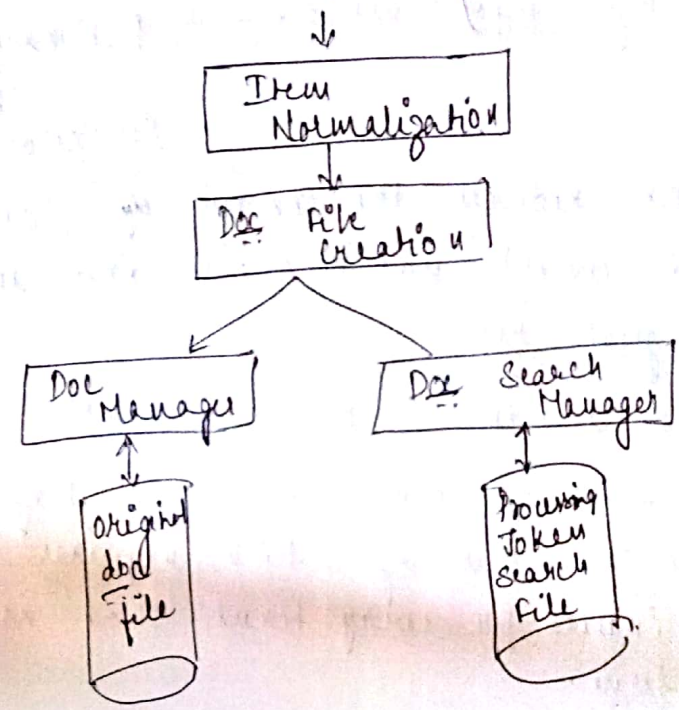
Module II

⇒ Data Structure:-

Any Info Sys uses 2 major data structures.

↳ One structure stores and manages the received items in their normalized form. called "document manager."

↳ The other DS contains the processing tokens and associated data to support search.



↳ The most common ds encountered in both database and info sys is the Inverted File System. This minimizes secondary storage access when multiple search terms are applied across the total data-base.

↳ A variant of the searchable ds is the N-gram structure that breaks processing tokens into smaller string units and uses the token fragments for search.

↳ PAT trees and arrays view the text of an item as a single long stream vs a juxtaposition of words.

↳ Signature files are based upon the idea of fast elimination of non-relevant items reducing the searchable items to a manageable subset.

↳ Hypertext structure allows the creator of an item to manually or automatically create imbedded links within one item or related items.

⇒ Stemming Algorithms :- Porter Stemming Alg
Dictionary Look up Stemmers
Successor Stemmers

Stemming reduces the ~~concept of~~ diversity of rep of a concept ~~discrimination into~~ to a canonical morphological rep.

↳ Causes decrease in precision while improves recall

↳ The stemming process creates one large index for the stem as Term Masking which requires the merging of the indexes for every term that matches the search term.

↳ Stemming alg are used to improve efficiency of the info sys. thereby improving recall.

↓

As long as a semantically consistent stem can be identified for a set of words, the generalization process of stemming does help in not missing potentially relevant items.

↳ Stemming can also cause problems for NLP sys by causing the loss of info needed for aggregate levels of NLP.

↳ Most stemming alg removes suffixes & prefixes, sometimes recursively, to derive the final stem.

↳ Stemmers such as Table lookup and successor stemming provide the alternatives that require additional overheads.

↳ Stemming is applied to the user query as well as to the incoming text.

→ (1) Porter Stemming Algorithm :-

↳ based upon a set of conditions of the stem, suffix and prefix and associated actions given the cond.

* The measure m, of a stem is a fn of sequences of vowels followed by a const. If V is a seq of vowels and C is a seq of consonants, then m is :

$C(VC)^m V$

m = 0 free, why

m = 1 frees, whose

m = 2 prologue, compute

* *<X> - stem ends with letter X.
* *V* - stem contains a vowel.
* *d - stem ends in double consonant
* *O - stem ends with consonant - vowel - consonant seq where the final consonant is not w, x or y.

→ (2) Dictionary look-up Stemmers :-

↳ These alg rely on the determination of a stem by using dictionary look-up mechanism.

↳ Here, the original term or stemmed version of the term is looked up in a dictionary and replaced by the stemp that best represents it.

↳ used by INQUERY & Retrieval Ware Sys.
↳ kstem.

↳ KStem is a morphological analyzer that conflates word variants to a root form.

Eg: Memorial, Memorize → memory

not synonyms

↳ KStem requires a word to be in a dictionary before it reduces one word form to another.
Eg:- 'Factorial' needs to be in dictionary or it is stemmed to "factory".

KStem uses the following 6 major data files to control & limit the stemming process:-

* Dictionary of words (lexicon)
* Supplemental list of words for the dictionary
* Exceptions list for those words that should retain an "e" at the end.
  eg: "suites" to "suite"
  but "suited" to "suit"
* Direct_Conflation: allows def of direct conflation via word pairs that override the stemming alg.
* County_Nationality: conflations b/w nationalities and countries ("British" maps to "Britain")
* Proper Nouns - a list of proper nouns that should not be stemmed.

↳ The strength of the "Retrieval Ware Sys" lies in its Thesaurus/Semantic Network support data structure that contains over 400,000 words.
↳ Dictionaries contain the morphological variants of words
↳ New words that are not spl forms are located

in the dictionary to determine simpler forms by stripping off suffixes and respelling plurals as defined in the dictionary.

⟶ (3) Successor Stemmers :-

↳ based upon the length of prefixes that optimally stem expansions of additional suffixes.

↳ determines the successor varieties for a word, use this info to divide a word into segments and select one of the segments as the stem.

↳ successor variety of a segment of a word in a set of words is the no of distinct letters that occupy the segment length plus one character.