**Information Retrieval Midterm Cheat Sheet**

**IR is** finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)

**General applications** of information retrieval: digital libraries, information filtering, media search, search engine. Other: - Adversarial information retrieval – Automatic document summarization – Cross-lingual retrieval – Document classification • Spam filtering – Question answering – Structured document retrieval – Topic detection and tracking

**Information Retrieval versus Databases: Data** structured vs unstructured, **fields** clear semantics vs no fields, **Queries**: defined vs free text, **recoverability**: critical vs downplayed, **matching**: exact vs imprecise

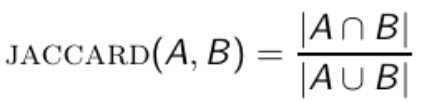
A **token** is an instance of a sequence of characters

**Text processing**: tokenization, normalization, lemmatization, stemming (porter, Lovins, paice), stop words

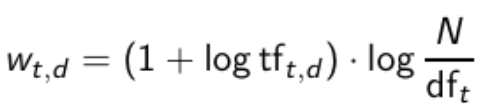
**Index**: skip pointers, positional index (phase and proximity queries), permuterm index, k-gram index

A positional index is 2–4 as large as a non-positional index, 35–50% of volume of original text

**Permuterm query processing**: Rotate query wild-card to the right. Now use B-tree lookup as before. Permuterm problem: ≈ quadruples lexicon size

**Scoring and ranking**

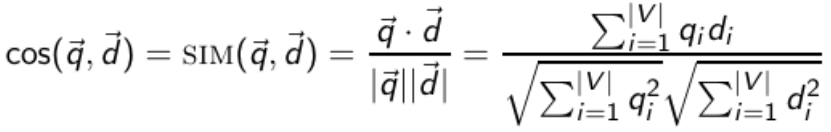
**Jaccard coefficient**:

**Bag of words model**: We do not consider the order of words in a document

dft is the document frequency, the number of documents that t occurs in. N is the number of documents in the collection.

**The Vector Space Model**

Terms are axes of the space. Documents are points or vectors in this space

Euclidean distance is a bad idea, because Euclidean distance is large for vectors of different lengths

**Speed up**: unweighted query terms

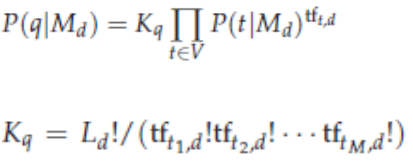
Index elimination: Cosine computation algorithm only considers docs containing at least one query term

Champion lists: Precompute for each dictionary term t, the r docs of highest weight in t’s postings. At query time, only compute scores for docs in the champion list of some query term.

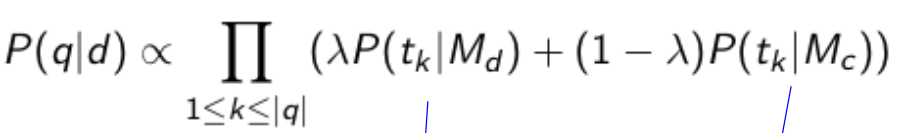
**Relevant and authoritative**

Examples of authority signals: Wikipedia among websites, Articles in certain newspapers, A paper with many citations, Many bitly’s or digs, (Pagerank)

A **zone** is a region of the doc that can contain an arbitrary amount of text, e.g. Title, Abstract, References

**Tiered indexes**: Break postings up into a hierarchy of lists

**Statistical Language Models**



Multinomial + Unigram

**Retrieve based on a language model**: Infer a LM for each document. Estimate P(q|Mdi). Rank the documents according to these probabilities

**LMs vs. vector space model**: Term frequency is directed in the model. Probabilities are inherently “length-normalized”. Mixing document and collection frequencies has an effect similar to idf.

Differences. LMs: based on probability theory. Vector space: based on similarity, a geometric/ linear algebra notion. Collection frequency vs. document frequency. Details of term frequency, length normalization etc.

**LM Assumptions:** Queries and documents are objects of same type. Terms are conditionally independent.

**Basic crawler operation**: Begin with known “seed” URLs. Fetch and parse them. (Extract URLs they point to. Place the extracted URLs on a queue.) Fetch each URL on the queue and repeat.

**Must do:** Be Polite: Respect implicit and explicit politeness considerations (Only crawl allowed pages. Respect robots.txt avoid hitting any site too often). Be Robust: Be immune to spider traps and other malicious behavior from web servers.

**Should do**: Be capable of distributed operation: designed to run on multiple distributed machines. Be scalable: designed to increase the crawl rate by adding more machines. Performance/efficiency: permit full use of available processing and network resources. Fetch pages of “higher quality” first. Continuous operation: Continue fetching fresh copies of a previously fetched page. Extensible: Adapt to new data formats, protocols.

**Processing steps**: Pick a URL from the frontier. Fetch the document at the URL. Parse the URL (Extract links from it to other docs (URLs)). Check if URL has content already seen (If not, add to indexes). For each extracted URL: Ensure it passes certain URL filter tests, Check if it is already in the frontier (duplicate URL elimination)

**Service-oriented architecture (SOA)** is a software design and software architecture design pattern based on distinct pieces of software providing application functionality as services to other applications.

**Searching on anchor text**: Represent each page by all the anchor text pointing to it

Paid placement ranking. Link-based ranking

**PageRank:** If page P with importance x has n outlines, each link gets x/n votes. Page P’s own importance is the sum of the vote on its inlinks.

P = (1- α) \* transition matrix + α \* teleporting matrix

**Web ranking**: PageRank, anchor text, indexing, zone weighting, phrases…