1. **Information Retrieval**
   1. Boolean Retrieval – “exact match” retrieval
   2. Most document vectors are sparse (most documents don’t have most words)
   3. TF-IDF
      1. 1. N = total # docs in collection C
         2. = # docs in C that contain
      2. Log
         1. = term k in document
         2. = freq of term in doc
         3. = total # of docs in collection C
         4. = # docs in C that contain
      3. Normalized to prevent longer docs from being given more weight
   4. Vector Space Similarity
      1. Text, letter

         Description automatically generated
      2. Text

         Description automatically generated
   5. Assessing rank quality
      1. Positives and Negatives
         1. True Positive – Relevant doc returned
         2. False Positive – Irrelevant doc returned
         3. True Negative – Irrelevant doc not returned
         4. False Negative – Relevant doc not returned
      2. Precision – fraction of retrieved docs that are relevant
      3. Recall – fraction of relevant docs that are retrieved
      4. Kendall’s tau – compute fraction of pairwise orderings that are consistent
         1. Diagram, schematic

            Description automatically generated
      5. Mean Reciprocal Rank – proximity of best result to the top of the search
2. **Link Analysis**
   1. PageRank
      1. is usually 0.85, represents probability of randomly surfing to a URL
      2. Normalize after each iteration
      3. Sink Nodes are connected to all input nodes
   2. HITS
      1. ,
      2. init all nodes to 1
      3. score normalize after both update
         1. Divide auth by
         2. Divide hub by
   3. HITS vs PageRank
      1. HITS depends on the initial query for the root and base set
      2. Both can be combined with vector model / tf-idf
   4. Search Engine Optimization
      1. TF-IDF manipulation – focus on a particular topic (spam keyword)
      2. Use text and minimize JS reliance
      3. Avoid links with form queries
3. **Scaling Web Search**
   1. Web Crawlers
      1. Start with seed pages, follow links like BFS
      2. Generate lots of traffic (some sites rate limit to prevent this)
      3. Some sites avoid indexing with robots.txt
   2. Deduplication
      1. Hashing algorithm
         1. Hash document – if it exists in hash table, it’s a duplicate
         2. O(N) with respect to O(1) per lookup
         3. Only works for identical documents
      2. Jaccard Similarity

         2. Compare k-shingles (sequences of k words)
         3. Comparing pairs of docs is
         4. Optimizations
            1. Hash shingles for integer comparison
            2. Select random shingle subset (i.e. select minimum)
            4. Select k samples
            5. Computing hash values is O(N) and set comparison is O(1)
            6. Precompute shingle hashes
      3. Inverted Indexes
         1. Inverted Index construction can take too much memory so use Block Merge (yay EECS 484 :/)
         2. Segment by document
            1. Easy to partition
            2. Easy to add new docs (sites)
            3. If machine fails, queries don’t die but docs can
         3. Segment by term
            1. Hard to partition (uneven)
            2. Hard to add new documents
            3. If machine fails, terms poof but documents do not
4. **Scaling Static Pages**
   1. IaaS – Infrastructure as a Serveice
      1. Benefits: don’t have to manage datacenter or backups, quick to rent more servers
   2. PaaS – Platform as a Service
      1. Data Center Service configures pre-installed software
   3. SaaS – Software as a Service
      1. Rent a web app completely maintained and built by service
   4. Domain Name System (DNS)
      1. Translates domain names into IPs
      2. Forms a tree structure with nodes grouped by authority servers
      3. DNS servers send resource records
      4. DNS clients use UDP to connect
      5. Cache DNS IPs for future use
         1. Throw out resource records whose TTLs expire
      6. IANA (Internet Assigned Numbers Authority)
         1. manages IP allocation and DNS roots
         2. US Government did not like IANA (Jon Postel) and so it became non-profit ICANN with for-profit registrars
      7. Security Vulnerabilities
         1. Remap domains/ DoS
         2. Cache Poisoning
         3. DNSSEC signs DNS entries with asymmetric cryptography
   5. CDNs
      1. Store static files in many locations to allow sites to be sent quickly over a shorter distance
5. **Scaling Dynamic Pages**
   1. Can’t combine multiprocessing, multithreading and asynchronous programming due to overhead
   2. Round Robin DNS
      1. Handles which server responds to requests using load balancer
      2. Load Balancer (Proxy)
         1. Forwards requests to backend servers
         2. Often are run in parallel to prevent bottleneck
   3. Hardware Virtualization
      1. Problems with IaaS
         1. Energy Efficiency
         2. Diverse Environments
         3. Security and Isolation
      2. Host – physical computer running OS
      3. Guest – OS being run
      4. Hypervisor – virtualization software
      5. Hardware emulation – sharing of host resources with guest
      6. Advantages
         1. Opposite costs of IaaS
         2. Scaling
         3. Replication for correctness (Dev = Production)
         4. Replication for disaster recovery (recover VMs)
      7. Disadvantages
         1. Lots of Memory Usage
         2. Slow Start
   4. Containerization
      1. Shared operating system, binaries, and libraries
      2. Containers contain code and dependencies
      3. Containers are stateless
6. **Scaling Storage**
   1. Sharding by Content
      1. Different users/tables are kept in different DBs
      2. Downside is database consistency (db has to communicate with others for joins)
      3. Better for consistency
   2. Database Replication
      1. Multiple copies of a database
      2. Problem is that all copies need to be the same
      3. Better for lots of reads
   3. CAP Theorem
      1. Consistency – read gets most recent write
      2. Availability – every query gets a quick response
      3. Partition tolerance – DB still works when network database nodes go down
      4. You can only have 2 of the above traits
      5. RDBMS prioritize consistency over scaling
      6. Mongo prioritizes scaling over consistency
   4. Media Storage
      1. Usually not dynamically changing, mostly reads
      2. Network File System
         1. Not very scalable and/or fast or fault tolerant
      3. Distributed File System
         1. Optimized for scale and reliability
7. **Recommender Systems**
   1. Data Collection
      1. Explicit (ratings, demographic info)
      2. Implicit (web logs, timing)
   2. User-based collaborative filtering
      1. Recommend things that are highly rated by similar users
      2. Nearest neighbor algorithm
         1. Cosine similarity on vectors
         2. Pearson correlation coefficient (between 1 and -1)
            1. Text

               Description automatically generated
      3. KNN – multiple nearest neighbors
         1. Downside is popular terms gets recommended
         2. Solution is to reweight other user scores
         3. Text, letter

            Description automatically generated
         4. Can use IDF for exclusive terms
      4. Problems
         1. Cold Start
         2. Scalability
         3. Sparsity
   3. Content based Filtering
      1. Recommendations based on past term interactions
      2. Problems
         1. New content does not appear
         2. Service isn’t familiar with a user prototype
   4. Hybrid Filtering – use both content based and collaborative filtering
8. **Ads and Auctions**
   1. Open Auction – everyone sees bids
      1. Ascending Price (English)
         1. Winner pays 2nd Highest bid + increment
      2. Descending Price (Dutch)
   2. Sealed-bid Auction – bids are secret
      1. First-price
         1. Results in underbidding
      2. Second-price (Vickrey)
         1. Winner pays 2nd highest bid
         2. Used in Proxy bidding
         3. Sniper bids raise prices right before sale and turns Proxy bidding into Vickrey
   3. Pay-per click auction
      1. Bidders purchase keywords
      2. If user clicks ad, the bidder pays bid
      3. Overture
         1. First-price, open
         2. Results in lots of bid changes
      4. Google PPC (AdWords)
         1. Sealed-bid, 2nd-price auction
         2. Rank ads by bid amount AND quality
9. **Blockchain**
   1. Distributed store of information with no central authority
   2. Bitcoin
      1. Owner of a bitcoin is a bitcoin address which is a public key (wallet)
      2. Owner of a bit coin cryptographically signs statements for transfers of bitcoin addresses
      3. All nodes must agree on a sequence of transactions and the process of agreeing is called mining