## Some Reminders for a Seamless Online Class…

- Please turn on your video
- Mute yourself (press and hold spacebar when you'd like to talk)
- Don't do anything you wouldn't do in an in-person class
- I will occasionally check the chat for messages if you'd like to share there instead
- Please say your name before you speak

## Logistics

- Midterm and Programming Deadline Extended
    - Midterm report next
- Zoom can record! We'll share those videos (please don't share those broadly — just for private consumption)
- Any other questions?

# Midterm Project Report (now due 3/30)

- From the FAQs:
  - "What goes in the midterm report?"

The midterm report is a 4-page summary outlining the project goals, its relevance to the course, key milestones/timeline, proposed design and architecture, the name of your project partner, and preliminary work done on the project. The midterm report will be evaluated based on the project description and proposed approach, justification of design and architecture, understanding of related work in this space, the feasibility of the projected plan, as well as promising results shown in the preliminary findings.

# Recap

- Data-savviness is the future!
- "Classical" relational databases
    - Notion of a DBMS
    - The relational data model and algebra: bags and sets
    - SQL Queries, Modifications, DDL
    - Database Design
    - Views, constraints, triggers, and indexes
    - Query processing & optimization
    - Transactions
- Non-classical data systems
    - Semi-structured data and document stores
    - Next: unstructured data and search engines

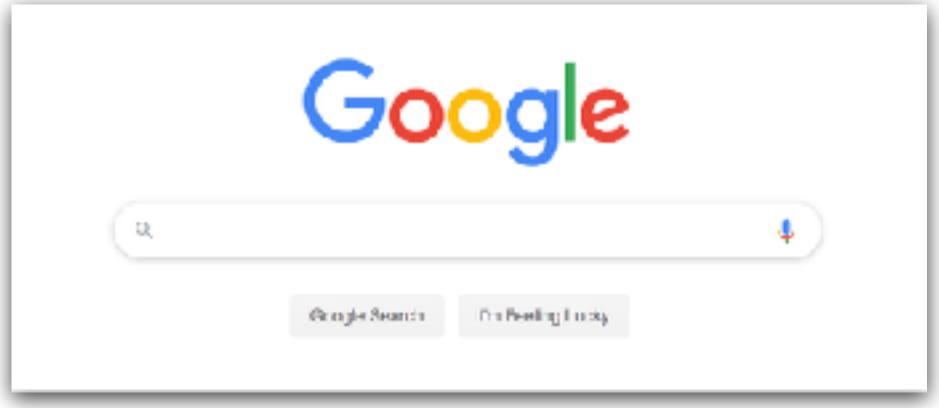
# Unstructured data

- Today, we will focus our attention on unstructured data that doesn't really have any "structure"
  - Unlike semistructured data, which does have some structure
  - It's a nested, flexible structure
- Unstructured data comes in many forms:
  - Text, images, video, audio
  - Our focus today is text — text is pretty general purpose:
    - Can capture "metadata" or "descriptions" for images and video
    - Can capture transcriptions for sound in video and audio too

## Querying Unstructured Data

- We've all queried unstructured data: search engines!
  - The field of designing better, more efficient search engines is called "information retrieval"
- We're going to get a bird's eye view of how to build a search engine …

# First: the interface



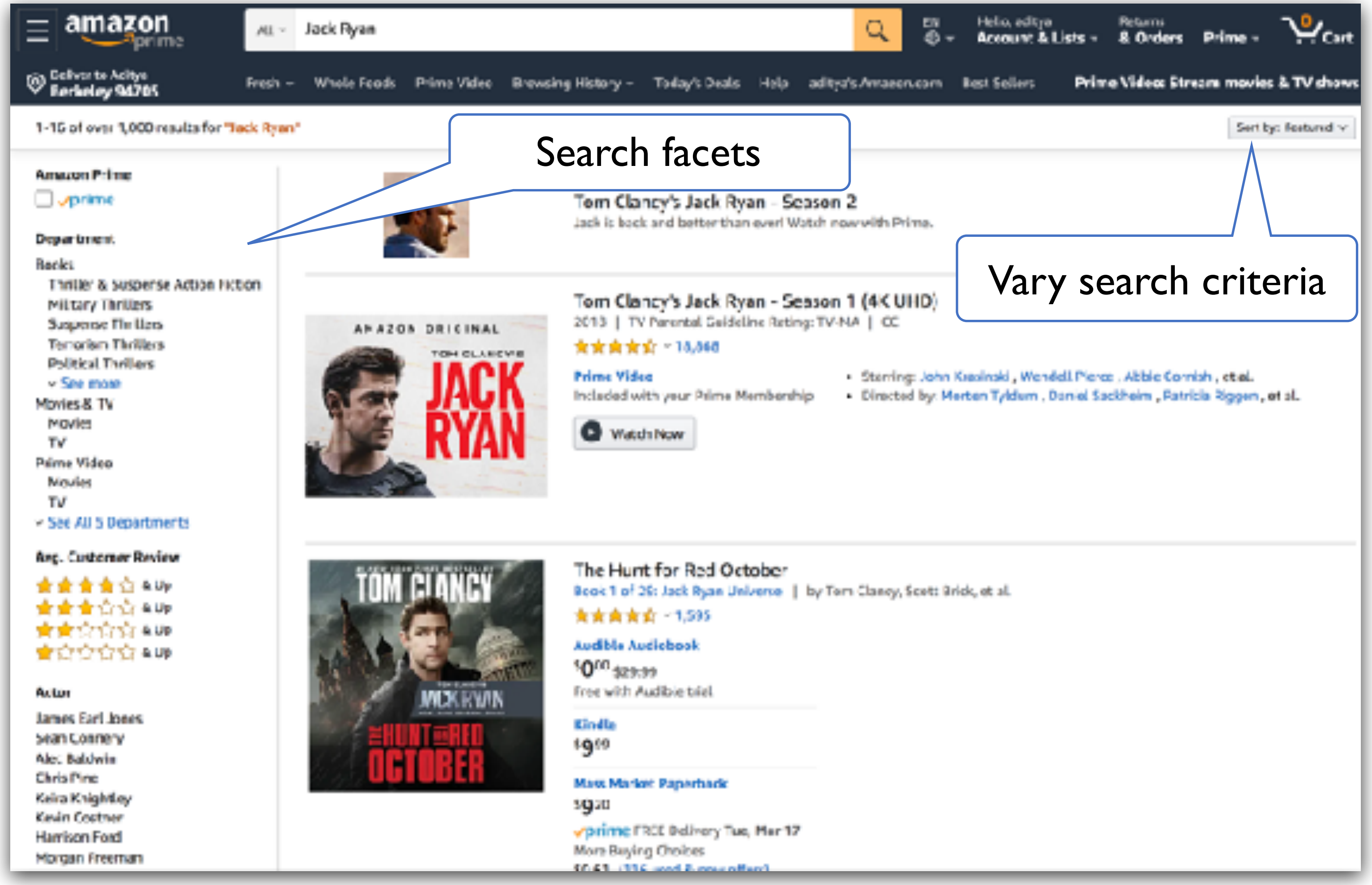
- A simple text (or keyword) search field
- Sequences of keywords

- Can be more advanced:
  - Can specify exact word or phrase using “”
    - “Jack Ryan"
  - Can specify one word or the other using OR
    - Jack OR Ryan
  - Can specify a word or phrase not appearing
    - -“Jack Ryan”
  - Can specify file type, region, language, domain, …

## Not Just Restricted to Internet Search

- Product search
  - Amazon, Ebay, Walmart, …
- Entertainment search
  - Netflix, Spotify, Youtube, …
- Social media search
  - Twitter, Facebook, …
- Email search

# Interfaces Can Vary: Refining User Intent via Knobs



amazon prime | All | Jack Ryan | EN | Hello, aditya Account & Lists | Returns & Orders | Prime | 0 Cart

Deliver to Aditya Berkeley 94705 | Fresh | Whole Foods | Prime Video | Browsing History | Today's Deals | Help | aditya's Amazon.com | Best Sellers | Prime Video: Stream movies & TV shows

1-16 of over 1,000 results for "Jack Ryan" | Sort by: Featured

**Amazon Prime**
prime

**Department**
Books
Thriller & Suspense Action Fiction
Military Thrillers
Suspense Thrillers
Terrorism Thrillers
Political Thrillers
See more
Movies & TV
Movies
TV
Prime Video
Movies
TV
See All 5 Departments

**Avg. Customer Review**
★★★★☆ & Up
★★★☆☆ & Up
★★☆☆☆ & Up
★☆☆☆☆ & Up

**Actor**
James Earl Jones
Sean Connery
Alec Baldwin
Chris Pine
Keira Knightley
Kevin Costner
Harrison Ford
Morgan Freeman

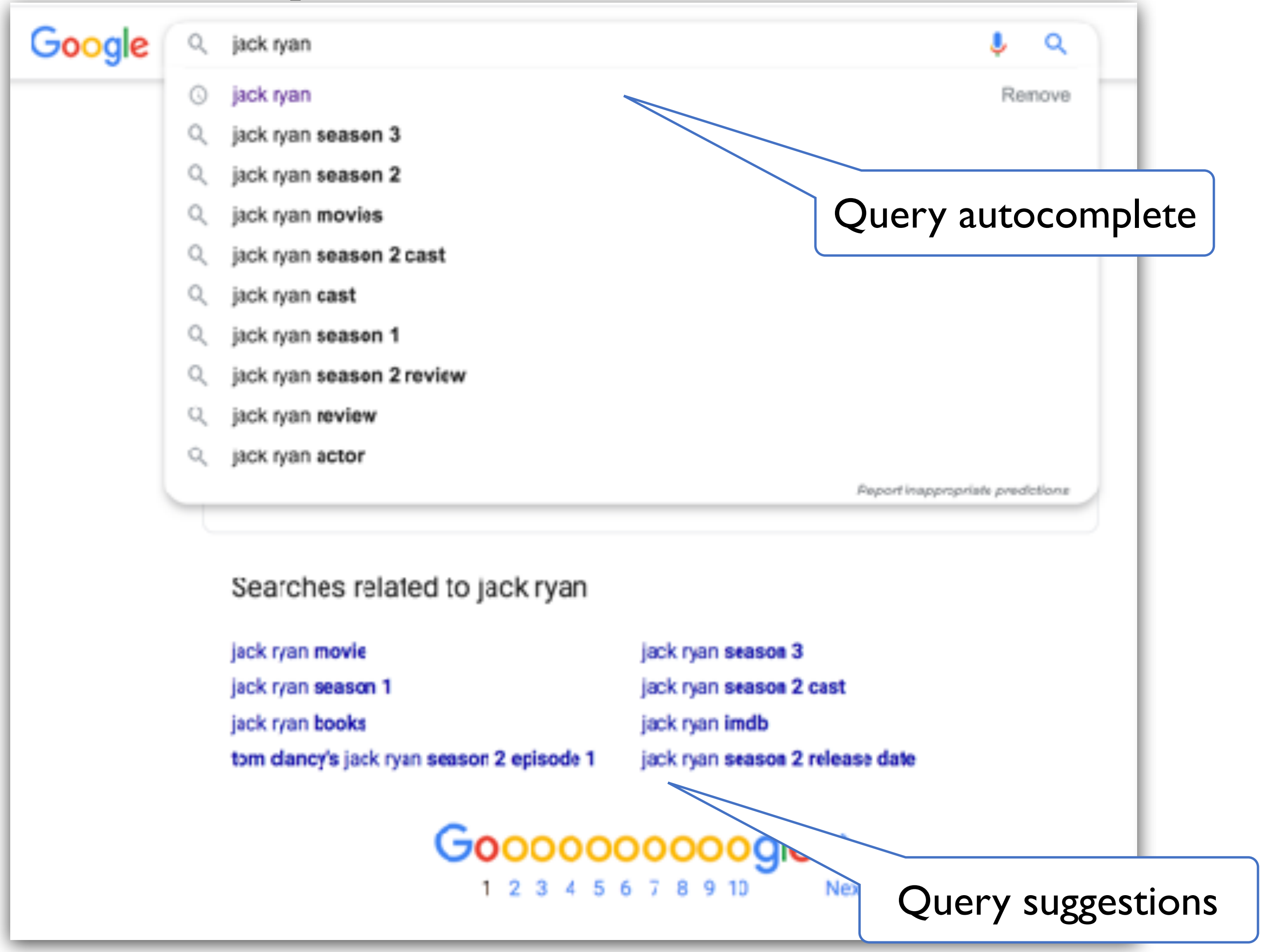
Tom Clancy's Jack Ryan - Season 2
Jack is back and better than ever! Watch now with Prime.

Tom Clancy's Jack Ryan - Season 1 (4K UHD)
2018 | TV Parental Guideline Rating: TV-MA | CC
★★★★☆ 18,568
Prime Video
Included with your Prime Membership
Watch Now
• Starring: John Krasinski , Wendell Pierce , Abbie Cornish , et al.
• Directed by: Morten Tyldum , Daniel Sackheim , Patricia Riggen , et al.

The Hunt for Red October
Book 1 of 20: Jack Ryan Universe | by Tom Clancy, Scott Brick, et al.
★★★★☆ 1,595
Audible Audiobook
$0.00 $29.99
Free with Audible trial
Kindle
$9.99
Mass Market Paperback
$9.20
prime FREE Delivery Tue, Mar 17
More Buying Choices
$6.61 (316 used & new offers)

- Search facets are ways of organizing the results for a search query
  - Allowing users to browse the results using various "facets" or properties
- Search criteria can be used to reorder the search results

# Interfaces Can Vary: Refining User Intent via Query Modification



Google

jack ryan

jack ryan — Remove
jack ryan season 3
jack ryan season 2
jack ryan movies
jack ryan season 2 cast
jack ryan cast
jack ryan season 1
jack ryan season 2 review
jack ryan review
jack ryan actor

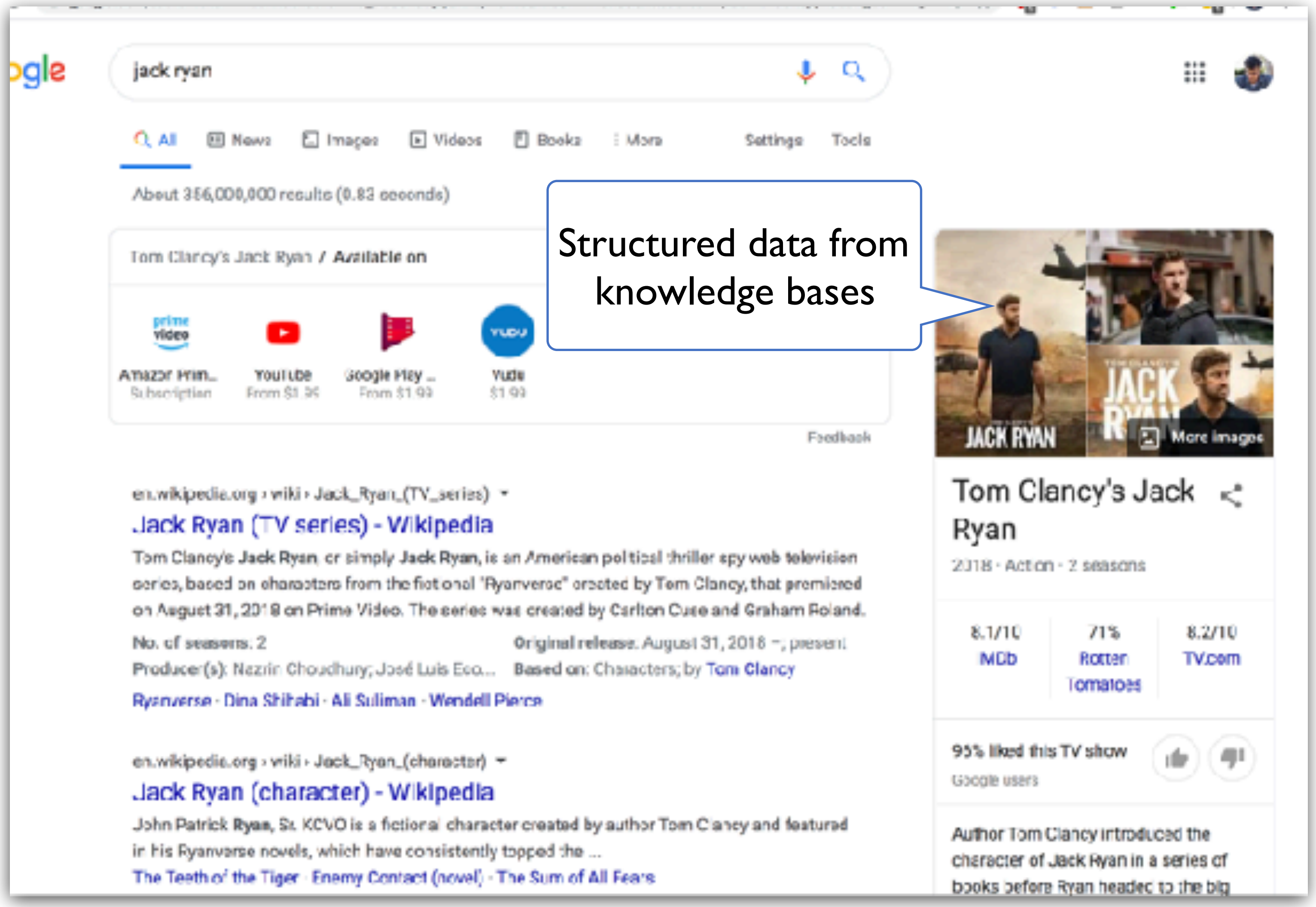
Report inappropriate predictions

Searches related to jack ryan

| | |
|---|---|
| jack ryan movie | jack ryan season 3 |
| jack ryan season 1 | jack ryan season 2 cast |
| jack ryan books | jack ryan imdb |
| tom clancy's jack ryan season 2 episode 1 | jack ryan season 2 release date |

- Search engines also suggest variations or completions of queries to assist users

# Interfaces Can Vary: Structured Information



About 356,000,000 results (0.83 seconds)

Tom Clancy's Jack Ryan / Available on

| Amazon Prim... | YouTube | Google Play ... | Vudu |
|---|---|---|---|
| Subscription | From $1.99 | From $1.99 | $1.99 |

Feedback

en.wikipedia.org › wiki › Jack_Ryan_(TV_series)

Jack Ryan (TV series) - Wikipedia

Tom Clancy's Jack Ryan, or simply Jack Ryan, is an American political thriller spy web television series, based on characters from the fictional "Ryanverse" created by Tom Clancy, that premiered on August 31, 2018 on Prime Video. The series was created by Carlton Cuse and Graham Roland.

No. of seasons: 2 Original release: August 31, 2018 – present

Producer(s): Nazrin Choudhury; José Luis Esc... Based on: Characters; by Tom Clancy

Ryanverse · Dina Shihabi · Ali Suliman · Wendell Pierce

en.wikipedia.org › wiki › Jack_Ryan_(character)

Jack Ryan (character) - Wikipedia

John Patrick Ryan, Sr. KCVO is a fictional character created by author Tom Clancy and featured in his Ryanverse novels, which have consistently topped the ...

The Teeth of the Tiger · Enemy Contact (novel) · The Sum of All Fears

Tom Clancy's Jack Ryan

2018 · Action · 2 seasons

| 8.1/10 | 71% | 8.2/10 |
|---|---|---|
| IMDb | Rotten Tomatoes | TV.com |

95% liked this TV show

Google users

Author Tom Clancy introduced the character of Jack Ryan in a series of books before Ryan headed to the big

Structured data from knowledge bases

- Knowledge bases (KBs) store information about various popular real-world entities
  - Can be used to provide structured information about these entities for quick lookup
  - Attributes like the fact that it is a movie, what it its IMDB score, etc.
- Imagine a KB to be a hierarchy of entities
- Most search engines maintain KBs and use it to enhance their search results

## Why the fuss?

- Q: Why do we have all these modules for understanding user intent better? Why not a simple text field and that's it?
    - After all, SQL is just "one query", and we have no autocompletion, refinement, faceting, …
- Considerable ambiguity in what the user wants and what they may find for a given text search query
    - e.g., Jack Ryan could refer to the books or the TV shows, but they may not get the books if they just typed "Jack Ryan"
    - Or the document/item could be an inexact match
- These are used by end-users, not by programmers or applications
    - If a web form is triggering a SQL query, this is usually rigorously tested

## Formal Setting for Information Retrieval

- Fixed input: a set of documents
- Variable input: a search keyword query

- Goal:
  - Retrieve documents that "match" the keyword query (or underlying information need)
- Metrics
  - Precision: fraction of documents that "match" returned out of the number of documents retrieved
  - Recall: fraction of documents that "match" returned out of the total number of documents that "match"
  - Often, when there are lots of matches, these metrics become somewhat meaningless
    - "Precision@k": proportion of retrieved documents that "match" out of first k

# OK… How do we make this happen

- Let’s talk about the data systems underlying search interfaces
- For now, simplifying the query:
  - “a” AND “b” AND NOT “c”
  - “Clinton” AND “Bush” AND NOT “Obama”
  - (I want to read articles that say compare Clinton to Bush, without talking about all three)
- Say we want to do this across a bunch of news articles (documents).
  - And we want to find all the matches
- One option: going through each document and checking what the word is — very slow!

# Another Approach: Term-document incidence matrix

- Precomputed matrix
- Simply look at rows corresponding to Obama, Clinton, Bush.
- Perform binary AND of Clinton entry, Bush entry, and negation of Obama entry
- Q: why is this not ideal?
- A: most of this matrix would be empty!

| | **Doc 1** | **Doc 2** | **Doc 3** | **…** |
|---|---|---|---|---|
| **Obama** | 0 | 0 | 1 | … |
| **Clinton** | 1 | 1 | 1 | … |
| **Bush** | 0 | 1 | 1 | … |
| **…** | … | … | … | … |

# Empty Term-Document Incidence Matrix

- If we have 1000 documents (each of length 1000 words) and there are 1M words, matrix has 1000 x 1M = 1B entries
- Q: But how many 1s does the matrix have (in worst case)?
  - 1000 x 1000 = 1M
- Q: What can we do?
  - Compression!

| | Doc 1 | Doc 2 | Doc 3 | … |
|---|---|---|---|---|
| **Obama** | 0 | 0 | 1 | … |
| **Clinton** | 1 | 1 | 1 | … |
| **Bush** | 0 | 1 | 1 | … |
| **…** | … | … | … | … |

# Compressing the Matrix

- If we have 1000 documents (each of length 1000 words) and there are 1M words, matrix has 1000 x 1M = 1B entries
- Compression approach:
  - For each word, maintain **sorted** list of documents that contain the word (more later) - called a **posting list**
  - Q: total size of posting list?
    - 1M entries, each of 10 bits
    - 10M

| | **Doc 1** | **Doc 2** | **Doc 3** | **…** |
|---|---|---|---|---|
| **Obama** | 0 | 0 | 1 | … |
| **Clinton** | 1 | 1 | 1 | … |
| **Bush** | 0 | 1 | 1 | … |
| **…** | … | … | … | … |

## How do Posting Lists Help?

- Obama -> 10, 24, 125, 259, 1025, 2314, …
- Bush -> 10, 15, 17, 259, 2001, 2547, …
- Clinton -> 10, 15, 24, 17, 125, 1005, 2001, 2347, …

- What approach would we use to find “Clinton” AND “Bush” AND NOT “Obama”?

- Essentially a “Merge”!

## Overall Structure

- A collection of postings lists, one per word is also called an **inverted index**
- Thus, an inverted index allows us to quickly retrieve all matching documents for queries involving boolean connectives between words
- Q: What are ways this is insufficient?

## Improving the Basic Process

- Inexact word specification
    - “modify” vs. “modified” vs. “modification” (roughly same meaning)
        - Stemming: replace all of “modify” “modifying” “modified” with “modif”
    - IL and Illinois, USA and United States
        - Canonicalization: pick a canonical form
- Some words are very frequent “to”, “and” etc.
    - These are called **stop words**
    - These words are omitted from the matching
    - Their posting lists will be very long, and a lot of time will be wasted on them!

# Improving the Basic Process

- Wanting the words to be close to each other
  - "jack" AND "ryan" may match an article about "jack black" and "ryan adams" (but you want them to be near each other!)
  - Q: how may we able to capture this?
  - A: post-processing may be really bad
    - Embed information about positions into posting list
      - Jack -> 10 (1, 10, 20), 24 (1, 15, 27), …
      - Ryan -> 10 (2, 15, 30), 24 (3, 25, 57), …
    - Or do "biword" posting lists
      - Can extend to many words A1 A2 A3 A4
        - A1 A2 and A2 A3 and A3 A4

## Scoring the Matches

- Say you found 100s of documents that matched the query.
- Now we need to rank it to show it to the user
- Typically use some combination of:
    - The inherent "quality" of a document
        - If on the web, this could be based on webpages that link to the webpage — page rank
    - The importance of the term to the document
        - One way to score this is called TF-IDF
        - Term Frequency X Inverse Document Frequency

# of term occ. in doc

$$\frac{f_t}{\sum_{t'} f_{t'}} \times \log \frac{N}{N_t}$$

# of docs with term

## Generating the Inverted Index

- Phase 1:
  - For each document, generate a "canonicalized term, documentID" pair for all non-stop-word terms in the document
  - Can encode more information with the pair, e.g., list of locations, frequency of occurrence, etc.
- Phase 2:
  - Sort these pairs based on the term, documentID pair
  - May need to use two or multi-pass sort (remember query processing!)
- Phase 3:
  - Read in the sorted pairs based on term, concatenate documentID together to form a posting list for that term
  - (We may be able to merge Phase 3 with Phase 2)

## Examples of Data Systems

- Lucene
  - Open source text indexing and search library (since 2000s)
- ElasticSearch
  - Combines Lucene-like text indexing/search with JSON querying capabilities (since 2010)
- Database systems also support text indexing and search
  - For example, PostgreSQL supports
    - GIN indexes (Generalized INverted Indexes) for querying text
    - Basic string stemming, canonicalization, tokenization capabilities

## Summary

- Unstructured data is everywhere
- The discipline of retrieving relevant unstructured data is called information retrieval
- Keyword search is the primary way to query unstructured data
    - Terms with boolean connectives
- Many variations in terms of the keyword search interface itself
- Inverted indexes as a key mechanism to support keyword search
- When coupled with various scoring mechanisms provide a good foundations for searching unstructured data