What are we searching for?

- What is search?
- Where do we use search?
- What are we searching for?
- How many searches are processed per day?
- What is the average number of words in text-based searches?
- What applications and varieties of search do we make use of?
- How do search engines and search functionality scale up?
Finding things

- Applications and varieties of search include:
  - Web search
  - Site search
  - Vertical search
  - Enterprise search
  - Desktop/mobile/local search
  - Proximity search
  - App search
  - People search (social media)
  - Location search (maps)
  - Text-based search
  - Image/video search
  - Siri; Alexa; Google Assistant
  - As-you-type search
  - Find-in-page search

Acquisition and indexing
How do we measure success?

- Relevance
  - Do the presented search results contain information that the user was actually looking for?
  - Problems of context and vocabulary mismatch often occur here (e.g., homonyms)

- User relevance
  - Search results relevant to one user may be completely irrelevant to another user
How do we measure success?

- **Timeliness**
  - Do the presented search results contain information that is current?
  - Are more recent results ranked higher?

- **Performance**
  - Users expect sub-second response times

- **Spam-resistance**

---

How do we measure success?

- **Freshness**
  - Do the presented search results contain information that is very recent (e.g., breaking news stories)

- **Age**
  - To what degree are the presented search results out of date?

- **How often should we crawl (and re-index) everything?**
How do we measure success?

- **Precision**
  - How precise are the presented search results?
  - Precision measures the proportion of retrieved documents that were relevant to the given query versus those that were not relevant
  - Focuses only on retrieved documents (i.e., not the entire corpus)

- **Recall**
  - Did we retrieve all of the relevant documents?
  - Recall measures the proportion of relevant documents actually retrieved versus all possible (indexed) documents
  - Includes all documents in the given corpus

How do we measure success?

- **Scalability**
  - How well does the search engine scale?
  - Can we easily increase the number of documents (or users or queries) by an order of magnitude (or more)?
  - What hardware and parallelization techniques can we use here?
  - What data indexing optimizations can we use here?
  - The goal here is to achieve a design that performs equally well as the system grows and expands *by orders of magnitude*
A day in the life of a crawler....

```
procedure CRAWLER THREAD(frontier)
  while not frontier.done() do
    website ← frontier.nextSite()
    url ← website.nextURL()
    if website.permitsCrawl(url) then
      text ← retrieveURL(url)
      storeDocument(url, text)
      for each url in parse(text) do
        frontier.addURL(url)
      end for
    end if
  end while
end procedure
```

Information retrieval (IR)

Information retrieval is “a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.”
– Gerard Salton (1968)

• Note that this is 1968, before the Internet, the Web, Unix, etc.
Structured information

- Structured information:
  - Often stored in a relational database or a set of related files
  - Organized via predefined tables, columns, keys, relationships, triggers, etc.
  - Queries are also structured, adhering to some implementation-specific flavor of SQL
  - Databases are often private and therefore not widely accessible to general search applications

Unstructured information

- Unstructured information:
  - Often the real or raw data that we wish to store, index, query, etc.
  - Consists of documents, images, video, audio (much of which uses textual tags and annotations)
  - Sometimes stored within a database (e.g., as a “blob” data type)
  - Contains information that humans can easily extract, but machines face major difficulties doing so (e.g., identifying headings, words, phrases, semantics)
Processing text

- Search and information retrieval has primarily focused on text processing and documents
- Search typically uses various statistical properties of text, including:
  - Word counts
  - Word frequencies
  - Phrases (i.e., n-grams, including bigrams, trigrams, etc.)
  - Linguistic and parts-of-speech features (e.g., nouns, verbs, etc.)
- The entire collection of documents is often called a corpus

What can text statistics tell us?

- English documents are rather predictable:
  - The top two most frequently occurring words are the and of, accounting for 10% of all word occurrences
  - The top six most frequently occurring words account for approximately 20% of all word occurrences
  - The top 50 most frequently occurring words account for approximately 50% of all word occurrences
  - Given all unique words in a (relatively large) document/corpus, approximately 50% occur only once
- Very similar predictions and observations can easily be made for other languages
Zipf's law

- Let’s count word occurrences, then rank words in order of decreasing frequency
- The rank $r$ of a word multiplied by its frequency $f$ is approximately equal to constant $k$:
  $$k = r \times f$$
- In other words, the frequency of the $r$th most common word is inversely proportional to $r$

Zipf's law

- Define probability of occurrence $P_r$ of a word as the word’s frequency divided by the total number of words in the document
- Zipf’s law is then: $r \times P_r = c$
- Here, $c$ is a language-specific constant (e.g., for English, $c \approx 0.1$)
Example word statistics

- AP89 is an extensive dataset containing all Associated Press (AP) news stories from 1989 (http://trec.nist.gov)
  - Total documents: 84,678
  - Total word occurrences: 39,749,179
  - Vocabulary size (i.e., unique words): 198,763
  - Words occurring more than 1000 times: 4169
  - Words occurring only once: 70,064

Top 50 words

- Here are the top 50 words from the AP89 dataset
- Statistics include:
  - Word frequencies
  - Rank \( r \)
  - Probability of occurrence \( P_r \)
  - What does \( r \times P_r \) tell us?
Vocabulary growth over time

- As the given corpus grows, so does vocabulary size
  - This growth slows down when the corpus becomes large
- The relationship between corpus size $n$ and vocabulary size $v$ was defined empirically by Herdan (1960), then by Heaps (1978)
- Heaps’ law or Herdan’s law:
  \[ v = k \times n^\beta \]
  - Here, constants $k$ and $\beta$ vary
  - Typically $10 \leq k \leq 100$ and $\beta \approx 0.5$
The GOV2 dataset contains Web pages crawled from .gov in early 2004.