INFORMATION RETRIEVAL SYSTEM

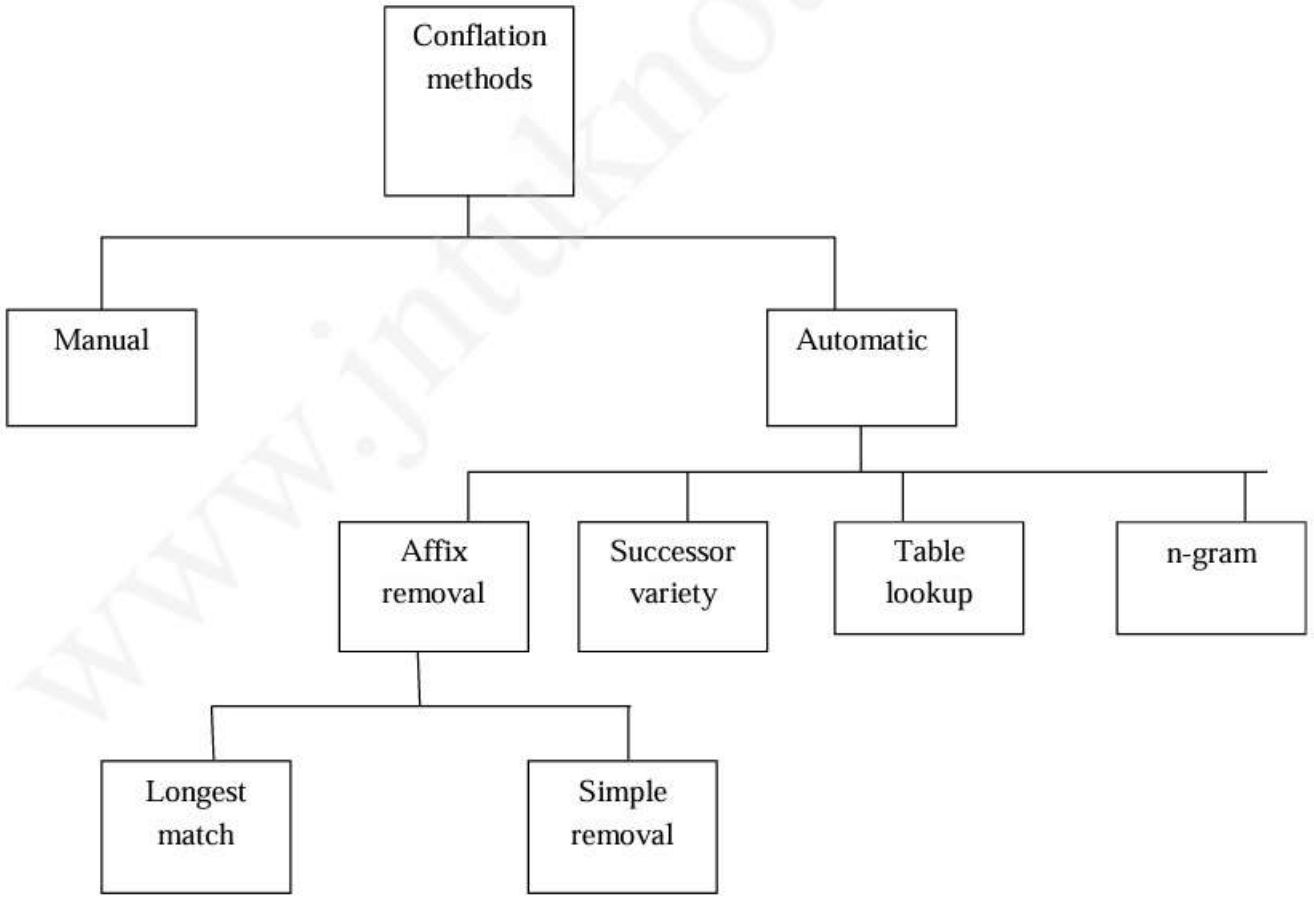
Unit IV: Stemming Algorithms and Thesaurus Construction:

- Types of Stemming algorithms
- Experimental Evaluations of Stemming
- Stemming to Compress Inverted Files.
- **Thesaurus Construction**: Features of Thesauri
- Thesaurus Construction
- Thesaurus construction from Texts
- Merging existing Thesauri.

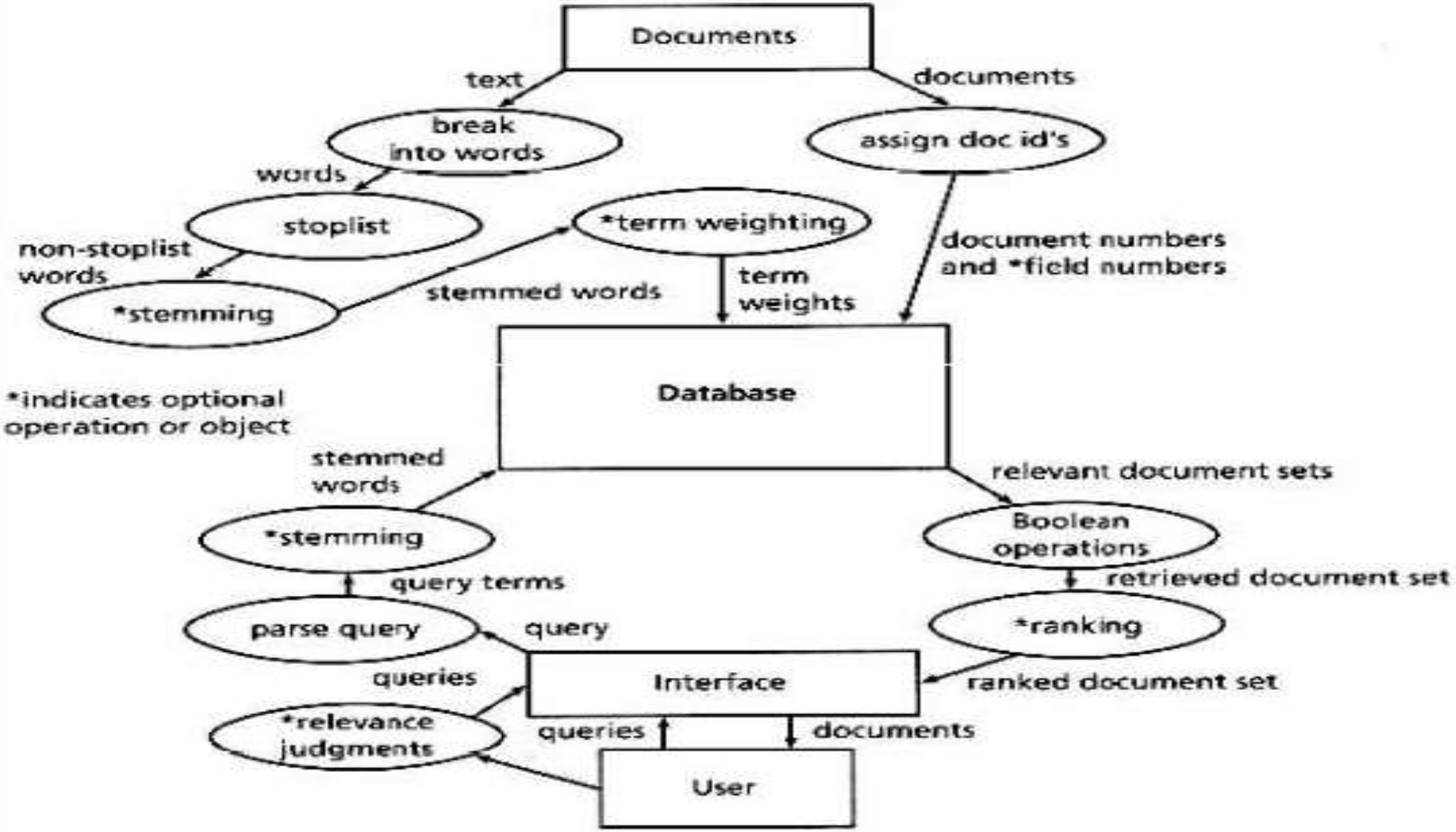
INTRODUCTION

- One technique for improving IR performance is to provide searchers with ways of finding morphological variants of search terms.
- **Example:** A searcher enters the term stemming as part of a query, it is likely that he or she will also be interested in such variants as stemmed and stem.
- **Conflation:** It is the term frequently used to refer to mapping multiple morphological variants to a single representation.
- Conflation can be either manual or automatic.
- **Manual conflation:** Using some kind of regular expressions.
- **Automatic conflation:** Using programs (stemmers).

- Terms can be stemmed at indexing time or at search time.



Functional View of Paradigm IR System:



- The stem carries the meaning of the concept associated with the word and the affixes (endings) introduce subtle modifications to the concept.
- **For example:** the stem “comput” could associate “computable, computability, computation, computational, computed, computing, computer, computerize” to one compressed word.
- Several criteria for judging stemmers:
  - Correctness
  - Retrieval effectiveness
  - Compression performance
- Two ways stemming can be incorrect:
  - Over stemming (too much removal)
  - Under stemming (too little removal)

Example of Stemmer Use in Searching

- In CATALOG, terms are stemmed at search time rather than at indexing time.
- CATALOG prompts for queries with the string "Look for:”
- At the prompt, the user types in one or more terms of interest.
- For example: Look for: system users will cause CATALOG to attempt to find documents about system users.
- CATALOG takes each term in the query, and tries to determine which other terms in the database might have the same stem.
- If any possibly related terms are found, CATALOG presents them to the user for selection.

- In the case of the query term "users," for example, CATALOG might respond as follows:

Term	Occurrences
user	15
users	1
used	3
using	2

TYPES OF STEMMING ALGORITHMS-TABLE LOOKUP

- There are several approaches to stemming. One way to do stemming is to store a table of all index terms and their stems. For example:

Term	Stem
engineering	engineer
engineered	engineer
engineer	engineer

- Terms from queries and indexes could then be stemmed via table lookup. Using a B-tree or hash table, such lookups would be very fast.

- There are problems with this approach are:
- There is no such data for English. Hence no standard database exist and some other stemming method would be needed.
- Storage overhead for such a table.

SUCCESSOR STEMMERS

- Successor stemmers are based upon the length of prefixes that optimally stem expansions of additional suffixes.
- The algorithm is based upon an analogy in structural linguistics that investigated word and morpheme boundaries based upon file distribution of phonemes, the smallest unit of speech that distinguish one word from another.
- The successor variety of a segment of a word in a set of words is the number of distinct letters that occupy the segment length plus one character.

- The successor varieties of a word are used to segment a word by applying one of the following four methods:

1. **Cutoff method:** Some cutoff value (Threshold) is selected for successor varieties and a boundary is reached.
2. **Peak and Plateau:** A segment break is made after a character whose successor variety exceeds that of the character immediately preceding it and the character immediately following it.
3. **Complete word method:** A break is made after a segment if the segment is a complete word in the corpus.

4. **Entropy method:**

The method works as follows:

- Let | Dαi| be the number of words in a text body beginning with the i length sequence of letters α.
- Let |Dαij| the number of words in |Dαi| with the successor j computed in step one.
- The probability that a word has the successor j is given by:

The entropy of $|D_{\alpha i}|$ is: $\frac{|D_{\alpha ij}|}{|D_{\alpha i}|}$

$$H_{\alpha i} = \sum_{j=1}^{26} -\frac{|D_{\alpha ij}|}{|D_{\alpha i}|} \cdot \log_2 \frac{|D_{\alpha ij}|}{|D_{\alpha i}|}$$

- A cutoff value is selected, and a boundary is identified whenever the cutoff value reached

- Example

Test Word :	*READABLE*
Corpus :	ABLE, APE, BEATABLE, FIXABLE, READ, READABLE, READING, READS, RED, ROPE, RIPE

- Cutoff method:
  - segment when successor variety >= threshold
  - consider threshold = 2 R|E|AD|ABLE

Table 1: Successor variety stem process

Prefix	Successor variety	Letters
R	3	E, I, O
RE	2	A, D
REA	1	D
READ	3	A, I, S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	(Blank

- Peak and plateau method : Break at the character whose successor variety is greater than both its preceding and following character READ|ABLE.

- Complete word method: Break is made if the segment is a complete word in the corpus (READ).

- Entropy method:

- for i = 2, α = RE, |Dαi| = 5

Choose i = 2, α = "RE"

That means: look at the 2-letter prefix "RE" in the corpus and check what letters can come **immediately after** it.

- for j = ‘A’, |Dαij| = 4
- for j = ‘D’, |Dαij| = 1
- Haij = - 1/5 * log2 (1/5) - 4/5 * log2 (4/5) = 0.46 + 0.26 = 0.72

This value is low because “REA…” appears four times!

The calculation shows that "RE" is **strongly biased towards successor "A"** (→ "REA..."). Therefore, entropy is low, meaning "RE" is a predictable sequence in this corpus.

N-Gram Stemmers

- A digram is a pair of consecutive letters.
- Since trigrams, or n-grams could be used, we have called it the n-gram method.
- Though we call this a "stemming method," this is a bit confusing since no stem is produced.

- In this approach, association measures are calculated between pairs of terms based on shared unique digrams.
- For example, the terms statistics and statistical can be broken into digrams as follows.
- statistics => st ta at ti is st ti ic cs
- unique digrams = at cs ic is st ta ti
- statistical => st ta at ti is st ti ic ca al
- unique digrams = al at ca ic is st ta ti

- Thus, "statistics" has nine digrams, seven of which are unique, and "statistical" has ten digrams, eight of which are unique.
- The two words share six unique digrams: at, ic, is, st, ta, ti.
- Once the unique digrams for the word pair have been identified and counted, a similarity measure based on them is computed. The similarity measure used was Dice's coefficient, which is defined as

$$S = \frac{2C}{A + B}$$

- where A is the number of unique digrams in the first word, B the number of unique digrams in the second, and C the number of unique digrams shared by A and B.

- For the example above, Dice's coefficient would equal

$$(2 \times 6)/(7 + 8) = .80.$$

- Such similarity measures are determined for all pairs of terms in the database, forming a similarity matrix.

- Since Dice's coefficient is symmetric ($S_{ij} = S_{ji}$), a lower triangular similarity matrix can be used as in the example below.

	$word_1$	$word_2$	$word_3$	. . .	$word_{n-1}$
$word_1$					
$word_2$	s_{21}				
$word_3$	s_{31}	s_{32}			
•					
•					
$word_n$	s_{n1}	s_{n2}	s_{n3}		$s_{n(n-1)}$

- Once such a similarity matrix is available, terms are clustered using a single link clustering method

AFFIX REMOVAL STEMMERS

- Affix removal algorithms remove suffixes and/or prefixes from terms leaving a *stem*.
- These algorithms sometimes also transform the resultant stem.
- A simple example of an affix removal stemmer is one that removes the plurals from terms.

- A set of rules for such a stemmer is as follows
- If a word ends in "ies" but not "eies" or "aies"

  Then "ies" -> "y"
- If a word ends in "es" but not "aes", "ees", or "oes"

  then "es" -> "e"
- If a word ends in "s", but not "us" or "ss"

  then "s" -> NULL

- Most stemmers currently in use are iterative longest match stemmers, a kind of affix removal stemmer irst developed by Lovins (1968).
- An iterative longest match stemmer removes the longest possible string of characters from a word according to a set of rules.
- This process is repeated until no more characters can be removed.
- Even after all characters have been removed, stems may not be correctly conflated.
- The word "skies," for example, may have been reduced to the stem "ski" which will not match "sky."
- There are two techniques to handle this--recoding or partial matching.

- Recoding is a context sensitive transformation of the form AxC -> AyC where A and C specify the context of the transformation, x is the input string, and y is the transformed string.
- For example, specify that if a stem ends in an "i" following a "k," then i -> y.
- In partial matching, only the n initial characters of stems are used in comparing them.
- Using this approach, we might say that two stems are equivalent if they agree in all but their last characters.

- The Porter algorithm is more compact than Lovins, to give retrieval performance comparable to the larger algorithms.
- The Porter algorithm consists of a set of condition/action rules.
- The conditions fall into three classes:
  - Conditions on the stem
  - Conditions on the suffix
  - Conditions on the rules.

- There are several types of **stem conditions**.

1. The *measure*, denoted *m*, of a stem is based on its alternate vowel-consonant sequences. Vowels are a, e, i, o, u, and y if preceded by a consonant. Consonants are all letters that are not vowels. Let *C* stand for a sequence of consonants, and *V* for a sequence of vowels. The measure *m*, then, is defined as

$$[C](VC)^m[V]$$

- The superscript *m* in the equation, which is the measure, indicates the number of *VC* sequences. Square brackets indicate an optional occurrence. Some examples of measures for terms follows.

Measure	Examples
m=0	TR, EE, TREE, Y, BY
m=1	TROUBLE, OATS, TREES, IVY
m=2	TROUBLES, PRIVATE, OATEN

2. * < X >--the stem ends with a given letter X

3. *v*--the stem contains a vowel

4. *d--the stem ends in a double consonant

5. *o--the stem ends with a consonant-vowel-consonant, sequence, where the final consonant is not w, x, or y.

- **Suffix conditions** take the form: (current_suffix == pattern).
- **Rule conditions** take the form: (rule was used).
- Actions are rewrite rules of the form:
- old_suffix -> new_suffix

- The rules for the steps of the stemmer are as follows.

Step 1a Rules

Conditions	Suffix	Replacement	Examples
NULL	sses	ss	caresses -> caress
NULL	ies	i	ponies -> poni
			ties -> tie
NULL	ss	ss	carress -> carress
NULL	s	NULL	cats -> cat

Step 1b Rules

Conditions	Suffix	Replacement	Examples
(m>0)	eed	ee	feed-> feed
			agreed -> agree
(*v*)	ed	NULL	plastered-> plaster
			bled -> bled
(*v*)	ing	NULL	motoring-> motor
			sing-> sing

- **Purpose of Step 1b1**
- After Step 1b removes “ed” or “ing”, Step 1b1 checks the resulting stem for:
- Specific endings like “at”, “bl”, “iz” → replace with “ate”, “ble”, “ize”.
- Double consonants at the end → remove one of the repeated letters.
- Short (CVC) words → add “e” at the end.

Step 1b1 Rules

Conditions	Suffix	Replacement	Examples
NULL	at	ate	conflat(ed) -> conflate
NULL	bl	ble	troubl(ing) -> trouble
NULL	iz	ize	siz(ed) -> size
(*d and not (*<L> or *<S> or *<Z>))	NULL	single letter	hopp(ing) -> hop tann(ed) -> tan fall(ing) -> fall hiss(ing) -> hiss fizz(ed) -> fizz
(m=1 and *o)	NULL	e	fail(ing) -> fail fil(ing) -> file

Step 1c Rules

Conditions	Suffix	Replacement	Examples
(*v*)	y	i	happy - > happi
			sky -> sky

Step 2 Rules

Conditions	Suffix	Replacement	Examples
(m>0)	ational	ate	relational -> relate
(m>0)	tional	tion	conditional -> condition
			rational -> rational
(m>0)	enci	ence	valenci -> valence
(m>0)	anci	ance	hesitanci -> hesitance
(m>0)	izer	ize	digitizer -> digitize
(m>0)	abli	able	conformabli -> conformable
(m>0)	alli	al	radicalli -> radical
(m>0)	entli	ent	differentli -> different
(m>0)	eli	e	vileli -> vile
(m>0)	ousli	ous	analogousli -> analogous
(m>0)	ization	ize	vietnamization -> vietnamize
(m>0)	ation	ate	predication -> predicate
(m>0)	ator	ate	operator -> operate
(m>0)	alism	al	feudalism -> feudal
(m>0)	iveness	ive	decisiveness -> decisive
(m>0)	fulness	ful	hopefulness -> hopeful
(m>0)	ousness	ous	callousness -> callous
(m>0)	aliti	al	formaliti -> formal
(m>0)	iviti	ive	sensitiviti -> sensitive
(m>0)	biliti	ble	sensibiliti -> sensible

Step 3 Rules

Conditions	Suffix	Replacement	Examples
(m>0)	icate	ic	triplicate -> triplic
(m>0)	ative	NULL	formative -> form
(m>0)	alize	al	formalize -> formal
(m>0)	iciti	ic	electriciti -> electric
(m>0)	ical	ic	electrical -> electric
(m>0)	ful	NULL	hopeful -> hope
(m>0)	ness	NULL	goodness -> good

Step 4 Rules

Conditions	Suffix	Replacement	Examples
(m>1)	al	NULL	revival -> reviv
(m>1)	ance	NULL	allowance -> allow
(m>1)	ence	NULL	inference -> infer
(m>1)	er	NULL	airliner-> airlin
(m>1)	ic	NULL	gyroscopic -> gyroscop
(m>1)	able	NULL	adjustable -> adjust
(m>1)	ible	NULL	defensible -> defens
(m>1)	ant	NULL	irritant -> irrit
(m>1)	ement	NULL	replacement -> replac
(m>1)	ment	NULL	adjustment -> adjust
(m>1)	ent	NULL	dependent -> depend
(m>1 and (*<S> or *<T>))	ion	NULL	adoption -> adopt
(m>1)	ou	NULL	homologou-> homolog
(m>1)	ism	NULL	communism-> commun
(m>1)	ate	NULL	activate -> activ
(m>1)	iti	NULL	angulariti -> angular
(m>1)	ous	NULL	homologous -> homolog
(m>1)	ive	NULL	effective -> effect
(m>1)	ize	NULL	bowdlerize -> bowdler

Step 5a Rules

Conditions	Suffix	Replacement	Examples
(m>1)	e	NULL	probate -> probat
			rate - > rate
(m=1 and not *o)	e	NULL	cease - > ceas

Step 5b Rules

Conditions	Suffix	Replacement	Examples
(m> 1 and *d and *<L>)	NULL	single letter	controll -> control
			roll -> roll

The algorithm is as follows.

```
{
step1a(word);
step1b(stem);
if (the second or third rule of step 1b was used)
step1b1(stem);
step1c(stem);
step2(stem);
step3(stem);
step4(stem);
step5a(stem);
step5b(stem);
}
```

- **Original Word: caresses**

Step 1a: Plural Rules

- Rule: SSES → SS
- caresses → caress
- ✓ Result: **caress**

Step 1b: -ed / -ing Removal

- Word: *caress* → does not end with *eed/ed/ing*
- ✗ No change.
- ✓ Result: **caress**

Step 1b1: Adjustments

- Only applies if Step 1b removed *ed/ing* → not applicable.
- ❌ No change.
- ✅ Result: **caress**

Step 1c: Replace Y with I

- Word ends in *ss*, not *y* → ❌ No change.
- ✅ Result: **caress**

Step 2: Long Suffix Replacement

- Look for suffixes like *ational, tional, enci, ization...*
- Word *caress* has none.
- ❌ No change.
- ✅ Result: **caress**

Step 3: Further Suffix Reduction

- Rules like *icate → ic, ative → "", alize → al...*
- Word *caress* has none.
- ❌ No change.
- ✅ Result: **caress**

Step 4: Common Suffix Removal

- Suffixes: *al, ance, ence, er, ic, able, ible, ant, ent, ism, ate, iti, ous, ive, ize*
- Word *caress* has none.
- ❌ No change.
- ✅ Result: **caress**

Step 5a: Final -e Removal

- Word ends with *ss*, not *e*.
- ❌ No change.
- ✅ Result: **caress**

Step 5b: Double L Rule

- Word does not end in *ll*.
- ❌ No change.
- ✅ Result: **caress**

Final Stemmed Word: **caress**

Full Example Walkthrough

Let's trace the algorithm with **word = "relational"**:

1. **Step 1a:** no plural → stays *relational*
2. **Step 1b:** no *ed/ing* → unchanged
3. **Step 1c:** no final *y* → unchanged
4. **Step 2:** *ational* → *ate* → *relate*
5. **Step 3:** no further suffix match
6. **Step 4:** no removal
7. **Step 5a/5b:** final *e* not removed → final stem = **relate**

Another Example: “Controllably”

Step	Action	Result
Step 1a	no plural suffix	controllably
Step 1b	no *ed/ing*	controllably
Step 1c	no *y* → *i*	controllably
Step 2	*ably* → *able*	controllable
Step 3	remove *able* → *NULL* (m>0)	controll
Step 4	check for other suffixes (none)	controll
Step 5a	no final e	controll
Step 5b	ends with *ll*, m>1 → remove one	**control**

EXPERIMENTAL EVALUATIONS OF STEMMING

- **Salton (1968)** examined the relative retrieval performance of fully stemmed terms against terms with only the suffix "s" removed.
- Three document collections were used in these studies: the **IRE-3 collection** consisting of 780 computer science abstracts and 34 search requests, the **ADI collection** consisting of 82 papers and 35 search requests, and the **Cranfield-1 collection** consisting of 200 aerodynamics abstracts and 42 search requests.
- Differences between the two conflation methods, fully stemmed and "s" stemmed, were calculated on 14 dependent variables for each query: rank recall, log precision, normalized recall, normalized precision, and precision for ten recall levels.
- This data was analyzed using both related group t tests(The same subjects are tested **twice** under **two conditions**.-before and after stemming) and sign tests(To check whether one condition tends to produce **larger** or **smaller** results than another. It only considers **+** or **–** signs of differences, ignoring how big they are.).

- **Hafer and Weiss (1974)** tested their stemmer against other stemming methods using the ADI collection, and the Carolina Population Center (CPC) Collection consisting of 75 documents and five queries. Comparisons were made on the basis of recall-precision plots. No statistical testing was reported.
- **Van Rijsbergen et al. (1980)** tested their stemmer (Porter 1980) against the stemmer described by Dawson (1974) using the Cranfield-1 test collection. Both of these stemmers are of the longest match type, though the Dawson stemmer is more complex. **They report that the performance of their stemmer was slightly better across ten paired recall-precision levels**, but that the observed differences are probably not meaningful. No statistical results of any kind are reported.

- **Katzer et al. (1982)** examined the performance of stemmed title-abstract terms against six other document representations: unstemmed title-abstract terms, unstemmed abstract terms, unstemmed title terms, descriptors, identifiers, and descriptors and identifiers combined. The stemmer used was a simple suffix removal stemmer employing 20 endings.
- **Lennon et al. (1981)** examined the performance of various stemmers both in terms of retrieval effectiveness and inverted file compression. For the retrieval comparisons, the Cranfield-1400 collection was used. This collection contains 1,396 documents, and also 225 queries. The titles of documents and the queries were matched, and the documents ranked in order of decreasing match value.

- **Frakes (1982)** did two studies of stemming. The first tested the hypothesis that the closeness of stems to root morphemes will predict improved retrieval effectiveness. Fifty-three search queries were solicited from users and searched by four professional searchers under four representations: title, abstract, descriptors, and identifiers.
- **Walker and Jones (1987)** did a thorough review and study of stemming algorithms. They used Porter's stemming algorithm in the study. The database used was an on-line book catalog (called RCL) in a library.
- **Harman (1991)** used three stemmers--Porter, Lovins, and S removal--on three databases- Cranfield 1400, Medlars, and CACM--and found that none of them significantly improved retrieval effectiveness in a ranking IR system called IRX. The dependent variable measure was again E, at b levels of .5, 1.0, and 2.0.

STEMMING TO COMPRESS INVERTED FILES

- Since a stem is usually shorter than the words to which it corresponds, storing stems instead of full words can decrease the size of index files. Lennon et al. (1981), for example, report the following compression percentages for various stemmers and databases.

- **Index Compression Percentages from Stemming**

Stemmer	Cranfield	National physical laboratory	INSPEC	Brown Corpus
INSPEC	32.1	40.7	40.5	47.5
Lovins	30.9	39.2	39.5	45.8
RADCOL	32.1	41.8	41.8	49.1
Porter	26.2	34.6	33.8	38.8
Frequency	30.7	39.9	40.1	50.5

Thesaurus Construction

Introduction:

- Thesauri are valuable structures for Information Retrieval systems.
- A thesaurus provides a precise and controlled vocabulary which serves to coordinate document indexing and document retrieval.
- In both indexing and retrieval, a thesaurus may be used to select the most appropriate terms.
- Additionally, the thesaurus can assist the searcher in reformulating search strategies if required.
- Two major approaches for automatic thesaurus construction have been selected for detailed examination.
- The first is on thesaurus construction from collections of documents, and the second, on thesaurus construction by merging existing thesauri.

- In IR systems, used to retrieve potentially relevant documents from large collections, the thesaurus serves to coordinate the basic processes of indexing and document retrieval.
- In indexing, a succinct representation of the document is derived, while retrieval refers to the search process by which relevant items are identified.
- The IR thesaurus typically contains a list of terms, where a term is either a single word or a phrase, along with the relationships between them.
- It provides a common, precise, and controlled vocabulary which assists in coordinating indexing and retrieval.

- A short extract from an alphabetical listing of thesaurus terms and their relationships in the INSPEC thesaurus.

```
cesium
        USE caesium

computer-aided instruction
        see also education
        UF  teaching machinse
        BT  educational computing
        TT  computer applications
        RT  education
            teaching
        CC  C7810C
        FC  7810Cf
```

Symbol	Meaning	Example
BT	Broader Term(More general)	AI → BT: Computer Science
NT	Narrower Term(More specific)	AI → NT: Machine Learning
RT	Related Term	AI → RT: Robotics
UF	Use For (non-preferred term)	AI → UF: Machine Intelligence
USE	Preferred term to use	Machine Intelligence → USE: Artificial Intelligence
TT	Top Term	Computer Science
See also	cross-referenced terms	

- **CC(Classification Code)** indicates the **subject category** or **broad area** where a term belongs.
- **FC (Facet Code)** indicates the **facet** or **contextual category** in which the term is used within that subject area.

Higher Education

↳ Bachelor of Technology (B.Tech)

↳ Branches:

- Computer Science and Engineering (CSE)

-Artificial Intelligence and Machine Learning

-Data Science

- Electronics and Communication Engineering (ECE)

-Communication Systems

-Signal Processing

- Mechanical Engineering (ME)

- Civil Engineering (CE)

- Electrical Engineering (EE)

USE Computer Science and Engineering (CSE)

UF CSE

BT Bachelor of Technology

TT Higher Education

RT Artificial Intelligence

Data Science

Computer Engineering

NT CSE - Artificial Intelligence and Machine Learning

CSE - Data Science

CC CSE001

FEATURES OF THESAURI:

- Coordination Level
- Term Relationships
- Number of Entries for Each Term
- Specificity of Vocabulary
- Control on Term Frequency of Class Members
- Normalization of Vocabulary

Coordination Level:

- Coordination refers to the construction of phrases from individual terms.
- Two distinct coordination options are recognized in thesauri: precoordination and post-coordination.
- A precoordinated thesaurus is one that can contain phrases. Consequently, phrases are available for indexing and retrieval.
- A postcoordinated thesaurus does not allow phrases. Instead, phrases are constructed while searching.

- The advantage in precoordination is that the vocabulary is very precise, thus reducing ambiguity in indexing and in searching.
- Also, commonly accepted phrases become part of the vocabulary.
- However, the disadvantage is that the searcher has to be aware of the phrase construction rules employed.
- Thesauri can adopt an intermediate level of coordination by allowing both phrases and single words.
- Some thesauri may emphasize two or three word phrases, while others may emphasize even larger sized phrases.
- Therefore, it is insufficient to state that two thesauri are similar simply because they follow precoordination.
- The level of coordination is important as well.
- It should be recognized that the higher the level of coordination, the greater the precision of the vocabulary but the larger the vocabulary size.
- It also implies an increase in the number of relationships to be encoded. Therefore, the thesaurus becomes more complex.

- The advantage in post-coordination is that the user need not worry about the exact ordering of the words in a phrase.
- Phrase combinations can be created as and when appropriate during searching.
- The disadvantage is that search precision may fall, as illustrated by the following well known example, from Salton and McGill (1983): the distinction between phrases such as "Venetian blind" and "blind Venetian" may be lost. A more likely example is "library school" and "school library."
- The problem is that unless search strategies are designed carefully, irrelevant items may also be retrieved.
- Precoordination is more common in manually constructed thesauri.
- Automatic thesaurus construction usually implies post-coordination.

Term Relationships:

- Term relationships provide vocabulary connections that are most valuable for retrieval.
- Many kinds of relationships are expressed in a manual thesaurus.
- Aitchison and Gilchrist (1972) specify three categories of term relationships:

1. Equivalence relationships
2. Hierarchical relationships
3. Nonhierarchical relationships.

- Equivalence relations include both synonymy and quasi-synonymy.
- Synonyms can arise because of trade names, popular and local usage, superseded terms, and the like.
  - **Trade names:** *Aspirin* ↔ *Acetylsalicylic acid*
  - **Popular vs. scientific terms:** *Heart attack* ↔ *Myocardial infarction*
  - **Local or regional usage:** *Lift* ↔ *Elevator*
  - **Superseded terms:** *Microcomputer* (old) ↔ *Personal computer* (new)
- **Synonyms** are **different words that have the same or very similar meanings**. In IR, these are treated as equivalent so that a search for one term also retrieves documents containing its synonym.
  - In a **thesaurus**, we can have:

    Automobile **USE** Car
    Film **USE** Movie
- **Quasi-synonyms** are **terms that do not mean exactly the same thing**, but are **close enough in meaning** that they can be treated as equivalent **for retrieval purposes**.
  - for example, "genetics" and "heredity," which have significant overlap in meaning. Also, the terms "harshness" and "tenderness," which represent different viewpoints of the same property continuum.

- A **hierarchical relation** shows a **"broader–narrower"** or **"parent–child"** relationship between two concepts.
- In a **thesaurus**, these are often marked as:
- **BT** (Broader Term)
- **NT** (Narrower Term)
- A typical example of a hierarchical relation is genus-species, such as "dog" and "german shepherd." “Dog” is a general class of animals; “German Shepherd” is a specific kind of dog.

- Nonhierarchical relationships also identify conceptually related terms.
- In a **thesaurus**, these are often marked as:
- **RT** – *Related Term*
- There are many examples including:

1. Thing–Part Relationship

- Example: **Bus – Seat**
- A *seat* is **part of** a *bus*, but it is not a *type of* bus.

2. Thing–Attribute Relationship

- Example: **Rose – Fragrance**
- A *rose* is known for having the attribute *fragrance*.

- Wang, Vandendorpe, and Evens (1985) provide an alternative classification of term relationships consisting of:

1. Parts—wholes
2. Collocation relations
3. Paradigmatic relations
4. Taxonomy and synonymy
5. Antonymy relations.

- Parts and wholes: This type links a **whole** with its **parts.**
  - Examples such as set—element(An element is a member of a set).
- Collocation relates words that frequently co-occur in the same phrase or sentence.
  - Artificial Intelligence-“Artificial” and “Intelligence” often appear together as a fixed phrase.
  - Data Mining-Commonly used word pair in computer science.
- Paradigmatic relations relate words that have the same semantic core and are somewhat similar to Aitchison and Gilchrist's quasi-synonymy relationship.
  - Example "moon" and "lunar" - Both relate to the same concept (the moon), though used in different grammatical forms.

- Taxonomy and synonymy are self-explanatory and refer to the classical relations between terms.
- **Taxonomy** defines hierarchical relations — *broader* and *narrower* terms (genus–species).
  - Animal-Dog, Dog- German Shepherd
- **Synonymy** defines equivalence — *same or similar meaning* terms.
  - Car-Automobile, Movie-Film
- **Antonymy** represents the **opposite meaning relationship** between two terms.
  - Hot – Cold (Opposite ends of the temperature scale)
  - True – False(Opposite truth values)

Number of Entries for Each Term:

- It is in general preferable to have a single entry for each thesaurus term.
- It is achieved due to the presence of homographs--words with multiple meanings.
- Also, the semantics of each instance of a homograph can only be contextually deciphered.
- Therefore, it is more realistic to have a unique representation or entry for each meaning of a homograph.
- This also allows each homograph entry to be associated with its own set of relations.
- The problem is that multiple term entries add a degree of complexity in using the thesaurus--especially if it is to be used automatically.
- Typically the user has to select between alternative meanings. In a manually constructed thesaurus such as INSPEC, this problem is resolved by the use of parenthetical qualifiers, as in the pair of homographs, bonds (chemical) and bonds (adhesive).
- However, this is hard to achieve automatically.

Specificity of Vocabulary:

- Specificity of the thesaurus vocabulary is a function of the precision associated with the component terms.
- A highly specific vocabulary is able to express the subject in great depth and detail. This promotes precision in retrieval.
- The disadvantage is that the size of the vocabulary grows since a large number of terms are required to cover the concepts in the domain.
- Also, specific terms tend to change (i.e., evolve) more rapidly than general terms.
- Therefore, such vocabularies tend to require more regular maintenance.

Control on Term Frequency of Class Members:

- This has relevance mainly for statistical thesaurus construction methods which work by partitioning the vocabulary into a set of classes where each class contains a collection of equivalent terms.
- In order to maintain a good match between documents and queries, it is necessary to ensure that terms included in the same thesaurus class have roughly equal frequencies.
- Further, the total frequency in each class should also be roughly similar.
- These constraints are imposed to ensure that the probability of a match between a query and a document is the same across classes.
- In other words, terms within the same class should be equally specific, and the specificity across classes should also be the same.

Term	Frequency
car	500
automobile	50
vehicle	480
doctor	300
physician	280
surgeon	310

Now, if we group:
•**Class 1** = {car, automobile, vehicle}
•**Class 2** = {doctor, physician, surgeon}
We can see:
•Within Class 1 → *automobile* (50) is far less frequent than *car* (500).
→ The terms are **not equally frequent**, so retrieval results could be skewed.
•Within Class 2 → All terms are around 300 → good balance.

Normalization of Vocabulary:

- Normalization of vocabulary terms is given considerable emphasis in manual thesauri.
- There are extensive rules which guide the form of the thesaural entries.
- A simple rule is that terms should be in noun form.
- A second rule is that noun phrases should avoid prepositions unless they are commonly known.
- Also, a limited number of adjectives should be used.
- There are other rules to direct issues such as the singularity of terms, the ordering of terms within phrases, spelling, capitalization, transliteration, abbreviations, initials, acronyms, and punctuation.
- In other words, manual thesauri are designed using a significant set of constraints and rules regarding the structure of individual terms.

- The advantage in normalizing the vocabulary is that variant forms are mapped into base expressions, thereby bringing consistency to the vocabulary.
- As a result, the user does not have to worry about variant forms of a term.
- The obvious disadvantage is that, in order to be used effectively, the user has to be well aware of the normalization rules used.
- In contrast, normalization rules in automatic thesaurus construction are simpler, seldom involving more than stoplist filters and stemming.

- **Main Rules in Manual Thesauri**
- **Use noun form:**
  Terms are usually written as **nouns**, since they represent *concepts or entities*.
  - ✓ *"Education"* (noun)
  - ✗ *"Educate"* (verb)
- **Avoid prepositions unless necessary:**
  - ✓ *"Water pollution"* (common, clear phrase)
  - ✗ *"Pollution of water"* (unnecessary preposition)
- **Limit adjectives:**
  Only include adjectives if they add real meaning.
  - ✓ *"Solar energy"*
  - ✗ *"Good energy"* (too vague)

•**Ensure singularity and order:**
•Use singular form for countable nouns:
✓ *“Computer network”*, not *“Computer networks”*
•Maintain logical order in noun phrases:
✓ *“Library automation system”*, not *“Automation library system”*
•**Control spelling, capitalization, and punctuation:**
•Use one accepted spelling (e.g., *color* or *colour*, not both).
•Use consistent capitalization (e.g., all lowercase except proper nouns).
•Avoid unnecessary punctuation.
•**Handle abbreviations and acronyms consistently:**
•Choose one preferred form,
•e.g., *UNESCO (preferred)*, *United Nations Educational, Scientific and Cultural Organization (use for definition)*.

THESAURUS CONSTRUCTION

Manual Thesaurus Construction

- First the boundaries of the subject area has to be defined.
- Boundary definition includes identifying central subject areas and peripheral ones since it is unlikely that all topics included are of equal importance.
- Once this is completed, the domain is generally partitioned into divisions or subareas.
- Once the domain, with its subareas, has been sufficiently defined, the desired characteristics of the thesaurus have to be identified.

- Now, the collection of terms for each subarea may begin.
- A variety of sources may be used for this including indexes, encyclopedias, handbooks, textbooks, journal titles and abstracts, catalogues, as well as any existing and relevant thesauri or vocabulary systems.
- Subject experts and potential users of the thesaurus should also be included in this step.
- After the initial vocabulary has been identified, each term is analyzed for its related vocabulary including synonyms, broader and narrower terms, and sometimes also definitions and scope notes.
- These terms and their relationships are then organized into structures such as hierarchies, possibly within each subarea.
- Once the initial organization has been completed, the entire thesaurus will have to be reviewed (and refined) to check for consistency such as in phrase form and word form.

- Once the thesaurus has been designed and implemented for use within a retrieval system, the next problem is that it needs to be maintained in order to ensure its continued viability and effectiveness.
- That is, the thesaurus should reflect any changes in the terminology of the area.
- The problem is that since the older documents are still within the system, the updated thesaurus must also retain the older information.
- Updates are typically slow and again involve several individuals who regularly review and suggest new and modified vocabulary terms as well as relationships.
- Therefore, typically a thesaurus evolves with time and slowly responds to changes in the terminology of the subject.

Automatic Thesaurus Construction

- The two major approaches selected here have not necessarily received equal attention in the literature.
- The first approach, on designing thesauri from document collections, is a standard one.
- The second, on merging existing thesauri, is better known using manual methods.
- A third automatic approach is discussed which is quite novel and interesting, although it is based on tools from expert systems and does not use statistical methods. In this approach, thesauri are built using information obtained from users.

From a Collection of Document Items:

- The idea is to use a collection of documents as the source for thesaurus construction. This assumes that a representative body of text is available.
- The idea is to apply statistical procedures to identify important terms as well as their significant relationships.
- It is semantic knowledge that is used by both indexer and searcher.
- Until more direct methods are known, statistical methods will continue to be used.

By Merging Existing Thesauri:

- This second approach is appropriate when two or more thesauri for a given subject exist that need to be merged into a single unit.
- If a new database can indeed be served by merging two or more existing thesauri, then a merger perhaps is likely to be more efficient than producing the thesaurus from scratch.
- The challenge is that the merger should not violate the integrity of any component thesaurus.

User Generated Thesaurus:

- In this third alternative, the idea is that users of IR systems are aware of and use many term relationships in their search strategies. The objective is to capture this knowledge from the user's search.
- This is the basis of TEGEN--the thesaurus generating system designed by Guntzer et al. (1988).
- They propose TEGEN as a viable alternative technique for automatic thesaurus construction.
- The procedure involves examining the types of Boolean operators used between search terms, the type of query modification performed by the user, and so on.
- User feedback is included to resolve any ambiguities and uncertainties.
- Feedback is also required to select the specific type of relationship between two terms once it has been decided that the terms are indeed related.
- Therefore, their approach requires considerable interaction with the user population.

THESAURUS CONSTRUCTION FROM TEXTS:

- The overall process may be divided into three stages:

  (1) Construction of vocabulary

  (2) Similarity computations between terms

  (3) Organization of vocabulary

Construction of Vocabulary:

- The first step is to identify an appropriate document collection.
- The collection should be sizable and representative of the subject area.
- The next step is to determine the required specificity for the thesaurus.
- If high specificity is needed, then the emphasis will be on identifying precise phrases; otherwise, more general terms can be sought.
- Terms can be selected from titles, abstracts, or even the full text of the documents if available.
- This initial set of vocabulary terms is now ready for normalization.
- The simplest and most common normalization procedure is to eliminate very trivial words such as prepositions and conjunctions.
- For this, an appropriate stoplist of trivial words needs to be constructed.

- Stemming reduces each word into its root form. For example, the terms "information," "informing," and "informed" could all be stemmed into the same root "inform.“
- The resulting pool of stems is now ready to be analyzed statistically with two objectives:

1. Stem evaluation and selection

There are a number of methods for statistically evaluating the worth of a term.

- ***Selection by Frequency of Occurrence***
- ***Selection by Discrimination Value (DV)***
- ***Selection by the Poisson Method***

Selection by Frequency of Occurence:

- The basic idea is that each term may be placed in one of three different frequency categories with respect to a collection of documents: **high, medium, and low frequency**.
- Terms in the mid-frequency range are the best for indexing and searching.
- Terms in the low-frequency range have minimal impact on retrieval, while high-frequency terms are too general and negatively impact search precision.
- Salton recommends creating term classes for the low-frequency terms; High-frequency terms are generally coordinated into phrases to make them more specific.

Selection by Discrimination Value (DV):

- The overall procedure is to compute the average interdocument similarity in the collection, using some appropriate similarity function.
- Next, the term *k* being evaluated is removed from the indexing vocabulary and the same average similarity is recomputed.
- The discrimination value (*DV*) for the term is then computed as:

$$DV(k) = (\text{Average similarity without } k) - (\text{Average similarity with } k)$$

- Good discriminators are those that decrease the average similarity by their presence, that is, those for which *DV* is positive.
- Poor discriminators have negative *DV*, while neutral discriminators have no effect on average similarity.
- Terms that are positive discriminators can be included in the vocabulary and the rest rejected.

Selection by the Poisson Method:

- The Poisson distribution is a discrete random distribution that can be used to model a variety of random phenomena including the number of typographical errors in a page of writing and the number of red cars on a highway per hour.
- In all the research that has been performed on the family of Poisson models, the one significant result is that trivial words have a single Poisson distribution, while the distribution of nontrivial words deviates significantly from that of a Poisson distribution.
- This result is used here to select nontrivial words as thesaurus terms.

2. Phrase construction

- Two methods are described
- ***Salton and McGill Procedure***
- ***Choueka Procedure***

Salton and McGill Procedure:

- The procedure is a statistical alternative to syntactic and/or semantic methods for identifying and constructing phrases.
- First, the component words of a phrase should occur frequently in a common context, such as the same sentence.
- The second general requirement is that the component words should represent broad concepts, and their frequency of occurrence should be sufficiently high.

Algorithm:

1. Compute pairwise co-occurrence for high-frequency words.

2. If this co-occurrence is lower than a threshold, then do not consider the pair any further.

3. For pairs that qualify, compute the cohesion value. Two formulas for computing cohesion are given below. Both ti and tj represent terms, and size-factor is related to the size of the thesaurus vocabulary.

COHESION (ti, tj) =

co-occurrence-frequency/sqrt(frequency(ti) * frequency(tj))

COHESION (ti, tj) =

size-factor * (co-occurrence-frequency/(total-frequency(ti) *

total-frequency(tj)))

4. If cohesion is above a second threshold, retain the phrase as a valid vocabulary phrase.

Choueka Procedure:

- The algorithm proposed is statistical and combinatorial and requires a large collection of documents to be effective.

1. Select the range of length allowed for each collocational expression. Example: two to six words.

2. Build a list of all potential expressions from the collection with the prescribed length that have a minimum frequency (again, a preset value).

3. Delete sequences that begin or end with a trivial word. The trivial words include prepositions, pronouns, articles, conjunctions, and so on.

- ✓ **Example:**
  Remove:
- "of information retrieval"
- "information retrieval of"
  Keep:
- "information retrieval system"

4. Delete expressions that contain high-frequency nontrivial words.

- ✓ **Example:**
  If *system* occurs in too many unrelated contexts:
- "system design" may be dropped
- But "information retrieval" may remain, since it's more specific.

5. Given an expression such as a b c d, evaluate any potential subexpressions such as a b c and b c d for relevance. Discard any that are not sufficiently relevant.

- ✅ **Example:**
  If we have:
- "information retrieval system design"
  Then subphrases:
- "information retrieval system"
- "retrieval system design"
  are tested — if one occurs rarely or is not meaningful, it's discarded.
- So, if *retrieval system design* rarely appears → it is dropped.

6. Try to merge smaller expressions into larger and more meaningful ones. For example, a b c d and b c d may merge to form a b c d. (Again, the exact criteria for allowing a merger are not given.)

- ✅ **Example:**
  If you have:
- "information retrieval system"
- "retrieval system design"
  They can be merged into the longer phrase:
- "information retrieval system design"
- This merging process captures more **specific and complete** multi-word concepts.

Similarity Computation:

- Once the appropriate thesaurus vocabulary has been identified, and phrases have been designed if necessary, the next step is to determine the statistical similarity between pairs of terms.

1. Cosine: which computes the number of documents associated with both terms divided by the square root of the product of the number of documents associated with the first term and the number of documents associated with the second.

2. Dice: which computes the number of documents associated with both terms divided by the sum of the number of documents associated with one term and the number associated with the other.

Formula:

$$\text{Cosine}(A, B) = \frac{n_{AB}}{\sqrt{n_A \times n_B}}$$

where:

- n_{AB} = number of documents containing **both** term A and term B
- n_A = number of documents containing **term A**
- n_B = number of documents containing **term B**

Formula:

$$\text{Dice}(A, B) = \frac{2n_{AB}}{n_A + n_B}$$

(Note: The version in your text omits the 2, but this is the most common form. Without the 2, it is equivalent to half of this value.)

Vocabulary Organization:

- Once the statistical term similarities have been computed, the last step is to impose some structure on the vocabulary which usually means a hierarchical arrangement of term classes.
- For this, any appropriate clustering program can be used.
- A standard clustering algorithm generally accepts all pairwise similarity values corresponding to a collection of objects and uses these similarity values to partition the objects into clusters or classes such that objects within a cluster are more similar than objects in different clusters.
- Some clustering algorithms can also generate hierarchies.