1.Web Search Engines

A search engine is a program designed to help find information stored on a computer system such as the World Wide Web, inside a corporate or proprietary network or a personal computer Elements- Spider = collect web pages recursively \ Indexer - create inverted index / Query Processor - serve query results frontend(query reformulation, word stemming, capitalization,optimization of Booleans), Backend(find matching document, rank documents)

Query Processing- Semantic - query language, removing stop words. Specific query type - personalities, cities, location, context remembering, user profile

2. Web servicing basics

Summary of Recent Trends in Web/Internet Development

Growth in number of users connected, Growth in Smartphone use, Growth in digital data, especially photos and video • Growth in Social Media as an advertising platform • Transition from desktop/laptop use to mobile, Growth in tablet usage over desktops/laptops, Decreased dominance of Microsoft Windows, Move away from server farms to cloud computing, Growth in voice communication with devices. The Apache TikaTM toolkit detects and extracts metadata and text content from various documents using existing parser libraries.Tika is useful for search engine indexing

3.SE Evaluation Metrics

1. Precision = tp/tp + fp(relevant/ total relevant) 2. Recall = tp/tp + fn(relevant/total retrieved) 3. Accuracy of engine (tp + tn) / (tp + fp + fn + tn)

For web applications,Precision is more important than Recall

In a good system, precision decreases as the number of docs retrieved (or recall) increases To find the harmonic mean of a set of n numbers 1. add the reciprocals of the numbers in the set 2. divide the sum by n 3. take the reciprocal of the result

Mean Average Precision (MAP)

§ Some negative aspects– If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero (this is actually reasonable) – Each query counts equally – MAP assumes user is interested in finding many relevant documents for each query – MAP requires many relevance judgments in the document collection

AM = sum/# GM = $\sqrt[n]{a_1 a_2 \cdots a_n}$ HM = (1/a+1/b+1/c…)

$$F_\beta = (\beta^2 + 1)RP/(R + \beta^2 P)$$

mean of the average precision scores for each query

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP(q)}}{Q}$$

Discounted Cumulative Gain - highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result

$$\text{DCG}_\text{p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

Gain is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks • Typical discount is 1/log (rank)

Scale	Metric	Measures	Drawbacks
Binary	Precision (P)	The relevance of the entire results set (gridded results display)	Doesn't account for position
Binary	Average Precision (AP)	Relevance to a user scanning results sequentially	Large impact of low-rank results
Graded	Cumulative Gain (CG)	Information gain from a results set	Same as *Precision* doesn't factor in position
Graded	Discount Cumulative Gain (DCG)	Information gain with positional weighting	Difficult to compare across queries
Graded	normalized DCG (nDCG)	How close the results are to the best possible	No longer shows information gain

A/B testing- Query Logs - • Used for both tuning and evaluating search engines contents of the query log files – User id – Query terms – List of URLs of results, ranks, and whether they were clicked – Timestamp(s)- Use of Query Logs - clickthough data to predict preferences between pairs of document

4.Web Crawling

Complications. Challenges– Handling/Avoiding malicious pages– Even non-malicious pages pose challenges– Maintain politeness – don't hit a server too often. Basic Search Strategies - BFS(FIFO), DFS(LIFO),Heuristically ordering(focused crawler), To determine if a new URL has already been seen - First hash on host/domain name, then Use a trie data structure, URLs are sorted lexicographically and then stored as a delta-encoded text file. Simplest (and worst) algorithm to determine if a new URL is in your set, For N URLs and maximum length K, time is O(NK). Normalization Of URL -> to lower case.Capitalize letters in escape sequences, Decode percent-encoded octets, Remove the default port A spider trap is when a crawler re-visits the same page over and over again. Second generation of spam used a technique called cloaking. A third generation, called a doorway page, contains text and metadata chosen to rank highly on certain search keywords, Cloaking and doorway pages are not permitted according to Google's webmaster suggestions. Distributed Crawling Approaches - centralized crawler,distributed set of crawlers, Independent, Dynamic assignment,Static assignment Policies-> selection policy, revisit policy,revisit policy(Uniform/Proportional), poiteness policy, parallellization

5.Deduplication - the identification of identical & nearly identical web pages & indexing only single version of those

Diff URLs directing to same page -distinct virtual hosts -distinct protocol -distinct pagename/path.Dom=> tree hierarchy

Mirroring - systematic replication of web pages across hosts (single largest cause)Exact duplicates if prevented: smart crawling,better analysis,reduce crawl time.Near duplicates if prevented:clustering,data extraction,plagiarism,spam detection.Duplicate-Solution:fingerprinting using cryptographic hashing. Hash stored in sorted order for log N access.Near Duplicate-Solution: syntactic similarity.properties of CryptoHash:easy,difficult to calculate,small change-different value,2 messages no same hash.ex;MD5,SHA-1,2,3.Simhash or Hamming Distance-compute fingerprint.Simhash is algo to test how similar 2 sets are.It is for near duplicates.1.obtain fingerprint.2 . Similar iff at most k-bits apart.Similar input give similar output(hash value)Shingle:contiguous subsequence of words.Near duplicates are found by comparing fingerprints and finding pairs with high overlap.Higher Jaccard Similarity means pages are near duplicate

6. IR - starts with keyword matching. Simple - query string appears in the doc.

Retrieval Methods: Document representation, query representation, retrieval function

IR Models - Boolean models(Simple AND/OR models), Easy to understand, efficient implementation for normal queries , very rigid, no complex user queries, difficult to rank, difficult to perform relevance feedback

Vector space models(very effective) - TF-IDF (term frequency , inverse(log(document frequency))

Common preprocessing steps - Strip unwanted words, break in to token , stem into root words, remove stopwords, detect common phrases, build inverted index. Similarity - It is possible to rank the retrieved documents in the order of presumed relevance.It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled. Normalised vector better as longer vector dont get more weight.

7.TEXT PROCESSING:-

Standing Queries - hand written text classifiers. Spam Filtering. How? Classification - how to represent and classify- on the basis of 'gamma' - classification function- classifier

1. Manual Classification- difficult, expensive to scale, used by yahoo! Original directory
2. Hand Coded rule based classifiers- high accuracy, need to keep changing rules
3. Supervised- model based. 1) Naive Bayes 2) k-NN 3) SVM
   How to sort on supervised? Feature Selection. Eg- BoW, URL. good for generalization, make runtime smaller, less training time
   Naive Bayes- best bhai, low storage, fast learning, robust. Measure? F1, Recall, etc

Similarity between vectors for the document d_j and query q can be computed as the vector inner product:

$$sim(d_j,q) = d_j \bullet q = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$

Vector Space Representation- Each doc is a vector, one component for each term. Jitna pass - utna relevant, jitna sparse utna irrelevant = good classification. Rocchio - centroid based. Nearest prototype based classification- convex voronoi regions

Nearest neighbor- 1 hi hua toh useless. So k NN best.

8.**INVERTED INDEXING:**

An **inverted index** is a vector containing all distinct words of text in lexicographical order (vocabulary), each word is mapped to a list of all the documents which have that word. Before indexing: 1) Case folding: all uppercase to lowercase. 2) Stemming 3) removing stop words

In-memory sorted dictionary points to an on-disk list of (document_id, term frequency).

Processing a new query: Matrix is sparse: sof or each term in the dictionary, create a postings list of all doc id that contain the term. Linked lists are generally preferred to arrays: 1) Dynamic space allocation 2)Easy insertion 3) pointer overhead is not serious.

Query processing in posting list: 1) extract search terms's postings and merge them (AND or OR). Speed up using skip pointers: Skip postings that will not be part of the search results. How: Added at indexing time to postings lists. Evenly spaced, based on (sqrt(P))

Phrase queries: queries like "stanford university" as a phrase. (<term: doc>) not enough. Biword: Consecutive pair of terms indexed. Solutions: 1) Each biword in the dictionary. Consequences: Biwords expand vocabulary and complicate longer queries. 2) Using Positional Indexes (Store doc ID and positions of each term) and merge according to query. 3) Part-of-Speech Tagging: Groups nouns and functional words for indexing.

N-grams are sequences of n consecutive words. 1) Follow a Zipf distribution 2) requires space.

Distributed Indexing: Use a master machine to assign indexing tasks to idle machines. Use parallel tasks.. Master assigns split to parser which gives (term: doc) and writes into j partitions. Inverter sorts in and writes in posting lists. (similar to mapreduce)

9.YOUTUBE PPT:

A video search engine crawls the web for video content. Indexing using meta-data (author, title date, duration, subtitles, etc).

Subtitles Types: 1) Open: Embedded in video. 2) Closed: Toggle on/off. 3)SDH: Includes sound descriptions. Youtube search engine issues: 1) Video Formats 2) display compatibility 3) Distribution: Utilizes a CDN 4) Monetization: contentID 5) Engagement: recommendation system.

Youtube business model includes Ads. ranking: 1) meta data 2) quality 3) views, likes 4) subtitles.

Youtube recommendation system: Association Rule Mining: Computes co-visitation counts to determine how often videos are watched together.

$$r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$$ Ci: occurrence count for video vi. Makes a directed graph.

Content delivery networks: Distributes video via a network of servers based on user location. **YouTube Video Delivery System**: Challenges: Identifying billions of videos and delivering efficiently. Solutions: Unique video IDs for identification and Google's data centers for distribution.

Upload Process: uploaded to central data center, transcoded into multiple formats, then distributed via CDN. supports multiple resolutions.

Download process: DNS guides the user to the nearest server for video streaming.

YouTube Delivery System Design: Comprises a flat video ID space, multi-layered server organization, and a 3 tiered cache system. Monetizing youtube: legal issues solved by spotting copyright content. Revenue shared by youtube.

ContentID system: detects copyright via audio and video fingerprinting system. Converts audio into a spectrogram and calculates hash for fingerprint. The spectrogram features three dimensions—time, frequency, and amplitude. For video, hash of sample frames. Identifies 99.5% of sound-related copyright issues, and 98% of claims across various media.

10.Query Processing:

Query Box default is AND=> Implicit AND

Google Rules: ignores stop words, query limit 32 words

1) search terms near each other 2) terms in the same order as in your query 3) NOT case sensitive 4) ignores some punctuation and special characters 5) The OR operator is acceptable in Google queries. OR>AND. ab OR c d is treated a AND (b OR c) AND. * is a wildcard character.

filetype: results to files ending in a specific file suffix, e.g ".doc"

inanchor: restrict the results to pages containing the query terms in the anchor text

allinanchor: restricts results to pages containing all query terms ain the anchor text

Failed Google experiments: 1. phonebook operator,rphonebook(residents),bphonebook(business)2. Reading level examples:The feature is based primarily on statistical models built with the help of teachers. Google paid teachers to classify pages for different reading levels, and then took their classifications to build a statistical model. With this model, they can compare the words on any webpage with the words in the model to classify reading levels.

Special content search engines: google patents,google books,google scholar

Auto-completion is a part of relevance feedback Google does automatic completion even after the user enters just the first character. When the second character is entered a totally different set of possibility.Yahoo does after 3rd character Bing does not even wait for anything to be entered,makes use of previous queries and gives suggestions.

Mean Reciprocal Rank measures how well a system ranks correct answers to queries. (average of inverse ranks)

For three queries with the correct answers ranked 3rd, 2nd, and 1st, the MRR would be calculated as MRR=⅓(⅓ + ½ +1) = 0.61