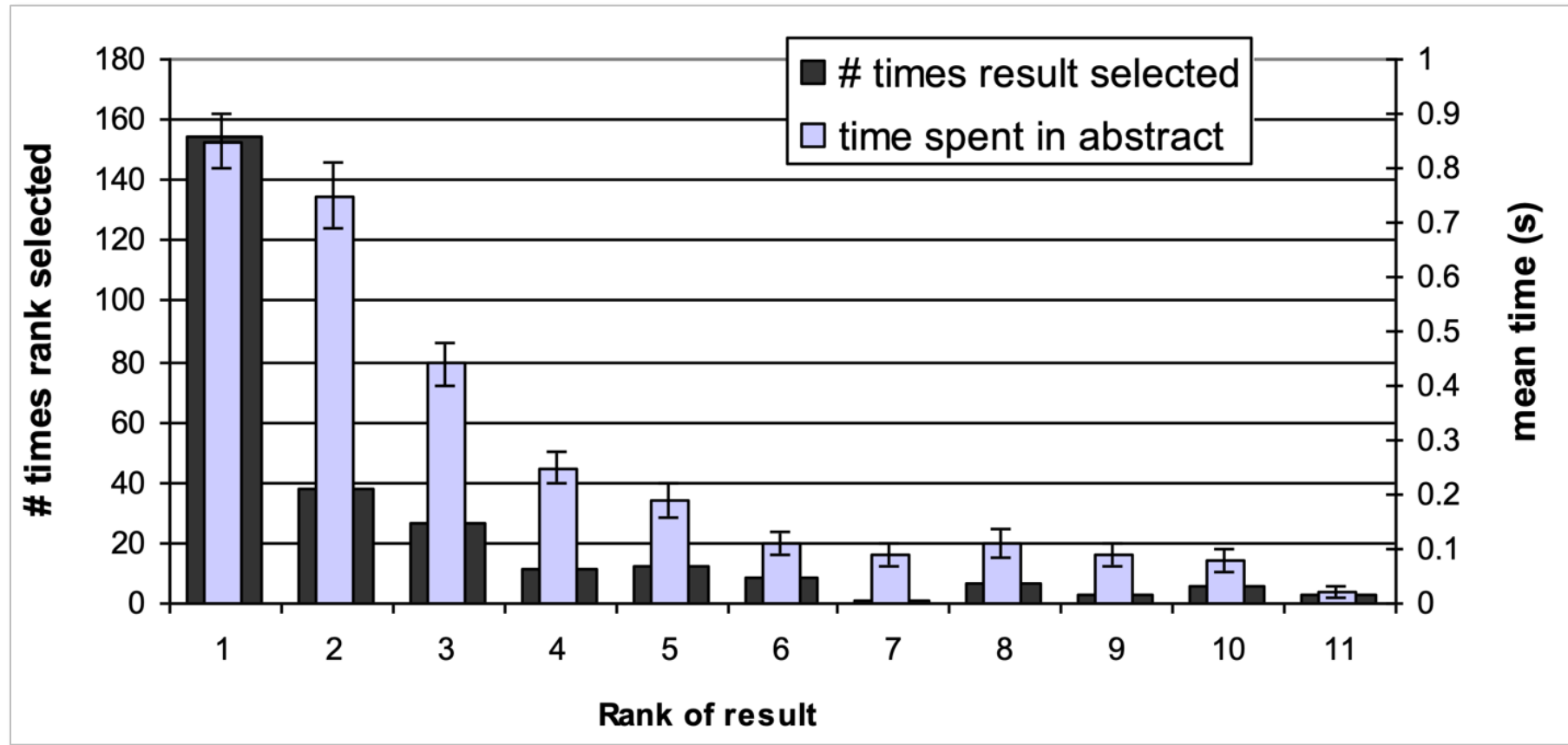
Eye tracking device



- Device to detect and record where and what people look at
 - **Fixations**: ~200-300ms, information is acquired
 - **Saccades**: extremely rapid movements between fixations

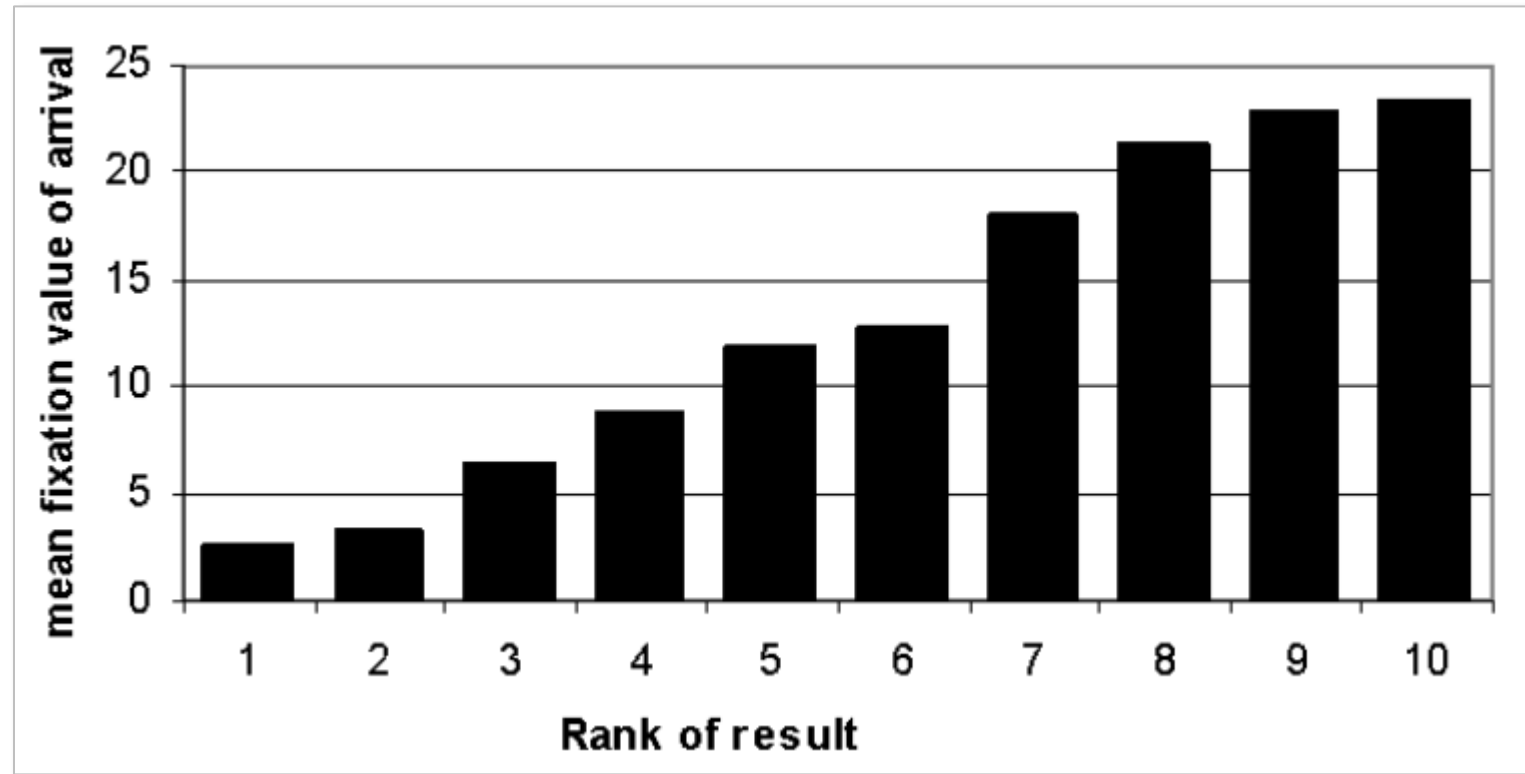


Viewing vs. Clicking



- Users view Documents 1 and 2 more thoroughly / often.
- Users click most frequently on Document 1.

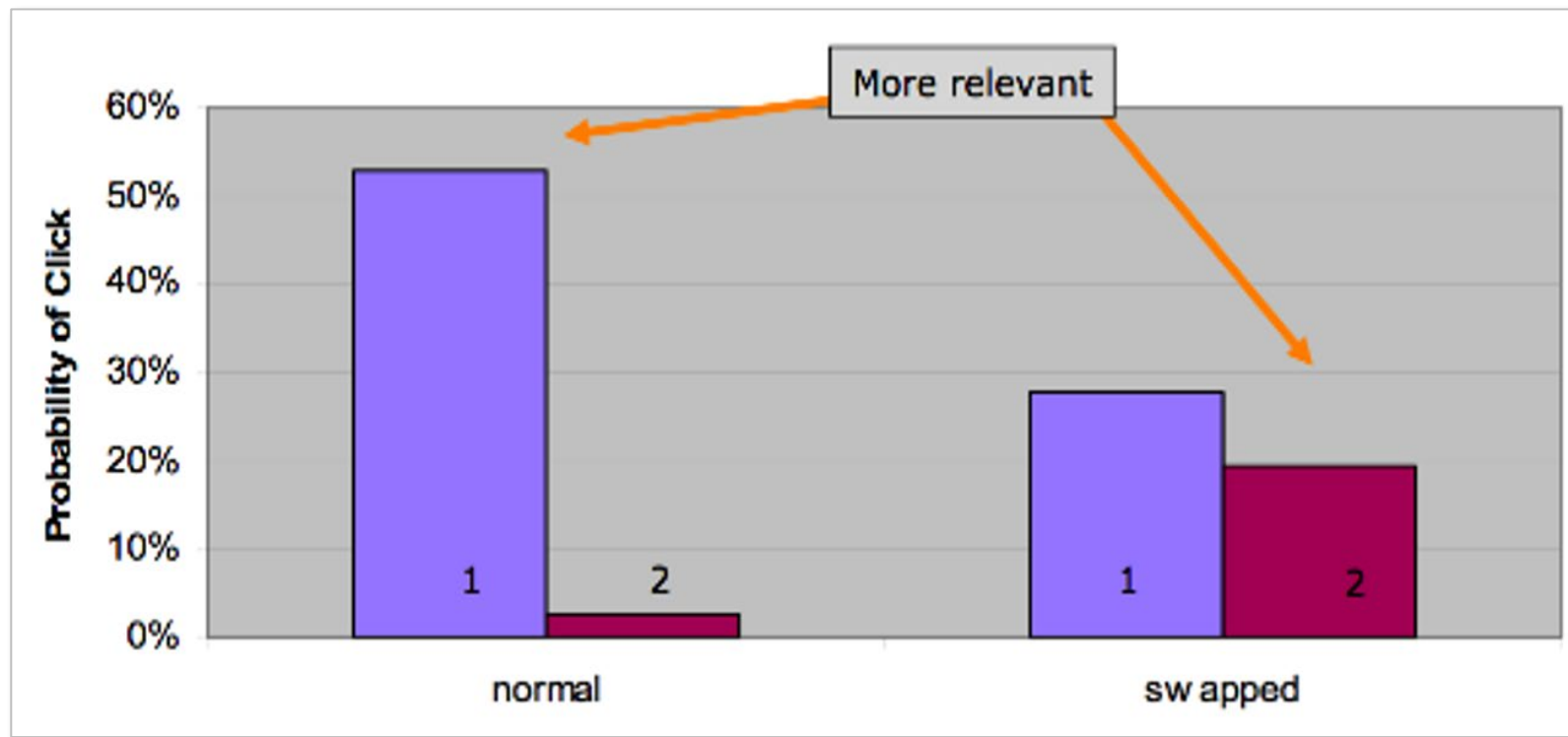
In which order are the results viewed?



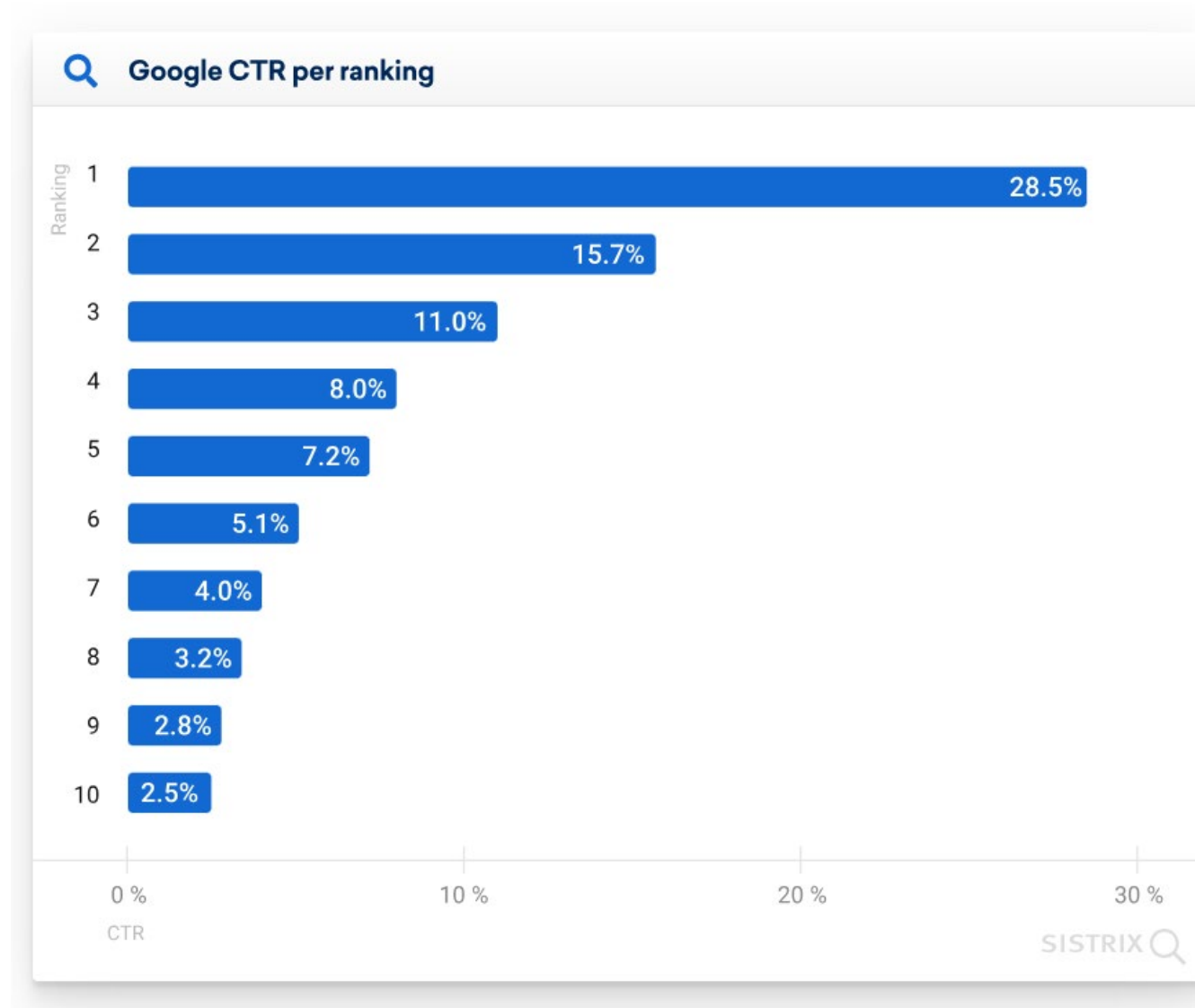
- Users tend to read the results in order.

Presentation Bias

- The top two results returned by Google were switched in order and then presented to the participants.
- Order of presentation influences where users look AND where they click.



Click-Through Rate (CTR) Per Ranking



<https://www.sistrix.com/blog/why-almost-everything-you-knew-about-google-ctr-is-no-longer-valid/>

- Accurate ranking increases the likelihood that the documents users click on are actually relevant.

Scoring as the Basis of Ranked Retrieval

- Let's try to build a scoring function:

$$\text{Score}(q, d) \in \mathbb{R}$$

to score every document d for a particular query q .

- Our hope is to design a scoring function so that the “best” documents (the most relevant, the best at satisfying the user) are scored highest.

What factors impact scoring?

- Query: “*information retrieval*”
- Document 1: “*Information retrieval is a core area of computer science. Modern information retrieval systems use ranking algorithms to improve search results. Deep learning has also been applied to information retrieval tasks. Classic information retrieval models include TF-IDF and BM25. Evaluation in information retrieval typically involves metrics like MAP and NDCG.*”
- Document 2: “*Natural language processing (NLP) has many applications, such as text classification, machine translation, and information retrieval. These tasks often require a deep understanding of both syntax and semantics. Recent advancements in large language models have significantly improved the performance of NLP systems across many benchmarks.*”
- Which document should be ranked higher? Why?

Term Frequency (TF)

- Score each term t in a document d by the number of times it occurs in the document, denoted as $tf_{t,d}$
 - **Intuition:** The more frequently a term appears in a document, the more important it is considered within that document.
- Example
 - Document d : “zebra any love any zebra”
 - $tf_{\text{love},d} = 1$
 - $tf_{\text{zebra},d} = 2$
 - $tf_{\text{dream},d} = 0$

Variants of TF

Term frequency	
n (natural)	$tf_{t,d}$
l (logarithm)	$1 + \log(tf_{t,d})$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$

- Why do we need these variants?
- If our scoring function is based **solely** on TF
 - Any monotonic transformation of TF is equivalent to TF in terms of ranking.
- If TF is **only one component** of our scoring function, we may want to control its “weight” in the function.
 - Do we want the growth of TF's weight to be equal in the following two cases?
 - $tf_{t,d}$ increases from 1 to 2
 - $tf_{t,d}$ increases from 100 to 101
 - “diminishing marginal gain”

Hans Peter Luhn (1896-1964)

- TF (1957)
- Researcher at IBM
- Invented the KWIC (Key Word In Context) indexing system, widely used in early digital libraries
- (Proposed an algorithm to checksum your credit card numbers!)



What factors impact scoring?

- Query: “mitochondria cell”
- Document 1: “The cell structure of an organism varies depending on the type of cell. In multicellular organisms, each cell has a specific function. Cell division plays an important role in growth and repair.”
 - $TF(\text{“cell”}, \text{Document 1}) = 4$; $TF(\text{“mitochondria”}, \text{Document 1}) = 0$
- Document 2: “Mitochondria are known as the powerhouse of the cell. They play a critical role in ATP production and cellular respiration. Damage to mitochondria can lead to metabolic disorders.”
 - $TF(\text{“cell”}, \text{Document 2}) = 1$; $TF(\text{“mitochondria”}, \text{Document 2}) = 2$
- Which document should be ranked higher? Why?

Inverse Document Frequency (IDF)

- Score each term in a document by how rare it is across all documents
 - **Intuition:** The rarer a term is across a collection, the more valuable it is.
 - “*mitochondria*” is more valuable than “*cell*”.
 - If a document matches the term “*mitochondria*” once, it should receive a greater reward than matching the term “*cell*” once.

$$\text{idf}_t = \log \frac{|\mathcal{D}|}{|d \in \mathcal{D}: t \in d|}$$

- $|\mathcal{D}|$: number of documents in the collection
- $|d \in \mathcal{D}: t \in d|$: number of documents in the collection that contain t

Inverse Document Frequency (IDF)

$$\text{idf}_t = \log \frac{|\mathcal{D}|}{|d \in \mathcal{D}: t \in d|}$$

- Example (We use 10 as the base of the logarithm)
 - Suppose you want to rank 100,000 documents.
 - “*cell*” appears in 1,000 of them
 - $\text{idf}_{\text{cell}} = \log \frac{100000}{1000} = \log 100 = 2$
 - “*mitochondria*” appears in 10 of them
 - $\text{idf}_{\text{mitochondria}} = \log \frac{100000}{10} = \log 10000 = 4$
- Can you roughly estimate the IDF value of “*a*”?

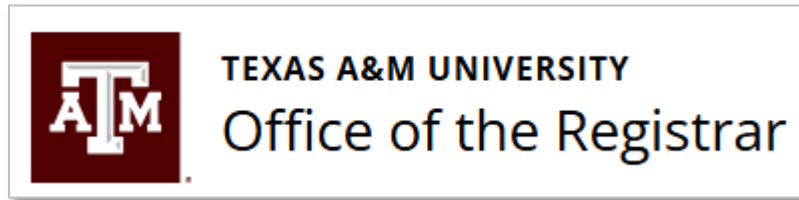
Karen Spärck Jones (1935-2007)

- IDF (1972)
- Professor at the University of Cambridge
- Advocated for the use of statistical methods in linguistics and IR, laying the groundwork for modern search engines
- Gerard Salton Award (1988)
 - Awarded by SIGIR, with one recipient every three years



What factors impact scoring?

- Query: “TAMU 2025 Fall Break”
- Document 1: <https://registrar.tamu.edu/academic-calendar/fall-2025>



- Document 2: A social media post written by an account with 10 followers mentioning the time of TAMU 2025 Fall Break
- Which document should be ranked higher? Why?
- How can we know the “reputation” of a website?
 - Next week!

Let's first consider the textual factors (TF and IDF) together.

- Given a query q and a document d , we want to calculate $\text{Score}(q, d)$.
- The query q may have one or more terms. For each term t ,

$$\text{tfidf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

- The overall score is the sum of the TF-IDF scores of all terms.

$$\text{Score}(q, d) = \sum_{t \in q} \text{tfidf}_{t,d}$$

- How should we interpret this formula?
 - A weighted sum of TF
 - The weight is IDF

Example

- Suppose you want to rank 100,000 documents. “*cell*” appears in 1,000 of them; “*mitochondria*” appears in 10 of them. (We use 10 as the base of the logarithm in IDF.)
- Query: “*mitochondria cell*”
- Document 1: “The *cell* structure of an organism varies depending on the type of *cell*. In multicellular organisms, each *cell* has a specific function. *Cell* division plays an important role in growth and repair.”
 - $\text{TF}(\text{“cell”}, \text{Document 1}) = 4; \text{IDF}(\text{“cell”}) = 2$
 - $\text{TF-IDF}(\text{“cell”}, \text{Document 1}) = 4 \times 2 = 8$
 - $\text{TF}(\text{“mitochondria”}, \text{Document 1}) = 0; \text{IDF}(\text{“mitochondria”}) = 4$
 - $\text{TF-IDF}(\text{“mitochondria”}, \text{Document 1}) = 0 \times 4 = 0$
 - $\text{TF-IDF}(\text{Query}, \text{Document 1}) = 8 + 0 = 8$

Example

- Suppose you want to rank 100,000 documents. “*cell*” appears in 1,000 of them; “*mitochondria*” appears in 10 of them. (We use 10 as the base of the logarithm in IDF.)
- Query: “*mitochondria cell*”
- Document 2: “*Mitochondria* are known as the powerhouse of the *cell*. They play a critical role in ATP production and cellular respiration. Damage to *mitochondria* can lead to metabolic disorders.”
 - $TF(\text{“cell”}, \text{Document 2}) = 1; IDF(\text{“cell”}) = 2$
 - $TF-IDF(\text{“cell”}, \text{Document 2}) = 1 \times 2 = 2$
 - $TF(\text{“mitochondria”}, \text{Document 2}) = 2; IDF(\text{“mitochondria”}) = 4$
 - $TF-IDF(\text{“mitochondria”}, \text{Document 1}) = 2 \times 4 = 8$
 - $TF-IDF(\text{Query}, \text{Document 1}) = 2 + 8 = 10$

Questions?

Vector Space Model (VSM)

Information Retrieval C.A. Montgomery
and Language Processing Editor

A Vector Space Model for Automatic Indexing

G. Salton, A. Wong
and C. S. Yang
Cornell University

In a document retrieval, or other pattern matching environment where stored entities (documents) are compared with each other or with incoming patterns (search requests), it appears that the best indexing (property) space is one where each entity lies as far away from the others as possible; in these circumstances the value of an indexing system may be expressible as a function of the density of the object space; in particular, retrieval performance may correlate inversely with space density. An approach based on space density computations is used to choose an optimum indexing vocabulary for a collection of documents. Typical evaluation results are shown, demonstrating the usefulness of the model.

- This paper is from 1975, but lots of earlier work in the 60s and early 70s
- P. Switzer, *Vector Images in Document Retrieval*, 1963.
- G. Salton, *Automatic Information Organization and Retrieval*, 1968.

Gerard Salton (1927-1995)

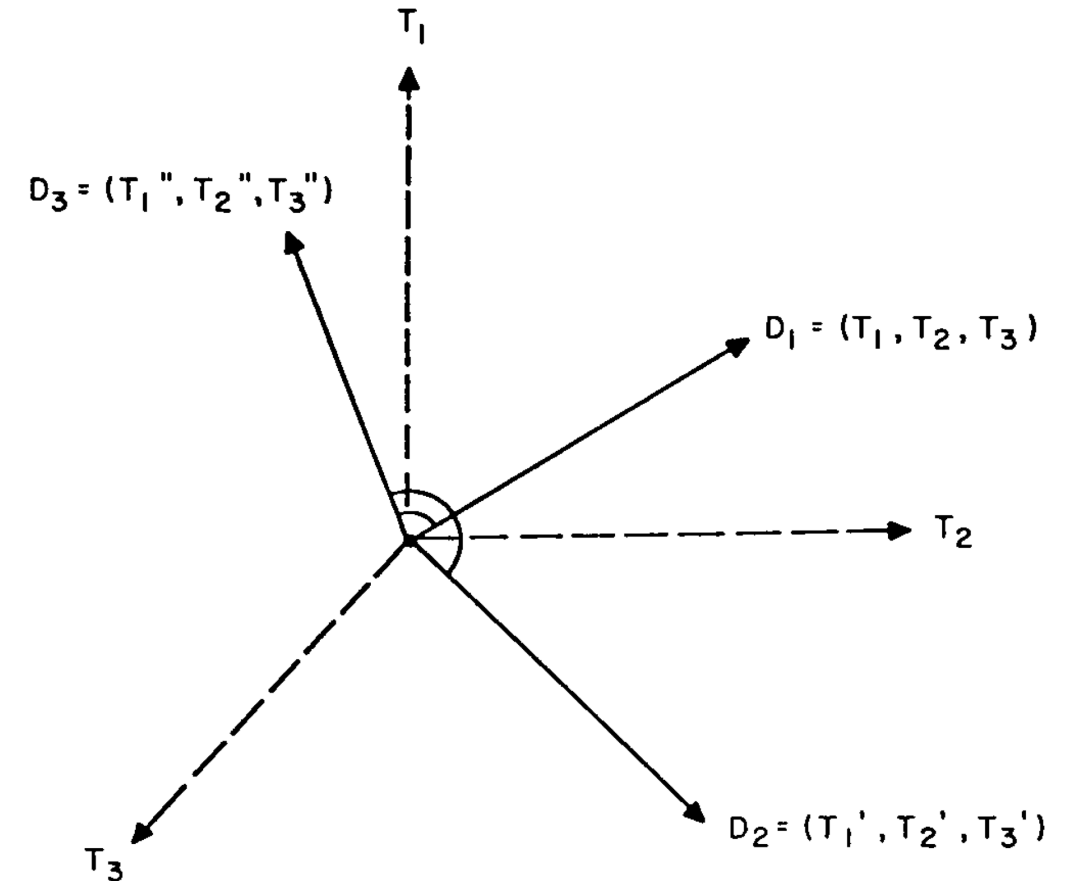
- Professor at Cornell University
- “*the father of Information Retrieval*”
- Laid the foundation for modern search engines and ranking algorithms
- Invented the Vector Space Model for representing documents and queries
- ACM SIGIR Award (1983)
 - Posthumously named the Gerard Salton Award



General Idea of VSM

- Represent documents and queries as vectors in a high-dimensional space
- A document is more **relevant** to a query if they are more “**similar**” in the vector space.
 - We will define “**similar**” later.
- What are the axes?
- How about each dimension representing **a word in the vocabulary**?

Fig. 1. Vector representation of document space.



VSM: Example 1

- If the word appears in a query/document, the corresponding entry is 1.
- Otherwise, the corresponding entry is 0.
- **Query** q : “any AND zebra”
- **Document** d : “zebra any love any zebra”

Vocabulary	Vector(q)	Vector(d)
any	1	1
believe	0	0
choose	0	0
...
love	0	1
starring	0	0
zebra	1	1

- What is the inner product of $\text{Vector}(q)$ and $\text{Vector}(d)$?
 - $\text{Vector}(q) \cdot \text{Vector}(d) = 2$
- In the setting of **Boolean retrieval**, d is relevant to this q **if and only if** $\text{Vector}(q) \cdot \text{Vector}(d) = 2$.

VSM: Example 1

- If the word appears in a query/document, the corresponding entry is 1.
- Otherwise, the corresponding entry is 0.
- **Query** q : “ t_1 AND t_2 AND ... AND t_N ” (There are N words that must appear in d .)
- In the setting of **Boolean retrieval**, d is relevant to this q **if and only if**
$$\text{Vector}(q) \cdot \text{Vector}(d) = N$$

VSM: Example 2

- Given a query q , the corresponding entry of word t is $tf_{t,q}$.
- Given a document d , the corresponding entry of word t is $tf_{t,d} \times idf_t$.
- Query q : “any any zebra”
- Document d : “zebra any love any zebra”

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

Vector(d)
$tf_{any,d} \times idf_{any}$
0
0
...
$tf_{love,d} \times idf_{love}$
0
$tf_{zebra,d} \times idf_{zebra}$

- What is the inner product of $\text{Vector}(q)$ and $\text{Vector}(d)$?

VSM: Example 2

- Query q : “any any zebra”
- Document d : “zebra any love any zebra”
- What is the inner product of $\text{Vector}(q)$ and $\text{Vector}(d)$?

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

Vector(d)
$\text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$\text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$\text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

$$\begin{aligned}
 \text{Vector}(q) \cdot \text{Vector}(d) &= 2 \times \text{tf}_{\text{any},d} \times \text{idf}_{\text{any}} + 1 \times \text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}} \\
 &= \text{tf}_{\text{any},d} \times \text{idf}_{\text{any}} + \text{tf}_{\text{any},d} \times \text{idf}_{\text{any}} + \text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}} \\
 &= \sum_{t \in q} (\text{tf}_{t,d} \times \text{idf}_t) \\
 &= \text{TF-IDF}(q, d)
 \end{aligned}$$

- TF-IDF is a special case of VSM.

Problems with Inner Product?

- Query q : “any any zebra”
- Document d_1 : “zebra any love any zebra”
- Document d_2 : “zebra any love any zebra zebra any love any zebra”

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

Vector(d_1)
$\text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$\text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$\text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

Vector(d_2)
$2 \times \text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$2 \times \text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$2 \times \text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

Problems with Inner Product?

- Query q : “any any zebra”
- Document d_1 : “zebra any love any zebra”
- Document d_{100} : “zebra any love any zebra zebra any love any zebra ... (repeat 100 times)”

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

Vector(d_1)
$\text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$\text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$\text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

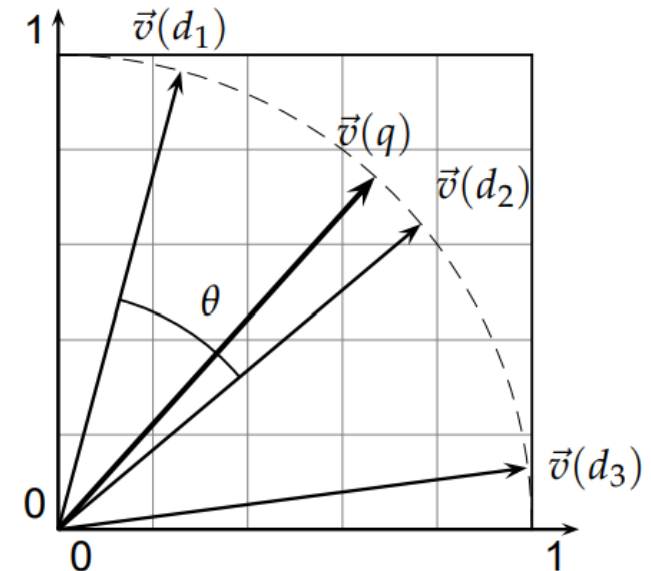
Vector(d_{100})
$100 \times \text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$100 \times \text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$100 \times \text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

Problems with Inner Product?

- $\text{Vector}(d_2) = 2 \text{Vector}(d_1)$
- $\text{Vector}(q) \cdot \text{Vector}(d_2) = 2 \text{Vector}(q) \cdot \text{Vector}(d_1)$
- $\text{Vector}(d_{100}) = 100 \text{Vector}(d_1)$
- $\text{Vector}(q) \cdot \text{Vector}(d_{100}) = 100 \text{Vector}(q) \cdot \text{Vector}(d_1)$
- ...
- We can make the inner product of q and d in the vector space as large as possible by just making d longer!
- How to take the document length factor into account?

Cosine Similarity

- $\mathbf{x} = [x_1, x_2, \dots, x_N]$
- $\mathbf{y} = [y_1, y_2, \dots, y_N]$
- $$\cos(\mathbf{x}, \mathbf{y}) = \frac{x_1y_1 + x_2y_2 + \dots + x_Ny_N}{\sqrt{x_1^2 + x_2^2 + \dots + x_N^2} \times \sqrt{y_1^2 + y_2^2 + \dots + y_N^2}} = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} = \left(\frac{\mathbf{x}}{||\mathbf{x}||} \right) \cdot \left(\frac{\mathbf{y}}{||\mathbf{y}||} \right)$$
- Equivalent to first **normalizing the vectors to unit length**, and then computing the dot product
 - \mathbf{x} , $2\mathbf{x}$, and $100\mathbf{x}$ will be the same vector after length normalization.
- The larger the cosine similarity, the smaller the **angle** between the two unit vectors (i.e., the more “similar” they are).



Example

- Query q : “any any zebra”
- Document d_1 : “zebra any love any zebra”
- Let's assume $\text{idf}_{\text{any}} = \text{idf}_{\text{love}} = 2$ and $\text{idf}_{\text{zebra}} = 4$.

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

Vector(d_1)
$\text{tf}_{\text{any},d} \times \text{idf}_{\text{any}}$
0
0
...
$\text{tf}_{\text{love},d} \times \text{idf}_{\text{love}}$
0
$\text{tf}_{\text{zebra},d} \times \text{idf}_{\text{zebra}}$

Vector(d_1)
$2 \times 2 = 4$
0
0
...
$1 \times 2 = 2$
0
$2 \times 4 = 8$

Example

- $||\text{Vector}(q)|| = \sqrt{2^2 + 1^2} = \sqrt{5}$
- $||\text{Vector}(d_1)|| = \sqrt{4^2 + 2^2 + 8^2} = \sqrt{84}$
- $\cos(\text{Vector}(q), \text{Vector}(d_1)) = \frac{2 \times 4 + 0 \times 2 + 1 \times 8}{\sqrt{5} \times \sqrt{84}} \approx 0.781$

Vocabulary
any
believe
choose
...
love
starring
zebra

Vector(q)
2
0
0
...
0
0
1

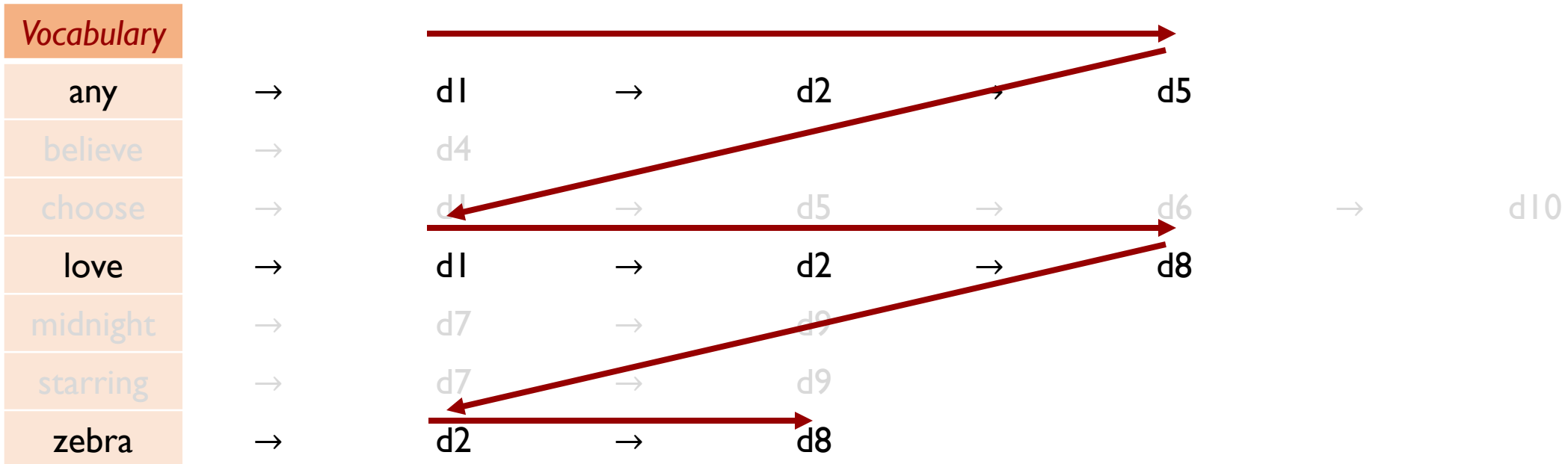
Vector(d_1)
4
0
0
...
2
0
8

What's the range of the cosine similarity in VSM?

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

How to calculate cosine with our index structure?

- **Term At A Time (TAAT):** Scores for all docs computed concurrently, one query term at a time
- E.g., Query: “any zebra love”



How to calculate cosine with our index structure?

- **Document At A Time (DAAT)**: Total score for each doc (including all query terms) computed before proceeding to the next
- E.g., Query: “*any zebra love*”

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					

How to calculate cosine with our index structure?

- **Document At A Time (DAAT):** Total score for each doc (including all query terms) computed before proceeding to the next
- E.g., Query: “any zebra love”

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			

How to calculate cosine with our index structure?

- **Document At A Time (DAAT):** Total score for each doc (including all query terms) computed before proceeding to the next
- E.g., Query: “any zebra love”

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					

How to calculate cosine with our index structure?

- **Document At A Time (DAAT):** Total score for each doc (including all query terms) computed before proceeding to the next
- E.g., Query: “any zebra love”

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					

How to calculate cosine with our index structure?

- **Document At A Time (DAAT):** Total score for each doc (including all query terms) computed before proceeding to the next
- E.g., Query: “any zebra love”

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

See MRS Chapter 7 for More Details

- One advantage of DAAT: can prune documents below a certain threshold
 - Exact vs. inexact top-k results
 - Champion lists
 - ...



Thank You!

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