CSCI 572 - Information Retrieval

# Comprehensive Midterm Study Guide

# Table of Contents

1. 1. High Priority Topics
2. 2. Key Concepts by Lecture
3. 3. Important Definitions
4. 4. Formulas & Algorithms
5. 5. Examples & Case Studies
6. 6. Study Checklist

# 1. High Priority Topics

* • SimHash is a method for determining near duplicates of web pages.
* • Near duplicates are determined by an f-bit fingerprint, where a pair of documents are near duplicates if their fingerprints are at most k-bits apart.
* • \*\*SimHash\*\*: a method developed by Moses Charikar for determining near duplicates of web pages.
* • \*\*f-bit fingerprint\*\*: a fingerprint obtained for each document using the SimHash method.
* • \*\*k-bits apart\*\*: a measure of the distance between two fingerprints, where a pair of documents are near duplicates if their fingerprints are at most k-bits apart.
* • Documents D1 and D2 are near duplicates iff Hamming-Distance(Simhash(D1), Simhash(D2)) ≤ κ
* • Boolean query
* • Advanced Search page
* • ANDing ( logical operator)
* • ORing (logical operator)
* • NOTing (logical operator)
* • Boolean query: a search method that allows users to enter specific keywords and operators to refine their search results

# 2. Key Concepts by Lecture

## Deduplication

* • 1. \*\*De-Duplication\*\* - (process of identifying and avoiding essentially identical web pages)
* • 2. \*\*Locker Storage\*\* - (strategy where only single copy of file is stored with multiple links to the single file)
* • 1. \*\*Deduplication\*\*: - The process of removing duplicate copies of data or URLs while maintaining their unique identities.
* • 2. \*\*Virtual hosts\*\*: - A feature that allows multiple hostnames to share the same document folder, but have different domain names.
* • 3. \*\*URL structure\*\*: - The components of a URL (protocol, hostname, path, page name) can be distinct yet still point to the same page.
* • Deduplication: removing duplicate data or records to improve efficiency and reduce storage needs.
* • Data Duplication
* • Deduplication (concept of removing duplicate data)
* • Similarity between web pages (differing slightly)
* • Deduplication (not explicitly mentioned in this slide, but relevant to the topic)
* • \* Mirroring ()
* • Apache mirrors
* • Deduplication ( implied by the context of the slide)
* • \*\*Deduplication\*\*: Not explicitly mentioned in the text, but implied to be related to finding software releases from Apache Software Foundation.
* • \* \*\*Deduplication\*\*: Avoiding or minimizing duplicate results in crawling
* • \* \*\*Smarter Crawling\*\*: Optimizing crawling to reduce resources and increase politeness
* • \* \*\*Better Connectivity Analysis\*\*: Combining in-links from multiple mirror sites for accurate PageRank
* • \*\*Deduplication\*\*: The process of identifying and removing duplicate or near-duplicate data.
* • \* Duplicate Problem: Exact match vs. Near-Duplicate Problem: Approximate match
* • \* Cryptographic hashing for exact match detection
* • \* Syntactic similarity with edit-distance measure for near-duplicate detection
* • Cryptographic hash function
* • Input Digest (a fixed-size alphanumeric string)
* • Hash value (also known as message digest, digital fingerprint, or checksum)
* • Properties of a cryptographic hash function:
* • \*\*Cryptographic hash functions\*\*: widely used for data deduplication and security.
* • \*\*Message-digest (MD) hash function\*\*: produces a fixed-size string of characters that represents the original input data.
* • \*\*SHA-1\*\*, \*\*SHA-2\*\*, and \*\*SHA-3\*\* families: types of cryptographic hash functions with different digest sizes.
* • \*\*RIPEMD-160\*\*: family of cryptographic hash functions that produces 160-bit digests.
* • \* \*\*Deduplication\*\*: process of identifying and eliminating duplicate documents
* • \* \*\*Hash function\*\*: a one-way function that takes input data and produces a fixed-size output, known as a hash value or digest
* • \* \*\*Bucketing\*\*: dividing documents into groups based on their hash values
* • Deduplication
* • Hash function
* • Signature
* • Fingerprint
* • Near duplicates
* • Distance measure must satisfy 4 properties:
* • \* Sets as unordered collections of objects (e.g., {a, b, c})
* • \* Distance between sets (d(A, B))
* • \* Similarity between sets (s(A, B))
* • \* Jaccard similarity
* • \* Clustering
* • \*\*Shingle\*\*: a contiguous subsequence of words in a document
* • \*\*Similarity Measures\*\*: Jaccard(A,B), Containment(A,B)
* • \*\*Resemblance\*\*: similarity measure between two documents (0 <= Resemblance <= 1)
* • \* Deduplication: a process of finding near duplicates in a large dataset
* • \* Tropical fish example: used to illustrate deduplication concept, not essential for exam
* • \* 3-shingles: sets of three consecutive words used as a basis for deduplication
* • \* Deduplication
* • \* Jaccard similarity of sets
* • \* Jaccard distance of sets
* • SimHash is a method for determining near duplicates of web pages.
* • Near duplicates are determined by an f-bit fingerprint, where a pair of documents are near duplicates if their fingerprints are at most k-bits apart.
* • 1. \*\*\*\* Ahash function: usually hashes different values to totally different hash values
* • 2. \*\*\*\* Simhash: a type of hashing where similar items are hashed to similar hash values (measured by bitwise Hamming distance)
* • 3. \*\*\*\* Bitwise Hamming distance: a measure of the similarity between two hash values
* • 1. \*\*Simhash\*\*: A technique for calculating a similarity score between two phrases
* • 2. The concept of simhash is useful for determining the similarity between two phrases
* • Deduplication: process of identifying and removing duplicate data or items.
* • \*\*Deduplication\*\*: The process of removing duplicate values from a list by sorting.
* • \* \*\*Deduplication\*\*: reducing runtime by checking adjacent pairs of a list instead of all combinations
* • \* \*\*Adjacent pair approach\*\*: improves efficiency by focusing on nearby elements
* • \* Bitwise Hamming distance preserves its value under permutation of bits
* • \* Sorting by fingerprint can be used to identify pairs of identical elements

## Info Retrieval

* • 1. \*\*\*\* Information Retrieval (IR) deals with indexing textual documents and retrieving relevant documents given a query.
* • 2. \*\*\*\* Searching for pages on the World Wide Web has become a primary application of IR.
* • 1. \*\*KEY CONCEPTS\*\* : Information Retrieval
* • + Boolean model of retrieval
* • + Vector-space model of retrieval
* • + Prof. Salton and his students' research at Cornell University
* • \* \*\*Creation of large document database systems\*\*
* • \* Searching FTP'able documents on the Internet
* • \* Archie
* • \* WAIS
* • \* Lycos
* • \* Yahoo
* • \* Altavista
* • \* Recommender Systems: computer programs that attempt to predict items users may be interested in, given some information about their profile.
* • \* Automated Text Categorization & Clustering Systems: useful for grouping news articles.
* • \* Link analysis for Web Search
* • \* Extension to retrieval of multimedia (images, music, video)
* • \* Question Answering systems that return an actual answer rather than a ranked list of documents
* • \* Information retrieval (IR) - process of searching for and retrieving relevant information from a large collection
* • \* Data analysis - process of extracting insights and patterns from data
* • \* Machine learning models - algorithms that enable systems to learn from data without being explicitly programmed
* • \* Focused on structured data stored in relational tables rather than free-form text
* • \* Efficient processing of well-defined queries in formal language (SQL)
* • \* Clearer semantics for both data and queries
* • + Human user aspects of interaction ()
* • + User interface ()
* • + Visualization ()
* • + Effective categorization of human knowledge ()
* • + Citation analysis ()
* • + Bibliometrics ()
* • + Digital libraries ()
* • \* Representation of knowledge, reasoning, and intelligent action
* • \* Formalisms for representing knowledge and queries
* • \* Integration with web ontologies and intelligent information agents
* • \* \*\*Syntactic analysis\*\*: focuses on syntax (phrase structure) in Natural Language Processing
* • \* \*\*Semantic analysis\*\*: focuses on semantics to retrieve meaning based on natural language text
* • \* \*\*Pragmatic analysis\*\*: analyzes the relationship between language and context
* • \* \*\*Extractive Summarization\*\*: a method of summarizing text by extracting key information
* • \* \*\*Abstractive Text Summarization\*\*: a method of summarizing text by generating new content
* • 1. \*\*Machine Learning\*\* - a branch of Artificial Intelligence that allows computers to evolve their behavior based on empirical data
* • 2. Machine learning is focused on developing computational systems that improve their performance with experience
* • 1. \*\*\*\* Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from various types of data.
* • 2. \*\*\*\* Data science is related to data mining, machine learning, and big data.
* • \*\*Notion of Relevance\*\*: Simplest notion of relevance is that the query string appears verbatim in the document.
* • \* Goes beyond using just keyword matching
* • \* Takes into account the meaning of the words used
* • \* Adapts to the user based on direct or indirect feedback
* • \* Indexing: The process of creating an index from a text database to enable fast search operations.
* • \* Search Initiation: Queries are used to initiate a search on the indexed documents.
* • \* Parsing forms index words (tokens)
* • \* Stopword removal
* • \* Stemming
* • \* Retrieval model specifies details of document representation, query representation, and retrieval function
* • \* Determines notion of relevance
* • \* Three major types of retrieval models: Boolean, Vector Space, and Probabilistic
* • \*\*Text Preprocessing\*\*: The process of preparing text data for information retrieval.
* • \*\*Inverted Index\*\*: A data structure that maps keywords to documents they appear in.
* • \* Document representation as a set of keywords
* • \* Boolean queries as expressions of keywords connected by AND, OR, and NOT operators
* • \* Scope indication using brackets
* • Popular retrieval model
* • Boolean models can be extended to include ranking
* • Reasonably efficient implementations possible for normal queries
* • \* Rigid Boolean query model
* • \* Difficulty in expressing complex user requests
* • \* Information retrieval challenges: too many matches for simple queries (e.g., "Lincoln")
* • \* Importance of query refinement for accurate results
* • \* \*\*Vector Space\*\*: A set of "orthogonal" terms (index terms or vocabulary) that form a vector space.
* • \* \*\*Dimension\*\*: The size of the vocabulary, equal to the number of index terms.
* • \*\*Vocabulary\*\*: consists of 3 terms (T)
* • \*\*Term weights\*\*: coefficients
* • \*\*Documents\*\*: D1 and D2
* • \*\*Query\*\*: Q = 0T + 0T + 2T
* • \* Term frequency in a document (more frequent terms are more important)
* • \* Normalizing term frequency across the entire corpus
* • Terms that appear in many different documents are less indicative of overall topic.
* • \* Term frequency (df)
* • \* Inverse Document Frequency (idf)
* • \* Combined term importance indicator
* • \* TF-IDF (Term Frequency-Inverse Document Frequency) weight
* • \* Query-document scoring
* • \* \*\*Term Frequency (tf)\*\*: represents the frequency of a term in a document
* • \* \*\*Inverse Document Frequency (idf)\*\*: measures the rarity of a term across the entire collection
* • \* Cosine of the angle between vectors is a similarity measure (not distance measure)
* • \* Similarity measure
* • \* Normalizing vectors ( )
* • \* Unit circle ( )
* • \* \*\*Similarity between vectors for the document d; and query q can be computed as the vector inner product\*\* ()
* • \* Similarity between vectors for documents and queries
* • \* Vector inner product for computing similarity
* • \* \*\*\*\* Vector space model
* • \* \*\*\*\* Binary vector representation
* • \* \*\*\*\* Weighted vector representation
* • \*\*Cosine Similarity\*\*: measures the cosine of the angle between two vectors
* • \*\*Vector Lengths\*\*: normalization factor in Cosine Similarity calculation ( Dj = sqrt(Dwj^2) )
* • \*\*Inner Product\*\*: a measure of similarity between two vectors
* • \*\*Vector Space Model\*\*: a method for representing documents and queries as vectors in a high-dimensional space.
* • \* Ranking: computing the documents in the corpus "nearest" to the query
* • \* Representing query and documents as weighted vectors:
* • Preprocessing
* • Preferred list
* • Cosine similarity
* • \* Mathematically based approach
* • \* Combination of local and global word occurrence frequencies
* • \* Missing semantic information (e.g. word sense)
* • \* Missing syntactic information (e.g. phrase structure, word order, proximity information)

## Inverted Indexing

* • \* Inverted index: a data structure used to improve search efficiency
* • \* Linked Inverted Index: an extension of inverted indexing that links words with their occurrences
* • \* Distributed Indexing: a technique for scaling inverted indexes across multiple machines
* • \* Inverted index is a data structure composed of a vocabulary (vector) containing distinct words in lexicographical order
* • \* Vocabulary includes lists of documents and text positions where each word occurs
* • 1. \*\*\*\* Inverted Indexing: A data structure used for efficient text search.
* • 2. \*\*\*\* Dictionary (Index File): The part of an inverted index that stores terms in alphabetical order with pointers to their postings lists.
* • 3. \*\*\*\* Postings List: The part of an inverted index that stores a list of document IDs where a term appears.
* • \* Inverted indexing
* • \* List of words by position
* • \* List of positions by word
* • 1. \*\* Inverted Index\*\*: Considered as a sparse matrix where rows represent terms and columns represent documents.
* • 2. \*\* Sparse Matrix\*\*: Used to represent an inverted index, where most entries are zero.
* • \* Inverted indexing ()
* • \* Bitwise AND operation in inverted indexing ()
* • \* Indexing: The process of creating a data structure that allows for efficient retrieval of specific information from a large dataset.
* • \* Inverted indexing ()
* • \* Sparsity of term-document matrix ()
* • Inverted Indexing: Storing a list of all documents that contain a specific term.
* • Inverted Indexing: A data structure used to store words and their corresponding documents.
* • 1. \*\*Inverted Indexing\*\*: An index data structure for efficient information retrieval
* • 2. Understanding of a corpus: Knowledge of the document collection before parsing
* • \*\*KEY CONCEPTS: \*\*
* • 1. \*\*Inverted Index\*\*: A data structure that stores a list of documents containing each unique term ()
* • 2. Importance of efficient indexing for search queries ()
* • \*\*Inverted Indexing\*\*: a data structure used to index and store terms in a document collection.
* • \*\*Dictionary file\*\*: contains a list of unique terms (index entries) with their corresponding frequencies and document numbers.
* • \*\*Postings file\*\*: contains the locations where each term appears in the documents, along with the frequency information.
* • + Inverted Indexing
* • + Dictionary (inverted index)
* • + Postings (document ids)
* • \* Inverted indexing ()
* • \*\*Inverted Indexing\*\*: The process of storing and retrieving documents based on their indices, allowing for efficient querying.
* • \*\*Query Processing\*\*: The process of evaluating a query to retrieve relevant documents from the index.
* • \* \*\*Inverted Indexing\*\*: not mentioned explicitly, but related to the technique described
* • \* \*\*Skip Pointers\*\*: main idea discussed in this slide
* • - \*\*Skip pointers\*\*:
* • - \*\*Posting augmentation\*\*:
* • \* Inverted Indexing
* • \* Posting lists
* • \* Successor (in the context of postings)
* • \* Skip successor (of a posting on a lower list)
* • 1. \*\*Skip Pointers\*\*: shortcuts added at indexing time to help with AND queries
* • 2. \*\*Static Corpus\*\*: when the corpus is relatively static, skip pointers are more useful
* • 3. \*\*Posting List Length (P)\*\*: the length of a postings list affects the placement of skip pointers
* • \* Inverted indexing
* • \* Phrase queries
* • + Biword (or 2-gram) concept: consecutive pair of terms in a text
* • + Indexing biwords to improve phrase searching
* • 1. Inverted Indexing
* • 2. Biwords
* • Inverted Indexing: a data structure used to store and retrieve documents based on their terms.
* • \* \*\* Inverted Indexing\*\*: Storing postings of the form docID: position1, position2, ..., where each position is a token index in the document.
* • \* \*\* Positional Index\*\*: Expanding required postings storage significantly, even if we compress position values/offsets.
* • \* Inverted indexing: a method for extracting and merging doc:position lists for multiple terms ()
* • \* Proximity searches: searching for adjacent words in a document ()
* • \* Frequency analysis in patent data\*
* • 1. N-grams are sequences of consecutive words
* • 2. N-grams can be identified at the time of parsing
* • 3. The inverted index will need pointers to all dictionary terms containing an n-gram (postings)
* • \* Inverted Indexing: A technique used to speed up information retrieval by storing an index of words in a document along with their locations.
* • \* N-grams: A sequence of n items (e.g., words, characters) from a given text.
* • 1. \*\*Inverted Indexing\*\*: The concept of inverted indexing is not explicitly mentioned in this slide, but it can be inferred from the context.
* • 2. \*\*N-gram statistics\*\*: The study focuses on analyzing n-grams (sequences of characters or words) in English and Chinese texts.
* • \* \*\*Distributed Computing Cluster\*\*: A pool of machines used for web-scale indexing to improve efficiency and reliability.
* • \* \*\*Fault-tolerance\*\*: The ability of individual machines in a distributed cluster to handle unpredictable slowdowns or failures.
* • Inverted Indexing
* • Parallel tasks (Parsers and Inverters)
* • Document corpus splitting
* • Inverted Indexing
* • Posting (refer to sign "assign")
* • \* Inverted indexing
* • \* Inverted indexing: a method for searching across multiple indices
* • \* Main index vs. auxiliary index: maintaining two separate indices for efficient search and updates
* • \* Big main index vs. small auxiliary index: trade-off between space and search efficiency
* • 1. \*\*\*\* Inverted Indexing: a data structure used to efficiently store and retrieve information from a document collection.

## Querying

* • \* Query language processing
* • \* Boolean operators (AND, BUT NOT)
* • \* Search engine functionality (returns unexpected results)
* • Boolean query
* • Advanced Search page
* • ANDing ( logical operator)
* • ORing (logical operator)
* • NOTing (logical operator)
* • \* \*\*Search Engine Query\*\*: Understanding how search engines process queries is crucial in querying.
* • \* Implicit AND: when searching for multiple keywords at once, Google/Bing will automatically search for pages that contain ALL of your keywords.
* • \* Searching for phrases in Google requires putting the phrase in quotes.
* • \* Using the exact phrase within quotes will show pages that contain the exact phrase, not just the individual words.
* • \* General principles of electricity supply systems
* • \* Law of electricity (no clear definition provided)
* • \* Basic principle of electric generator
* • \* Theory of electromagnetism (implied through discussion on magnetism and electromotive force)
* • Querying: searching for pages that contain words similar to search terms
* • Word Variations or Automatic Stemming: feature of Google that finds pages with similar words (e.g. "child" and "children")
* • \* Google favors results that have search terms near each other
* • \* Google gives higher priority to pages with terms in the same order as the query
* • \* Google is NOT case sensitive
* • \* Case sensitivity in querying (e.g., "bush" vs. "Bush")
* • 1. \*\*Querying\*\*: - The process of retrieving data from a database or system.
* • 2. \*\*Database Management\*\*: - A field that deals with storing, organizing, and managing large amounts of data in databases.
* • 3. \*\*Disney+\*\*: - A streaming service provided by The Walt Disney Company.
* • \* Boolean OR operator
* • \* Precedence of Boolean operators (OR > AND)
* • \* All query terms are implicitly ANDed
* • \* OR has higher precedence than AND
* • \* Full-word wildcard query
* • \* Google search functionality for exact phrases
* • \* Use of quotes to search for exact phrases
* • \* Stop words and their effect on search queries
* • \* Query modifiers Search
* • \* Search Operators
* • \* Restricting search results to specific file types using filetype:
* • \* Understanding that the "dot" in file extension is optional
* • \* `inanchor` operator restricts results to pages containing specific query terms in anchor text or links
* • \* `allinanchor` operator restricts results to pages containing all specified query terms in anchor text on links
* • \* Query types: By text, URLs, and titles
* • \* Importance of searching body text only
* • \* Intitle: operator restricts search results to documents containing a particular word in its title.
* • \* Restricting search results to documents containing a particular word in its URL using `inurl:\*` and `inurl:disney` ()
* • \* Restricting search results to a specific site using `site:` operator
* • \* Importance of exact matching between `site:` and domain
* • \* Using site: in conjunction with another search term or phrase.
* • \* Deepfaking the Mind
* • \* Brain-Computer Interfaces (BCIs)
* • \* Related lists web pages that are "similar" to a specified web page
* • \* Querying (note: this may not be explicitly stated in the slide, but it's implied as the topic of discussion)
* • \* Google's information retrieval system (implying the concept of a search engine)
* • \* Specific information about particular web pages (suggesting the idea of searching for specific data)
* • \* Querying with specific keywords can trigger special processing in search engines
* • \* Special processing in search engines for specific keyword queries, e.g., "stocks:"
* • \* Google operators: special symbols used to refine search results
* • \* Using math expressions in searches (e.g., 12 + 34+ 10 \* (150/ 7))
* • \* Utilizing dictionary definitions in searches (e.g., define:antidisestablishmentarianism)
* • \* Tracking numbers and airline flight numbers for specific search results
* • \*\*KEY CONCEPTS \*\*
* • 1. \*\*Google Phonebook Operators\*\*: - Different types of Google phonebook search operators.
* • 2. \*\*Privacy Violations\*\*: - Concerns raised about Google phonebook feature.
* • \*\*Statistical models\*\*: The lecture mentions that Google built a statistical model with the help of teachers to classify reading levels.
* • \*\*Reading level classification\*\*: The feature is based on classifying webpages into different reading levels using statistical models.
* • \* Google introduced the Wonder Wheel in 2009, a flash-based interface that provided possible interpretations for search queries.
* • \* The Wonder Wheel was removed in 2011 but restored in 2012 with a renaming of the "wheel of possible interpretations".
* • \* In 2014, Google re-focused the Wonder Wheel to help advertisers choose keywords.
* • \* Google Code Search was a free beta product that allowed web users to search for open-source code on the internet.
* • \* It used a regular expression engine to search for code in various formats, including tar.gz, CVS, and Subversion.
* • \* The service employed a methodology that combined trigram indexing with a custom-built regular expression engine.
* • \* It supported POSIX extended regular expression syntax.
* • \*\*Patent search\*\*: The process of searching for patents using various parameters.
* • \* Google Books: a digital database of scanned books and magazines
* • \* Optical Character Recognition (OCR): process of converting scanned texts to digital text
* • \* Copyright violations: potential issue with scanning and storing copyrighted materials
* • \* Editing errors: OCR process may introduce many errors into scanned texts
* • Querying
* • Full View
* • Snippet View
* • Limited View
* • Querying: HIGH (Understanding querying concepts is crucial for exam)
* • \* Dynamic character grouping
* • \* Consistency contains toporapic moos (Note: This appears to be a typo or unclear concept, but I'll keep it as is)
* • \* Relevance Feedback
* • \* Query Expansion
* • \* Peer-to-peer processing ()
* • \* Feedback mechanism for relevance of retrieved documents ()
* • \* Query processing on peer-to-peer networks ()
* • \*\*Enhanced Related Searches\*\*: Google has improved its related searches feature to provide more relevant results.
* • \* Querying
* • \* Search engines (Yahoo!, Bing)
* • \* Related searches
* • \*\*Query Optimization\*\*: The process of improving the performance of a query by optimizing its execution plan.
* • \* Auto-completion is a form of relevance feedback
* • \* Predicting word or phrase that the user wants to type in without actually typing it in completely
* • \* \*\*Autocomplete\*\*: Google's feature that provides suggestions based on user input
* • \* \*\*Experimental feature\*\*: Auto-complete was initially an experimental feature in 2004
* • \* Autocomplete feature in Google search
* • \* Intellisense or suggestion feature in Google search
* • 1. \*\*Autocomplete feature\*\* - The ability of a search engine to display results for a partially typed query, including links to related searches.
* • \* Auto-complete suggestions
* • \* Limitations of auto-complete systems (e.g., running out of alternatives)
* • \* Querying and search results (mark with )
* • \* Search engine algorithms and ranking systems (mark with )
* • + Query: A request for information submitted to a search engine (implied by )
* • + Search result: The output provided by a search engine in response to a query (implied by )
* • 1. \*\*Auto-Completion\*\*: Bing's use of previous queries to make suggestions before user enters a single character
* • 2. \*\*Querying\*\*: Process of searching for information on a search engine like Bing
* • \* \*\*Querying\*\*: Bing will offer results using corrected spelling and include a link for the user to correct their query
* • \* \*\*Search Engine Results\*\*: Displaying multiple sources, including images, videos, and links
* • \* Statistical measure for evaluating processes that produce lists of possible responses to samples of queries
* • \* Mean Reciprocal Rank (MRR) is a statistical measure

## Se-Basics

* • 1. KEY CONCEPTS:
* • \* \*\*Evolution of Search Engines\*\*: The development of search engines from 1991 to present day
* • \* \*\*Non-Web Search Engines\*\*: Early search engines that were not web-based (Gopher, Archie, Veronica)
* • \* \*\*Web-Based Search Engines\*\*: Transition to web-based search engines in the mid-to-late 1990s
* • \* The Internet's early search tools were developed in the late 1980s and early 1990s
* • \* Archie, Veronica, and Jughead were three major search tools that emerged during this period
* • \* The Gopher protocol was a TCP/IP application layer protocol designed for distributing, searching, and retrieving documents over the Internet
* • \* Statistical analysis of word relationships to make searching more efficient
* • \* Use of statistical analysis in search software development
* • \* World Wide Web Wanderer (marked as )
* • \* Potential for general-purpose WWW search engine (marked as )
* • \* ALIWEB (Archie-Like Indexing of the Web) - a Web search engine created in 1993
* • \* Importance of meta information in indexing web pages
* • \*\*AltaVista\*\*: A significant search engine in the early days of the web.
* • \*\*Natural Language Queries\*\*: Allowing users to search using everyday language, rather than specific keywords or syntax.
* • \*\*Advanced Searching Techniques\*\*: Features that enable more precise and effective searching, such as filtering and sorting results.
* • Lycos search engine
* • Ranked relevance retrieval
* • Prefix matching
* • Word proximity bonuses
* • \* \*\*Hierarchical listing\*\*: A way to organize links into a topical hierarchy
* • \* \*\*Portal\*\*: A website that acts as an entry point for other websites or services (e.g. Email, Finance, Groups)
* • \* \*\*Search feature\*\*: A way to search through all of the links on the Yahoo homepage
* • LookSmart was a search engine that competed with Yahoo! Directory in terms of inclusion rates.
* • Pay-per-click (PPC) business model, where listed sites pay a flat fee per click.
* • \* Inktomi Corporation: A search engine company that came into existence in 1996
* • \* Hotbot: The search engine developed by Inktomi
* • \* Paid inclusion model: A business model where websites pay a fee to guarantee display on certain search terms
* • \* Natural language search engine
* • \* Human editors matching search queries
* • \* Subject Specific Popularity
* • \* Google is a play on the word Googol, which refers to 1 followed by 100 zeros ()
* • \* A googol is bigger than the number of atoms in the universe ()
* • \* Algorithmic Yahoo
* • \* Search Era Lycos
* • \* Paid Search Era
* • \* Search Engine Basic Behavior
* • \* Providing access to heterogeneous, distributed information that is publicly available on the World Wide Web
* • \* Information comes in many different formats
* • \* Most of the information has not been screened for accuracy
* • \* The World Wide Web as a source of new opportunities in marketing
* • Search engine: program designed to help find information stored on computer systems.
* • \* The User (element 1)
* • \* The Web (element 2)
* • \* The Crawler/Spider (element 3)
* • \* The Indexer (element 4)
* • \* The Query Processor (element 6)
* • \* \*\*Spider (a.k.a. crawler/robot)\*\*: builds corpus
* • \* \*\*Indexer\*\*: creates inverted indexes
* • \* \*\*Query processor\*\*: serves query results
* • \* Distributed content creation and linking
* • \* Truth, lies, obsolete information, contradictions on the web
* • \* Diverse user backgrounds and training methods
* • \* Users' difficulty in distinguishing between search bar and URL address field
* • \* Importance of key results being at or near the top due to users rarely using scroll bars
* • \* Diverse access methodologies (high bandwidth connectivity, mobile limitations)
* • \* Poor comprehension of syntax in current search engines
* • Note that some content might overlap between categories. For instance, "Diverse user backgrounds and training methods" could be both a and a [PRIORITY] item. However, I've tried to categorize each point according to its primary significance for studying.
* • \* \*\*Types of user intentions\*\* (Informational, Navigational, Transactional)
* • 1. Query processing involves more than just matching query terms with document terms
* • 2. Semantic analysis of queries is a crucial step in search engine functionality
* • \* - User interface customization (e.g., "moar Nsagauavesane")
* • \* Search Engine Basics (SEB) - a fundamental concept in understanding how search engines work
* • \* Google - one of the most popular search engines
* • \* Las Vegas is the most populous city in the U.S. state of Nevada.
* • \* Las Vegas is officially known as the City of Las Vegas and has a population.
* • \* The official website for Las Vegas travel information is VEGAS.com.
* • \* Las Vegas has various hotels, shows, casinos, restaurants, maps, and attractions.
* • \* Hotel website structure: main pages include map, address, phone number, price of room, photos, features & amenities, directions, make reservation, special offers.
* • \* Online booking process: includes searching hotel website, selecting dates, and making a reservation.
* • \* Google Web History: a feature that allows users to view their search history
* • \* Personalization of search results: only the user can see their own search history
* • \* Google's dominance in the search engine revenue market
* • \* Baidu, Yahoo, and Bing's revenue trends over the years
* • \* Google's dominance in search engine market
* • \* Anti-Trust violations
* • \* Importance of maintaining a large index of the web

## Se-Evaluation

* • Search Engine Evaluation
* • Information Retrieval (IR)
* • Query Understanding
* • Ranking Algorithms
* • \* Evaluation metrics for search engines
* • \* Search result quality guidelines
* • \* Using log files for evaluation
* • \* Elements of good search results
* • \* Search Engine Evaluation
* • 1. Measuring search engine quality
* • 2. Importance of evaluating search engines
* • Relevant vs. Irrelevant information
* • True Positive (tp), False Positive (fp), True Negative (tn), and False Negative (fn)
* • Precision, Recall, and Accuracy as measures of classification performance
* • Set of relevant documents
* • Set of retrieved documents
* • Precision and Recall metrics
* • 1. \*\*Recall vs. Precision\*\*: High recall can lead to low precision ()
* • 2. The relationship between precision and number of documents retrieved: "precision decreases as the number of docs retrieved (or recall) increases" ()
* • \* There are three Pythagorean means: arithmetic mean, geometric mean, harmonic mean
* • \* These means are useful in analyzing data, with each having different applications and interpretations
* • 1. \*\*\*\* F-score (or F-measure): a harmonic mean of precision and recall used to evaluate algorithms and systems
* • 2. \*\*\*\* Harmonic mean: emphasizes the importance of small values, unlike arithmetic mean which is affected by outliers
* • \*\*Recall\*\*: The ratio of relevant items retrieved to all relevant items.
* • \*\*Precision\*\*: The ratio of relevant items retrieved to all items retrieved.
* • \*\*Relevant documents\*\*: Documents that are actually relevant to the query or task at hand.
* • \* Ranking #1 and #2:
* • \* Recall and Precision:
* • \* Average precision across multiple queries for relevant documents ( )
* • \* Ranking #1 docs: average precision calculation ( )
* • 1. \*\*Mean Average Precision (MAP)\*\* : A measure of the effectiveness of a search system that averages the average precision scores for each query.
* • Information Retrieval Evaluation
* • Recall and Precision metrics for evaluating search results
* • \* Mean Average Precision (MAP)
* • 1. \*\*\*\* Average over large document collection: The idea of using a large collection of documents to determine relevance.
* • 2. \*\*\*\* Query ensembles: Using multiple queries or search terms to assess relevance.
* • 3. \*\*\*\* Human relevance assessments: Assessing the relevance of search results based on human judgment.
* • 1. \*\*\*\* Discounted Cumulative Gain (DCG) is a measure that penalizes highly relevant documents appearing lower in search result lists.
* • 2. \*\*\*\* Graded Relevance value is reduced logarithmically proportional to the position of the result.
* • \*\*Discounting in Document Ranking\*\*: We want high weights for high-ranked documents because searchers are likely to inspect them, and low weights for low-ranked documents.
* • \* Information Retrieval evaluation measures (e.g., Precision, Recall, F-score)
* • \* Gridded results display vs. sequential scanning of results
* • \* Different types of metrics (Binary, Graded, Cumulative)
* • 1. \*\*\*\* Search engines have test collections of queries and hand-ranked results
* • 2. \*\*\*\* Recall is difficult to measure on the web
* • 3. \*\*\*\* Precision at top positions (e.g., top 10) is a common evaluation metric
* • 1. Google relies on raters to evaluate search results and search experience. (HIGH)
* • 2. Data generated by raters is statistically analyzed to give a view of the quality of search results and search experience. (HIGH)
* • 3. Ability to measure the effect of proposed changes to Google's search algorithms is crucial. (MEDIUM)
* • 1. \*\*Search Quality Evaluator Guidelines\*\* - The document provides guidelines for evaluators to rate search results.
* • 2. \*\*Rating Scale Categories\*\* - There are six categories used to evaluate search result relevance.
* • \* : Method for evaluating search result quality (HIGH)
* • \* : Method for testing search algorithm changes on a small scale (MEDIUM)
* • \* : Final evaluation and release of improved search results (HIGH)
* • \* A/B testing is comparing two versions of a web page to see which one performs better ()
* • \* Single innovation test ()
* • \*\*Using user clicks for evaluation\*\*: This concept highlights the idea that user interactions (clicks) can be used to assess website performance, usability, and overall effectiveness.
* • \* Conference on Information and Knowledge Management (CIKM)
* • \* International forum for presentation and discussion of research on information and knowledge management
* • \* \*\*Information and Knowledge Management\*\*: Provides an international forum for presentation and discussion of research on information and knowledge management.
* • \* \*\*CIKM Conference\*\*: A conference that brings together researchers to present and discuss research on information and knowledge management.
* • \* Query log files used for both tuning and evaluating search engines
* • \* Query logs contain various techniques such as query suggestion
* • \* Clicks are not relevance judgments
* • 1. \*\*Display Improvements\*\* () - The concept of improving search results display to provide more relevant information to users.
* • 2. \*\*autocomplete anticipations\*\* () - A feature that suggests possible search queries based on a user's input.
* • 3. \*\*Extensions to More Data\*\* () - The idea of incorporating additional data sources into search results, such as books, news, images, patents, and air schedules.
* • 4. \*\*Featured Snippets\*\* () - A summary of the most relevant information from a webpage, displayed at the top of the search results page.
* • 5. \*\*Knowledge Graph\*\* () - A knowledge base that provides additional information about entities such as people, places, and organizations.
* • \* Comparison of web search engines (marked as )

## Text Processing

* • Information Retrieval (IR) vs. text classification
* • Standing queries: periodic search for new relevant documents
* • Relevant vs. not relevant classification
* • \*\*Tokenization\*\*: The process of breaking down text into individual tokens or words (implied in the first tweet by @Robertoross)
* • \*\*Natural Language Processing (NLP)\*\*: A field of study that focuses on interactions between computers and human language (mentioned in various parts of the email)
* • - Real estate investment ()
* • - No-money-down property purchase ()
* • + Real estate investment
* • + No-money-down property purchase
* • \*\*Representation of Text Documents\*\*: How to represent text documents in a way that can be processed by computers.
* • \*\*Bag of Words\*\*: A type of high-dimensional space used to represent text documents.
* • \*\*Classification Functions\*\*: Also known as "classifiers", these are functions that determine the category of a document.
* • 1. \*\*Test language\*\*
* • 2. \*\*Data: Cet proof\*\* (likely referring to "CET" or "Certification")
* • 3. \*\*Artificial Intelligence (AI)\*\*
* • 4. \*\*Multimedia, Machine Learning, Programming, and Intelligence\*\*
* • Rule-based classifiers
* • Hand-coded rule-based classifiers
* • Text processing
* • \* Hand-coded rule-based classifiers
* • \*\*Text Classification\*\*: The process of assigning a class label to a document based on its content.
* • \*\*Supervised Learning\*\*: requires hand-classified training data
* • \*\*No Free Lunch\*\*: implies that each learning method has its own strengths and weaknesses
* • \*\*Mixture of Methods\*\*: commercial systems often use a combination of different machine learning methods
* • \* Supervised learning classifiers can use various features in text processing
* • \* Bag of words view of documents
* • \* Text collections have a large number of features
* • \* Selection can make particular classifiers feasible
* • \* Reduces training time
* • \* Makes runtime models smaller and faster
* • \* Can improve generalization (performance)
* • 1. \*\*Simplest Feature Selection Method\*\*: The most common terms can be used for feature selection with no particular foundation.
* • 2. \*\*Well-Estimatable Terms\*\*: Words that can be well-estimated are often available as evidence.
* • Naive Bayes has found a home in spam filtering
* • Spam filters often use features beyond just words
* • Document as a vector
* • High-dimensional vector space
* • Classification in high-dimensional space
* • \* Vector space classification
* • \* Labeled set of points (equivalently, vectors)
* • \* Documents in the same class form contiguous region of space
* • \* Documents from different classes don't overlap (much)
* • Text classification or categorization
* • + Text categorization ( Government, Science, Arts)
* • \* Centroid: the vector space representation of a set of documents
* • \* Rocchio forms a simple representative for each class: the centroid/prototype
* • \* Classification: nearest prototype/centroid
* • \* kNN (k-Nearest Neighbor) algorithm
* • \* Classification of documents using nearest neighbors
* • \*\*Voronoi Diagram\*\*: A way of partitioning a plane into regions based on proximity to points.
* • \* \*\*Just store the labeled training examples\*\*
* • \* \*\*kNN (k-Nearest Neighbors)\*\*
* • \* \*\*Contiguity hypothesis\*\*
* • \* 1-Nearest Neighbor (1NN) algorithm
* • \* Robustness of k-Nearest Neighbors algorithm
* • \* No feature selection necessary
* • \* Scales well with large number of classes

## Web Crawling

* • \* Categorize them accordingly with the corresponding marks (, [DEFINITION], [FORMULA], [ALGORITHM], [EXAMPLE], and [PRIORITY])
* • \* Web crawler: a computer program that visits web pages in an organized way
* • \* Web crawlers are sometimes called spiders or robots
* • \* Importance of understanding web crawlers for web crawling
* • 1. \*\*Quality in web crawling\*\*: finding the "Best" pages first
* • 2. \*\*Efficiency in web crawling\*\*: avoiding duplication or near duplication
* • 3. \*\*Etiquette in web crawling\*\*: behaving politely to not disturb website performance
* • Web Crawling
* • Seed pages (known pages to start with)
* • Fetching and parsing web pages
* • Database storage of crawled pages
* • \* Web crawling
* • \* Indexes
* • 1. - Crawling the entire web is not feasible with one machine (distributed processing)
* • 2. - Handling/Avoiding malicious pages
* • 3. - Latency/bandwidth to remote servers can vary widely
* • Protocol for web crawler limitations
* • Robots.txt file defines crawling permissions
* • \*\*robots.txt directives\*\*: rules that control which pages a robot can crawl on a website
* • \*\*User-agent\*\*: identifies the type of robot visiting the domain (e.g., Slurp)
* • \*\*Disallow\*\*: specifies URLs that should not be crawled
* • \* Robots.txt files
* • \* User-agent directives
* • 1. \*\*Determining bias to search engines from robots.txt \*\*
* • 2. \*\*Favored and disfavored robots \*\*
* • 3. \*\*Robot names in robots.txt files \*\*
* • 4. \*\*Association between robot names and AP(r) values \*\*
* • + BREADTH-FIRST SEARCH: A web crawling algorithm that examines all pages at a given level before moving on to the next level.
* • \*\*Depth-first Search\*\*: A traversal algorithm used to explore a graph or tree data structure.
* • Web Wide Crawl
* • PageRank algorithm developed by Google for determining page value
* • Breadth-First Search (BFS) crawling brings in high-quality pages early
* • \* Web crawling process
* • \* Crawling a webpage involves downloading its contents and extracting links to other relevant pages
* • How new links are added to the queue determines the search strategy
* • Search strategies can be breadth-first (FIFO) or depth-first (LIFO)
* • Focused crawlers direct their search towards "interesting" pages
* • Heuristic ordering of URLs in the queue can improve crawling efficiency
* • \* The web is a graph, not a tree, which means that links can be bidirectional and cyclic.
* • \* An Accrawler must efficiently index URLs as well as already visited pages.
* • \* To determine if a URL has already been seen, the crawler must store URLs in a standard format and develop a fast way to check if the URL has already been seen.
* • Finding all links in a page and extracting URLs is crucial for web crawling.
* • \* URLs are often long and storing all unique URLs can require large amounts of storage space
* • \* Hashing on host/domain name to determine uniqueness
* • \* Trie data structure for efficient lookup
* • \* Delta-encoded text file for compact storage
* • \*\*Tries\*\*: A data structure that can store multiple "words" with the same prefix.
* • \* \*\*URL Normalization\*\*: The process of standardizing URLs to eliminate variations.
* • \* URL Normalization
* • \* Scheme and host components of a URL are case-insensitive
* • \* Percent-encoding triplet is case-insensitive
* • \* Default port (port 80 for the "http" scheme) can be removed from or added to a URL
* • Spider Trap: A situation where a crawler re-visits the same page over and over again.
* • Session ID Management: The use of unique IDs to keep track of visitors, often used in J2EE, ASP, .NET, and PHP.
* • URL Length Monitoring: A technique to avoid spider traps by monitoring the length of URLs and stopping if it gets too long.
* • \* First generation of spam web pages (use of repeated terms)
* • \* Second generation of spam web pages (cloaking)
* • \* Third generation of spam web pages (doorway pages)
* • \*\*DNS Caching Server\*\*: A server that stores DNS responses to reduce the number of requests made to the DNS resolver.
* • \*\*UDP for DNS\*\*: The use of User Datagram Protocol (UDP) for transmitting DNS requests and responses.
* • \*\*Parallel Threads Waiting\*\*: The ability of a web crawler to perform multiple tasks concurrently using parallel threads.
* • Measuring and tuning a crawler for peak performance involves improving URL parsing speed, network bandwidth speed, and fault tolerance.
* • DNS lookup implementation
* • DNS caching server
* • \* Network delay in downloading individual pages is a bottleneck in web crawling
* • \* Having multiple threads running in parallel can improve throughput
* • \* A thread of execution is the smallest sequence of programmed instructions that can be managed independently by the scheduler
* • \* \*\*Centralized crawler control\*\*: a system where one main crawler controls multiple parallel crawlers running on a LAN
* • \* \*\*Distributed crawling\*\*: a system where multiple crawlers run on widely distributed machines with or without cross communication
* • \* Distributed crawlers must periodically update master index
* • \* \*\*Scalability\*\*: ability to handle large-scale web-crawls
* • \* \*\*Network-load dispersion and reduction\*\*: distributing network load by dividing the web into regions
* • \* \*\*\*\* Three strategies for web crawling: Independent, Dynamic assignment, Static assignment
* • Inter-partition links
* • Handling inter-partition links in web crawling
* • Firewall mode, Cross-over mode, Exchange mode for handling inter-partition links
* • \* Exchange mode limitations
* • \* Batch communication
* • \* Replication in web crawling
* • \* URL-hash based partitioning
* • \* Site-hash based partitioning
* • \* Hierarchical partitioning (e.g., by TLD)
* • \* Firewall crawlers and their benefits
* • \* Cross-over approach for 100% quality
* • + Combination of policies for Web crawler behavior
* • + Selection policy
* • + Re-visit policy
* • + Politeness policy
* • + Parallelization policy
* • \* \*\*Dynamic Web\*\*: The web is dynamic, with many new pages, updated pages, deleted pages, etc.
* • \* \*\*Page Update Tracking\*\*: Periodically check crawled pages for updates and deletions to maintain freshness.
* • 1. \*\*\*\* Steady crawler: runs continuously without pause
* • 2. \*\*\*\* Shadowing: implies new set of pages are collected and stored separately
* • 3. \*\*\*\* In-place updating: updates index current by replacing old versions with new ones
* • 4. \*\*\*\* Multiple crawlers: typically used by search engines to improve crawling efficiency
* • \* Re-visiting policies ()
* • \* Uniform policy ()
* • \* Proportional policy ()
* • \* Average freshness ()
* • \* Cho and Garcia-Molina's result that uniform policy outperforms proportional policy in terms of average freshness ()
* • \* Sitemap is a list of pages accessible to crawlers
* • \* Helps search engine crawlers find pages on the site
* • Use consistent, fully-qualified URLs to ensure Google crawls your site correctly.
* • Sitemaps can be posted anywhere on your site but only affect descendants of the parent directory.
* • Session IDs should not be included in sitemap URLs.
* • \* Multiple crawlers used by Google (e.g., Googlebot, AdsBot)
* • \* Different types of crawlers for specific tasks (e.g., images, news, video)
* • \* Importance of understanding how Googlebot sees a website
* • 1. \*\*Multiple Crawlers Used by Google\*\* (HIGH)
* • 3. \*\*Importance of Understanding How Googlebot Sees a Website\*\* (HIGH)
* • \* Googlebot cannot see within Flash files, audio/video tracks, and content within programs
* • \* Many versions of Googlebot are run on multiple machines located near the site they are indexing
* • \* Importance of creating an empty robots.txt file to prevent "File not found" in website error log
* • \*\*Web Crawling\*\*: The process of automatically scanning and indexing web pages to gather information.
* • \*\*IP Address Verification\*\*: Verifying the IP address of a web crawler to ensure it is legitimate.
* • \* Crawl rate: The number of requests per second Googlebot makes to a site when crawling it.
* • \* Googlebot must understand and execute JavaScript code to extract meaningful features from web pages.
* • \* Browsers render HTML hierarchy, but also include transformations via CSS and JavaScript.

## Web Serving Basics

* • \* Web Trends
* • \* Measurements
* • + The web has undergone significant changes over the last 30+ years
* • + Understanding the different dimensions of the web is crucial for building a web search engine today
* • + Scale, complexity, and growth are important factors to consider
* • \* The internet is used by a significant portion of the world's population ()
* • \* The internet penetration rate can be expressed as a percentage of the total population ()
* • 1. Global Internet Properties () - refers to popular websites with a global reach
* • 2. China's growing influence on the internet ()
* • 3. Top 10 Internet Properties by Global Monthly Unique Visitors ()
* • Mobile internet users growth rate in China
* • Year-over-year (Y/Y) growth comparison
* • 1. \*\*\*\* China Mobile Internet Usage Leaders: Tencent, Alibaba, Baidu dominate 71% of mobile time spent
* • 2. \*\*\*\* Share of Mobile Time Spent: WeChat leads with ~200 minutes per user, average QQ
* • \* The amount of global digital information created and shared is increasing exponentially
* • \* Digital info has grown 9x in five years to nearly 2 zettabytes in 2011, per IDC.
* • \* The growth of photo sharing remains robust despite new platforms emerging
* • \* Photos uploaded and shared daily number is doubling every year (since 2005-2014)
* • Online Video & Entertainment
* • Hours of video uploaded to YouTube every minute (as a key metric)
* • Growth rate of online video content between 2014 and 2020
* • \* Mobile devices (excluding tablets) generate a significant portion of global website traffic.
* • \* The percentage of mobile device-generated traffic has consistently hovered around 50% since 2017.
* • 1. \*\*\*\* Tablet growth is more rapid than smartphone growth, specifically iPad growth is ~3x iPhone growth.
* • 2. \*\*\*\* The rate of tablet adoption (iPad) vs. smartphone adoption (iPhone) can be compared using cumulative unit shipments.
* • \* Product Finding = Often Starts @ Search (Amazon + Google...)
* • \* Technology Cycles: refers to the pattern of technological innovation and adoption in computing devices
* • \* 10-Year Cycle: a common trend where new technologies tend to last about 10 years before being replaced or surpassed
* • 1. Re-Imagination of Computing Operating Systems ()
* • 2. Global Market Share of Personal Computing Platforms by Operating System Shipments ()
* • Market Share of Cloud Hosting Providers
* • Leading Cloud Hosting Providers (Microsoft Azure, Alibaba Group, Google Cloud Platform, etc.)
* • \* Amazon Web Services (AWS) is leading the cloud charge
* • \* Cloud Revenue Re-Accelerating
* • \* Volume Effects
* • \* Voice-related commands have increased significantly since 2008 after the launch of iPhone and Google Voice Search.
* • \* Voice-Based Mobile Platform Front-Ends: The idea that voice can replace typing in mobile interactions
* • \* Natural / Conversational Language: The use of natural language processing (NLP) to understand user requests
* • \* Voice assistants (e.g., Amazon Echo)
* • \* Growth in number of users connected
* • \* Transition from desktop/laptop use to mobile
* • \* Move away from server farms to cloud computing
* • \* Decreased dominance of Microsoft Windows
* • \* The World Wide Web is dynamic and hard to describe accurately over time
* • \* The total number of websites on the internet (approximately 1.7 billion)
* • \* The popularity of different web servers (Apache, nginx, Microsoft IIS)
* • \* The percentage of inactive/parked websites (around 75%)
* • \* The number of websites in the world has been growing rapidly over the years ()
* • \* Websites growth rate from 1991 to 2021 is significant ()
* • \* Domain Count Statistics for TLDs (TLD = Top-Level Domains)
* • \* The importance of having a record count that represents all domains known about, which is usually more accurate than other sources
* • \*\*KEY CONCEPTS \*\*
* • 1. \*\*Language diversity\*\*: Estimated 40,000 languages created by humans, with only 6,000-9,000 still in use.
* • 2. \*\*Internet language shift\*\*: Decline of English as primary language among internet users from 80% to 40%.
* • \*\*Web Serving Basics\*\*: The fundamental principles of serving web content.
* • \*\*Java Applet/Servlet\*\*: Java-based components used to extend the functionality of web servers.
* • \* Content types have a wide range (16,000 to 51,000)
* • \* Parsing content types is necessary for various applications
* • \* The Web has approximately 86 billion websites
* • \* The distribution of websites across TLDs (e.g., .com) is uneven, with 72% in .com
* • \* The number of web pages is estimated to be around 30 trillion unique URLs
* • \* HTML, PDF, Word, Excel, PPT, and others are examples of content types
* • \* The Web graph follows a power law distribution for in-degree and out-degree
* • + The Web has approximately 86 billion websites ()
* • + Distribution of websites across TLDs is uneven ()
* • + Number of web pages estimated to be around 30 trillion unique URLs ()
* • + HTML, PDF, Word, Excel, PPT, and others are examples of content types ()
* • + Storage required to hold a single snapshot of the Web: 100 petabytes ()
* • Human editors were used by Yahoo! in the past to assemble large directories.
* • The European Car Webzine focuses on prestige marques, includes articles, web broadcasting, screen savers, dealer business & economy.
* • 1. \*\*ODP - Open Directory Project\*\*: A collaborative effort to organize the web (HIGH)
* • 2. \*\*Ontology\*\*: A systematic arrangement of concepts or entities in a particular field or domain (MEDIUM)
* • 3. \*\*Distributed directory\*\*: A system where data is stored and managed across multiple locations or servers (MEDIUM)
* • 4. \*\*RDF format\*\*: A standard format for representing data on the web using resource description framework (LOW)
* • \* Open Directory - a online directory of web content
* • \* Mozilla Firefox - a web browser
* • \* The Internet Archive's snapshot of the World Wide Web
* • \* Apache Nutch
* • \* Wayback Machine
* • \* Crawler algorithm
* • 1. \*\*\*\* Multilingual Databases: Databases that store information in multiple languages to cater to diverse user needs.
* • 2. \*\*\*\* Deep Web: A part of the internet that is not indexed by search engines and can only be accessed through specific browsers or tools, designed for anonymity (e.g., Tor).
* • 3. \*\*\*\* Dark Web: A subset of the Deep Web, often associated with illicit activities.
* • 4. \*\*\*\* Government Resources: Information stored in databases that are restricted to authorized personnel or require special access.

## Youtube

* • 1. \*\*Video Search Engine\*\*: A web-based search engine that crawls the web primarily for video content. (Slide 1)
* • 2. \*\*Indexing of Video Content\*\*: The process of acquiring metadata associated with a video, such as author, title, creation date, duration, coding quality, tags, description, subtitles, and transcription.
* • 3. \*\*Ranking of Videos\*\*: The process of ordering videos under a query based on relevance, user preferences, date of upload, number of views, or user rating.
* • + Video search engines
* • + Web-wide video search engine
* • + All-content search engine
* • + Integrated universal search engine for science-oriented videos
* • \* Video hosting is highly concentrated on a small number of websites due to large file sizes involved
* • \* YouTube.com has become the defacto site for uploading videos
* • \* Subscription video on demand services (e.g., Hulu, Netflix)
* • \* Ownership structure of Hulu (jointly owned by major media companies)
* • \*\*Video Subtitling Services\*\*: There are three main types of video subtitling services: open caption, closed caption, and SDH (Subtitles for the Deaf and Hard of Hearing).
* • \*\*Speech Recognition\*\*: used to extract phrases from audio transcripts for better indexing.
* • \*\*Text Recognition\*\*: uses OCR on video slides to detect words.
* • \* YouTube is a video hosting website
* • \* The site allows users to upload, view, rate, share, add to favorites, report, and comment on videos
* • \* Google acquired YouTube in 2006 for $1.65 billion
* • \* YouTube is a search engine ()
* • \* YouTube processes more than 3 billion searches per month ()
* • \* YouTube is transforming and has surpassed other search engines like Bing ()
* • 1. \*\*\*\* YouTube traffic growth rate: 60 hours of video uploaded every minute
* • 2. \*\*\*\* Large-scale data storage capacity: estimated to be 1 sextillion gigabytes
* • YouTube's major hurdles (beyond crawling, indexing, and ranking)
* • \* YouTube requires a registered user to upload videos
* • \* Channels include thumbnails of uploaded videos, members subscribed, favorite videos, friends' lists
* • \* Having 1 million subscribers as a YouTuber can earn between $300,000 - $2 million per year
* • \* To be in the top 1000 YouTubers, you must have approximately 1.8 million subscribers
* • \* YouTube captures videos
* • \* Upload process on YouTube (making a video live)
* • \* Video management on YouTube (adding more videos, custom thumbnail)
* • \* Video/Audio quality
* • \* Encoding into streamable file format for faster video/audio quality
* • \*\*Monetization\*\*: YouTube allows creators to specify how they want to be monetized (paid promotions, sponsorships, etc.)
* • \*\*Age Restrictions\*\*: YouTube enables age restrictions on videos, allowing creators to control who can view their content
* • \*\*License and Ownership\*\*: The Standard YouTube License governs the use of licensed content on the platform
* • 1. \*\*YouTube Search Algorithm\*\*
* • 2. \*\*Vevo\*\*
* • Computer Science Education: The importance of computer science education and its challenges.
* • Online Resources for Learning: The role of online platforms like YouTube in providing educational resources for computer science students.
* • + FEATURES
* • 1. \*\*FILTERS\*\*: YouTube's search filters
* • 2. \*\*FEATURES ALGORITHMS\*\*: Mentioned but not explicitly explained
* • YouTube uses a set of metrics to rank search results
* • Video quality, metadata, views, likes, shares, links, subtitles/closed captions are all ranking factors
* • \* YouTube supports multiple video formats for uploading (MOV, MP4, AVI, WMV, FLV, 3GP, MPEGPS, WebM)
* • \* Aspect Ratio matters when uploading videos to YouTube
* • \* Video file size limit on YouTube is 128GB
* • \* Default video length limit on YouTube is 15 minutes (can be extended)
* • \* YouTube videos are played in the browser, assuming it supports HTML5
* • \* No native support for running YouTube videos on some devices (e.g. Apple products), requires separate app or transcoding
* • \* Different video standards used by various devices (e.g. H.264)
* • \* Computer algorithms play a crucial role in YouTube's recommendation system to maximize watch time.
* • \* OUTube Recommendation Algorithm ()
* • \* Query ()
* • \* Selection ()
* • Association Rule Mining
* • Co-visitation counts
* • Relatedness (r(vi,vj))
* • Normalization function (f(v; vj))
* • \* Video-rich snippet
* • \* YouTube's dominance in video-rich snippets
* • A Content Distribution Network (CDN) consists of a large set of content servers and means for dynamically selecting servers based on knowledge of the location of the user and possibly the content being requested.
* • + Identifying billions of videos
* • + Efficiently delivering video to desktop/mobile device
* • YouTube Video Cache Locations
* • Geographical distribution of video content
* • \*\*Video Transcoding\*\*: The technique of converting video into multiple different formats and resolutions to make it playable across different devices and bandwidths.
* • \*\*Content Distribution Network (CDN)\*\*: A system that sends transcoded copies of videos to various locations for faster playback.
* • [DEFINITION] \*\*Video Transcoding\*\*:
* • [DEFINITION] \*\*Content Distribution Network (CDN)\*\*:
* • \* \*\*DNS Resolution\*\*: Local DNS server resolves domain name (www.youtube.com) to IP address
* • \* \*\*YouTube Video Delivery Process\*\*: 4-step process for delivering YouTube video:
* • \* YouTube video delivery system design consists of three components: flat video ID space, multi-layered logical server organization, and 3-tiered physical cache hierarchy.
* • \* Video ID space is "flat" and has five "anycast namespaces" (with two unicast namespaces).
* • \* Complicated re-direction scheme to find the nearest data center to serve the video
* • \* Minimizing Round Trip Time (RTT)
* • \* YouTube CDN (Content Delivery Network)
* • \* YouTube had no way of making money in its early days
* • \* YouTube's infrastructure was expensive to maintain
* • \* Copyright infringement issues led to lawsuits against YouTube
* • \* Content ID: a fingerprint database of copyrighted content used by YouTube
* • \* Copyright infringement detection: the process of identifying and addressing copyright violations on YouTube
* • 1. \*\*\*\* Content ID is based on audio and video samples uploaded by rights holders to YouTube. (Slide 33)
* • 2. \*\*\*\* Video processing involves transcoding into multiple formats, including HTML5, H.264, WebM VP8, HD, non-HD, and others.
* • 3. \*\*\*\* Content ID uses a spectrogram-based approach for audio identification.
* • \* Acoustic Fingerprint: A unique representation of an audio signal
* • \* Spectrogram: Time-frequency graph representing three dimensions of audio (frequency, amplitude, and time)
* • \*\*Content ID\*\*: A system used by YouTube to automatically resolve copyright issues related to sound recordings.
* • \*\*Hashing algorithm\*\*: The process of identifying unique digital fingerprints to detect copyrighted content on YouTube.
* • \* Exponential growth of storage capacity
* • \* Kryder's Law (not explicitly defined)
* • \* Storage needs of YouTube
* • \* Data compression and re-encoding for storage efficiency
* • \* Scalability of data storage in cloud-based platforms like YouTube
* • 1. \*\*Content Delivery Network (CDN)\*\* : The backbone of YouTube's rapid content delivery.
* • 2. \*\*Load Balancer\*\* : Efficiently distributes incoming requests across multiple servers.
* • 3. \*\*Transcoding Servers\*\* : Converts and optimizes video files into various formats.
* • 4. \*\*Metadata Database\*\* : Stores crucial metadata associated with videos.
* • 5. \*\*Content Delivery Network (CDN)\*\* as the backbone of YouTube's rapid content delivery, ensuring seamless streaming for users worldwide.
* • 1. \*\*\*\* YouTube System Design: Understanding the architecture of a large-scale video sharing platform like YouTube is crucial for this topic.
* • 2. \*\*\*\* Content Delivery Network (CDN): A distributed network of servers that cache and serve content to users based on their geographical location.
* • 3. \*\*\*\* Load Balancer: A mechanism that distributes incoming traffic across multiple web servers to improve responsiveness, reliability, and scalability.
* • 4. \*\*\*\* Media Storage (S3): Amazon Simple Storage Service is a cloud-based object storage system for storing large amounts of data.

# 3. Important Definitions

• 1. \*\*De-Duplication\*\* - (identifying and eliminating duplicate web pages)

(Source: deduplication)

• 2. \*\*Locker Storage\*\* - (storage strategy for maintaining a single copy of a file)

(Source: deduplication)

• 1. \*\*Virtual hosts\*\*: - A method for hosting multiple websites on a single server, where each website has its own domain name and is accessible through different hostnames.

(Source: deduplication)

• 2. \*\*Deduplication\*\*: - The process of eliminating duplicate data or URLs while preserving their unique characteristics.

(Source: deduplication)

• \*\*SCOP (Structural Classification of Proteins)\*\*: a database that classifies proteins based on their structural characteristics.

(Source: deduplication)

• \*\*Domain\*\*: a unit of protein structure that performs a specific function.

(Source: deduplication)

• \*\*Fold\*\*: a common three-dimensional structure found in multiple proteins.

(Source: deduplication)

• \*\*Superfamily\*\*: a group of related folds with similar structures.

(Source: deduplication)

• \*\*Family\*\*: a group of related superfamilies with similar structures.

(Source: deduplication)

• Deduplication: The process of identifying and eliminating duplicate copies of data.

(Source: deduplication)

• Snapshot: A copy of a webpage at a particular point in time.

(Source: deduplication)

• \*\*Document Object Model (DOM)\*\*: a tree-based structure for representing an HTML document

(Source: deduplication)

• \* Mirroring: systematic replication of web pages across hosts ()

(Source: deduplication)

• Apache mirrors: a collection of servers that mirror or replicate content from a central server (implied by the context)

(Source: deduplication)

• Regions: geographic areas where Apache mirrors are located (listed in the "regions" section)

(Source: deduplication)

• \* \*\*PageRank (measure of importance)\*\*: A measure of a webpage's importance based on its in-links

(Source: deduplication)

• \* \*\*Mirror Sites\*\*: Copies of websites hosted on different servers

(Source: deduplication)

• \* \*\*Clustering\*\*: Grouping similar documents together.

(Source: deduplication)

• \* \*\*Plagiarism detection\*\*: Identifying pairs of documents that have significantly borrowed from each other.

(Source: deduplication)

• \* \*\*Spam detection\*\*: Using near-similarity techniques to identify spam emails.

(Source: deduplication)

• \* Duplicate Problem: A problem where duplicate data needs to be identified.

(Source: deduplication)

• \* Near-Duplicate Problem: A problem where approximate matching is required, e.g., detecting similar documents.

(Source: deduplication)

• \* Cryptographic hashing: A technique used for generating unique digital fingerprints of data.

(Source: deduplication)

• Cryptographic hash function: A hash function that takes an input (message) and returns a fixed-size alphanumeric string (hash value)

(Source: deduplication)

• Hash value: Also known as message digest, digital fingerprint, or checksum

(Source: deduplication)

• Input Digest: A fixed-size alphanumeric string resulting from a cryptographic hash function

(Source: deduplication)

• \*\*MD5 (message-digest)\*\*: a widely used cryptographic hash function producing a 128-bit (16-byte) hash value.

(Source: deduplication)

• \*\*SHA-1\*\*: a type of cryptographic hash function producing a 160-bit (20-byte) hash value.

(Source: deduplication)

• \*\*SHA-2\*\*: a family of algorithms that produce digests of size 224, 256, 384, and 512 bits.

(Source: deduplication)

• Distance measure: a method to compute similarity between two objects or documents.

(Source: deduplication)

• Euclidean distance: a type of distance measure that calculates the straight-line distance between two points in n-dimensional space.

(Source: deduplication)

• Jaccard distance: a type of distance measure that calculates the ratio of the sizes of the intersection and union of sets.

(Source: deduplication)

• Cosine distance: a type of distance measure that calculates the angle between two vectors.

(Source: deduplication)

• Edit distance: a type of distance measure that calculates the minimum number of insertions and deletions needed to transform one string into another.

(Source: deduplication)

• Hamming distance: a type of distance measure that calculates the number of components in which two Boolean vectors differ.

(Source: deduplication)

• \* d(A, B) = distance between two sets A and B

(Source: deduplication)

• \* s(A, B) = similarity between two sets A and B

(Source: deduplication)

• \* Intersection: size of intersection between two sets / size of union

(Source: deduplication)

• \* Union: size of intersection between two sets / size of union (same as above, but note that it's not a standard definition)

(Source: deduplication)

• \*\*Shingle Set (S(D,w))\*\*: a set of shingles for a document D with width w

(Source: deduplication)

• \*\*Jaccard(A,B)\*\*: similarity measure defined as size of (S(A,w) intersect S(B,w)) / size of (S(A,w) union S(B,w))

(Source: deduplication)

• \*\*Containment(A,B)\*\*: similarity measure defined as size of (S(A,w) intersect S(B,w)) / size of (S(A,w))

(Source: deduplication)

• 1. \*\*Shingle\*\*: a fixed-size substring of a document

(Source: deduplication)

• 2. \*\*Collision\*\*: when two different documents match each other's shingles (exact wording preserved)

(Source: deduplication)

• 3. \*\*Stop words\*\*: common words that are typically omitted in deduplication

(Source: deduplication)

• \* Hash value: a unique numerical representation of a string or sequence

(Source: deduplication)

• \* Fingerprints: selected hash values that are considered representative of the original data

(Source: deduplication)

• \* Jaccard similarity (J(A,B)) = |A ∩ B| / |A ∪ B|

(Source: deduplication)

• \* Jaccard distance = 1 - J(A,B)

(Source: deduplication)

• \* k-shingles

(Source: deduplication)

• \* Fingerprint

(Source: deduplication)

• \*\*SimHash\*\*: a method developed by Moses Charikar for determining near duplicates of web pages.

(Source: deduplication)

• \*\*f-bit fingerprint\*\*: a fingerprint obtained for each document using the SimHash method.

(Source: deduplication)

• \*\*k-bits apart\*\*: a measure of the distance between two fingerprints, where a pair of documents are near duplicates if their fingerprints are at most k-bits apart.

(Source: deduplication)

• 1. \*\*\*\* Ahash function: a type of hash function that produces unique hash values for different inputs

(Source: deduplication)

• 2. \*\*\*\* Simhash: a type of hashing that measures similarity between items using bitwise Hamming distance

(Source: deduplication)

• 1. \*\*Shingles\*\*: Breaking down an input phrase into smaller sub-phrases (features)

(Source: deduplication)

• 2. \*\*Hash function\*\*: A mathematical function that takes a feature as input and produces a fixed-size hash value

(Source: deduplication)

• 3. \*\*Bitwise Hamming distance\*\*: The number of positions at which two binary strings differ

(Source: deduplication)

• 8-bit hash values: a way to represent text as a binary number using 8 bits (0s and 1s).

(Source: deduplication)

• Fingerprint formed from 8-bit hash values: a unique representation of the original text.

(Source: deduplication)

• Weights: frequencies or importance of each word in the original text.

(Source: deduplication)

• \*\*Bitwise Hamming Distance (hdist)\*\*: A measure of the difference between two numbers represented in binary, calculated as the number of positions at which the corresponding bits are different.

(Source: deduplication)

• \* \*\*Hamming distance\*\*: the number of positions at which two strings are different (implied by "low Hamming distance")

(Source: deduplication)

• \* Bitwise Hamming distance: a measure of the number of different bits between two numbers

(Source: deduplication)

• \* Permutation: rearrangement of bits or elements in a specific order

(Source: deduplication)

• \* Rotating bits: shifting bits left and replacing lowest order bit with highest order bit

(Source: deduplication)

• 1. \*\*\*\* Corpus: A set of documents to be indexed.

(Source: info\_retrieval)

• 2. \*\*\*\* Indexing: The process of creating an organized system of information from a corpus.

(Source: info\_retrieval)

• - Publishes legal, tax and regulatory information for legal, corporate, government and academic markets

(Source: info\_retrieval)

• - Contains data from more than 1.4 billion unique records of key information

(Source: info\_retrieval)

• - National Library of Medicine health information database

(Source: info\_retrieval)

• \* FTP'able documents: Documents that can be accessed via File Transfer Protocol (FTP)

(Source: info\_retrieval)

• \* Archie: A search engine developed in 1990 to index and search FTP sites

(Source: info\_retrieval)

• \* WAIS: An information retrieval system that uses a database of articles and documents

(Source: info\_retrieval)

• \* TREC (Text REtrieval Conferences): a series of organized competitions sponsored by NIST to evaluate IR systems.

(Source: info\_retrieval)

• \* Collaborative filtering algorithm: an approach used in recommender systems to predict user preferences based on the behavior of similar users.

(Source: info\_retrieval)

• \* Document Object Model (DOM): provides some clues about web page structure

(Source: info\_retrieval)

• - Human user aspects: The focus on how humans interact with information systems.

(Source: info\_retrieval)

• - User interface: The means by which users interact with a system or application.

(Source: info\_retrieval)

• - Visualization: The use of visual elements to represent and communicate information.

(Source: info\_retrieval)

• \* \*\*First-order Predicate Logic\*\*: a formal system that uses quantified variables over a specified domain of discourse

(Source: info\_retrieval)

• \* \*\*Bayesian Networks\*\*: directed acyclic graph model that represents a set of random variables and their dependencies

(Source: info\_retrieval)

• \* \*\*Web Ontology Language (OWL)\*\*: a family of knowledge representation languages for authoring ontologies

(Source: info\_retrieval)

• \* \*\*Entity Recognition\*\*: the process of identifying specific entities in natural language text (e.g. people, places, organizations)

(Source: info\_retrieval)

• \* \*\*Natural Language Processing (NLP)\*\*: a field of study that focuses on the interaction between computers and human language

(Source: info\_retrieval)

• \* \*\*Natural Language Generation (NLG)\*\*: the process of generating human-like language from a computer system

(Source: info\_retrieval)

• 1. \*\*Supervised Learning\*\* - automated classification of examples based on learning concepts from labeled training examples

(Source: info\_retrieval)

• 2. \*\*Unsupervised Learning\*\* - automated methods for clustering unlabeled examples into meaningful groups

(Source: info\_retrieval)

• 3. \*\*Data Mining\*\* - the discovery of previously unknown properties of given data

(Source: info\_retrieval)

• 1. \*\*\*\* A data scientist is someone who knows how to extract meaning from and interpret data.

(Source: info\_retrieval)

• 2. \*\*\*\* The role of a data scientist requires both tools and methods from statistics and machine learning, as well as human review.

(Source: info\_retrieval)

• None explicitly stated

(Source: info\_retrieval)

• \* Text Database: A collection of unstructured text data that is used as input for information retrieval systems.

(Source: info\_retrieval)

• \* User Interface: The part of the system that allows users to interact with the search engine, submit queries, and view results.

(Source: info\_retrieval)

• \* Inverted index

(Source: info\_retrieval)

• \* Query token

(Source: info\_retrieval)

• \* Relevance metric

(Source: info\_retrieval)

• \* Binary relevance: a binary (yes/no) notion of relevance

(Source: info\_retrieval)

• \* Continuous relevance: ranked retrieval, where documents are ranked in order of relevance

(Source: info\_retrieval)

• \* Set theoretic model: Boolean model (chapter 1 in Manning et al)

(Source: info\_retrieval)

• \* Statistical/algebraic model: Vector Space model (chapter 2 in Manning et al)

(Source: info\_retrieval)

• \* Chapter 11 in Manning et al refers to Probabilistic models

(Source: info\_retrieval)

• \*\*Tokenization\*\*: Breaking down text into individual words or tokens.

(Source: info\_retrieval)

• \*\*Stemming\*\*: Reducing words to their root form (e.g. "running" becomes "run").

(Source: info\_retrieval)

• \*\*Stopwords\*\*: Common words that do not carry much meaning, such as "a", "the", etc.

(Source: info\_retrieval)

• \* Keyword: a word representing a document

(Source: info\_retrieval)

• \* Query: a request for information, expressed as a Boolean expression of keywords

(Source: info\_retrieval)

• \* AND operator: combines two or more keywords to search for documents containing all specified keywords

(Source: info\_retrieval)

• \* OR operator: combines two or more keywords to search for documents containing at least one of the specified keywords

(Source: info\_retrieval)

• \* NOT operator: excludes specific keyword from search results

(Source: info\_retrieval)

• \* AND: means all; OR: means any (in the context of Boolean queries)

(Source: info\_retrieval)

• \* \*\*Index Terms\*\*: Distinct terms remaining after preprocessing.

(Source: info\_retrieval)

• \* \*\*Vocabulary\*\*: Set of index terms.

(Source: info\_retrieval)

• \* \*\*Document Vector\*\*: A vector representing a document in the vector space, with each element being the weight of an index term.

(Source: info\_retrieval)

• \*\*Vocabulary\*\*: a set of terms used to represent documents or queries

(Source: info\_retrieval)

• \*\*Term weights\*\*: numerical values assigned to each term in the vocabulary

(Source: info\_retrieval)

• \*\*Document\*\*: a unit of information represented as a set of weighted terms

(Source: info\_retrieval)

• \*\*Query\*\*: a request for information, represented as a set of weighted terms

(Source: info\_retrieval)

• \* tf: term frequency in a document (number of times a term appears in a document)

(Source: info\_retrieval)

• \* df; = document frequency of term

(Source: info\_retrieval)

• \* = number of documents containing term i

(Source: info\_retrieval)

• \* N: total number of documents

(Source: info\_retrieval)

• + df; = document frequency of term

(Source: info\_retrieval)

• + = number of documents containing term i

(Source: info\_retrieval)

• + N: total number of documents

(Source: info\_retrieval)

• \* df: document frequency, number of documents containing a term

(Source: info\_retrieval)

• \* idf: inverse document frequency, measures importance of a term in the collection

(Source: info\_retrieval)

• Note that I didn't mark any of these examples as [EXAMPLE] since they are not concrete examples, but rather illustrative cases. Instead, I categorized them under .

(Source: info\_retrieval)

• \* tf-idf: a method for calculating the importance of a term in a document

(Source: info\_retrieval)

• \* N/df: the number of documents divided by the frequency of the term (denominator)

(Source: info\_retrieval)

• \* IDF: Inverse Document Frequency, a measure of how rare a term is across all documents

(Source: info\_retrieval)

• \* Term Frequency (tf): ratio of the frequency of a term in a document to the total number of terms in the document

(Source: info\_retrieval)

• \* Inverse Document Frequency (idf): logarithm of the ratio of the total number of documents to the number of documents containing the term

(Source: info\_retrieval)

• \* Distance between vectors d, and d, captured by cosine of the angle between them

(Source: info\_retrieval)

• \* A similarity measure is a function that computes the degree of similarity between two vectors

(Source: info\_retrieval)

• \* For binary vectors, the inner product is the number of matched query terms in the document (size of intersection) (Hamming distance) ()

(Source: info\_retrieval)

• \* For weighted term vectors, it is the sum of the products of the weights of the matched terms. ()

(Source: info\_retrieval)

• \* Binary vectors: vectors where each element is either 0 or 1 (representing matched or non-matched terms)

(Source: info\_retrieval)

• \* Weighted term vectors: vectors where each element represents the weight of a term

(Source: info\_retrieval)

• \* Hamming distance: the number of matched query terms in the document

(Source: info\_retrieval)

• \* \*\*\*\* o: term in the document or query (denoted by 1 if present, 0 if not)

(Source: info\_retrieval)

• \* \*\*\*\* S: size of vocabulary (number of unique terms)

(Source: info\_retrieval)

• None extracted, but note that "Cosine Similarity" and "Vector Lengths" are key concepts with implications for understanding the definition.

(Source: info\_retrieval)

• \*\*idf (Inverse Document Frequency)\*\*: a measure of how rare a word is in the document collection.

(Source: info\_retrieval)

• \*\*cosSim (Cosine Similarity)\*\*: a measure of similarity between two vectors, used to compute the score of each document.

(Source: info\_retrieval)

• \* \*\*Weighted vector\*\*: A vector where each component represents the importance of a particular feature or term, often calculated using techniques like TF-IDF.

(Source: info\_retrieval)

• \* Term independence: the assumption that terms in a query are independent of each other

(Source: info\_retrieval)

• \* Inverted index: an index data structure used to store the location(s) of each word in a document collection

(Source: inverted\_indexing)

• \* Linked Inverted Index: an extension of inverted indexing that stores pointers to linked words with their occurrences

(Source: inverted\_indexing)

• \* Skip Pointers: pointers that allow for efficient merging of sorted lists

(Source: inverted\_indexing)

• \* Case folding: converting all uppercase letters to lowercase

(Source: inverted\_indexing)

• \* Stemming: reducing words to their morphological roots

(Source: inverted\_indexing)

• \* Stop words: words that are so common they provide no information

(Source: inverted\_indexing)

• 1. \*\*\*\* Term Frequency: Not explicitly defined, but implied as the frequency of a term in a particular document (e.g., "ae" has a certain frequency in the system 1 document).

(Source: inverted\_indexing)

• 2. \*\*\*\* Document Frequency (df): The number of documents that contain a particular term.

(Source: inverted\_indexing)

• 3. \*\*\*\* Inverted Index: A data structure that stores terms and their corresponding postings lists.

(Source: inverted\_indexing)

• \* \*Inverted file\*: A list of positions by word

(Source: inverted\_indexing)

• \* Each entry in the inverted file represents a word and its corresponding positions

(Source: inverted\_indexing)

• 1. \*\* Term\*\*: A single word or phrase in a document (e.g., "Antony", "Brutus", etc.)

(Source: inverted\_indexing)

• 2. \*\* Document\*\*: A unit of text that is indexed by the inverted index (e.g., "Antony and Cleopatra", "Julius Caesar", etc.)

(Source: inverted\_indexing)

• + Complemented vector (implied as "complemented" in the text) (): a vector that has been modified to represent the absence of certain terms.

(Source: inverted\_indexing)

• \* Term-document matrix: a matrix representing the presence or absence of terms in documents ()

(Source: inverted\_indexing)

• Linked lists: Data structures that allow for dynamic space allocation and easy insertion of terms into documents.

(Source: inverted\_indexing)

• Inverted indexing: A technique used to efficiently store and retrieve information in search engines and databases.

(Source: inverted\_indexing)

• [PRIORITY] HIGH for Inverted Indexing concept and Linked lists

(Source: inverted\_indexing)

• \*\*Document ID\*\*: Unique identifier assigned to each document.

(Source: inverted\_indexing)

• \*\*Modified Token\*\*: Modified form of a word, e.g., "Caesar" becomes "caesar".

(Source: inverted\_indexing)

• \*\*Sequence of (Modified token, Document ID) pairs\*\*: A list of tuples containing the modified word and its corresponding document ID.

(Source: inverted\_indexing)

• 1. Inverted File: A sorted list of terms with their corresponding documents

(Source: inverted\_indexing)

• \*\*DEFINITIONS: \*\*

(Source: inverted\_indexing)

• 1. \*\*Term\*\*: A word or phrase in a document ()

(Source: inverted\_indexing)

• 2. \*\*Document Frequency (DF)\*\*: The number of documents containing a particular term ()

(Source: inverted\_indexing)

• 3. \*\*Term Frequency (TF)\*\*: The frequency of a term within a document ()

(Source: inverted\_indexing)

• \*\*Term Frequency (TF)\*\*: the number of times a term appears in a single document.

(Source: inverted\_indexing)

• \*\*Document Frequency (DF)\*\*: the number of documents that contain a particular term.

(Source: inverted\_indexing)

• \*\*Inverted Index\*\*: a data structure used to efficiently retrieve the locations and frequencies of terms in a document collection.

(Source: inverted\_indexing)

• + Dictionary: a data structure that stores words and their corresponding postings

(Source: inverted\_indexing)

• + Postings: the document ids associated with a word in the dictionary

(Source: inverted\_indexing)

• \* \*\*Postings\*\*: A collection of document IDs that contain a specific term.

(Source: inverted\_indexing)

• \* \*\*AND Operator\*\*: A logical operator that combines two or more terms, returning only documents that contain all specified terms.

(Source: inverted\_indexing)

• \* Inverted Indexing: A data structure that stores references to words in documents, allowing for efficient retrieval.

(Source: inverted\_indexing)

• \* Posting lists: Lists of postings that contain specific word or phrase occurrences.

(Source: inverted\_indexing)

• \* Successor (in the context of postings): The next posting after a given posting.

(Source: inverted\_indexing)

• 1. \*\*Skip Pointers\*\*: shortcuts that help for AND queries and are useful when the corpus is relatively static

(Source: inverted\_indexing)

• 2. \*\*Posting List\*\*: a list of documents containing a particular term

(Source: inverted\_indexing)

• \* Biword (or 2-gram): A consecutive pair of terms in a text

(Source: inverted\_indexing)

• \* Dictionary term: Each bi-word is treated as a single dictionary term

(Source: inverted\_indexing)

• 1. Vocabulary database : a collection of words or phrases used in indexing

(Source: inverted\_indexing)

• 2. Boolean query on biwords : a search query broken down into smaller segments (biwords)

(Source: inverted\_indexing)

• \* \*\*Inverted Entry\*\*: an entry in the inverted index containing information about a specific term, including:

(Source: inverted\_indexing)

• \* \*\* Posting\*\*: A record of the occurrences of a term in a document, including its frequency and positions.

(Source: inverted\_indexing)

• \* \*\* Token index\*\*: The index of a token (word) within a document.

(Source: inverted\_indexing)

• \* POS tagging: Part-of-Speech tagging, a process of automatically assigning grammatical categories to words in a text\*

(Source: inverted\_indexing)

• 1. N-grams: any sequence of consecutive words

(Source: inverted\_indexing)

• 2. Postings: pointers to all dictionary terms containing a given n-gram

(Source: inverted\_indexing)

• \* Token: An individual item in a dataset (e.g., word, character).

(Source: inverted\_indexing)

• \* Bigram: A sequence of two items (e.g., words).

(Source: inverted\_indexing)

• \* Trigram: A sequence of three items (e.g., words).

(Source: inverted\_indexing)

• \* Four-gram: A sequence of four items (e.g., words).

(Source: inverted\_indexing)

• 1. \*\*N-gram\*\*: A sequence of n items (characters, words, etc.) from a text.

(Source: inverted\_indexing)

• 2. \*\*Uni-gram\*\*, \*\*Bi-gram\*\*, \*\*3-gram\*\*, \*\*4-gram\*\*: Specific types of n-grams with different lengths.

(Source: inverted\_indexing)

• \* \*\*Master Machine\*\*: A central machine that directs the indexing job and assigns tasks to idle machines in the pool.

(Source: inverted\_indexing)

• \* \*\*Indexing Job\*\*: The process of creating an inverted index for efficient searching and retrieval of data.

(Source: inverted\_indexing)

• Parser: reads documents and emits (term, doc) pairs

(Source: inverted\_indexing)

• Inverter: complements the parser to complete index inversion

(Source: inverted\_indexing)

• Split: a subset of documents assigned to an idle parser machine

(Source: inverted\_indexing)

• Posting: a data structure that stores the locations of words in a document

(Source: inverted\_indexing)

• Inverted Index: an index data structure used for fast and efficient searching of text documents

(Source: inverted\_indexing)

• \* Partition: a subset of documents or terms (not explicitly defined, but implied)

(Source: inverted\_indexing)

• \* Invalidating bit-vector: a data structure used to mark deleted documents in the auxiliary index

(Source: inverted\_indexing)

• 1. \*\*\*\* Inverted Index: a mapping of keywords or phrases in a document collection to the documents that contain them.

(Source: inverted\_indexing)

• 2. \*\*\*\* Document Collection: a set of documents that are being indexed and searched.

(Source: inverted\_indexing)

• Query box: A text field in a web search engine where users can enter keywords or phrases to initiate a search.

(Source: querying)

• Boolean query: a search method that allows users to enter specific keywords and operators to refine their search results

(Source: querying)

• \* \*\*SOY cose som\*\* ( acronym or term not explicitly defined)

(Source: querying)

• \* \*\*Google apple orchard computer\*\*: Example of a query

(Source: querying)

• \* AND operator: used to combine two or more keywords in a search query (e.g., "disney AND disneyland AND pirates").

(Source: querying)

• \* Electromotive force (EMF) - "The pressure that is put on free electrons that causes them to flow"

(Source: querying)

• \* Magnetism

(Source: querying)

• \* Electric generator

(Source: querying)

• Stop Words: common words like "the," "and," etc. that are ignored in searches

(Source: querying)

• \* Case sensitivity: Google's ability to treat uppercase and lowercase letters the same way

(Source: querying)

• \* None explicitly stated

(Source: querying)

• 1. \*\*Streaming Service\*\*: - A type of service that provides access to content on-demand over the internet.

(Source: querying)

• 2. \*\*Database\*\*: - A collection of organized data stored in a way that allows for efficient retrieval and manipulation.

(Source: querying)

• \* Boolean OR: an operator used to combine keywords in a search query, allowing for multiple possible matches

(Source: querying)

• \* OR always in all caps

(Source: querying)

• \* Implicit ANDing: The default behavior of search engines to treat all query terms as ANDed, unless specified otherwise

(Source: querying)

• \* Precedence: The order in which operations or operators are performed in a query

(Source: querying)

• \* \*\*Wildcard\*\*: a symbol used in search queries to represent one or more characters (e.g. +)

(Source: querying)

• \* \*\*Stop word\*\*: a common word that is ignored by the search engine (e.g. "the", "and")

(Source: querying)

• \* daterange: Service - no definition provided ( likely a search operator)

(Source: querying)

• \* filetype: allinanchor:, allintext:, allintitle:, allinurl:, cache:, define:, jnanchor: - no clear definition, appears to be list of search operators

(Source: querying)

• \* filetype: restricts search results to files with a specific suffix

(Source: querying)

• \* Filetype: should not have spaces between filetype and the file extension

(Source: querying)

• \* Optional inclusion of the dot (.) in file extensions

(Source: querying)

• \* Anchor text: the text on links to a page

(Source: querying)

• \* Intext: A query type that searches for the exact phrase within the body text (e.g., "Pirates of the Caribbean")

(Source: querying)

• \* intitle: - an operator that restricts search results to documents containing a particular word in its title.

(Source: querying)

• \* `site:` operator - restricts search results to a specific website or domain

(Source: querying)

• \* Domain - the name of a website (e.g. google.com, cs.stanford.edu)

(Source: querying)

• \* None in this specific content, but it's worth noting that "web cache" is a term related to web searching and caching. However, since there are no specific definitions provided, I won't mark it as .

(Source: querying)

• \* Qos: aan + There can be no space between related: and the URL.

(Source: querying)

• \* "Related:" is a search operator that lists web pages similar to a specified page

(Source: querying)

• \* Stock ticker symbols

(Source: querying)

• \* Antidisestablishmentarianism: a term used to demonstrate the limits of the English language

(Source: querying)

• \* Tracking number: a unique code provided by shipping companies like FedEx or UPS

(Source: querying)

• \*\*DEFINITIONS \*\*

(Source: querying)

• 1. \*\*Phonebook:\*\* - Searches the entire Google phonebook.

(Source: querying)

• 2. \*\*rphonebook:\*\* - Searches residential listings only.

(Source: querying)

• 3. \*\*bphonebook:\*\* - Searches business listings only.

(Source: querying)

• \*\*Statistical model\*\*: A mathematical representation of a system or process that uses statistical methods to make predictions or classify data.

(Source: querying)

• \*\*Reading level\*\*: The complexity or difficulty of written content, often measured by factors such as vocabulary, sentence structure, and grammar.

(Source: querying)

• \* Trigram index: A method of indexing data to enable fast searching.

(Source: querying)

• \* Regular expression engine: A software component that searches for patterns in text or code using regular expressions.

(Source: querying)

• \* POSIX extended regular expression syntax: A standard for regular expressions used in Unix-like operating systems.

(Source: querying)

• \* OCR: Optical Character Recognition, the process of recognizing and extracting text from images or scans

(Source: querying)

• Querying: The process of searching for specific information in a database or repository.

(Source: querying)

• Full View: A mode of viewing search results that displays the entire content of each result.

(Source: querying)

• Snippet View: A mode of viewing search results that displays a brief summary or excerpt of each result.

(Source: querying)

• Limited View: A mode of viewing search results that restricts the amount of information displayed for each result.

(Source: querying)

• Full View, Snippet View, and Limited View: MEDIUM (Knowledge of these modes is important but not as critical as querying concepts)

(Source: querying)

• \* Google Scholar: freely accessible search engine that indexes the full text or metadata of scholarly literature

(Source: querying)

• \* Peer-to-peer processing: "a way to utilize peer-to-peer networking for distributed computing" ()

(Source: querying)

• \* Distributed computing: not explicitly defined, but implied as a concept related to peer-to-peer processing ()

(Source: querying)

• \*\*Category-based search\*\*: Searching for a category, such as "rock bands" or "german cars", to get related searches and top members.

(Source: querying)

• \*\*Index\*\*: A data structure that enables fast lookup, insertion, and deletion of data in a database.

(Source: querying)

• \* Auto-completion: predicting word or phrase that the user wants to type in without actually typing it in completely

(Source: querying)

• \* Relevance feedback: a form of auto-completion

(Source: querying)

• \* \*\*Autocomplete\*\*: A feature that predicts the next character or word a user is likely to type

(Source: querying)

• 1. \*\*Related search\*\*: A search that is suggested by the autocomplete feature based on the user's input.

(Source: querying)

• \* None explicitly stated, but implied concepts include:

(Source: querying)

• 1. \*\*Bing Auto-Completion\*\*: Bing's feature that suggests possible queries or keywords based on previous searches and the user's input

(Source: querying)

• 2. \*\*Previous Queries\*\*: Searches made by users that are stored and used to make suggestions for future queries

(Source: querying)

• \* Reciprocal rank: the multiplicative inverse of the rank of the first correct answer

(Source: querying)

• 2. DEFINITIONS:

(Source: se-basics)

• \* \*\*Indexing\*\*: The process of creating a database of web pages for searching

(Source: se-basics)

• \* \*\*Query Matching\*\*: The process of matching user queries with relevant web pages

(Source: se-basics)

• \* FTP (File Transfer Protocol) - an anonymous method of transferring files over the internet

(Source: se-basics)

• \* Archie - a tool that assembled lists of files available on many FTP servers

(Source: se-basics)

• \* Veronica - a search tool for text files available through Gopher servers

(Source: se-basics)

• \* Jughead - a search tool for text files available through Gopher servers (not explicitly defined in the slide, but implied)

(Source: se-basics)

• \* The Gopher protocol - a TCP/IP application layer protocol designed for distributing, searching, and retrieving documents over the Internet

(Source: se-basics)

• \* Web crawler: A Perl-based program that crawled the web and collected URLs (marked as )

(Source: se-basics)

• \* Crawling: collecting data from web pages

(Source: se-basics)

• \* Bandwidth: the amount of data transferred over a network

(Source: se-basics)

• \* ALIWEB: an alternative to traditional Web search engines that uses user-submitted meta information for indexing

(Source: se-basics)

• \*\*Overture\*\*: A company that owned AltaVista and was later acquired by Yahoo!.

(Source: se-basics)

• \*\*Yahoo! Search\*\*: The search engine developed by Yahoo! using some of the technology from AltaVista.

(Source: se-basics)

• PPC (Pay-Per-Click): A business model where advertisers pay for each ad click on their site.

(Source: se-basics)

• Syndication: The practice of distributing content, such as paid listings, to multiple websites or portals.

(Source: se-basics)

• \* Spam sites: Websites that are considered irrelevant or unwanted content

(Source: se-basics)

• Web search engine: searches for information on the public Web.

(Source: se-basics)

• Enterprise search engines: searches on intranets (internal corporate networks).

(Source: se-basics)

• Personal search engines: searches individual personal computers.

(Source: se-basics)

• \* Crawler/Spider : Software program that traverses the web, discovers new pages, and updates existing indexes

(Source: se-basics)

• \* Indexer : Component responsible for creating and maintaining search indexes

(Source: se-basics)

• \* Query Processor : Module that processes user queries, retrieves relevant documents, and generates ranked results

(Source: se-basics)

• \* \*\*Corpus\*\*: a collection of web pages built by the spider

(Source: se-basics)

• \* \*\*Inverted index\*\*: an index created by the indexer to speed up queries

(Source: se-basics)

• \* Semi-structured data storage in tables

(Source: se-basics)

• \* Structured data storage in databases (NS)

(Source: se-basics)

• \* Scale of larger than previous text corpora

(Source: se-basics)

• \* \*\*Informational intent\*\*: want to learn about something (~40%)

(Source: se-basics)

• \* \*\*Navigational intent\*\*: want to go to that page (~25%)

(Source: se-basics)

• \* \*\*Transactional intent\*\*: want to do something (web-mediated) (~35%)

(Source: se-basics)

• 1. Stop words: unnecessary words filtered from the query (e.g. "the", "and")

(Source: se-basics)

• \* Actor - a person who performs in plays, films, or television shows

(Source: se-basics)

• \* Filmmaker - a person who creates and produces films

(Source: se-basics)

• \* Golden Globe Awards - awards given to recognize excellence in film and television

(Source: se-basics)

• \*\*Evaluation metrics\*\*: Measures used to assess the performance of a search engine, e.g., precision, recall, F1-score.

(Source: se-evaluation)

• \*\*Relevance\*\*: The degree to which a document is relevant to a user's query.

(Source: se-evaluation)

• \*\*Precision\*\*: The ratio of true positives (relevant documents) to total number of retrieved documents.

(Source: se-evaluation)

• \* Precision/recall: Measures of relevance and recall in information retrieval

(Source: se-evaluation)

• \* Mean Average Precision (MAP): A metric to evaluate the ranking quality of a search engine

(Source: se-evaluation)

• \* Harmonic Mean (HM) and Measure: Metrics for evaluating multiple performance measures

(Source: se-evaluation)

• 1. \*\*Precision\*\*: # (relevant items retrieved) divided by #(all retrieved items)

(Source: se-evaluation)

• 2. \*\*Recall\*\*: # (relevant items retrieved) divided by #(all relevant items)

(Source: se-evaluation)

• \*\*True Positive (tp)\*\*: Correctly identified relevant information

(Source: se-evaluation)

• \*\*False Positive (fp)\*\*: Incorrectly identified irrelevant information

(Source: se-evaluation)

• \*\*True Negative (tn)\*\*: Correctly identified irrelevant information

(Source: se-evaluation)

• \*\*False Negative (fn)\*\*: Incorrectly identified relevant information

(Source: se-evaluation)

• Relevant documents: a set of documents that are relevant to the search query (implied, not explicitly stated)

(Source: se-evaluation)

• Retrieved documents: a set of documents returned by the search system or algorithm (implied, not explicitly stated)

(Source: se-evaluation)

• 1. \*\*Recall\*\*: Not explicitly defined, but implied as a measure of how well a system retrieves relevant documents ()

(Source: se-evaluation)

• 2. \*\*Precision\*\*: Not explicitly defined, but implied as a measure of how accurate the retrieved documents are ()

(Source: se-evaluation)

• \* Arithmetic mean: not explicitly defined, but mentioned as a well-known concept

(Source: se-evaluation)

• \* Geometric mean: the nth root of the product of numbers

(Source: se-evaluation)

• \* Harmonic mean: strongly tends toward the least element of the list, making it useful in search engine results analysis

(Source: se-evaluation)

• 1. \*\*\*\* Precision: (no definition provided, but mentioned as one of the components of F-score)

(Source: se-evaluation)

• 2. \*\*\*\* Recall: (no definition provided, but mentioned as one of the components of F-score)

(Source: se-evaluation)

• 3. \*\*\*\* F-measure: a measure that combines precision and recall

(Source: se-evaluation)

• 4. \*\*\*\* F-score: another name for the F-measure

(Source: se-evaluation)

• \*\*Recall (#/6)\*\*: The number of relevant documents retrieved out of a total of 6 (example used in the slide).

(Source: se-evaluation)

• \*\*Precision (#/4)\*\*: The number of relevant documents retrieved out of a total of 4 (example used in the slide).

(Source: se-evaluation)

• \* Precision: not explicitly defined, but implied as the ratio of relevant documents to total documents ( )

(Source: se-evaluation)

• \* Recall: not explicitly defined, but implied as the ratio of relevant documents retrieved to total relevant documents ( )

(Source: se-evaluation)

• 1. \*\*Average Precision (AveP(q))\*\* : The area under the precision-recall curve, representing the ratio of relevant documents to total documents at different recall levels.

(Source: se-evaluation)

• 2. \*\*Relevance Judgments\*\* : Assessments of which documents are relevant to a particular query.

(Source: se-evaluation)

• \*\*Recall\*\*: The proportion of relevant documents retrieved out of all relevant documents ( exact formula not provided )

(Source: se-evaluation)

• \*\*Precision\*\*: The proportion of relevant documents retrieved out of all documents retrieved ( exact formula not provided )

(Source: se-evaluation)

• \* Assuming precision of zero for a relevant document that never gets retrieved is "reasonable"

(Source: se-evaluation)

• 1. \*\*\*\* Binary assessment: An assessment that categorizes search results as relevant or not relevant (no nuance).

(Source: se-evaluation)

• 2. \*\*\*\* Heavily skewed by collection/authorship: A result that is biased towards a particular collection of documents or author.

(Source: se-evaluation)

• 1. \*\*\*\* DCG: Discounted Cumulative Gain

(Source: se-evaluation)

• 2. \*\*\*\* CG (Cumulative Gain): The sum of graded relevance values of search results.

(Source: se-evaluation)

• 3. \*\*\*\* rel\_i (Graded Relevance): The graded relevance value of the result at position i.

(Source: se-evaluation)

• \*\*Rank\*\*: The position of a document in the search results.

(Source: se-evaluation)

• \*\*Discount factor\*\*: A value used to divide the relevance grade, commonly chosen as log2(rank + 1).

(Source: se-evaluation)

• \* Binary Precision (P): "The relevance of the top-ranked result"

(Source: se-evaluation)

• \* Average Precision (AP): "Relevance to user scanning low-rank results sequentially"

(Source: se-evaluation)

• \* Graded Cumulative Gain (CG): "Information gain from a set of results"

(Source: se-evaluation)

• \* Discount Cumulative Gain (DCG): "Information gain with positional weighting"

(Source: se-evaluation)

• \* Normalized DCG (nDCG): "How close the results are to the best possible"

(Source: se-evaluation)

• 1. \*\*\*\* Recall: ability of a search engine to retrieve all relevant documents from a collection

(Source: se-evaluation)

• 2. \*\*\*\* Precision: measure of how accurate the search results are, often calculated at the top positions (e.g., top 10)

(Source: se-evaluation)

• 1. Raters: Individuals who evaluate search results and search experience.

(Source: se-evaluation)

• 2. General Guidelines: Overarching principles used by raters to evaluate search results.

(Source: se-evaluation)

• 1. \*\*Vital\*\* - A special rating category that requires further review (Section 4.1 of the Rating Guidelines).

(Source: se-evaluation)

• 2. \*\*Useful\*\* - A page that is very helpful for most users.

(Source: se-evaluation)

• 3. \*\*Relevant\*\* - A page that is helpful for many or some users.

(Source: se-evaluation)

• 4. \*\*Slightly Relevant\*\* - A page that is somewhat related to the query, but not very helpful for most users.

(Source: se-evaluation)

• 5. \*\*Off-Topic or Useless\*\* - A page that is helpful for very few or no users.

(Source: se-evaluation)

• 6. \*\*Unrateable\*\* - A page that cannot be evaluated (Section 4.6 of the Rating Guidelines).

(Source: se-evaluation)

• \* Variants: Two different versions of a web page being compared in an A/B test ()

(Source: se-evaluation)

• \* \*\*CIKM (Conference on Information and Knowledge Management)\*\*: An international forum for presentation and discussion of research on information and knowledge management.

(Source: se-evaluation)

• \* \*\*CIKM Conference\*\*: A conference that provides an international forum for presentation and discussion of research on information and knowledge management.

(Source: se-evaluation)

• \* Query log files: records of user interactions with a search engine, including queries submitted, results clicked on, and timestamps

(Source: se-evaluation)

• \* Correlation between clicks and relevance judgments

(Source: se-evaluation)

• 1. \*\*Autocomplete\*\* () - A feature that suggests possible search queries based on a user's input.

(Source: se-evaluation)

• 2. \*\*Directions\*\* () - A type of search result that provides directions to a location.

(Source: se-evaluation)

• 3. \*\*Knowledge Graph traffic\*\* () - The information provided by the Knowledge Graph, such as entity relationships and attributes.

(Source: se-evaluation)

• IR: Information Retrieval

(Source: text\_processing)

• Text classification: classification of documents as relevant or not relevant

(Source: text\_processing)

• Standing queries: periodic search for new documents on a specific topic

(Source: text\_processing)

• \*\*Parser\*\*: A component of NLP that attempts to assign a syntactic structure to a given input sentence (mentioned in the second tweet by @Robertoross)

(Source: text\_processing)

• \*\*Tagger\*\*: A component of NLP that assigns a part-of-speech tag to each word in a sentence (mentioned in the second tweet by @Robertoross)

(Source: text\_processing)

• \*\*Document Representation\*\*: The way in which text documents are represented in a computer.

(Source: text\_processing)

• \*\*Bag of Words Space\*\*: A high-dimensional space where each dimension represents a word in the vocabulary.

(Source: text\_processing)

• \*\*Classification Function\*\*: A function that takes a document as input and outputs its category.

(Source: text\_processing)

• 1. \*\*NA intelligence\*\* ( likely referring to "Natural Language Processing" or "NLP")

(Source: text\_processing)

• 2. \*\*Temporal semantics collection\*\*

(Source: text\_processing)

• 3. \*\*Optimization network\*\*

(Source: text\_processing)

• \* Manual classification: Used by original Yahoo! Directory, Looksmart, about.com, ODP, PubMed

(Source: text\_processing)

• IDE (Integrated Development Environment): a software tool for writing rules for hand-coded rule-based classifiers.

(Source: text\_processing)

• \*\*Document\*\*: A piece of text that is being classified.

(Source: text\_processing)

• \*\*Class (C)\*\*: A fixed set of categories or labels that documents can be assigned to (e.g. Cp, Cy, ..., Cf).

(Source: text\_processing)

• \*\*Training Set\*\*: A collection of labeled documents used to train a classifier.

(Source: text\_processing)

• \*\*Naive Bayes\*\*: a simple and common supervised learning algorithm

(Source: text\_processing)

• \*\*k-Nearest Neighbors (k-NN)\*\*: a simple and powerful supervised learning algorithm

(Source: text\_processing)

• \*\*Support-vector machines (SVMs)\*\*: a newer and generally more powerful supervised learning algorithm

(Source: text\_processing)

• \* Feature (in supervised learning): any sort of characteristic used to describe a document or text

(Source: text\_processing)

• \* URL, email address, punctuation, capitalization, dictionaries, network features: examples of features that can be used in text processing

(Source: text\_processing)

• 1. \*\*Feature Selection\*\*: The process of selecting a subset of relevant features from the original set of features.

(Source: text\_processing)

• Spam filtering (no definition provided)

(Source: text\_processing)

• \* Vector: each document is represented by one component for each term (word)

(Source: text\_processing)

• \* Dimensionality: 10,000+ dimensions or even 100,000+

(Source: text\_processing)

• + "Documents" can be considered as : Input data points or instances in the text classification problem

(Source: text\_processing)

• + "Classes" can be considered as : Categories or labels assigned to documents (e.g., spam/not spam, positive/negative review)

(Source: text\_processing)

• Government

(Source: text\_processing)

• Sci (Science)

(Source: text\_processing)

• Arts

(Source: text\_processing)

• Copyright Ellis Horowitz 2011-2012

(Source: text\_processing)

• \* \*\*D\*\*: the set of all documents that belong to class

(Source: text\_processing)

• \* Centroid/Prototype: A simple representative formed by Rocchio for each class

(Source: text\_processing)

• \* k-neighborhood: The set of the k-nearest neighbors to a document d

(Source: text\_processing)

• \*\*Copyright\*\*: Ownership or rights over original work (relevant for the Voronoi diagram citation).

(Source: text\_processing)

• \* \*\*Testing instance\*\* (an example to be classified)

(Source: text\_processing)

• \* \*\*Database D\*\* (a set of labeled training examples)

(Source: text\_processing)

• \* Atypical example: a single example that is significantly different from the others in its category.

(Source: text\_processing)

• \* Noise: an error in the category label of a single training example.

(Source: text\_processing)

• \* Googlebot: Google's crawler

(Source: web\_crawling)

• \* Yahoo! Slurp: Yahoo's former web crawler (now retired)

(Source: web\_crawling)

• \* Bingbot, Adidxbot, MSNbot, MSNBotMedia, BingPreview: Bing's five crawlers

(Source: web\_crawling)

• 1. \*\*Coverage\*\*: percentage of the web that should be covered

(Source: web\_crawling)

• 2. \*\*Relative Coverage\*\*: comparison of coverage between competitors

(Source: web\_crawling)

• Queue: a data structure that holds URLs to be fetched

(Source: web\_crawling)

• Fetch: retrieving a web page from the internet

(Source: web\_crawling)

• Parse: analyzing the content of a web page

(Source: web\_crawling)

• \* Unseen Web

(Source: web\_crawling)

• \* Seed URLs

(Source: web\_crawling)

• \* Frontier (of crawled pages)

(Source: web\_crawling)

• 1. - Spider traps: dynamically generated pages that can trap crawlers

(Source: web\_crawling)

• 2. - Robots.txt stipulations: rules set by webmasters to control crawling behavior

(Source: web\_crawling)

• 3. - Politeness: avoiding hitting a server too often

(Source: web\_crawling)

• Robotstxt.org: a protocol that defines limitations for web crawlers

(Source: web\_crawling)

• Robots.txt file: a file placed in the root directory to announce crawling requests

(Source: web\_crawling)

• Crawling: the process of a web crawler visiting and retrieving data from websites

(Source: web\_crawling)

• \*\*robots.txt file\*\*: a text file placed in the root directory of a website to control crawling

(Source: web\_crawling)

• \*\*Allow\*\*: specifies URLs that can be crawled (opposite of Disallow)

(Source: web\_crawling)

• \*\*User-agent directive\*\*: a rule that applies to a specific type of robot

(Source: web\_crawling)

• \* Robots.txt file: a text file placed at the root of a website's domain that contains instructions for web crawlers

(Source: web\_crawling)

• \* User-agent: a software program that acts on behalf of a user, such as a web crawler

(Source: web\_crawling)

• 1. \*\*AP(r)\*\*: Not explicitly defined, but it appears to be a measure or score associated with each robot name (e.g., -0.0291).

(Source: web\_crawling)

• 2. \*\*robot.txt\*\*: A file used by web servers to communicate with crawlers and spiders about which parts of the website should not be crawled.

(Source: web\_crawling)

• + Level: The hierarchical structure of web pages, where each page has a level (e.g., level 0 for the starting page, level 1 for its direct neighbors, etc.)

(Source: web\_crawling)

• \*\*At each step move to page down the tree\*\*: This phrase describes the behavior of the Depth-first Search algorithm, where it moves down one level in the tree at each iteration.

(Source: web\_crawling)

• PageRank: an algorithm for determining the value of a page

(Source: web\_crawling)

• BFS (Breadth-First Search): a graph traversal algorithm that visits nodes level by level

(Source: web\_crawling)

• \* Queue (Q): a data structure used to hold URLs to be crawled, with the front of the queue being the next URL to be processed

(Source: web\_crawling)

• \* Inverted index: an index that maps words or phrases to their corresponding documents or webpages

(Source: web\_crawling)

• \* Accrawler: A type of web crawler that efficiently indexes URLs as well as already visited pages.

(Source: web\_crawling)

• \* Web graph: The structure of links between web pages, which is bidirectional and cyclic.

(Source: web\_crawling)

• \* \*\*Relative URL\*\*: A URL that must be completed to form a complete absolute URL.

(Source: web\_crawling)

• \* \*\*Absolute URL\*\*: A complete URL that can be used as is, without any additional processing.

(Source: web\_crawling)

• \* \*\*Terabyte (TB)\*\*: 1 trillion bytes = 1,000 GB

(Source: web\_crawling)

• \* \*\*Petabyte (PB)\*\*: 1 million terabytes = 1,000 TB

(Source: web\_crawling)

• \* \*\*Trie data structure\*\*: a compact digital trie, or prefix tree, used to store and retrieve strings

(Source: web\_crawling)

• \* \*\*Delta-encoded text file\*\*: a method of storing URLs as the difference between consecutive URLs

(Source: web\_crawling)

• \*\*Endmarker symbol\*\*: A special character ($), used to indicate the end of a word.

(Source: web\_crawling)

• \*\*Viterbi algorithm\*\*: Not explicitly defined, but mentioned as having time complexity O(NK).

(Source: web\_crawling)

• \* \*\*Hash\*\*: A unique string of characters generated from a URL, used to identify it.

(Source: web\_crawling)

• \* \*\*Canonicalization\*\*: The process of selecting a single, preferred version of a URL when there are multiple variations.

(Source: web\_crawling)

• \* Scheme: refers to the protocol used in a URL (e.g., http, https)

(Source: web\_crawling)

• \* Host: refers to the domain name or IP address of a website

(Source: web\_crawling)

• \* Percent-encoding triplet: a sequence of characters preceded by a percentage sign (%)

(Source: web\_crawling)

• Spider Trap: A situation where a crawler re-visits the same page over and over again due to unique IDs in URLs.

(Source: web\_crawling)

• Session ID: A unique identifier used to keep track of visitors.

(Source: web\_crawling)

• \* Keyword stuffing: using high frequency of repeated terms to score high on search engines

(Source: web\_crawling)

• \* Cloaking: technique used by spammers to return different page to crawlers than users

(Source: web\_crawling)

• \* Doorway page: a page designed to rank highly for certain keywords but returns commercial page when browser requests it

(Source: web\_crawling)

• \*\*DNS Resolver\*\*: A process that resolves domain names to IP addresses.

(Source: web\_crawling)

• \*\*Client-Server Architecture\*\*: The design pattern where a client requests and receives data from a server.

(Source: web\_crawling)

• \*\*Caching\*\*: Storing frequently accessed data in a temporary storage area to reduce the number of requests made to the original source.

(Source: web\_crawling)

• \* Refresh Strategies: the frequency at which the crawling process is restarted.

(Source: web\_crawling)

• \* Duplicate pages: pages that have been previously crawled and do not need to be recrawled.

(Source: web\_crawling)

• \* Mirror sites: websites that contain identical or similar content as another website.

(Source: web\_crawling)

• DNS lookup: the process of resolving domain names to IP addresses

(Source: web\_crawling)

• DNS caching: storing previously resolved IP-domain name mappings for future use

(Source: web\_crawling)

• Pre-fetching client: making DNS resolution requests while parsing a page

(Source: web\_crawling)

• UDP (User Datagram Protocol): a protocol used for DNS resolution

(Source: web\_crawling)

• \* Thread: the smallest sequence of programmed instructions that can be managed independently by the scheduler

(Source: web\_crawling)

• \* Process: a component of which threads are part

(Source: web\_crawling)

• \* \*\*Parallel crawler\*\*: a process that consists of multiple crawling processes communicating via local network (intra-site parallel crawler)

(Source: web\_crawling)

• \* Incremental update: generally "cheap" due to compression and differential updates

(Source: web\_crawling)

• \* \*\*\*\* Independent strategy: "no coordination, every process follows its extracted links"

(Source: web\_crawling)

• \* \*\*\*\* Dynamic assignment: "a central coordinator dynamically divides the web into small partitions and assigns each partition to a process"

(Source: web\_crawling)

• \* \*\*\*\* Static assignment: "Web is partitioned and assigned without central coordinator before the crawl starts"

(Source: web\_crawling)

• Inter-partition links: Links between different partitions in a web crawling system

(Source: web\_crawling)

• Firewall mode: A method of handling inter-partition links where the process does not follow them

(Source: web\_crawling)

• Cross-over mode: A method of handling inter-partition links where the process follows them and discovers more pages

(Source: web\_crawling)

• Exchange mode: A method of handling inter-partition links where processes exchange URLs

(Source: web\_crawling)

• \* Exchange mode: a method of communication in web crawling where processes exchange information

(Source: web\_crawling)

• \* Batch communication: sending multiple URLs at once from one process to another

(Source: web\_crawling)

• \* Replication: duplicating popular URLs at each process to reduce exchange overhead

(Source: web\_crawling)

• + Selection policy: states which pages to download

(Source: web\_crawling)

• + Re-visit policy: states when to check for changes to the pages

(Source: web\_crawling)

• + Politeness policy: states how to avoid overloading websites

(Source: web\_crawling)

• + Parallelization policy: states how to coordinate distributed web crawlers

(Source: web\_crawling)

• \* \*\*LastModified indicator\*\*: A timestamp indicating the last time a page was modified.

(Source: web\_crawling)

• 1. \*\*\*\* Shadowing: a method of collecting and storing new pages separately from the current database

(Source: web\_crawling)

• 2. \*\*\*\* In-place updating: updating the index by replacing old versions with new ones without separating them

(Source: web\_crawling)

• \* Uniform policy: re-visiting all pages with the same frequency, regardless of change rate ()

(Source: web\_crawling)

• \* Proportional policy: re-visiting pages more often based on their estimated change frequency ()

(Source: web\_crawling)

• \* Sitemap: A list of pages of a web site accessible to crawlers

(Source: web\_crawling)

• \* XML (Extensible Markup Language): Used as the standard for representing sitemaps

(Source: web\_crawling)

• Fully-qualified URL: A URL that includes the domain name and path (e.g., https://example.com/path/to/page).

(Source: web\_crawling)

• \*\*Reverse DNS Lookup\*\*: A process of looking up an IP address in the DNS system to retrieve its corresponding domain name.

(Source: web\_crawling)

• \*\*Forward DNS Lookup\*\*: A process of looking up a domain name in the DNS system to retrieve its corresponding IP address.

(Source: web\_crawling)

• \* Recrawl: Requesting Google to crawl new or updated content on a site.

(Source: web\_crawling)

• \* Googlebot: A program that crawls the web for Google search engine.

(Source: web\_crawling)

• \* Googlebot: a search engine's software agent that crawls the web and indexes content.

(Source: web\_crawling)

• \* DOM (Document Object Model): a hierarchical representation of an HTML document.

(Source: web\_crawling)

• \* Internet penetration rate: The percentage of people in a region who use the internet ()

(Source: web\_serving\_basics)

• \* Region: A geographic area such as North America, Europe and Central Asia, etc. ()

(Source: web\_serving\_basics)

• 1. Baidu - Chinese search engine ()

(Source: web\_serving\_basics)

• 2. Tencent - Chinese holding company of internet properties ()

(Source: web\_serving\_basics)

• 3. Sohu.com Inc. - Chinese online media and community service provider ()

(Source: web\_serving\_basics)

• 4. comScore - a company that tracks website traffic and user behavior ()

(Source: web\_serving\_basics)

• Y/Y growth rate: a measure of change from one year to another, calculated as the percentage increase or decrease over the same period.

(Source: web\_serving\_basics)

• 1. \*\*\*\* WeChat: a messaging and social media app in China

(Source: web\_serving\_basics)

• 2. \*\*\*\* Tencent Video: an online video streaming service owned by Tencent

(Source: web\_serving\_basics)

• 3. \*\*\*\* Baidu Browser: a web browser developed by Baidu

(Source: web\_serving\_basics)

• 4. \*\*\*\* AliPay: a mobile payment platform owned by Alibaba

(Source: web\_serving\_basics)

• \* Zettabyte: 1 zettabyte = 1,024 exabytes

(Source: web\_serving\_basics)

• \* Exabyte: 1 exabyte = 1,024 petabytes

(Source: web\_serving\_basics)

• \* Petabyte: 1 petabyte = 1,024 terabytes

(Source: web\_serving\_basics)

• \* Terabyte: 1 terabyte = 1,024 gigabytes

(Source: web\_serving\_basics)

• \* StatCounter: A tool used to track and analyze website traffic data (not explicitly defined, but implied)

(Source: web\_serving\_basics)

• 1. \*\*\*\* Cumulative unit shipments: the total number of units shipped over a certain period of time (in this case, 12 quarters).

(Source: web\_serving\_basics)

• 2. \*\*\*\* Post-launch: referring to the period after a product is launched in the market.

(Source: web\_serving\_basics)

• \* Mainframe: an early type of computer that served as a centralized system for processing data and applications

(Source: web\_serving\_basics)

• \* Mini: refers to the miniaturization of computers in the 1970s, which led to smaller and more portable devices

(Source: web\_serving\_basics)

• \* Personal Desktop: a type of computer designed for individual use, popularized in the 1980s

(Source: web\_serving\_basics)

• \* Internet Mobile: refers to the shift towards mobile internet access and computing devices (smartphones, tablets)

(Source: web\_serving\_basics)

• 1. Wintel ()

(Source: web\_serving\_basics)

• 2. Amiga ()

(Source: web\_serving\_basics)

• 3. TRS-80 ()

(Source: web\_serving\_basics)

• \* S3: AWS' storage product, used as a proxy for AWS scale/growth

(Source: web\_serving\_basics)

• \* TLD: Top-Level Domain (e.g. .com, .tk, etc.)

(Source: web\_serving\_basics)

• \*\*DEFINITIONS \*\*

(Source: web\_serving\_basics)

• 1. \*\*Content languages for websites\*\*: Refers to the primary language used on a website.

(Source: web\_serving\_basics)

• 2. \*\*Primary language\*\*: The language in which most content is written on a website or spoken by its users.

(Source: web\_serving\_basics)

• \*\*HTML (Hypertext Markup Language)\*\*: A standard markup language used to create structure and content on the web.

(Source: web\_serving\_basics)

• \* Apache Tika toolkit: detects and extracts metadata and text content from documents using existing parser libraries

(Source: web\_serving\_basics)

• \* Indexing technology: used to organize and retrieve data efficiently (e.g., Lucene, Solr)

(Source: web\_serving\_basics)

• \* TLD (Top-Level Domain): e.g., .com, .org, etc.

(Source: web\_serving\_basics)

• \* In-degree and out-degree distribution: refers to the number of incoming and outgoing links in a network

(Source: web\_serving\_basics)

• + Languages in which documents are written: English (55%), French, German, Spanish, Chinese ()

(Source: web\_serving\_basics)

• Yahoo! Directory Search Results: a list of search results from Yahoo!'s directory.

(Source: web\_serving\_basics)

• Umbrella organization for car sharing companies in Europe: an organization that oversees and coordinates the activities of car sharing companies in Europe.

(Source: web\_serving\_basics)

• 1. \*\*Open Directory Project (ODP)\*\*: A collaborative effort to organize the web, started by Netscape (HIGH)

(Source: web\_serving\_basics)

• 2. \*\*Distributed directory\*\*: A system where data is stored and managed across multiple locations or servers (MEDIUM)

(Source: web\_serving\_basics)

• \* Science category in the Open Directory (104,420 entries)

(Source: web\_serving\_basics)

• \* Computers: Computer Science category in the Open Directory (2,971 entries)

(Source: web\_serving\_basics)

• \* Petabytes: a unit of digital information storage (approximately 1 petabyte = 1,000 terabytes)

(Source: web\_serving\_basics)

• \* Seed sites: initial websites used as starting points for web crawling

(Source: web\_serving\_basics)

• 1. \*\*\*\* Deep Web Technologies: Refers to the technologies used to access and navigate the Deep Web.

(Source: web\_serving\_basics)

• 2. \*\*\*\* Tor: A browser designed for anonymity, used to access the Deep Web.

(Source: web\_serving\_basics)

• 3. \*\*\*\* Dark Web: A subset of the Deep Web characterized by illicit activities.

(Source: web\_serving\_basics)

• 1. \*\*Metadata\*\*: Data associated with a video, such as author, title, creation date, duration, coding quality, tags, description, subtitles, and transcription.

(Source: youtube)

• 2. \*\*Video Recognition\*\*: The process of identifying and extracting metadata from a video.

(Source: youtube)

• + CastTV: A web-wide video search engine founded in 2006 (no longer existing)

(Source: youtube)

• + Munax: An all-content search engine that powers both nationwide and worldwide search engines with video search

(Source: youtube)

• + ScienceStage: An integrated universal search engine for science-oriented videos

(Source: youtube)

• \* Subscription video on demand service: a service that allows users to access content for a fee

(Source: youtube)

• \* Major media companies: large corporations involved in the production and distribution of media content (e.g., Disney, Comcast)

(Source: youtube)

• \*\*SRT\*\*: stands for “SubRip Subtitle” file, a common subtitle/caption file format in text format.

(Source: youtube)

• \* Video hosting website: a platform that stores and delivers video content (not explicitly defined in the text, but implied)

(Source: youtube)

• \* Web traffic analysis company: a company that analyzes online traffic patterns (specifically mentioned as Alexa Internet)

(Source: youtube)

• \* Content Distribution Network (CDN): a system for distributing videos worldwide

(Source: youtube)

• \* ContentID system: YouTube's monetization system

(Source: youtube)

• \* Registered user: a user who has created an account on YouTube to upload videos

(Source: youtube)

• \* Channel: a type of account on YouTube that includes uploaded video thumbnails, subscribed members, favorite videos, and friends' lists

(Source: youtube)

• \* YouTuber: an individual with a YouTube channel

(Source: youtube)

• \* Upload status: current state of uploaded video (Select language >)

(Source: youtube)

• \*\*Syndication\*\*: The process of reusing or redistributing content (audio/video) on other platforms

(Source: youtube)

• \*\*Captioning\*\*: Adding text to videos for accessibility and comprehension

(Source: youtube)

• \*\*Embedding\*\*: Incorporating YouTube videos into external websites or platforms

(Source: youtube)

• 1. \*\*Vevo\*\*: A video hosting service that specializes in music videos

(Source: youtube)

• View Count: A measure of the number of views a video has received on YouTube.

(Source: youtube)

• Ranking Algorithm: The method used by YouTube to determine the order and visibility of search results.

(Source: youtube)

• Meta Data: video titles, descriptions, tags that are core ranking factors

(Source: youtube)

• HD (High Definition) videos rank higher than low quality videos

(Source: youtube)

• \* Aspect ratio: the ratio of width to height of a video

(Source: youtube)

• \* YouTube Search: A feature that returns search results based on user queries.

(Source: youtube)

• \* None explicitly mentioned, but "query" can be considered a definition: A query is likely referring to the user's search or request for video recommendations.

(Source: youtube)

• \*\*Co-visitation count\*\*: Number of times two videos are co-watched.

(Source: youtube)

• \*\*Relatedness (r(vi,vj))\*\*: Measure of how related two videos are, based on co-visitation counts.

(Source: youtube)

• \*\*Normalization function (f(v; vj))\*\*: Function that takes into account the global popularity of both seed and candidate videos.

(Source: youtube)

• \* Video-rich snippet

(Source: youtube)

• \* Web search

(Source: youtube)

• CDN: Content Distribution Network

(Source: youtube)

• Server: A computer that provides services or resources to other computers over a network

(Source: youtube)

• \* Unique identifier assigned by YouTube: A fixed-length, 11 character string, base 64 identifier

(Source: youtube)

• \*\*Metadata\*\*: Additional information about a video, such as its title, description, and tags.

(Source: youtube)

• \* \*\*DNS (Domain Name System)\*\*: System for translating domain names into IP addresses

(Source: youtube)

• \* \*\*HTTP GET Request\*\*: Type of request sent to a server to retrieve data

(Source: youtube)

• \* \*\*Front-end Web Server\*\*: Server that handles requests and delivers web content

(Source: youtube)

• \* Anycast namespace: a mechanism that allows packets to be sent to one of multiple destinations.

(Source: youtube)

• \* Unicast namespace: a mechanism that allows packets to be sent to a single destination.

(Source: youtube)

• \* CDN (Content Delivery Network): a system of distributed servers that store and deliver content to users based on geographic proximity

(Source: youtube)

• \* RTT (Round Trip Time): the time it takes for data to travel from a user's device to a server and back

(Source: youtube)

• \* Fingerprinting: a technique used to identify unique characteristics of digital content, such as videos or audio files

(Source: youtube)

• 1. \*\*\*\* Content ID: A system used by YouTube to identify copyright infringement.

(Source: youtube)

• 2. \*\*\*\* Transcoding: The process of converting video into multiple formats.

(Source: youtube)

• 3. \*\*\*\* Spectrogram: A visual representation of audio frequencies over time.

(Source: youtube)

• \* Digitized audio signal: An audio signal converted into a digital format

(Source: youtube)

• \* Amplitude: The magnitude or strength of a sound wave

(Source: youtube)

• \*\*Finite-state transducers\*\*: A mathematical model used for computing hash functions, suggested as the proprietary method used by YouTube (referring to Eugene Weinstein and Pedro J. Moreno's 2007 ICASSP paper).

(Source: youtube)

• \* None mentioned, but Kryder's Law is linked to a Wikipedia article for further information.

(Source: youtube)

• \* TB (Terabyte): a unit of digital information storage, equivalent to 1 trillion bytes (TB = 1,000 GB)

(Source: youtube)

• \* PB (PetaByte): a unit of digital information storage, equivalent to 1 quadrillion bytes (PB = 1,000 TB)

(Source: youtube)

• 1. \*\*Load Balancer\*\*: A network device that distributes incoming requests across multiple servers to optimize performance and prevent bottlenecks.

(Source: youtube)

• 2. \*\*Transcoding Server\*\*: A server that converts and optimizes video files into various formats, accommodating diverse user devices and network conditions.

(Source: youtube)

• 1. \*\*\*\* CDN: A network of servers that cache and serve content to users based on their geographical location.

(Source: youtube)

• 2. \*\*\*\* Load Balancer: A mechanism that distributes incoming traffic across multiple web servers to improve responsiveness, reliability, and scalability.

(Source: youtube)

• 3. \*\*\*\* Transcoding Servers: Specialized servers that convert video formats for smooth playback on different devices.

(Source: youtube)

# 4. Formulas & Algorithms

## Formulas:

• \* \*\*O(log n)\*\*: time complexity for searching in sorted order (Note: no actual formula or equation is given in this content)

(Source: deduplication)

• Euclidean distance: D([X}..-Xn], [¥1.--sYnl) = sqrt(Sum(x;-y;)\*2) i=1...0

(Source: deduplication)

• Jaccard distance: D(x,y) = 1 — SIM(x,y)

(Source: deduplication)

• Cosine distance: (usually represented as an angle between 0 and 180 degrees)

(Source: deduplication)

• \* d(A, B) = 0 if A and B are the same

(Source: deduplication)

• \* d(A, B) ∈ [0, ∞] (distance is in the range [0, infinity])

(Source: deduplication)

• \* s(A, B) = 1 if A and B are the same

(Source: deduplication)

• \* s(A, B) ∈ [0, 1] (similarity is in the range [0, 1])

(Source: deduplication)

• \* d(A, B) = 1 - s(A, B) (relationship between distance and similarity)

(Source: deduplication)

• \* JS(A, B) = size( intersection A ∩ B )/size( union A ∪ B )

(Source: deduplication)

• \* SISau(A, B) = IS(A∪B) = size( ( {C1, C3} intersect {C1, C2, C3, C4} )/( {C1, C3} union {C1, C2, C3, C4})

(Source: deduplication)

• \*\*Jaccard(A,B) = size of (S(A,w) intersect S(B,w)) / size of (S(A,w) union S(B,w))\*\*

(Source: deduplication)

• \*\*Containment(A,B) = size of (S(A,w) intersect S(B,w)) / size of (S(A,w))\*\*

(Source: deduplication)

• \* Jaccard similarity (J(A,B)) = |A ∩ B| / |A ∪ B|

(Source: deduplication)

• \* Jaccard distance = 1 - J(A,B) or {(union B) - (intersect B)}/(union-B)

(Source: deduplication)

• Documents D1 and D2 are near duplicates iff Hamming-Distance(Simhash(D1), Simhash(D2)) ≤ κ

(Source: deduplication)

• 1. V[i] = 1 if bit i of hash is set, otherwise V[i] = -1

(Source: deduplication)

• 2. simhash bit i is 1 if V[i] > 0, otherwise it is 0

(Source: deduplication)

• Replace each 0 by -1; multiply each 1 or -1 by weight (freq), sum these for each column:

(Source: deduplication)

• \* \*\*O(n\*(n-1)/2)\*\*: original runtime calculation for checking all combinations

(Source: deduplication)

• \* \*\*O(log n) + O(n) + O(log n)\*\*: improved runtime calculation using adjacent pair approach

(Source: deduplication)

• \* \*\*Dv = (di1, d12, ..., dim)\*\*: Document D represented as a vector of index terms, where di is the weight of the j-th term in the document.

(Source: info\_retrieval)

• idf, = log, (W/ df)

(Source: info\_retrieval)

• + idf, = log, (W/ df)

(Source: info\_retrieval)

• \* idf = log(N/df) , where N is the total number of documents (1,000,000)

(Source: info\_retrieval)

• \* w = (1 + log(tf))\*log(N/df)

(Source: info\_retrieval)

• Note that there are no mathematical formulas or step-by-step algorithms in this content, so I did not mark anything as or [ALGORITHM]. However, the properties and procedures mentioned may be useful to study for the midterm exam.

(Source: info\_retrieval)

• \* a|~ = √(a\_1^2 + ... + a\_n^2) ( ) - implied formula for calculating vector length

(Source: info\_retrieval)

• \* cos(θ)=d⋅d' / (√(d^2) \\* √(d'^2)) ( ) - explicit cosine formula

(Source: info\_retrieval)

• + sim(d,q) = dsq = ∑ w\_id \\* w\_qi

(Source: info\_retrieval)

• \* sim(d,q) = dsq = ∑ w\_id \\* w\_qi

(Source: info\_retrieval)

• \* (for binary vectors) Hamming distance: size of intersection between query and document terms

(Source: info\_retrieval)

• \* \*\*\*\* Size of vector = size of vocabulary = 7

(Source: info\_retrieval)

• \* \*\*\*\* similarity(D, Q) = inner product of D and Q vectors

(Source: info\_retrieval)

• \* \*\*\*\* sim(D, , Q) = weighted sum of term frequencies

(Source: info\_retrieval)

• CosSim(d, q) = (d · q) / (√(d^2) \* √(q^2))

(Source: info\_retrieval)

• dj = sqrt(Dwj^2)

(Source: info\_retrieval)

• CosSim(D1, Q) = 10 / V(4+9+25)(0+0+44) = 0.81

(Source: info\_retrieval)

• CosSim(D2, Q) = 2 / V(9+4941)(040+44) = 0.13

(Source: info\_retrieval)

• Note that there are no mathematical formulas or algorithms in the provided content, so I couldn't mark any as or [ALGORITHM]. The priority ratings are subjective and based on the assumption that a midterm exam would focus on understanding how an inverted index is created and used.

(Source: inverted\_indexing)

• \* 110100 AND 110111 AND 101111 = 100100 ()

(Source: inverted\_indexing)

• \* The size of the term-document matrix is approximately 500K x 1M = half-trillion elements, but with no more than one billion 1's ()

(Source: inverted\_indexing)

• Note that there are no mathematical formulas or equations related to inverted indexing in this content, so I did not mark any as .

(Source: inverted\_indexing)

• \*\*FORMULAS/EQUATIONS: \*\*

(Source: inverted\_indexing)

• \* O(m+n) operations for merge ()

(Source: inverted\_indexing)

• Note that I did not mark any of the content as since there are no mathematical formulas presented.

(Source: inverted\_indexing)

• Note that the mathematical formulas are minimal in this content, so there is no category.

(Source: inverted\_indexing)

• \* 12 + 34+ 10 \* (150/ 7) = 260.285714 (example math expression)

(Source: querying)

• Note: There are no mathematical formulas or equations in this content, so there's no need to mark any with .

(Source: querying)

• None

(Source: querying)

• \* MRR = ∑(1/rank) / n, where n is the number of queries and rank is the reciprocal rank of each query response

(Source: querying)

• 3. FORMULAS/EQUATIONS:

(Source: se-basics)

• Note: Since there are no mathematical formulas, I didn't mark any as .

(Source: se-basics)

• \* Mean Average Precision (MAP): P(R|q) = 1/N ∑ [p(r) \* rel(r)] where N is the number of relevant documents, p(r) is the precision at rank r, and rel(r) is the relevance of the document at rank r

(Source: se-evaluation)

• \* Harmonic Mean (HM): HM = 2 \* MAP \* Precision

(Source: se-evaluation)

• Precision = tp / (tp + fp)

(Source: se-evaluation)

• Recall = tp / (tp + fn)

(Source: se-evaluation)

• Accuracy = (tp + tn) / (tp + fp + fn + tn)

(Source: se-evaluation)

• Precision = |A| / (|A| + |AN B|)

(Source: se-evaluation)

• Recall = |A| / |AN B|

(Source: se-evaluation)

• \* Geometric mean formula: nth-root( product of numbers ) = √[n] (product of numbers)

(Source: se-evaluation)

• \* Formula for harmonic mean calculation: (1/number 1 + 1/number 2 + ... + 1/number n) / (n-1) = 1/(sum of reciprocals / (n-1))

(Source: se-evaluation)

• 1. \*\*\*\* F = 2RP / (R + P): formula for calculating F-score

(Source: se-evaluation)

• 2. \*\*\*\* Fg = ((α+1)RP / (R + αP)): more general form of F-measure with parameter α

(Source: se-evaluation)

• \*\*Recall = # Relevant items Retrieved / Total # Relevant Items\*\*

(Source: se-evaluation)

• \*\*Precision = # Relevant items Retrieved / Total # Items Retrieved\*\*

(Source: se-evaluation)

• \* (Ranking #1): (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) /6 = 0.78

(Source: se-evaluation)

• \* (Ranking #2): (0.5 + 0.4+0.5 + 0.57 + 0.56 + 0.6) /6 = 0.52

(Source: se-evaluation)

• Reca 0.2 02 04 04 04 06 06 06 08 1.0 5 +.4+.43)/8 = 0.55 ( )

(Source: se-evaluation)

• 1. \*\*MAP Formula\*\* : MAP = (1/n) \\* Σ AveP(q)

(Source: se-evaluation)

• Recall = number of relevant documents retrieved / total number of relevant documents

(Source: se-evaluation)

• Precision = number of relevant documents retrieved / total number of documents retrieved

(Source: se-evaluation)

• 1. \*\*\*\* pcG = ∑ rel\_i / log2(i+1) - log2(j+1)

(Source: se-evaluation)

• 2. \*\*\*\* DCG, = ∑ (rel\_i / log2(z+1)) for z=0 to i-1

(Source: se-evaluation)

• \*\*Discount factor calculation\*\*: log2(rank + 1)

(Source: se-evaluation)

• \*\*Weighting formula\*\*: (weight) = (relevance grade) / (discount factor)

(Source: se-evaluation)

• Note that there are no mathematical formulas (algorithms or equations) in the given content, so there is no category.

(Source: text\_processing)

• \* P(c|d) ≈ #(c)/ for larger k (Note: P(c|d) represents the probability of class c given document d, and #(c) is the number of documents in class c)

(Source: text\_processing)

• Note that there are no mathematical formulas or algorithms in this content, so there is no or [ALGORITHM] category.

(Source: web\_crawling)

• Note that there are no mathematical formulas or examples in this content, so the categories and [EXAMPLE] remain empty. However, since this algorithm is a key concept in web crawling, it's essential to focus on understanding how it works and applying it correctly.

(Source: web\_crawling)

• \*\*Search time for trie\*\*: O(K)

(Source: web\_crawling)

• \*\*Search time for binary search tree\*\*: O(K \\* log N)

(Source: web\_crawling)

• \* No formulas or equations are explicitly stated, so no label.

(Source: web\_crawling)

• Note that there are no mathematical formulas or algorithms in this content, so I did not mark anything with or [ALGORITHM].

(Source: web\_serving\_basics)

• Note that there are no mathematical formulas in the provided content, so I did not mark any as .

(Source: youtube)

• r(vi,vj) = cij / √(ci × cj)

(Source: youtube)

• \* 24TB \* 4x (for profiles) \* 365 days = 35PB/year

(Source: youtube)

• \* 86MB \* 4 (for profiles) \* 1,000,000,000 = 320PB

(Source: youtube)

## Algorithms:

• The process of deduplication involves identifying and removing duplicate data or records, which can be achieved through various methods such as:

(Source: deduplication)

• 1. Deconstructing web page structure by focusing on content blocks

(Source: deduplication)

• \* \*\*Finding the right download for a project\*\*: Start at the project's webpage or resource page, rather than browsing links directly on Apache.org.

(Source: deduplication)

• \* \*\*Smarter Crawling Algorithm\*\*:

(Source: deduplication)

• \* \*\*Better Connectivity Analysis Algorithm\*\*:

(Source: deduplication)

• None explicitly mentioned in the slide content.

(Source: deduplication)

• \* Compute fingerprints using cryptographic hashing

(Source: deduplication)

• \* Syntactic similarity with edit-distance measure

(Source: deduplication)

• 1. Deduplication process using shingling:

(Source: deduplication)

• \* Step 1: Divide text into 3-shingles (sets of three consecutive words)

(Source: deduplication)

• \* Step 2: Calculate hash value for each 3-shingle

(Source: deduplication)

• \* Step 3: Select a subset of hash values that meet certain criteria (e.g., divisible by some number)

(Source: deduplication)

• \*

(Source: deduplication)

• \* Pick a hash size (e.g., 32 bits)

(Source: deduplication)

• \* Initialize an array V of length hash size with zeros

(Source: deduplication)

• \* Break down the input phrase into shingles

(Source: deduplication)

• \* Hash each feature using a normal 32-bit hash algorithm (e.g., MD5 or SHA)

(Source: deduplication)

• \* For each hash, update V[i] accordingly

(Source: deduplication)

• \* If bit i of hash is set, add 1 to V[i], otherwise subtract 1 from V[i]

(Source: deduplication)

• Vector formed by summing weights:

(Source: deduplication)

• \* \*\*Adjacent Pair Approach\*\*:

(Source: deduplication)

• \* Rotate bits (bit shift left and replace lowest order bit with highest order bit) to create new fingerprints

(Source: deduplication)

• \* Sort by fingerprint to identify pairs of identical elements

(Source: deduplication)

• \* \*\*Statistical models\*\*: used to infer meaning from large amounts of data

(Source: info\_retrieval)

• \* \*\*Machine learning models\*\*: used to infer meaning from large amounts of data

(Source: info\_retrieval)

• \* Taking into account the order of words in the query

(Source: info\_retrieval)

• \* Search Process:

(Source: info\_retrieval)

• 2. Stemming

(Source: info\_retrieval)

• Note that there is no explicit mention of "algorithms" or step-by-step processes, so I did not mark anything as . Similarly, there are no concrete examples or case studies mentioned, so I did not mark any content as [EXAMPLE].

(Source: info\_retrieval)

• \* None explicitly mentioned, but the process of normalizing term frequency is implied

(Source: info\_retrieval)

• \* Query-document scoring:

(Source: info\_retrieval)

• \* Normalizing a vector by dividing each component by its length ( )

(Source: info\_retrieval)

• \* To do efficient ranking:

(Source: info\_retrieval)

• 1. Compute cosine similarity score between query and document vectors that contain the query term:

(Source: info\_retrieval)

• 2. Rank documents by score:

(Source: info\_retrieval)

• Note that I didn't include "Search" as an since it's more of a process description rather than a step-by-step procedure. If you'd like to treat it as an algorithm, let me know!

(Source: info\_retrieval)

• \* Partial matching and ranked results: This implies that the algorithm is capable of returning multiple results, ordered by their relevance or importance.

(Source: info\_retrieval)

• \* Boolean model limitations (e.g., requiring term to appear in document)

(Source: info\_retrieval)

• \* Processing Query on Linked Inverted Index: an algorithmic process to retrieve relevant documents from a linked inverted index

(Source: inverted\_indexing)

• \* Distributed Indexing: a distributed processing algorithm for scaling inverted indexes across multiple machines

(Source: inverted\_indexing)

• + For each word, listing all documents and text positions where it occurs

(Source: inverted\_indexing)

• \* Creating an inverted index from a file involves creating a list of words by position, where each entry is a word and its position in the original file

(Source: inverted\_indexing)

• \* To create an inverted file, create a list of positions by word, where each entry is a word and its corresponding positions in the original file

(Source: inverted\_indexing)

• \* [PRIORITY] MEDIUM: Creating an inverted index from a file and creating an inverted file

(Source: inverted\_indexing)

• 1. \*\*\*\* The process of creating an inverted index can be represented as a sparse matrix, where each row corresponds to a term and each column corresponds to a document.

(Source: inverted\_indexing)

• 1. Take the vectors for the relevant documents (e.g., Brutus, Caesar, and Calpurnia) ()

(Source: inverted\_indexing)

• 2. Complement the vectors (implied as necessary for the AND operation) ()

(Source: inverted\_indexing)

• 3. Perform bitwise AND on the complemented vectors to retrieve matching documents ()

(Source: inverted\_indexing)

• Dynamic space allocation process:

(Source: inverted\_indexing)

• [PRIORITY] MEDIUM for Dynamic space allocation process

(Source: inverted\_indexing)

• 1. \*\*Sorting of Inverted Index\*\*: Sorting the list of terms by frequency or other criteria

(Source: inverted\_indexing)

• 2. \*\*Updating of Inverted Index\*\*: Updating the index when new documents are added to the corpus

(Source: inverted\_indexing)

• \*\*ALGORITHMS: \*\*

(Source: inverted\_indexing)

• 1. \*\*Inverted Index Construction Algorithm\*\*: A step-by-step process to create an inverted index from a set of documents ()

(Source: inverted\_indexing)

• \*\*Building an Inverted Index:\*\*

(Source: inverted\_indexing)

• \* Merge the two postings (postings are document ids)

(Source: inverted\_indexing)

• \* Walk through two postings simultaneously in linear time ()

(Source: inverted\_indexing)

• \*\*AND Operator Algorithm\*\*:

(Source: inverted\_indexing)

• \* \*\*Technique of Skip Pointers for merging postings\*\*:

(Source: inverted\_indexing)

• \* Stepping through lists to process postings until a certain point (e.g., processing 8 on each list).

(Source: inverted\_indexing)

• \* Identifying the successor of a posting.

(Source: inverted\_indexing)

• \* Determining the skip successor of a posting based on its position in a lower list.

(Source: inverted\_indexing)

• 1. \*\*Simple Heuristic for Placing Skips\*\*: use sqrt(P) evenly-spaced skip pointers for postings list of length P

(Source: inverted\_indexing)

• 1. Breaking queries longer than 2 words into biword segments

(Source: inverted\_indexing)

• 2. Matching the query to terms in the index

(Source: inverted\_indexing)

• \* Inverted Indexing Algorithm:

(Source: inverted\_indexing)

• \*

(Source: inverted\_indexing)

• 1. \*\*Analyzing N-gram statistics\*\*: The study involved analyzing the frequency distributions of n-grams in English and Chinese texts, but no specific algorithm is mentioned.

(Source: inverted\_indexing)

• 2. \*\*Calculating the ratio of Chinese characters to English words\*\*: Although not explicitly stated as an algorithm, the conclusion that 1.5 Chinese characters correspond to 1 English word can be seen as a result of analyzing n-gram statistics.

(Source: inverted\_indexing)

• \* \*\*Task Assignment\*\*: Break up indexing into sets of parallel tasks, and assign each task to an idle machine from the pool using a master machine as the coordinator.

(Source: inverted\_indexing)

• \* \*\*Fault-tolerance Mechanism\*\*: Implementing mechanisms to handle unpredictable slowdowns or failures in individual machines.

(Source: inverted\_indexing)

• Assigning postings to a sign (assign...-| Master ]-----...)

(Source: inverted\_indexing)

• \* Deletion process:

(Source: inverted\_indexing)

• \* Periodic re-indexing into one main index

(Source: inverted\_indexing)

• 1. \*\*\*\* Building an inverted index involves the following steps:

(Source: inverted\_indexing)

• \* \*\*Advanced Search\*\*: Not explicitly defined as an algorithm, but mentioned as a feature in search engines

(Source: querying)

• \* None explicitly stated, but the example of querying different variations (e.g., "yahoo/news" vs. "Yahoo!News") implies a process or procedure for handling queries.

(Source: querying)

• \* \*\*\*\*

(Source: querying)

• \* Step-by-step process to avoid common query words in URLs and links:

(Source: querying)

• + Example: "pirates of the caribbean"

(Source: querying)

• \* "cere Tec oe rin that Google has" is not a clear algorithm, but it could be interpreted as an informal description of the process. However, it's likely referring to Google's search algorithm, which is a complex system.

(Source: querying)

• \*\*ALGORITHMS \*\*

(Source: querying)

• \*\*Google's reading level classification algorithm\*\*: While not explicitly described, the lecture mentions that Google used a statistical model to classify webpages into different reading levels.

(Source: querying)

• \* The methodology employed by Google Code Search combined trigram indexing with a custom-built regular expression engine.

(Source: querying)

• \* The service supported fast indexed regular expression searches over local code.

(Source: querying)

• \* \*\*Patent search algorithm\*\*: No specific steps mentioned, but it's implied that there is a process for searching patents.

(Source: querying)

• The process of querying is not explicitly described as an algorithm in this slide. However, based on the context:

(Source: querying)

• Search process:

(Source: querying)

• \*\*B-tree Indexing\*\*:

(Source: querying)

• + Spelling corrections algorithms to assist in making guesses

(Source: querying)

• \* The challenge is to search large index or long list of popular queries in very short amount of time so the results pop up while the user is typing

(Source: querying)

• \* Google's autocomplete algorithm

(Source: querying)

• \* Correction and suggestion algorithm for misspelled words

(Source: querying)

• \* None explicitly stated, but implied concept is the process of generating auto-complete suggestions based on user input.

(Source: querying)

• \* \*\*Query Correction Algorithm\*\*: Bing's algorithm for correcting spelling errors in user queries

(Source: querying)

• 4. ALGORITHMS:

(Source: se-basics)

• \* \*\*Indexing and Query Processing\*\*: Not explicitly described, but implied to be a key aspect of search engine development

(Source: se-basics)

• \* \*\*Link-Based Ranking\*\*: Mentioned as a feature of Google's ranking algorithm

(Source: se-basics)

• \* None explicitly mentioned, but the process of crawling the web and collecting URLs can be considered an algorithmic process (marked as )

(Source: se-basics)

• Lycos' indexing process (mentioned as identifying and cataloging documents)

(Source: se-basics)

• \* \*\*Spider's process\*\*:

(Source: se-basics)

• \* \*\*Indexer's process\*\*: creates inverted indexes with various policies (e.g., word stemming, capitalization, support for Unicode)

(Source: se-basics)

• 6. Maintaining a user profile

(Source: se-basics)

• \*\*Ranking Algorithm\*\*: A step-by-step process for ranking documents based on relevance and other factors.

(Source: se-evaluation)

• 2. Take the nth root of the product

(Source: se-evaluation)

• 3. Take the reciprocal of the result

(Source: se-evaluation)

• Note: There are no explicit algorithms or step-by-step processes mentioned in this slide, so I did not mark any content as . Also, there are no concrete examples or case studies provided, so I did not mark any content as [EXAMPLE].

(Source: se-evaluation)

• \*\*Ranking documents with discounting\*\*: Assign high weights to high-ranked documents and low weights to low-ranked documents using the discount factor.

(Source: se-evaluation)

• 5. Evaluate with automatic measure like click-through on first result ()

(Source: se-evaluation)

• \*\*Clickstream analysis\*\*: A process of analyzing the sequence of user clicks on a website to identify patterns, trends, and areas for improvement. (Note: This is not a mathematical formula, but rather an algorithmic process.)

(Source: se-evaluation)

• \* Using clickthrough data to predict preferences between pairs of documents (implied, but not explicitly stated as an algorithm)

(Source: se-evaluation)

• 1. \*\*Search algorithm\*\* () - An algorithm that determines the order and relevance of search results based on user input.

(Source: se-evaluation)

• 2. \*\*Spelling correction algorithm\*\* () - An algorithm that corrects spelling errors in search queries.

(Source: se-evaluation)

• Not explicitly stated, but implied:

(Source: text\_processing)

• \*\*CKY parser\*\*: A bottom-up parsing algorithm for context-free grammars, implemented as an assignment for Stanford's nip-class (mentioned in the third link)

(Source: text\_processing)

• 1. \*\*Training learning planning programming garbage...\*\* ( likely referring to a step-by-step process in machine learning or AI)

(Source: text\_processing)

• The process of creating rules for hand-coded rule-based classifiers, which involves writing specific criteria to classify text.

(Source: text\_processing)

• \* Simplest bag of words view of documents:

(Source: text\_processing)

• 1. \*\*Simple Feature Selection Method\*\*: Use the most common terms for feature selection, but no particular foundation is required.

(Source: text\_processing)

• Normalization of vectors to unit length (no specific steps mentioned)

(Source: text\_processing)

• \* Build surfaces to delineate classes in the space

(Source: text\_processing)

• None explicitly mentioned in this slide content

(Source: text\_processing)

• \* The Rocchio algorithm forms a centroid/prototype for each class

(Source: text\_processing)

• \* Classification is performed by finding the nearest prototype/centroid

(Source: text\_processing)

• \*

(Source: text\_processing)

• + Assign the category of the most similar example in D.

(Source: text\_processing)

• \* 1-Nearest Neighbor (1NN) algorithm: uses only the closest example to make predictions, prone to errors due to atypical examples and noise.

(Source: text\_processing)

• \* k-Nearest Neighbors algorithm: finds examples and returns the majority category of these, typically with an odd value of k to avoid ties.

(Source: text\_processing)

• \* Crawling process:

(Source: web\_crawling)

• + BREADTH-FIRST SEARCH algorithm:

(Source: web\_crawling)

• \*\*Depth-first Search Algorithm:\*\*

(Source: web\_crawling)

• BFS crawling process (although not explicitly described, it's mentioned as a method that brings high-quality pages early)

(Source: web\_crawling)

• Web Crawling Algorithm:

(Source: web\_crawling)

• Focused Crawling: New Approach by S. Chakrabarti et al (algorithm for re-ordering URLs in the queue based on their relevance and frequency of change)

(Source: web\_crawling)

• \* To determine if a URL has already been seen:

(Source: web\_crawling)

• \* To determine if a new page has already been seen:

(Source: web\_crawling)

• HIGH

(Source: web\_crawling)

• 2. Use trie data structure to determine if path/resource is same as one in URL database

(Source: web\_crawling)

• 1. Store each entry as the difference (delta) between current and previous URL

(Source: web\_crawling)

• + Checkpointing: store full URL periodically

(Source: web\_crawling)

• \*\*Viterbi algorithm\*\*: Not explicitly defined, but mentioned as having time complexity O(NK).

(Source: web\_crawling)

• \*\*Greedy algorithm (grep)\*\*: Not explicitly defined, but mentioned as being used to determine if a new URL is in the set.

(Source: web\_crawling)

• \* URL Normalization:

(Source: web\_crawling)

• \*\*Web Crawler Algorithm\*\*:

(Source: web\_crawling)

• \*\*DNS Request-Response Process\*\*:

(Source: web\_crawling)

• 1. DNS caching:

(Source: web\_crawling)

• \* Distribute URL's to threads to guarantee equitable distribution of requests across different hosts

(Source: web\_crawling)

• \* Early Google spider had multiple coordinated crawlers with about 300 threads each, downloading over 100 pages per second in 2010

(Source: web\_crawling)

• \* \*\*Crawl strategy\*\*: a method for determining which pages to crawl next

(Source: web\_crawling)

• \* \*\*\*\* None explicitly stated, but implied that each strategy has its own algorithm for web crawling

(Source: web\_crawling)

• \*

(Source: web\_crawling)

• \*\*Method 1 for Verifying Googlebot's IP Address\*\*

(Source: web\_crawling)

• \*\*Method 2 for Verifying Googlebot's IP Address\*\*

(Source: web\_crawling)

• \* 3-phase process for Googlebot to process web pages with JavaScript:

(Source: web\_crawling)

• \* The process of using voice-based mobile platform front-ends to replace typing, but no specific steps are provided.

(Source: web\_serving\_basics)

• \*\*HTTP Request-Response Cycle\*\*: A step-by-step process of how a client's request is processed by a server, including:

(Source: web\_serving\_basics)

• + Unifies parsers under a single interface

(Source: web\_serving\_basics)

• 3. Identify language they belong to (using N-grams)

(Source: web\_serving\_basics)

• + Automatically opens links on those pages and archives content

(Source: web\_serving\_basics)

• \* Wayback Machine's database growth rate: approximately 100TB of data per month

(Source: web\_serving\_basics)

• 1. \*\*Indexing Algorithm\*\*: Acquiring meta-data associated with the video (e.g., author, title, creation date, duration, coding quality, tags, description).

(Source: youtube)

• The YouTube Recommendation System (no specific details provided, but mentioned as a key concept)

(Source: youtube)

• + Add custom thumbnail

(Source: youtube)

• \* Encoding process: encode into streamable file format for faster video/audio quality (no specific steps provided)

(Source: youtube)

• 1. No specific algorithm mentioned, but "search algorithms" or "optimization techniques" might be relevant ( Potential answer: "Search optimization algorithm")

(Source: youtube)

• \* The process of maximizing watch time through recommendations is an algorithmic approach.

(Source: youtube)

• \* YouTube's recommendation algorithm uses computer algorithms to choose the first result (based on "ee eo" query).

(Source: youtube)

• + Selection ()

(Source: youtube)

• + Subsequent selection of videos (not specified as a separate algorithm, but rather part of the overall process) ()

(Source: youtube)

• To determine related videos:

(Source: youtube)

• \* \*\*YouTube Video Delivery Process\*\*:

(Source: youtube)

• \* Complicated re-direction scheme (not explicitly described, but mentioned as "complicated")

(Source: youtube)

• \* Not explicitly mentioned in the content

(Source: youtube)

• \* The process of checking new video uploads against the Content ID database and flagging copyright violations if a match is found

(Source: youtube)

• 1. \*\*\*\*

(Source: youtube)

• 2. \*\*\*\*

(Source: youtube)

• 3. \*\*\*\*

(Source: youtube)

• 1. \*\*\*\* Content Delivery Network (CDN) process:

(Source: youtube)

• 2. \*\*\*\* Load Balancer process:

(Source: youtube)

# 5. Examples & Case Studies

• 1. \*\*Web Crawling with De-Duplication\*\* - (identifying identical and nearly identical web pages and indexing only a single version to return as search result)

(Source: deduplication)

• 1. \*\*Same page with different URLs\*\*: - Two URLs (http://espn.go.com, http://www.espn.com) can point to the same page due to virtual hosts and distinct URL structures.

(Source: deduplication)

• 2. \*\*Virtual hosts example\*\*: - A website with multiple hostnames sharing the same document folder, but having different domain names.

(Source: deduplication)

• The SCOP database provides an example of deduplication in action, where multiple URLs point to the same page, demonstrating the removal of duplicate data.

(Source: deduplication)

• Two web pages from www.nytimes.com with slight differences in content (ads and photo).

(Source: deduplication)

• Analyzing a web page's DOM structure using HTML tags (e.g., Document, Head, Body)

(Source: deduplication)

• List of countries with Apache mirrors (e.g. "fi", "ge", "gr", etc.)

(Source: deduplication)

• Number of sites in 55 regions (281)

(Source: deduplication)

• Update frequency (4 hours)

(Source: deduplication)

• \*\*Apache Software Foundation projects\*\*: Various software releases listed, including:

(Source: deduplication)

• \* \*\*“If that fails you can try: <mirror>/samepath”\*\*: An example of adding redundancy in result listings

(Source: deduplication)

• \* Identifying related articles describing the same event.

(Source: deduplication)

• \* Extracting and categorizing information from a collection of similar pages (e.g., movie reviews).

(Source: deduplication)

• \* Identifying near-duplicates arising out of revisions, modifications, copying or merging of documents within a domain.

(Source: deduplication)

• \* URL matching using cryptographic hashing

(Source: deduplication)

• \* Detecting identical web pages using hash values

(Source: deduplication)

• Example of how a small change in the input text can produce a major difference in the output hash value (referenced to Wikipedia article: https://en.wikipedia.org/wiki/Cryptographic\_hash\_function)

(Source: deduplication)

• \*\*Verisign\*\*: an example of a certificate authority that uses cryptographic hash functions.

(Source: deduplication)

• \* \*\*Web pages with common prefix\*\*: example of a problem where simple hashing approaches fail (e.g., all web pages start with "<HTML>")

(Source: deduplication)

• \* Consider sets A = {0, 1, 2, 5, 6} and B = {0, 2, 3, 5, 7, 9}

(Source: deduplication)

• \* Suppose we divide our items into four clusters...

(Source: deduplication)

• Shingleing example: "rose is rose is rose" produces a set S(D,w) with 5 items

(Source: deduplication)

• \* Tropical fish text with 3-shingles and their corresponding hash values

(Source: deduplication)

• \* Example of selecting a subset of hash values (e.g., those divisible by some number)

(Source: deduplication)

• \* Testing if two pages are near duplicates using Jaccard similarity (e.g., greater than 0.9)

(Source: deduplication)

• 1. \*\*\*\* Example 1: Three strings "the cat sat on the mat", "the cat sat on mat", and "we all scream for ice cream" are hashed using Ahash and Simhash functions

(Source: deduplication)

• 2. \*\*\*\* Analysis of bitwise Hamming distance between similar items (p1,p2) and dissimilar items (p1,p3) and (p2,p3)

(Source: deduplication)

• Tropical fish text example, illustrating how to create a fingerprint from 8-bit hash values:

(Source: deduplication)

• \*\*Sorting and Deduplication Example\*\*: The given example illustrates how sorting a list of numbers can lead to adjacent pairs with low bitwise Hamming distance. Specifically:

(Source: deduplication)

• \* \*\*Low Hamming distance issue\*\*: a pair with low Hamming distance that ends up apart due to sorting only picking up differences in lower order bits

(Source: deduplication)

• + Hamming distance: same value under rotation and sorting

(Source: deduplication)

• + Hamming distance: same value under rotation and sorting

(Source: deduplication)

• - (2,4), (3,6), (5,8)

(Source: deduplication)

• 1. \*\*\*\* None explicitly mentioned, but the content references a video by Jurafsky and Manning, which may provide examples or case studies.

(Source: info\_retrieval)

• 5. \*\*EXAMPLES\*\* : The list (1. Docl, 2. Doc2, 3. Doc3) could be an example of documents being retrieved, with "Doc" possibly representing a document ID.

(Source: info\_retrieval)

• - \*\*Lexis-Nexis\*\*: http://www.lexisnexis.com/ (MEDIUM [PRIORITY])

(Source: info\_retrieval)

• - \*\*Dialog\*\*: http://www.dialog.com/ (LOW [PRIORITY])

(Source: info\_retrieval)

• - \*\*MEDLINE\*\*: http://www.medlineplus.gov/ (MEDIUM [PRIORITY])

(Source: info\_retrieval)

• \* YouTube's recommendation system

(Source: info\_retrieval)

• \* Amazon's recommendation system

(Source: info\_retrieval)

• \* TREC (Text REtrieval Conference) has had a Question/Answer track since 1999 (http://trec.nist.gov/data/qa.html)

(Source: info\_retrieval)

• \* Web pages are mostly unstructured

(Source: info\_retrieval)

• \* The DOM can provide some clues about web page structure

(Source: info\_retrieval)

• \* Bayesian Network that represents the probabilistic relationships between diseases and symptoms

(Source: info\_retrieval)

• 1. \*\*\*\* Watch "The Beauty of Data Visualization" (18 min) to learn about data visualization.

(Source: info\_retrieval)

• 2. \*\*\*\* Data scientists spend a lot of time collecting and cleaning data, as data is never clean.

(Source: info\_retrieval)

• \* Sample Boolean query with explicit AND, OR, NOT operators: [[Rio & Brazil] | [Hilo & Hawaii]] & hotel & !Hilton]

(Source: info\_retrieval)

• \* Advanced Search example using AND, OR, and NOT operators in Google

(Source: info\_retrieval)

• \* Simple query: "Lincoln" (too many matches)

(Source: info\_retrieval)

• \* More detailed query: "President AND Lincoln" (returns incorrect results)

(Source: info\_retrieval)

• \* Even more detailed query: "president AND Lincoln AND NOT (automobile OR car)" (better, but still not ideal)

(Source: info\_retrieval)

• \* Successful query refinement: "President AND lincoln AND (biography OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)"

(Source: info\_retrieval)

• \*\*Term weights and document representation\*\*: D1 = 2T + 3T + 5T; Q = 0T + 0T + 2T

(Source: info\_retrieval)

• \* Calculations for A, B, and C terms with given frequencies and document frequencies

(Source: info\_retrieval)

• \* \*\*\*\* Binary vector representation: oS Ss ws (example of binary vector with 7 terms)

(Source: info\_retrieval)

• \* \*\*\*\* Weighted vector representations:

(Source: info\_retrieval)

• \* \*\*\*\* Calculating similarity using weighted vectors: sim(D, , Q) = 10 and sim(D, , Q) = 2

(Source: info\_retrieval)

• D1 = [3T, 7T, 1T]; CosSim(D1, Q) = 2 / V(9+4941)(040+44) = 0.13

(Source: info\_retrieval)

• D2 is 6 times better than D1 using cosine similarity but only 5 times better using inner product

(Source: info\_retrieval)

• \* Two-term query "B" may prefer document containing frequently but not B, over document that contains both and B, but both less frequently

(Source: info\_retrieval)

• \* Examples of Inverted Indices: real-world applications or scenarios where inverted indexing is used

(Source: inverted\_indexing)

• \* biwords: an example of using n-grams to improve search results

(Source: inverted\_indexing)

• 1. \*\*\*\* System 1 document with term "ae" and its frequency.

(Source: inverted\_indexing)

• 2. \*\*\*\* Postings list for the term "system 1", which would contain a list of document IDs where this term appears.

(Source: inverted\_indexing)

• - Last entry: "°" at last word

(Source: inverted\_indexing)

• 1. \*\*\*\* The slide provides a concrete example of an inverted index represented as a sparse matrix:

(Source: inverted\_indexing)

• \* Querying multiple documents simultaneously using bitwise AND ()

(Source: inverted\_indexing)

• + Example: 110100, 110111, and 101111 (vectors for Brutus, Caesar, and Calpurnia) ()

(Source: inverted\_indexing)

• \* Antony and Cleopatra, Act Ill, Scene ii: This example illustrates a situation where Agrippa observes Antony's emotional response to the death of Julius Caesar.

(Source: inverted\_indexing)

• \* Hamlet, Act Ill, Scene ii: This example shows Lord Polonius misinterpreting Brutus' actions in relation to the death of Julius Caesar.

(Source: inverted\_indexing)

• Caesar —\_—\_> (example of a term with associated documents)

(Source: inverted\_indexing)

• Document parsing: "Tem Doct and these are saved with the document ces ID i. sequence of (Modified token, peesar : ...)" is an example of how documents are parsed to extract words.

(Source: inverted\_indexing)

• Example of document ID pairing: "(brutus 1, Doc 1), (Caesar was killed Caesar. The noble ——— ' 1 , noble) becomes ((brutus, 1), (doc\_1)), ((caesar, 2), (doc\_2))"

(Source: inverted\_indexing)

• \*\*EXAMPLES: \*\*

(Source: inverted\_indexing)

• The provided example illustrates a simplified inverted index for a small document collection, showing how terms are indexed and their frequencies stored.

(Source: inverted\_indexing)

• + Merged postings: combined document ids for both words

(Source: inverted\_indexing)

• \* Example of merging postings: "If the list lengths are m and n, the merge takes O(m+n) operations" ()

(Source: inverted\_indexing)

• \*\*Query Example\*\*:

(Source: inverted\_indexing)

• \* Processing postings until reaching a specific point, such as getting to 16 on the top list and realizing its successor is 32.

(Source: inverted\_indexing)

• \* Identifying the skip successor of a posting (8 on the lower list) as 31, allowing for skipping ahead.

(Source: inverted\_indexing)

• + friends romans

(Source: inverted\_indexing)

• + romans countrymen

(Source: inverted\_indexing)

• 1. Querying a 4-word phrase (stanford university palo alto) and breaking it down into Boolean queries on biwords

(Source: inverted\_indexing)

• \* \*\*\*\* Document 1 has the following postings for term "be":

(Source: inverted\_indexing)

• \* \*\*\*\* Document 2 has the following postings for term "be":

(Source: inverted\_indexing)

• \*

(Source: inverted\_indexing)

• \* TREC Patent Frequency — Phrase Frequency Phrase\*

(Source: inverted\_indexing)

• 1. Common n-grams are usually made up of stop words (e.g. "and the", "there is")

(Source: inverted\_indexing)

• \* The Google n-gram dataset, which contains 1 trillion tokens, is an example of a large-scale inverted indexing system.

(Source: inverted\_indexing)

• \* The number of bigrams (314,843,401) and trigrams (977,069,902) in the Google n-gram sample are examples of how inverted indexing can be used to analyze text data.

(Source: inverted\_indexing)

• 1. \*\*Study on 200 million randomly sampled English and Chinese Web pages\*\*: The study provides an example of analyzing large text datasets.

(Source: inverted\_indexing)

• 2. \*\*Frequency distributions of uni-grams, bi-grams, 3-grams, and 4-grams in English and Chinese texts\*\*: The study provides examples of frequency distributions for different types of n-grams.

(Source: inverted\_indexing)

• Document corpus splitting example:

(Source: inverted\_indexing)

• The use of posting and inverted indexing for fast searching (no specific example given)

(Source: inverted\_indexing)

• 1. \*\*\*\* Inverted Index for the given document collection:

(Source: inverted\_indexing)

• \* The query "apple AND orchard BUT NOT computer" still returns Apple Computer results

(Source: querying)

• \* Google search result showing unexpected results (image of Google search engine with a red X marked through it)

(Source: querying)

• Entering a Boolean query in Google's Advanced Search page:

(Source: querying)

• \* \*\*Google apple orchard computer\*\*: Example query

(Source: querying)

• \* \*\*sows wer Apple Orchard -Computer\*\*: Example of a modified query with operators

(Source: querying)

• \* \*\*Whally’ Orchard\*\*: Example query result or entity related to an orchard

(Source: querying)

• \* Searching for multiple keywords at once: "disney disneyland pirates" is the same as searching for "disney AND disneyland AND pirates".

(Source: querying)

• \* Searching for the phrase "pirates of the Caribbean" within quotes will show pages that contain the exact phrase.

(Source: querying)

• \* Using the AND operator, searching for "disney" AND "disneyland" AND "pirates of the Caribbean" will show pages that contain all three words.

(Source: querying)

• Query [child bicycle helmet]: finds pages with similar words to search terms

(Source: querying)

• Query with only Stop Words, e.g. [the who]: gets treated as significant and returns relevant results (e.g. the Rock Group)

(Source: querying)

• \* The query [snake grass] finds pages about plants;

(Source: querying)

• \* The query [snake in the grass] finds pages about sneaky people

(Source: querying)

• \* [Red Cross ], [ red cross ], and [ RED CROSS ] return the same results.

(Source: querying)

• \* Querying examples:

(Source: querying)

• 1. \*\*Disney+ Streaming Service\*\*: - An example of a streaming service provided by The Walt Disney Company.

(Source: querying)

• 2. \*\*Database Management\*\*: - Using Disney's database to manage and retrieve data about their streaming service.

(Source: querying)

• \* Example 1: disney disneyland OR “pirates of the caribbean”

(Source: querying)

• 1. abORc dis treated as AND ( OR c) AND

(Source: querying)

• 2. aORb OR dis treated as ( OR b) AND ( OR d)

(Source: querying)

• 3. aOR “ c” dis treated as ( OR (“ c”)) AND

(Source: querying)

• \* Searching for "it' small world" instead of "it's small world"

(Source: querying)

• \* Using quotes to search for exact phrases (e.g. "It' Wonderful World")

(Source: querying)

• \* Effect of stop words on search queries:

(Source: querying)

• \* Using advanced search operators (e.g. daterange:, filetype:) to refine search results

(Source: querying)

• \* \*\*\*\*

(Source: querying)

• \* Example 1: filetype:doc will NOT return .docx files

(Source: querying)

• \* Explanation of how to use filetype: to restrict search results to specific file types, e.g. filetype:doc for Microsoft Word documents

(Source: querying)

• \* Example 1: `[restaurants inanchor:gourmet]` returns pages where anchor text contains "gourmet" and page contains "restaurants."

(Source: querying)

• \* Example 2: `[allinanchor: best museums sydney]` returns only pages where anchor text contains all three words "best", "museums", and "sydney".

(Source: querying)

• \* Example 1: Searching for "Pirates of the Caribbean" with intext: "Disney.com"

(Source: querying)

• \* Example 2: Understanding the difference between searching by text, URLs, and titles

(Source: querying)

• \* Searching with intitle: operator: "intitle:pirates" or "intitle: 'pirates of the caribbean'"

(Source: querying)

• \* Searching for "masters site:cs" to restrict results to Stanford University's Computer Science department

(Source: querying)

• \* Importance of exact matching between `site:` and domain (e.g. "masters site:cs.stanford.edu")

(Source: querying)

• \* pirates site:disney.com

(Source: querying)

• \* pirates -site:disney.com

(Source: querying)

• \* Pirates -site:com

(Source: querying)

• \* Pirates site:edu

(Source: querying)

• \* "Could Improve Brain-Computer Interfaces for People with 24 Disabilities"

(Source: querying)

• \* Using Ctrl+ or ⌘- (Mac) and the find bar to quickly search on a webpage

(Source: querying)

• \* Google's related: feature

(Source: querying)

• \* Various university computer science department websites listed as examples of "related:" results

(Source: querying)

• \* Searching for specific information on a particular web page (illustrating how users interact with the search engine)

(Source: querying)

• \* Google stocks query with "a0pl", "£aQ s#@", and "Co" as stock ticker symbols

(Source: querying)

• \* Using @ symbol to search social media: e.g., @twitter

(Source: querying)

• \* Searching within a price range using $ symbol: e.g., camera $50..$100

(Source: querying)

• \* Searching for synonyms using tilde symbol: e.g., ~car repair

(Source: querying)

• \*\*EXAMPLES \*\*

(Source: querying)

• 1. \*\*Google Phonebook Operators Example\*\*: - Different Google phonebook search operators (phonebook, rphonebook, bphonebook) explained.

(Source: querying)

• \*\*Ovarian cancer information webpage\*\*: The lecture uses this example to demonstrate how the feature classifies content into different reading levels.

(Source: querying)

• \* In 2009, Google introduced the Wonder Wheel as a flash-based interface that provided possible interpretations for search queries.

(Source: querying)

• \* The Wonder Wheel was removed in 2011 but restored in 2012 with a renaming of the "wheel of possible interpretations".

(Source: querying)

• \* Searching for open-source code using the "file:" operator to match common file extensions for a language.

(Source: querying)

• \* Using regular expressions in queries to search for specific patterns in code.

(Source: querying)

• \*\*Google Patents Page example\*\*: The slide shows an image of a Google Patents page with various patent results. This could be considered an example of how to use the patent search system.

(Source: querying)

• \* 0 & Secure itps:/fchola google.com #2 (Note: This appears to be an example URL, but its relevance and importance are unclear without further context)

(Source: querying)

• \* Google's use of peer-to-peer processing (mentioned as an example) ()

(Source: querying)

• + Storage Device Evolution: General-Purpose Peer Processing ()

(Source: querying)

• \*\*Related Searches for German Cars\*\*: If you search for "german cars", Google will show the most popular members of that category, e.g. Audi, Volkswagen, BMW, etc.

(Source: querying)

• \*\*Related Searches for Rock Bands\*\*: If you search for "rock bands", related searches will include top rock bands such as Metallica and Led Zeppelin.

(Source: querying)

• \* Yahoo!'s 3rd result for query "jaguar" is Bing, indicating a ranking or algorithmic decision.

(Source: querying)

• \* Extensive ads for the car at the top and side of search results indicate that searches for the animal are rarer.

(Source: querying)

• \*\*Query Example\*\*: A user searches for all customers in a database who live in a specific city. The query optimizer uses an index on the customer's address to quickly locate the relevant data.

(Source: querying)

• \* Browser filling in your name, address and/or email in a form

(Source: querying)

• \* Search engines using past history, phonetic Soundex algorithms, and spelling corrections algorithms to assist in making guesses

(Source: querying)

• \* Google's auto-complete suggesting "Anniversary Gifts" when typing "A"

(Source: querying)

• \* Auto-complete offering different possibilities after entering the second character

(Source: querying)

• \* Examples of websites with autocomplete features (e.g., Google Maps, Amazon)

(Source: querying)

• + "amazon kindle"

(Source: querying)

• + Squiggly red line under "anaheim duks" suggesting the correct spelling "Anaheim Ducks"

(Source: querying)

• 1. \*\*Autocomplete example\*\*: When searching for "Mikhail Gorbachev", the search engine also suggests related searches for "Mike Gorbachev" and displays links to relevant profiles on social media platforms.

(Source: querying)

• \* Yahoo's auto-complete system running out of alternatives after a certain number of characters are typed.

(Source: querying)

• \* Google's auto-complete system starting to provide suggestions from the first character typed.

(Source: querying)

• \* Examples of websites (e.g., "mad mike garbage pail kids") that have implemented auto-complete systems.

(Source: querying)

• \* Yahoo search results for "Gorbachev" vs. "gorbachev" (mark with )

(Source: querying)

• \* Search results for "Mike Gorbachev" (mark with )

(Source: querying)

• 1. \*\*Example of Bing Auto-Completion\*\*: After entering "mike garbac", Bing finally comes up with the correct spelling, showing how it uses previous queries to improve search results

(Source: querying)

• \* \*\*Bing Search Results\*\*:

(Source: querying)

• \* \*\*Corrected Spelling\*\*: Bing offers corrected spelling suggestions for user queries

(Source: querying)

• \* Example 1: A system tries to translate English words to their plurals, with three sample queries and their corresponding results (cats, tori, virus)

(Source: querying)

• 5. EXAMPLES:

(Source: se-basics)

• \* \*\*Early Search Engines (1991-1994)\*\*: List of specific early search engines (Gopher, Archie, Veronica, Wanderer, ALIWeb, Excite, etc.)

(Source: se-basics)

• \* \*\*Successful Search Engines (1995-2000s)\*\*: List of successful search engines that emerged in the late 1990s and early 2000s (Infoseek, Metacrawler, SavvySearch, LookSmart, Inktomi, HotBot, AskJeeves, Goto)

(Source: se-basics)

• \* In 1990, Alan Emtage, P. Deutsch, et al of McGill Univ. developed Archie

(Source: se-basics)

• \* Veronica and Jughead were developed in 1993 to search names of text files available through Gopher servers

(Source: se-basics)

• \* Excite came from the project Architext, which was started in February 1993 by six Stanford undergrad students

(Source: se-basics)

• \* Excite was bought by @Home for $6.5 billion and later filed for bankruptcy

(Source: se-basics)

• \* World Wide Web Wanderer as a concrete example of an early web robot (marked as )

(Source: se-basics)

• \* Submission of web pages by users with their own page descriptions using ALIWEB

(Source: se-basics)

• \* Use case where people did not know how to submit their site, highlighting a limitation of ALIWEB

(Source: se-basics)

• \*\*AltaVista's features\*\*: The slide describes several examples of AltaVista's innovative features, such as natural language queries, advanced searching techniques, and inbound link checking.

(Source: se-basics)

• Lycos going public with a catalog of 54,000 documents on July 20, 1994

(Source: se-basics)

• Lycos reaching 394,000 documents in August 1994

(Source: se-basics)

• Lycos indexing over 60 million documents by November 1996

(Source: se-basics)

• \* \*\*Yahoo's early days\*\*: The story of David Filo and Jerry Yang posting web pages with links on them in 1994.

(Source: se-basics)

• \* \*\*Yahoo's decline and acquisition\*\*: The fact that Yahoo was purchased by Verizon in 2017 for $4.48 billion after years of decline.

(Source: se-basics)

• WiseNut search engine was bought by LookSmart in March 2002, but failed to gain traction.

(Source: se-basics)

• \* Inktomi's database of spam sites was accidentally made public in 2001, containing over 1 million URLs

(Source: se-basics)

• \* Inktomi sold out to Yahoo! for approximately $235 million in December 2003

(Source: se-basics)

• \* The launch of Ask Jeeves in 1997 as a natural language search engine

(Source: se-basics)

• \* The use of DirectHit technology by Ask Jeeves, which proved vulnerable to spam

(Source: se-basics)

• \* The release of Teoma search engine and its clustering technology

(Source: se-basics)

• \* Google

(Source: se-basics)

• \* Overture (goto.com)

(Source: se-basics)

• \* Multi-billion dollar business

(Source: se-basics)

• \* Strains the boundaries of trademark and intellectual property laws

(Source: se-basics)

• \* N/A (no concrete examples provided on this slide)

(Source: se-basics)

• \* Dynamic generation of content on the web

(Source: se-basics)

• \* Growing and expanding nature of web content

(Source: se-basics)

• \* Chrome conflating search bar with URL address field

(Source: se-basics)

• \* \*\*Informational intent example\*\*: Low hemoglobin

(Source: se-basics)

• \* \*\*Navigational intent example\*\*: United Airlines

(Source: se-basics)

• \* \*\*Transactional intent examples\*\*:

(Source: se-basics)

• \* - Customizing bookmarks (e.g., "weather los angeles ca- 6")

(Source: se-basics)

• \* George Clooney's career as an actor and filmmaker

(Source: se-basics)

• \* The search engine results page (SERP) for "George Clooney" showcasing various news articles, images, and videos

(Source: se-basics)

• \* Sheraton Times Square Hotel website example: illustrating main pages, features, and amenities.

(Source: se-basics)

• \* Sofitel New York Hotel in Manhattan near Times Square: another hotel website example.

(Source: se-basics)

• \* Viewing search history on Google Web History page

(Source: se-basics)

• \* Filtering search history by category (e.g., Today's searches, Yesterday's searches)

(Source: se-basics)

• \* Yahoo's revenue fluctuations from 2013 to 2021

(Source: se-basics)

• \* Baidu's revenue growth from 2014 to 2021

(Source: se-basics)

• \* Google's sweetheart deal with Apple

(Source: se-basics)

• \* Other search engines like DuckDuckGo relying on Bing's index

(Source: se-basics)

• \* Searching for:

(Source: se-evaluation)

• 1. Viewing multiple pages of Google results often does not improve precision at all ()

(Source: se-evaluation)

• \* Calculate the arithmetic mean for the numbers 3, 6, 9, and 12: (3+6+9+12)/4 = 7.5

(Source: se-evaluation)

• \* Calculate the geometric mean for the numbers 3, 6, 9, and 12: nth-root(1944) = 6.64

(Source: se-evaluation)

• 2. Take the reciprocal of the result: 1/.17 = 5.88

(Source: se-evaluation)

• \*\*Bing vs. Google\*\*: An example used to compare the recall and precision of two search engines.

(Source: se-evaluation)

• \*\*3/6 vs. 1/2\*\*: Examples used to illustrate different ratios for recall and precision.

(Source: se-evaluation)

• + Recall scores: 0.17, 0.17, 0.33, 0.5, 0.67, 0.83, 0.83, 0.83, 0.83, 1.0

(Source: se-evaluation)

• + Precision scores: 1.0, 0.5, 0.67, 0.75, 0.8, 0.83, 0.71, 0.63, 0.56, 0.6

(Source: se-evaluation)

• 1. \*\*\*\* Click-through on first result as a non-relevance-based measure of evaluation

(Source: se-evaluation)

• 2. \*\*\*\* Studies of user behavior in lab settings and A/B testing

(Source: se-evaluation)

• 1. The document provides examples and guidelines for evaluators to rate search results.

(Source: se-evaluation)

• \* : search results (HIGH)

(Source: se-evaluation)

• \* : 118,812 people are shown two different sets of search results and asked which they prefer (HIGH)

(Source: se-evaluation)

• \* Comparing two web pages to see which one gives a better conversion rate ()

(Source: se-evaluation)

• \*\*Case study:\*\* Analyzing click data from an e-commerce website showed that users who clicked on product images were more likely to make a purchase than those who did not.

(Source: se-evaluation)

• \* CIKM 2008 conference held in Napa Valley, California

(Source: se-evaluation)

• \* CIKM 2007 conference held in Lisbon, Portugal, with a best interdisciplinary paper award

(Source: se-evaluation)

• \* CIKM 2009 conference held in Hong Kong

(Source: se-evaluation)

• \* \*\*Case Study: CIKM 2007\*\*: The best interdisciplinary paper award at CIKM 2007 went to Fei Wu and Daniel Weld for Autonomously Semantifying Wikipedia.

(Source: se-evaluation)

• \* \*\*Conference Locations\*\*: Examples of locations where the CIKM conference has been held, such as Napa Valley Marriott Hotel & Spa in California (2008) and Hong Kong (2009).

(Source: se-evaluation)

• 1. \*\*Searching by voice\*\* () - An example of how users can search using voice commands.

(Source: se-evaluation)

• 2. \*\*People Also Ask boxes\*\* () - A feature that displays related questions and answers below the main search results.

(Source: se-evaluation)

• \* Wikipedia link provided for comparison of web search engines (marked as )

(Source: se-evaluation)

• Unrest in the Niger delta region: a specific topic of interest

(Source: text\_processing)

• Google Alerts: modern mass instantiation of standing queries

(Source: text\_processing)

• \*\*Tokenization using PTBTokenizer\*\*: The first tweet by @Robertoross provides an example of tokenizing a text file using the Java class edu.stanford.nlp.process.PTBTokenizer

(Source: text\_processing)

• - A 22-year-old person buying 6 properties using the ebook's methods ()

(Source: text\_processing)

• + The anecdotal claim of buying 6 properties at age 22 ()

(Source: text\_processing)

• \* Accurate when job is done by experts (this is an example of a scenario where manual classification is effective)

(Source: text\_processing)

• News agencies and intelligence agencies use hand-coded rule-based classifiers.

(Source: text\_processing)

• Vendors provide "IDE" for writing such rules.

(Source: text\_processing)

• \*\*Building up data by amateurs\*\*: can refine and improve the training dataset over time

(Source: text\_processing)

• \*\*Commercial systems using mixture of methods\*\*: often combine multiple machine learning techniques to achieve best results

(Source: text\_processing)

• 1. \*\*EXAMPLE\*\*: - A review of a movie (with satirical humor, romantic, etc.) that discusses its characteristics.

(Source: text\_processing)

• \* Some classifiers can't deal with 1,000,000 features

(Source: text\_processing)

• \* Training time for some methods is quadratic or worse in the number of features

(Source: text\_processing)

• Paul Graham's Plan for Spam (http://www.paulgraham.com/spam.html) is an example of how Naive Bayes is used in spam filtering.

(Source: text\_processing)

• None provided in this slide content

(Source: text\_processing)

• \* The example given is not a concrete example, but rather an explanation of the concept

(Source: text\_processing)

• \* The 1NN algorithm is prone to errors due to atypical examples (e.g., a single outlier) and noise in the category label of a single training example.

(Source: text\_processing)

• \* Classes can influence each other - Small changes to one class can have ripple effect

(Source: text\_processing)

• \* List of web crawlers at http://en.wikipedia.org/wiki/Web\_crawler

(Source: web\_crawling)

• \* Google's crawler is called googlebot (http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072)

(Source: web\_crawling)

• \* Yahoo!'s web crawler is/was called Yahoo! Slurp (http://wikipedia.org/wiki/Yahoo\_Search)

(Source: web\_crawling)

• \* Online edition of "Our textbook" (cited from Cambridge UP, 2009)

(Source: web\_crawling)

• Ticketmaster.com/robots.txt: an example of a website's robots.txt file

(Source: web\_crawling)

• \*\*robots.txt example\*\*:

(Source: web\_crawling)

• \* ScienceDirect's robots.txt file (https://www.sciencedirect.com/robots.txt)

(Source: web\_crawling)

• 1. \*\*Table 2: Top 10 favored and disfavored robots\*\*: A concrete example showing the top 10 robot names, their frequencies, and AP(r) values.

(Source: web\_crawling)

• 2. \*\*Figure 2: Most frequently used robot names\*\*: An illustration of the most common robot names in robots.txt files.

(Source: web\_crawling)

• \*

(Source: web\_crawling)

• \* Convert "HTTP://www.Example.com/" to lowercase: http://www.example.com/

(Source: web\_crawling)

• \* Capitalize letters in escape sequences: "%3A" -> "%C2%"

(Source: web\_crawling)

• \* Decode percent-encoded octets of unreserved characters: http://www.example.com/%7Eusername/ -> http://www.example.com/~username/

(Source: web\_crawling)

• \* Remove default port: http://www.example.com:80/bar.html -> http://www.example.com/bar.html

(Source: web\_crawling)

• www.webmasterworld.com/page.php?id=2646844 13484654: An example URL with a unique ID that can lead to a spider trap.

(Source: web\_crawling)

• \* Example of keyword stuffing: "We sell custom cigar humidors. Our frequency custom cigar humidors are handmade."

(Source: web\_crawling)

• \* Explanation of cloaking technique

(Source: web\_crawling)

• \* Google downloads ~50,000 pages/second or a billion+ pages in a day (as estimated in 2021)

(Source: web\_crawling)

• \* Early Google spider's coordinated crawlers with multiple threads

(Source: web\_crawling)

• \* \*\*Scenario 1: Centralized crawler control on LAN\*\*: describes a system where one main crawler controls multiple parallel crawlers running on a local area network

(Source: web\_crawling)

• \* \*\*Scenario 2: Distributed crawling on widely distributed machines\*\*: describes a system where multiple crawlers run on widely distributed machines with or without cross communication

(Source: web\_crawling)

• \* Organizing crawlers by country, region, or available bandwidth as a strategy for managing distributed crawlers

(Source: web\_crawling)

• \* \*\*\*\* No specific example given, but understanding of the three strategies can be applied to various scenarios

(Source: web\_crawling)

• \* Example XML sitemap for a three-page website

(Source: web\_crawling)

• Hreflang tags can be used to indicate alternate URLs for different languages or regions:

(Source: web\_crawling)

• \* The mention of specific examples (e.g., Googlebot Images, AdsBot Mobile Web) can be considered as examples

(Source: web\_crawling)

• 2. \*\*Examples of Specific Crawlers\*\*: Googlebot Images, AdsBot Mobile Web, Googlebot Video, etc.

(Source: web\_crawling)

• \* Creating an empty robots.txt file to prevent "File not found" in website error log

(Source: web\_crawling)

• \* Using "nofollow" meta tag to prevent Googlebot from following any links on a page

(Source: web\_crawling)

• \* Adding "rel='nofollow'" attribute to individual links to prevent Googlebot from following them

(Source: web\_crawling)

• \* Crawl rate example: 5 requests per second.

(Source: web\_crawling)

• \* URL for requesting recrawl: https://developers.google.com/search/docs/advanced/crawling/ask-google-to-recrawl

(Source: web\_crawling)

• \* None explicitly mentioned on this slide, but it's likely that the lecture will provide examples of web trends and measurements later on.

(Source: web\_serving\_basics)

• + Building a web search engine today requires understanding the different dimensions of the web

(Source: web\_serving\_basics)

• + Many early slides come from Mary Meeker, a prominent venture capital firm (Kleiner Perkins Caufield & Byers)

(Source: web\_serving\_basics)

• \* The chart shows the share of population using the internet for different regions ()

(Source: web\_serving\_basics)

• \* The fact that 5.03 billion people use the internet today is an example of the internet's widespread usage ()

(Source: web\_serving\_basics)

• \* Amazon.com ()

(Source: web\_serving\_basics)

• 2. China's growing influence on the internet, with >86% of users outside America ()

(Source: web\_serving\_basics)

• The comparison of China Mobile Internet Users' growth rate between +9% and +8%.

(Source: web\_serving\_basics)

• 1. \*\*\*\* Tencent dominates mobile internet usage in China with 71% market share

(Source: web\_serving\_basics)

• 2. \*\*\*\* WeChat leads with ~200 minutes per user, average QQ

(Source: web\_serving\_basics)

• \* The growth of digital information from documents to pictures to tweets in online information.

(Source: web\_serving\_basics)

• \* Yahoo making a major upgrade to Flickr (+500 million)

(Source: web\_serving\_basics)

• \* Instagram being purchased by Facebook for $1 billion (2010)

(Source: web\_serving\_basics)

• \* Snapchat's photo messaging application developed by Stanford students ($9B valuation)

(Source: web\_serving\_basics)

• The example of YouTube uploading 500 hours of video every minute in February 2020

(Source: web\_serving\_basics)

• \* Q1 2015 to Q2 2022: Period of time considered in the analysis (example of a specific timeframe)

(Source: web\_serving\_basics)

• 1. \*\*\*\* The comparison of iPhone and iPad cumulative unit shipments over 12 quarters post-launch, with specific numbers provided.

(Source: web\_serving\_basics)

• 2. \*\*\*\* The illustration of tablet growth (iPad) being ~3x faster than smartphone growth (iPhone).

(Source: web\_serving\_basics)

• \* 36% Search Engine

(Source: web\_serving\_basics)

• \* Wearables Coming on Strong: an example of a technology cycle accelerating faster than the typical 10-year trend

(Source: web\_serving\_basics)

• 1. Global Market Share of Personal Computing Platforms by Operating System Shipments ()

(Source: web\_serving\_basics)

• \* Amazon AWS = Microsoft Azure = Google Cloud (note: this is more of a comparison than an example)

(Source: web\_serving\_basics)

• \* Google Assistant: An example of a voice-based mobile platform front-end

(Source: web\_serving\_basics)

• \* Nearly 70% of Requests are Natural / Conversational Language: A statistic illustrating the prevalence of natural language processing in user requests

(Source: web\_serving\_basics)

• \* Amazon Echo as an example of a voice assistant

(Source: web\_serving\_basics)

• \* Measuring the Web by various methods, such as number of websites, languages of web pages, rate of change of pages, etc.

(Source: web\_serving\_basics)

• \* The graph showing the growth of websites from 1991 to 2021 is an example of how website numbers have increased over time ()

(Source: web\_serving\_basics)

• \* Tokelau (a country) has a high number of domains in the .tk TLD

(Source: web\_serving\_basics)

• \*\*EXAMPLES \*\*

(Source: web\_serving\_basics)

• 1. \*\*Study by the United Nations\*\*: Examined pages on search engines to identify primary languages, resulting in 0-80% distribution of content languages among websites as of 2014.

(Source: web\_serving\_basics)

• 2. \*\*English language dominance (1996-2008)\*\*: Occupied roughly 80% of web pages during this period.

(Source: web\_serving\_basics)

• \* Google's storage requirement: 24 petabytes per day

(Source: web\_serving\_basics)

• \* Internet Archive's storage size: over 10 petabytes

(Source: web\_serving\_basics)

• + Categories of Content: pornography, spam, mirrors ()

(Source: web\_serving\_basics)

• + Cost of storing 1 petabyte: under $1,000 ()

(Source: web\_serving\_basics)

• European Car Sharing Business and Economy - an example of a structured directory of car sharing companies in Europe.

(Source: web\_serving\_basics)

• EuroNCAP (European New Car Assessment Programme) - aims to provide consumers with realistic and independent assessment of the safety performance of cars sold in Europe.

(Source: web\_serving\_basics)

• 1. \*\*DMOZ's categorization structure\*\*: An example of how DMOZ organizes websites into categories, such as Arts, Reference, Regional, etc. (HIGH)

(Source: web\_serving\_basics)

• 2. \*\*RDF format usage\*\*: An example of how RDF format is used to represent data on the web (LOW)

(Source: web\_serving\_basics)

• \* The Open Directory listing for "Science" and its subcategories

(Source: web\_serving\_basics)

• \* The number of entries in the Science category (104,420)

(Source: web\_serving\_basics)

• \* The Internet Archive has been taking snapshots every two months since 1997

(Source: web\_serving\_basics)

• \* Over 412 billion web pages saved over time

(Source: web\_serving\_basics)

• \* Rapid growth rate of the Wayback Machine's database (approximately 100TB/month)

(Source: web\_serving\_basics)

• 1. \*\*\*\* Medical Records: Example of sensitive information stored in databases, which may be part of the Deep Web.

(Source: web\_serving\_basics)

• 2. \*\*\*\* Competitor Websites: Examples of websites that may provide valuable insights for businesses or organizations.

(Source: web\_serving\_basics)

• 3. \*\*\*\* Social Media: Platforms where individuals and organizations share information, potentially including subscription-based content.

(Source: web\_serving\_basics)

• + CastTV (no longer existing)

(Source: youtube)

• + Munax (released their first version in 2005 and no longer active)

(Source: youtube)

• + ScienceStage (no longer active)

(Source: youtube)

• + Blinkx/RhythmOne (uses speech recognition and visual analysis to process downloaded video)

(Source: youtube)

• \* Vimeo.com, the first to support HD video, focuses on short, arty films

(Source: youtube)

• \* Vevo.com, a joint venture of major music companies, hosts high-quality videos

(Source: youtube)

• \* Dailymotion.com, owned by Vivendi, hosts high-quality videos

(Source: youtube)

• \* Hulu's ownership structure (jointly owned by Walt Disney, 21st Century Fox, Comcast, and Time Warner)

(Source: youtube)

• \* Netflix's evolution from delivering DVDs to developing original content

(Source: youtube)

• \* Recent entry of Disney+ into the market

(Source: youtube)

• \*\*TalkMiner System\*\*: an example of Text Recognition using OCR on video slides to detect words. (see https://www.youtube.com/watch?v=7N6L\_m9LywM)

(Source: youtube)

• \* YouTube generated revenue of $19.8 billion in 2020

(Source: youtube)

• \* The website was ranked as the second most popular site by Alexa Internet in January 2022

(Source: youtube)

• \* Pscrib With 1 million subscribers as a YouTuber

(Source: youtube)

• \* Wop seagull Ellis Horowitz's YouTube channel

(Source: youtube)

• \* Example of a YouTube upload status ("Select language >")

(Source: youtube)

• \*\*Paid Promotion\*\*: A creator specifies that their video contains paid product placement, sponsorships, or endorsements

(Source: youtube)

• 1. Search results for "Katy Perry" on YouTube

(Source: youtube)

• 2. Vevo's role in hosting music videos for artists like Katy Perry

(Source: youtube)

• Computer Science Education Challenges: The video "Computer science education: why does it suck so much and what can be done about it?" highlights the challenges faced by computer science students, including lack of resources and outdated curriculum.

(Source: youtube)

• Successful Online Courses: The MIT 6.0 Introduction to Computer Science course with over 420,000 views demonstrates the effectiveness of online learning platforms in reaching a large audience.

(Source: youtube)

• \* Recent tennis match with short life cycle (example of a video with limited lifespan)

(Source: youtube)

• \* YouTube apps exist for Android and iPhone devices

(Source: youtube)

• \* Examples of other devices that can play YouTube videos:

(Source: youtube)

• \* A search for "computer algorithms" returns approximately 521,000 results.

(Source: youtube)

• \* The article mentioned at the end of the slide ("Algorithms Take Over YouTube's Recommendations...") is an example of how algorithms affect human behavior.

(Source: youtube)

• \* Searching for "tutorial on bitcoin"

(Source: youtube)

• Some sights operate their own CDN, e.g. Google, YouTube

(Source: youtube)

• Third-party companies that offer CDN services such as Akamai, Limelight and Level 3 Communications (now part of Century Link)

(Source: youtube)

• Google's data centers map (https://www.google.com/about/datacenters/inside/locations/index.html)

(Source: youtube)

• Geographic distribution of YouTube video cache locations across the world (Africa, Atlantic Ocean, etc.)

(Source: youtube)

• \* Rare video requested in California: first request came from the Netherlands, but future requests were served from California

(Source: youtube)

• \* YouTube has paid $1 billion to rights holders via Content ID since 2007 (highlighting the effectiveness of the system)

(Source: youtube)

• 1. \*\*\*\* The introduction of YouTube's new version of Content ID, Copyright Match (recent development).

(Source: youtube)

• \* Graph with two dimensions: time and frequency; third dimension represented by intensity or color (no specific example given)

(Source: youtube)

• \*\*98% of copyright claims are automatically resolved\*\*: This stat shows the effectiveness of Content ID, but it's not an example of a process or procedure.

(Source: youtube)

• \* $100 can buy 5000 gigabytes of storage (illustrating exponential growth)

(Source: youtube)

• \* YouTube storing approximately 35 PB of new data every year

(Source: youtube)

• \* Estimated total storage needs of YouTube to date (320 PB)

(Source: youtube)

• 1. The example of YouTube's \*\*Content Delivery Network (CDN)\*\* ensuring seamless streaming for users worldwide.

(Source: youtube)

• 2. The example of \*\*Transcoding Servers\*\* converting and optimizing video files into various formats.

(Source: youtube)

# 6. Study Checklist

* □ Review all key concepts from each lecture
* □ Memorize important definitions
* □ Practice all formulas and algorithms
* □ Work through examples and case studies
* □ Understand relationships between topics
* □ Create your own examples for each concept
* □ Practice applying algorithms step-by-step
* □ Review high-priority topics multiple times