L01: Search Engine (SE)

Names for different retrieval

Information (IR) | Document (DR) | Text (TR)

Specific SE

Vertical	One type of data (Job search, news search)
Site	Just one site
Custom	narrow search to small set of websites
Enterprise	Corporate intranet w/ collections, metadata, roles, security

Federated search organization: Meta search "only" option

Federated search (union)	Meta search (calling hosts)
Each node full-func SE	It itself is not a SE
Agreement obtained	No agreement: may block
Standard query/res repr.	needed and parsing
SEs collaborate	SEs do their own thing

Difficulties of IR

Unstructured text/media	Scale(ds)	Semantics	Diversity, Personalize

Cloud computing advantages (central SE server)

Unlimited scaling	Many replicas and shards
Indexing for pros	
high cost	Inconsistent / inefficient

Timeline of IR gen 0 to 3

1960: exact bool | 1990: stat, crawler | 1997: link | 2010: advanced

L02: Business model (SE)

Software licensing	Sell the software for intranet search
Search service	Outsourced content portals, monetize by ads (e.g. YouTube)
Sponsored	Do the hw/sw/index dirty work for clients
Advertisement	kw-based ads/organic res, popup & banners
Charge end users	refuse to pay unless unique valuable info
Charge websites	by queries, for indexing, for ranking

Notes

kw ads	AdWords (buy kw), AdSense (page contents)
(Content) Portal	SE used to need portals; now SE portals; need ad channels
SEM (Marketing)	SEM company help get visitors, find kw rank high
SEP (placement)	Pay-by-impression (cost per mille = 1000); Pay-per-click
SEO (optimization)	Make web page rank high in algorithm
Trademarked kw	Potential legal problems such as lawsuits

retrieval models

Performance metrics

precision: % relevancy in model (noise in retrieval)	recall: % of relevant docs retrieved (completeness)

model	contains	details
doc repr.	query repr.	retrieval function

Current class of models

bool model	vector space (sim)	probabilistic

L03: Boolean model

Doc repr.	Set of kw (Bag-of-words BoW)
Query repr.	Bool expr of kw (AND, OR, NOT, brackets)
retrieval fn.	satisfy?
pros.	simple, interpretable, efficient, extendable to include ranking
Cons	too rigid, no complex requests; AND/OR/interpretation; no control of docs, no ranking, no auto relevance feedback

L03/L04: Statistical model

Doc. repr	Set of kw w/ stat info; DB = store 1.0, db 0.5, info 0.2; weight = ranking, indicates imp of word; weighted kw as row vector, 0 for missing
Query repr.	kw with optional weights: no bool ops; 1 if present, else 0
retrieval	for similarity
challenges	word importance (in the doc, also coll.); degree of sim b/w doc & q; (many) what is collect?: lots, construct, format info; weight with error

L03: TF-IDF: word importance (within doc TF & entire coll. IDF)

word importance (intra-doc) TF	Principle: repeat = important; Issue: unbounded; fix: bound by division; TF = n/X, # of (unique) words; ... : 0.5 ~ 1
word importance inter-doc, Inverse doc freq (IDF)	principle: rare = discriminant; formula: $\log_e(N/df)$; issue: N=0; fix: log(N/df) ranges from log(N) to log(N+1) ... 0
BM25 weight (OkaPi)	avg doc len affect ranking; param tuning is expensive
TF×IDF	TF & IDF: anti-corr; TF×IDF: max when high local ...

$\ln\left(\frac{N - df + 0.5}{df + 0.5}\right) \cdot \frac{(k+1)tf}{\ldots}$, k = 2, b = 0.75

L04: Vector space model | sim b/w doc & q

sim func

Inner product (sim)	formula: term weights multiply & add together; unbounded, favors long doc
cos sim	Formula: IP / |D| / |Q| (cosine of angle)
Jaccard	Formula: IP / (|D|^2 + |Q|^2 − IP); intersection / union
Dice	Formula: 2IP / (|D|^2 + |Q|^2); × Δ inequality

op on vectors

relevance feedback: Q − D or Q + D

Flexible use

D-Q: SE (typical)	D-D: cluster	Q-Q: suggestion

Disadvantage: reduce ... poor ... assumptions ...

Inner prod unbounded, the ... comparison with ... unmatched terms ... in ...

	Inner	cos	Jaccard
D-D	ok	ok	ok
D-Q	ok	ok	ok

Issues

Term independence not realistic

Synonyms	false negative
unbalanced property	favor unbalanced data, penalize ...
one-time value	Bool → sim

L05: Inverted index

Doc vectors: sparse, ... word list

index: ...

inv. file (doc ...)

Data struct	Hash file, B-tree, tries, etc.
Data	key: term, value: list (doc) postings list

Inv idx

Dict	key: doc, value: list of terms, locations, frequency

Bool query: fetch all postings and perform set ops; opt: AND, start intermediate res

Advantages of inv. idx

filter docs that contain query term for fast retrieval	flexible struct (postings) with info (doc, term)	complex retrieval, e.g. word dist & phrase & proximity
large storage overhead: 50%-100%-300%	high update costs, expensive adds, dels	poor bool, inverted ...

Extensions on inv. index

adj.: phrase, ...	...
term	compression
Suffix/prefix	B+-tree ✓, hash file ✗

overhead in inv. idx on addition

... unique, ... postings lists.

unsorted: fast append, slow retrieval

sorted: slow ins., fast retrieval

insert: by ... × 1.5

speed impact: search, insertion (doc has some words) overhead in ...

p-norm operators

sup: ∧ OR: ∨

$1-\left(\frac{(1-x)^2+(1-y)^2}{2}\right)^{1/2}$ $=\left(\frac{x^2+y^2}{2}\right)^{1/2}$

p-norm: 2 ≥ p

p=1: similar to inner product; p=inf: fuzzy logic

... Pursuit (MIT, 1996)

..., not ... for ranking & ...

Dual Path	$S_{i,j}^{sp} = \frac{1}{2^{[illegible]}} + \frac{1}{2^{[illegible]}}$