* In the context of IR, how do search engines address the challenge of balancing precision and recall.....

In information retrieval, precision and recall are two important matrices —

Precision: refers to the proportion of relevant results among all results that are retrieved by the search engine.

Recall: refers to the proportion of relevant result that are retrieved by the search engine.

How search engine address this balance:

① Query expansion:

→ Adds synonyms or related terms to the user's query.

→ Example: Expanding 'car' to also include 'automobile'.

(2) Relevance feedback:

→ uses user behavior to refine results.

→ improves precision by learning what users consider relevant.

(3) Ranking algorithm:

→ Algorithm like BM25, TF-IDF and neural ranking models sort documents by relevance score.

(4) personalization:

→ uses ~~user~~ user history, location or preferences to rank results.

→ increases precision by tailoring result to user intent.

$$TF = \left(\frac{\text{number of times the term appears in doc}}{\text{Total terms in the doc}} \right)$$

$$IDF = \log\left(\frac{\text{Total Number of doc}}{\text{Number of docs that include the term}} \right)$$

* How do search engine handle dupliate content?

Search engine detect duplicate content using algorithms that analyze content similarity, canonical tags, URL structures and hashing. Only one version (canonical) is usually indexed and ranked to avoid redundancy.

PageRank: A link analysis algorithm by google that ranks web pages based on the number and quality of backlinks.

$$PR(A) = (1-d) + d \times \left(\sum \frac{PR(B)}{L(B)} \right)$$

$PR(A)$ = pageRank of page A

d = dumping factor (0.85)

B = pages linking to A

$L(B)$ = Number of outbound links on page B.

* How do search engine prioritize which page to crawl first. - - - - - ——

Crawling priority factors:

(i) pageRank or authority
(ii) Update frequency
(iii) Sitemaps and robots.txt instruction
(iv) Inbound links
(v) Crawl budget

Determining freshness:

(i) Last modified header

(ii) New or update content

(iii) Timestamps on the page.

(iv) Backlink activity and social signals.

Importance of freshness in ranking:

(i) Fresh content is more likely to be relevant

(ii) Especially critical for time sensitive queries (news, events, trends, articles)

(iii) Helps improve user satisfaction and trust in search results.

Web crawler:

A web crawler (spider / Bot) is a software agent that systematically browses the web to index content for search engines.

Hard focused crawling:

Strictly follows a specific topic, only pages highly relevant to the topic are crawled.

- Ex: Crawling only 'machine learning' articles.

Soft focused crawling:

Allows a broader scope and include pages with partial relevance.

Ex: Crawling pages that mention 'AI' 'deep learning' etc.

Index compression:

A technique used to reduce the size of the inverted index to save storage and increase retrieval speed.

Common methods include dictionary compression and postings list compression.

(variable byte ~~coding~~ encoding, gamma coding)

+ Why is compression needed?

Compression is used to reduce data size so it can be stored and transferred faster and more efficiently. web crawlers handle huge amount of data, so compression helps:-

→ Save storage space
→ Speed up data transfer
→ Improve processing speed.
→ lower costs.

Distributed crawling:

Distributed crawling means using multiple computers to crawl the web at the same time. Each crawler handles a different part of the web, making the process —

→ faster
→ Scalable
→ more reliable.

Google uses many distributed crawlers.

Desktop crawl:

A desktop crawl checks how a website appears on a desktop computer. It uses a desktop browser setting and screen size. To make sure search engines index content as users see it on desktop.

+Explain Zipf's law and give an example to illustrate the law:

Zipf's law is a statistical principle that describes how the frequency of words in a natural language is inversely proportional to their rank in a frequency table.

The most frequent word occurs twice as often as the second most frequent word, three times as often as the third most frequent, and so on.

$$f(r) \propto \frac{1}{r}$$

where, $f(r)$ is the frequency of the word with rank r.

r = rank of the word when sorted by frequency.

Suppose you analyze a book and find the following word frequencies:

Rank	word	frequency
1	the	1000
2	of	500
3	and	333
4	to	250

Here,

'the' is the most frequent word

'of' is the second most frequent word occurs about half as often as "the"

'and' occurs about one third as often as "the".

Zipf's law helps in:-

→ understanding word distribution in documents.

→ Optimizing search engines.

* Heaps' Law:

Heaps' law describes how the nomber of distinct words (vocabulary size) in a document collection grows with the size of the collection.

It shows that, as you process more text, the vocabulary size increases, but at a slowere scale.

mathematical formula:

$$V(n) = K \cdot n^{\beta}$$

- $V(n)$ = number of unique words in document
  size of n (total words)

- K and β are constants,
  typically, $30 \leq K \leq 100$, and $0.4 \leq \beta \leq 0.6$

Example:

Suppose you are analyzing documents

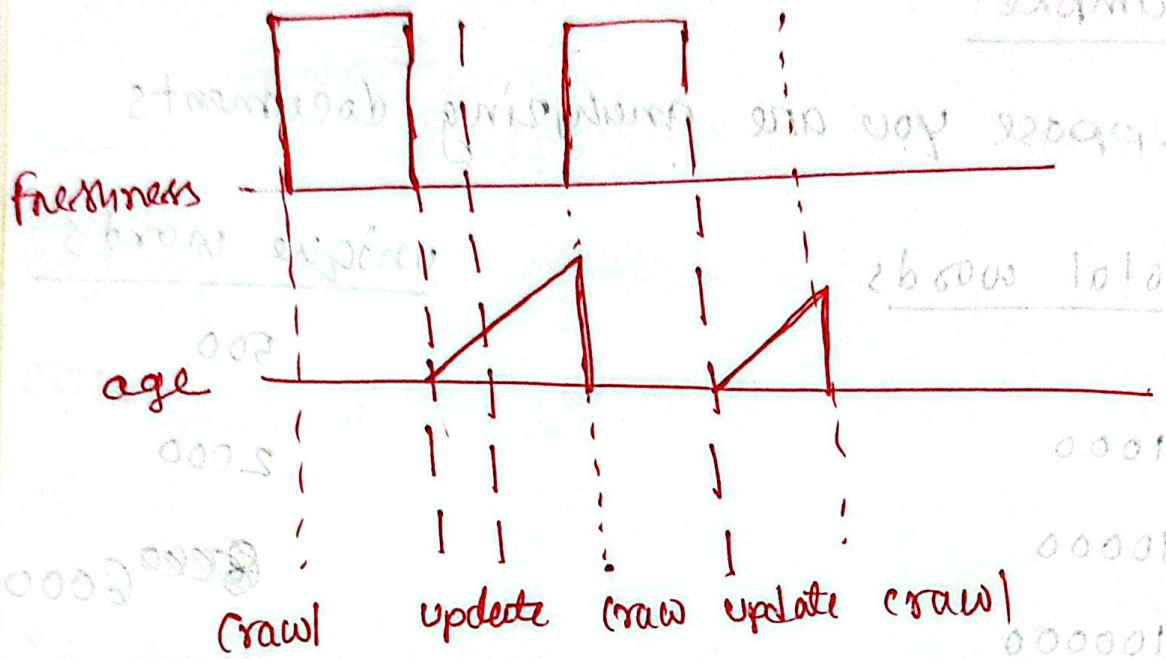
Total words	unique words
1000	500
10000	2000
100000	6000

As the total number of words increase by 100x, the vocabulary only grows by about 12x.

Heaps' law is useful for:-

→ Estimating the size of index for large text collection.

→ planning storage and processing resources in search engine.



* what are the main challenges in tokenizing?

Tokenization means breaking text into small pieces, usually words, so that a search engine can understand and index them.

Challenges:

① pantuation confusion: It's hard to tell that if pantuation is part of the word or not

U.S.A — should it be one word or three?

don't → one word or two (do and not)

(2) No spaces between words:

Some languages (like Chinese or Japanese) don't use spaces, making it hard to know where words start and end.

(3) Special symbol:

Symbols like @, # and & can be meaningful

- #Throughout should stay as one token
- C&A → A brand name, should not be split.

(4) Capital letters:

Should we treat capital and small letters the same? Apple → (Brand) apple – (fruit)

(5) Emails, links and hashtags.

These look like many parts but should be kept as one.

naimul123@gmail.com → should stay as one token.

https://openai.com → kept it whole.

(6) Mixed languages.

Some texts used more then one language.

"Ami office-e jabo", → Bangla + English

Needs special handling.

* Stopping vs Stemming:

Criterea	Stopping	Stemming
Definition.	Removing common words from text	Reducing words to their base form
purpose	To eliminate unimportant words	To group similar words with the same meaning
Example	"the", "is", "and" removed	"running", "runs" → run.
Used for	reducing noise and data size	matching different forms of a word

Impact on search effectiveness:

Stopping

positive effect:

→ Reduces index size and speed up search.

→ Removes words that don't affect meaning.

Negative Impact:

→ might hurt result when stop words are meaningful.

→ Ex: "The Who", removing "the" changes (Brand name) meaning.

Stemming:

Positive:

→ Improve recall by matching word variations.

→ Searching for "connect" also finds "connecting" and "connected".

Negative:

→ Can reduce precision due to over-stemming.

→ "organization" and "organ" may get wrongly grouped.

How porter stemmer works?

The porter stemmer is a popular algorithm used to reduce english words to their roots or stem form by removing common suffix like "ing", 'ed', 'ly' etc.

step 1: Remove "ing", "ed", "s" etc.

→ "running" → "runn"

step 2: Remove suffixes like "-ational", "tional".

e.g: "relational" → relat

step 3: Remove suffixes like "ness" "ful" etc.

e.g: "usefulness" → "useful → use"

final: Remove double letters or unnecessary endings. "hopping" → "hop"

advantages:

(i) Simple and efficient: use a set of clear rules, no need for a dictionary.

(ii) Lightweight: fast and consumes low resources.

(iii) Improves recall: Groups similar words together improves recall.

Disadvantages:

(i) Over Stemming: may cut too much and merge unrelated words.

→ "university" and "universe" both become "univers"

(ii) Under Stemming: May not cut enough and miss grouping.

→ "sky" and "skies" → "sky" and "ski" (wrong cut)

(iii) Language specific: works only for english language, not suitable for other languages.

How krovetz stemmer works?

The krovetz stemmer is a dictionary - based stemming algorithm that aims to produce more accurate and meaningful stems compared to aggresive stemmers like porter.

process:

① Check dictionary first:

→ If the word is in the dictionary, no stemming is done.

→ Ex: "run" already a valid word, keep as it is.

(ii) Apply light suffix rule:

→ if the words is not in the dictionary it tries to remove suffix like —

"s", "ed", "ing", "ly", "ness" etc.

Ex: "running" → "run"

(iii) Check modified word in the dictionary:

→ if the modified word after suffix removal exists in the dictionary, use that.

→ if not, keep the original word.

Advantages:

(i) more accurate

(ii) Reduces over-stemming

(iii) keep meanings.

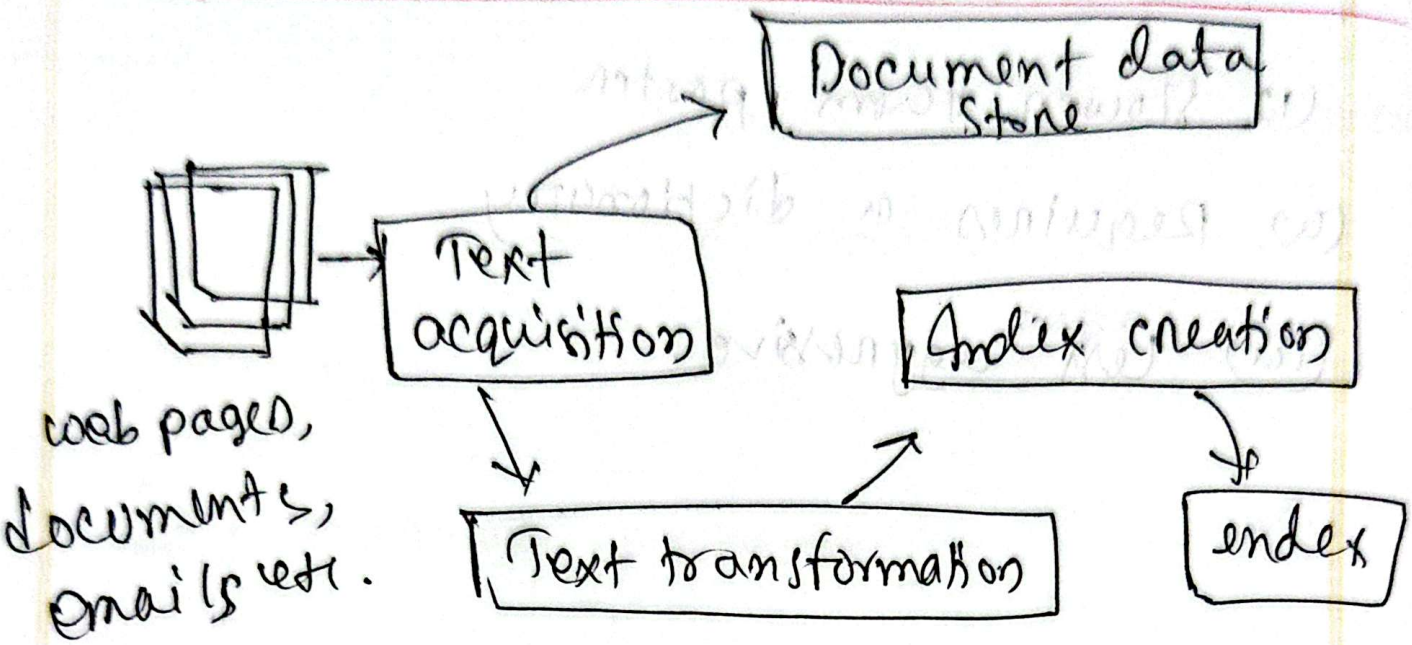
Disadvantages:

(i) Slower than porter

(ii) Requires a dictionary

(iii) Less aggressive.

Indexing process in search engine:



Search engines follow main three steps to build an index:

(1) Text acquisition:

→ The search engine's crawler visits web pages and download their content.

→ It collects HTML, text, multimedia, and metadata from millions of pages across the web.

(2) Text transformation:

The colleted content is cleaned and processed. Steps include:-

(i) Tokenization

(ii) Stopping

(iii) Stemming

(iv) Removing HTML tags & noise.

(3) Index creation:

→ The processed words are stored in a data structure called ~~inverted~~ inverted index.

→ It maps each words to the list of documents where it appears.