

\* Learn Python \*

C2F

Assignment / Paper reading

/ Research paper — 40%.

Mid term — 20%.

End term — 40%.

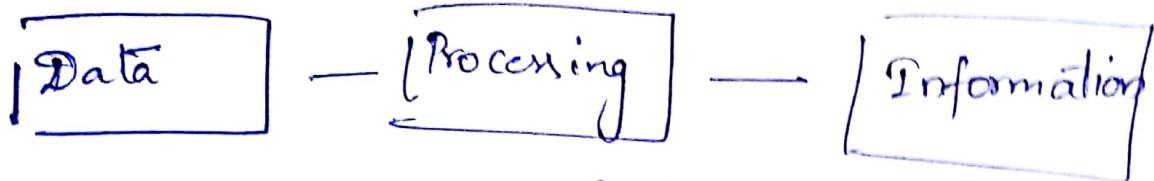
Book:— Christopher D. Manning —

Intro to IR.

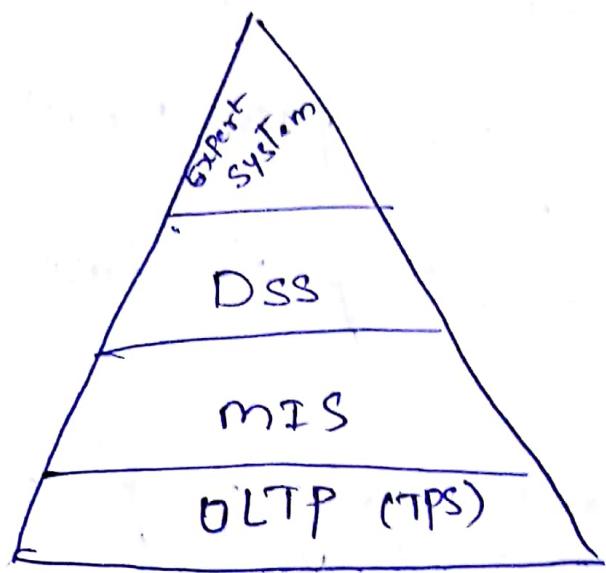
Student meeting hrs:— Tuesday :- 4 to 6  
=

Wednesday:— 4 to 6.

Friday:— after 3 o'clock



- collection
- classification
- coding
- sorting
- validations
- verifications
- calculations
- storage
- Retrieval.



classification of clients

DSS — Decision Support System.

Types of IR :-

- ① Boolean Retrieval system.
- ② Vector Space Model.

frequency is basic unit in this model.

Semantic analysis & Syntactic analysis

End of analysis

and tags needed - 20

## Role of an IR System

Thesauri → Set of string patterns. (topic Specific)

### Index :-

- ① first find out keywords.
- ② Index the Keywords
- ③ Retrieve the information

### Facet organisation

- Each document is described by a set of attributes Values (facet) .

## Basic IR Models

### ① Boolean retrieval models :-

Like query's in database.

displays if query is correct or else not.  
use 'and, or, not'

### ② Vector Space Model :-

$$d_1 = \{ dt_1, dt_2, dt_3, \dots \}$$

$$d_2 = \{ dt_5, dt_7, \dots, dt_n \}$$

$$d_3 = \{ \dots \}$$

- may be
- binary
  - frequency
  - weight

represented as keywords.

③ Cosine Similarity :-

We find similarity between the documents. Then score is given. And which documents displayed in descending order of score.

Tokenization and Indexing:-

② Precision

$$\text{no. of docs} = 1000$$

$$\text{no. of relevant docs} = 350$$

$$\text{no. of retrieved docs} = 500$$

$$\text{no. of relevant retrieved docs} = 200$$

$$\text{Precision} = \frac{\text{no. of relevant retrieved}}{\text{total no. of retrieved}} = \frac{200}{500}$$

$$\text{Recall} = \frac{\text{no. of relevant retrieved}}{\text{total no. of relevant}} = \frac{200}{350}$$

Precision :- calculating the accuracy of relevant docs retrieved

Recall :- calculation of coverage of relevant doc

$$F\text{-Measure} = \frac{2PR}{P+R}$$

Stemming :- Making a set of words into a group and selecting a root word.

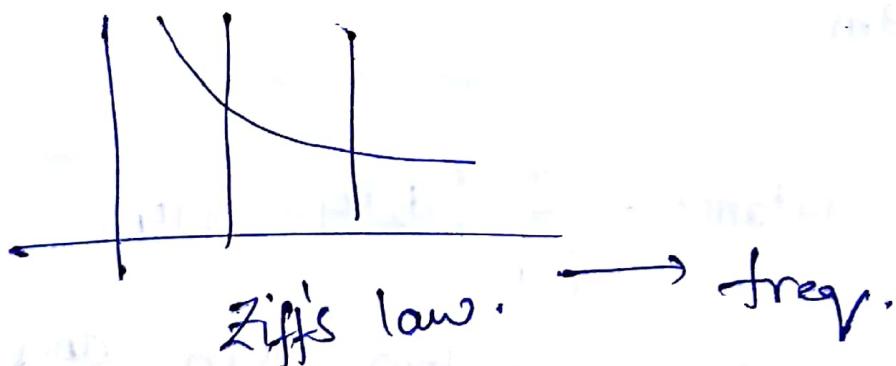
Ex:- Play, played, plays | go, went, gone

Tokenization :- Separate out all words

Normalization :- Need to normalize terms into same form.

## Indexing

Assignment:- Take 500 docs on 5 diff topics, ~~per~~ Tokenise all docs, find unique terms in all docs. How to extract keywords from unique terms? Stemming f - - - . find frequency of each term. and put a threshold. use hashmap. many imp terms may have low freq. Many unimp terms may have high freq.



Keywords



Doc No.

↓

National academy of Sciences [Pattern opp]

DT's also

## Similarity Measures / Metrics

How to find Similarity between two docs:

$d_1 \text{ ft}_1$

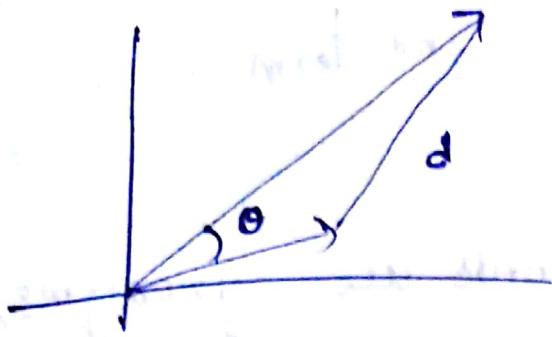
$d_2 \text{ ft}_1$

$d_3 \text{ ft}_1$

$d_m$

$$\text{Euclidean distance} = \sum_{i=1}^m (d_i \text{ ft}_i - d_m \text{ ft}_i)^2$$

If Euclidean distance is low, then docs are more similar.



Cosine Similarity :- if angle is small, the documents are more similar.

dot product of two vectors.

Jaccard  
Coefficient

Jaccard coefficient :- frequency doesn't play significant role in similarity of documents.

$$J(d_1, d_2) = \frac{|d_1 \cap d_2|}{|d_1| + |d_2| - |d_1 \cap d_2|}$$

hyponym :-



Colour is hyponym of red, blue, Green.

Meronym:- 1 term is kind of 2<sup>nd</sup> term.

Semantically related :- if words are Synonyms, Antonyms, hyponyms / hypernym, meronym.

### F-Score

$$F = \frac{2 \cdot P \cdot R}{P + R}, \quad P - \text{precision}$$

R - recall.

### Pearson correlation coefficient.

Measure of linear dependence between two variables

Values      -1 to +1



most  
dissimilar



most  
similar

## Assignment - 2

Query 1: Anna Hazare anti.

Query 1: Anna Hazare anti Land Acquisition Bill.

Download 100 docs on each topic and apply  
Similarity Measure.

### Problem 1

Masquerade User Problem Detection -

By detecting unusual commands by users and  
declaring masquerade.

	$C_1$	$C_2$	$\dots$	$\dots$	$\dots$	$C_{635}$
$d_1$	$d_{11}$	$d_{12}$	$\vdots$	$\vdots$	$\vdots$	$d_{1635}$
$d_2$	$d_{21}$	$d_{22}$	$\vdots$	$\vdots$	$\vdots$	$d_{2635}$
$d_3$	$d_{31}$	$d_{32}$	$\vdots$	$\vdots$	$\vdots$	$d_{3635}$
$d_4$	$d_{41}$	$d_{42}$	$\vdots$	$\vdots$	$\vdots$	$d_{4635}$
$d_5$	$d_{51}$	$d_{52}$	$\vdots$	$\vdots$	$\vdots$	$d_{5635}$
$d_6$	$d_{61}$	$d_{62}$	$\vdots$	$\vdots$	$\vdots$	$d_{6635}$
$d_7$	$d_{71}$	$d_{72}$	$\vdots$	$\vdots$	$\vdots$	$d_{7635}$
$d_8$	$d_{81}$	$d_{82}$	$\vdots$	$\vdots$	$\vdots$	$d_{8635}$
$d_9$	$d_{91}$	$d_{92}$	$\vdots$	$\vdots$	$\vdots$	$d_{9635}$
$d_{10}$	$d_{101}$	$d_{102}$	$\vdots$	$\vdots$	$\vdots$	$d_{10635}$
$d_{11}$	$d_{111}$	$d_{112}$	$\vdots$	$\vdots$	$\vdots$	$d_{11635}$
$d_{12}$	$d_{121}$	$d_{122}$	$\vdots$	$\vdots$	$\vdots$	$d_{12635}$
$d_{13}$	$d_{131}$	$d_{132}$	$\vdots$	$\vdots$	$\vdots$	$d_{13635}$
$d_{14}$	$d_{141}$	$d_{142}$	$\vdots$	$\vdots$	$\vdots$	$d_{14635}$
$d_{15}$	$d_{151}$	$d_{152}$	$\vdots$	$\vdots$	$\vdots$	$d_{15635}$
$d_{16}$	$d_{161}$	$d_{162}$	$\vdots$	$\vdots$	$\vdots$	$d_{16635}$
$d_{17}$	$d_{171}$	$d_{172}$	$\vdots$	$\vdots$	$\vdots$	$d_{17635}$
$d_{18}$	$d_{181}$	$d_{182}$	$\vdots$	$\vdots$	$\vdots$	$d_{18635}$
$d_{19}$	$d_{191}$	$d_{192}$	$\vdots$	$\vdots$	$\vdots$	$d_{19635}$
$d_{20}$	$d_{201}$	$d_{202}$	$\vdots$	$\vdots$	$\vdots$	$d_{20635}$
$d_{21}$	$d_{211}$	$d_{212}$	$\vdots$	$\vdots$	$\vdots$	$d_{21635}$
$d_{22}$	$d_{221}$	$d_{222}$	$\vdots$	$\vdots$	$\vdots$	$d_{22635}$
$d_{23}$	$d_{231}$	$d_{232}$	$\vdots$	$\vdots$	$\vdots$	$d_{23635}$
$d_{24}$	$d_{241}$	$d_{242}$	$\vdots$	$\vdots$	$\vdots$	$d_{24635}$
$d_{25}$	$d_{251}$	$d_{252}$	$\vdots$	$\vdots$	$\vdots$	$d_{25635}$
$d_{26}$	$d_{261}$	$d_{262}$	$\vdots$	$\vdots$	$\vdots$	$d_{26635}$
$d_{27}$	$d_{271}$	$d_{272}$	$\vdots$	$\vdots$	$\vdots$	$d_{27635}$
$d_{28}$	$d_{281}$	$d_{282}$	$\vdots$	$\vdots$	$\vdots$	$d_{28635}$
$d_{29}$	$d_{291}$	$d_{292}$	$\vdots$	$\vdots$	$\vdots$	$d_{29635}$
$d_{30}$	$d_{301}$	$d_{302}$	$\vdots$	$\vdots$	$\vdots$	$d_{30635}$
$d_{31}$	$d_{311}$	$d_{312}$	$\vdots$	$\vdots$	$\vdots$	$d_{31635}$
$d_{32}$	$d_{321}$	$d_{322}$	$\vdots$	$\vdots$	$\vdots$	$d_{32635}$
$d_{33}$	$d_{331}$	$d_{332}$	$\vdots$	$\vdots$	$\vdots$	$d_{33635}$
$d_{34}$	$d_{341}$	$d_{342}$	$\vdots$	$\vdots$	$\vdots$	$d_{34635}$
$d_{35}$	$d_{351}$	$d_{352}$	$\vdots$	$\vdots$	$\vdots$	$d_{35635}$
$d_{36}$	$d_{361}$	$d_{362}$	$\vdots$	$\vdots$	$\vdots$	$d_{36635}$
$d_{37}$	$d_{371}$	$d_{372}$	$\vdots$	$\vdots$	$\vdots$	$d_{37635}$
$d_{38}$	$d_{381}$	$d_{382}$	$\vdots$	$\vdots$	$\vdots$	$d_{38635}$
$d_{39}$	$d_{391}$	$d_{392}$	$\vdots$	$\vdots$	$\vdots$	$d_{39635}$
$d_{40}$	$d_{401}$	$d_{402}$	$\vdots$	$\vdots$	$\vdots$	$d_{40635}$
$d_{41}$	$d_{411}$	$d_{412}$	$\vdots$	$\vdots$	$\vdots$	$d_{41635}$
$d_{42}$	$d_{421}$	$d_{422}$	$\vdots$	$\vdots$	$\vdots$	$d_{42635}$
$d_{43}$	$d_{431}$	$d_{432}$	$\vdots$	$\vdots$	$\vdots$	$d_{43635}$
$d_{44}$	$d_{441}$	$d_{442}$	$\vdots$	$\vdots$	$\vdots$	$d_{44635}$
$d_{45}$	$d_{451}$	$d_{452}$	$\vdots$	$\vdots$	$\vdots$	$d_{45635}$
$d_{46}$	$d_{461}$	$d_{462}$	$\vdots$	$\vdots$	$\vdots$	$d_{46635}$
$d_{47}$	$d_{471}$	$d_{472}$	$\vdots$	$\vdots$	$\vdots$	$d_{47635}$
$d_{48}$	$d_{481}$	$d_{482}$	$\vdots$	$\vdots$	$\vdots$	$d_{48635}$
$d_{49}$	$d_{491}$	$d_{492}$	$\vdots$	$\vdots$	$\vdots$	$d_{49635}$
$d_{50}$	$d_{501}$	$d_{502}$	$\vdots$	$\vdots$	$\vdots$	$d_{50635}$
$d_{51}$	$d_{511}$	$d_{512}$	$\vdots$	$\vdots$	$\vdots$	$d_{51635}$
$d_{52}$	$d_{521}$	$d_{522}$	$\vdots$	$\vdots$	$\vdots$	$d_{52635}$
$d_{53}$	$d_{531}$	$d_{532}$	$\vdots$	$\vdots$	$\vdots$	$d_{53635}$
$d_{54}$	$d_{541}$	$d_{542}$	$\vdots$	$\vdots$	$\vdots$	$d_{54635}$
$d_{55}$	$d_{551}$	$d_{552}$	$\vdots$	$\vdots$	$\vdots$	$d_{55635}$
$d_{56}$	$d_{561}$	$d_{562}$	$\vdots$	$\vdots$	$\vdots$	$d_{56635}$
$d_{57}$	$d_{571}$	$d_{572}$	$\vdots$	$\vdots$	$\vdots$	$d_{57635}$
$d_{58}$	$d_{581}$	$d_{582}$	$\vdots$	$\vdots$	$\vdots$	$d_{58635}$
$d_{59}$	$d_{591}$	$d_{592}$	$\vdots$	$\vdots$	$\vdots$	$d_{59635}$
$d_{60}$	$d_{601}$	$d_{602}$	$\vdots$	$\vdots$	$\vdots$	$d_{60635}$
$d_{61}$	$d_{611}$	$d_{612}$	$\vdots$	$\vdots$	$\vdots$	$d_{61635}$
$d_{62}$	$d_{621}$	$d_{622}$	$\vdots$	$\vdots$	$\vdots$	$d_{62635}$
$d_{63}$	$d_{631}$	$d_{632}$	$\vdots$	$\vdots$	$\vdots$	$d_{63635}$
$d_{64}$	$d_{641}$	$d_{642}$	$\vdots$	$\vdots$	$\vdots$	$d_{64635}$
$d_{65}$	$d_{651}$	$d_{652}$	$\vdots$	$\vdots$	$\vdots$	$d_{65635}$
$d_{66}$	$d_{661}$	$d_{662}$	$\vdots$	$\vdots$	$\vdots$	$d_{66635}$
$d_{67}$	$d_{671}$	$d_{672}$	$\vdots$	$\vdots$	$\vdots$	$d_{67635}$
$d_{68}$	$d_{681}$	$d_{682}$	$\vdots$	$\vdots$	$\vdots$	$d_{68635}$
$d_{69}$	$d_{691}$	$d_{692}$	$\vdots$	$\vdots$	$\vdots$	$d_{69635}$
$d_{70}$	$d_{701}$	$d_{702}$	$\vdots$	$\vdots$	$\vdots$	$d_{70635}$
$d_{71}$	$d_{711}$	$d_{712}$	$\vdots$	$\vdots$	$\vdots$	$d_{71635}$
$d_{72}$	$d_{721}$	$d_{722}$	$\vdots$	$\vdots$	$\vdots$	$d_{72635}$
$d_{73}$	$d_{731}$	$d_{732}$	$\vdots$	$\vdots$	$\vdots$	$d_{73635}$
$d_{74}$	$d_{741}$	$d_{742}$	$\vdots$	$\vdots$	$\vdots$	$d_{74635}$
$d_{75}$	$d_{751}$	$d_{752}$	$\vdots$	$\vdots$	$\vdots$	$d_{75635}$
$d_{76}$	$d_{761}$	$d_{762}$	$\vdots$	$\vdots$	$\vdots$	$d_{76635}$
$d_{77}$	$d_{771}$	$d_{772}$	$\vdots$	$\vdots$	$\vdots$	$d_{77635}$
$d_{78}$	$d_{781}$	$d_{782}$	$\vdots$	$\vdots$	$\vdots$	$d_{78635}$
$d_{79}$	$d_{791}$	$d_{792}$	$\vdots$	$\vdots$	$\vdots$	$d_{79635}$
$d_{80}$	$d_{801}$	$d_{802}$	$\vdots$	$\vdots$	$\vdots$	$d_{80635}$
$d_{81}$	$d_{811}$	$d_{812}$	$\vdots$	$\vdots$	$\vdots$	$d_{81635}$
$d_{82}$	$d_{821}$	$d_{822}$	$\vdots$	$\vdots$	$\vdots$	$d_{82635}$
$d_{83}$	$d_{831}$	$d_{832}$	$\vdots$	$\vdots$	$\vdots$	$d_{83635}$
$d_{84}$	$d_{841}$	$d_{842}$	$\vdots$	$\vdots$	$\vdots$	$d_{84635}$
$d_{85}$	$d_{851}$	$d_{852}$	$\vdots$	$\vdots$	$\vdots$	$d_{85635}$
$d_{86}$	$d_{861}$	$d_{862}$	$\vdots$	$\vdots$	$\vdots$	$d_{86635}$
$d_{87}$	$d_{871}$	$d_{872}$	$\vdots$	$\vdots$	$\vdots$	$d_{87635}$
$d_{88}$	$d_{881}$	$d_{882}$	$\vdots$	$\vdots$	$\vdots$	$d_{88635}$
$d_{89}$	$d_{891}$	$d_{892}$	$\vdots$	$\vdots$	$\vdots$	$d_{89635}$
$d_{90}$	$d_{901}$	$d_{902}$	$\vdots$	$\vdots$	$\vdots$	$d_{90635}$
$d_{91}$	$d_{911}$	$d_{912}$	$\vdots$	$\vdots$	$\vdots$	$d_{91635}$
$d_{92}$	$d_{921}$	$d_{922}$	$\vdots$	$\vdots$	$\vdots$	$d_{92635}$
$d_{93}$	$d_{931}$	$d_{932}$	$\vdots$	$\vdots$	$\vdots$	$d_{93635}$
$d_{94}$	$d_{941}$	$d_{942}$	$\vdots$	$\vdots$	$\vdots$	$d_{94635}$
$d_{95}$	$d_{951}$	$d_{952}$	$\vdots$	$\vdots$	$\vdots$	$d_{95635}$
$d_{96}$	$d_{961}$	$d_{962}$	$\vdots$	$\vdots$	$\vdots$	$d_{96635}$
$d_{97}$	$d_{971}$	$d_{972}$	$\vdots$	$\vdots$	$\vdots$	$d_{97635}$
$d_{98}$	$d_{981}$	$d_{982}$	$\vdots$	$\vdots$	$\vdots$	$d_{98635}$
$d_{99}$	$d_{991}$	$d_{992}$	$\vdots$	$\vdots$	$\vdots$	$d_{99635}$
$d_{100}$	$d_{1001}$	$d_{1002}$	$\vdots$	$\vdots$	$\vdots$	$d_{100635}$
$d_{101}$	$d_{1011}$	$d_{1012}$	$\vdots$	$\vdots$	$\vdots$	$d_{101635}$
$d_{102}$	$d_{1021}$	$d_{1022}$	$\vdots$	$\vdots$	$\vdots$	$d_{102635}$
$d_{103}$	$d_{1031}$	$d_{1032}$	$\vdots$	$\vdots$	$\vdots$	$d_{103635}$
$d_{104}$	$d_{1041}$	$d_{1042}$	$\vdots$	$\vdots$	$\vdots$	$d_{104635}$
$d_{105}$	$d_{1051}$	$d_{1052}$	$\vdots$	$\vdots$	$\vdots$	$d_{105635}$
$d_{106}$	$d_{1061}$	$d_{1062}$	$\vdots$	$\vdots$	$\vdots$	$d_{106635}$
$d_{107}$	$d_{1071}$	$d_{1072}$	$\vdots$	$\vdots$	$\vdots$	$d_{107635}$
$d_{108}$	$d_{1081}$	$d_{1082}$	$\vdots$	$\vdots$	$\vdots$	$d_{108635}$
$d_{109}$	$d_{1091}$	$d_{1092}$	$\vdots$	$\vdots$	$\vdots$	$d_{109635}$
$d_{110}$	$d_{1101}$	$d_{1102}$	$\vdots$	$\vdots$	$\vdots$	$d_{110635}$
$d_{111}$	$d_{1111}$	$d_{1112}$	$\vdots$	$\vdots$	$\vdots$	$d_{111635}$
$d_{112}$	$d_{1121}$	$d_{1122}$	$\vdots$	$\vdots$	$\vdots$	$d_{112635}$
$d_{113}$	$d_{1131}$	$d_{1132}$	$\vdots$	$\vdots$	$\vdots$	$d_{113635}$
$d_{114}$	$d_{1141}$	$d_{1142}$	$\vdots$	$\vdots$	$\vdots$	$d_{114635}$
$d_{115}$	$d_{1151}$	$d_{1152}$	$\vdots$	$\vdots$	$\vdots$	$d_{115635}$
$d_{116}$	$d_{1161}$	$d_{1162}$	$\vdots$	$\vdots$	$\vdots$	$d_{116635}$
$d_{117}$	$d_{1171}$	$d_{1172}$	$\vdots$	$\vdots$	$\vdots$	$d_{117635}$
$d_{118}$	$d_{1181}$	$d_{1182}$	$\vdots$	$\vdots$	$\vdots$	

What should be size of similarity matrix.

$$Y(c_i, c_j) = \frac{\sum_{k=1}^{50} f_{ik} f_{jk} - \overline{f_{ik}} \overline{f_{jk}}}{\sqrt{\left( \sum_{k=1}^{50} f_{ik}^2 - \left( \frac{\sum_{k=1}^{50} f_{ik}}{n} \right)^2 \right) \left( \sum_{k=1}^{50} f_{jk}^2 - \left( \frac{\sum_{k=1}^{50} f_{jk}}{n} \right)^2 \right)}}$$

Any command pair having highest similarity score are removed from matrix and are considered as one command.

Our threshold of joining commands are 5.

Now calculate  $c_1 c_3 c_6$ ,  $c_2 c_3 c_6$ ,  $c_4 c_3 c_6$ ,  $c_5 c_3 c_6$ .

Again repeat the above process. ① Calc Max, ② Join Commands.

frequency of  $c_3 c_6$  can be sum, avg, max or min of frequencies. [expt].

We also need to put a threshold to the score of similarity.

This is called Tuning Parameters.  
(the process of thresholding).

$P_1, P_2, \dots, P_{40} \xrightarrow{\text{Similarity}} \text{Patterns of commands obtained}$

Now we divide 10,000 commands into 100 blocks of 100 commands each.

Now check for each block that these patterns exists in that block.

We need to put a threshold, like if 70+ commands match with pattern, then user is valid or else manipulated.

$$b_1, b_2, \dots, b_{100}$$

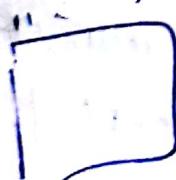
$d_1$

$d_2$

$d_3$

0 → if block is normal

1 → if manipulated



→ Dynamic programming can be applied to Similarity Matrix for calculating similarity measure of grouped commands.

## Text Summarization:-

Why do we summarise :— If a student of history is to be evaluated, then text summarisation can be used.

For keywords finding.

A Single document.

$t \rightarrow$  unique terms  
 $p \rightarrow$  Paragraphs.

	$t_1$	$t_2$	$\dots$	$t_m$
$p_1$	$p_{1f_{t_1}}$	$p_{1f_{t_2}}$	$\dots$	$p_{1f_{t_m}}$
$p_2$	1			
$p_3$		1		
$p_n$			$\dots$	$p_nf_{t_m}$

Now we find similarity between paragraphs, using Similarity measure.

$$P_1 \quad P_2 \quad - \quad - \quad - \quad P_n$$

$$P_1 \quad P_1 P_2 \quad | \quad | \quad | \quad P_1 P_n$$

$$P_2 \quad | \quad | \quad | \quad |$$

$$\vdots \quad | \quad | \quad | \quad |$$

$$P_n.$$

In place of using paragraphs, Sentences are more appropriate.

### Weighting Methods:-

#### tf-idf:-

$$\underset{\text{doc no}}{i,j} = \underset{\text{term.}}{t_{f_{i,j}}}$$

$t_f \rightarrow$  term frequency  
 $\text{idf} \rightarrow$  inverse document frequency

$$W_{t_{f_{i,j}}} = t_{f_{i,j}} \times \log \frac{N}{\underset{\text{document}}{df_{t_j}}}$$

$N \rightarrow$  total number of documents.

document frequency.

document frequency  $< N$ .

<u>Ex1</u>	Term	df <sub>t</sub>	idf <sub>t</sub>	total no. of words
Car	18,165	1.65		docs (N) = 806,791
auto	6723	2.08		Renters data Set.
insurance	19241	1.65		
best	25235	1.5		

<u>Ex2</u>	Term	DOC1	DOC2	DOC3
Car	27	4	24	
auto	3	33	0	
insurance	0	33	29	
best.	14	0	17	

## Variant of tf-idf

$$r_{ft} = \frac{f_{tij}}{\max \cdot x d_{f_{tij}}}$$

$$\max \cdot x d_{f_{tij}}$$

$$r_{ftij} = \begin{cases} 1 + \log f_{tij} & f_{tij} > 0 \\ 0 & \text{Otherwise} \end{cases}$$

$$idf = \log \frac{N}{dft}$$

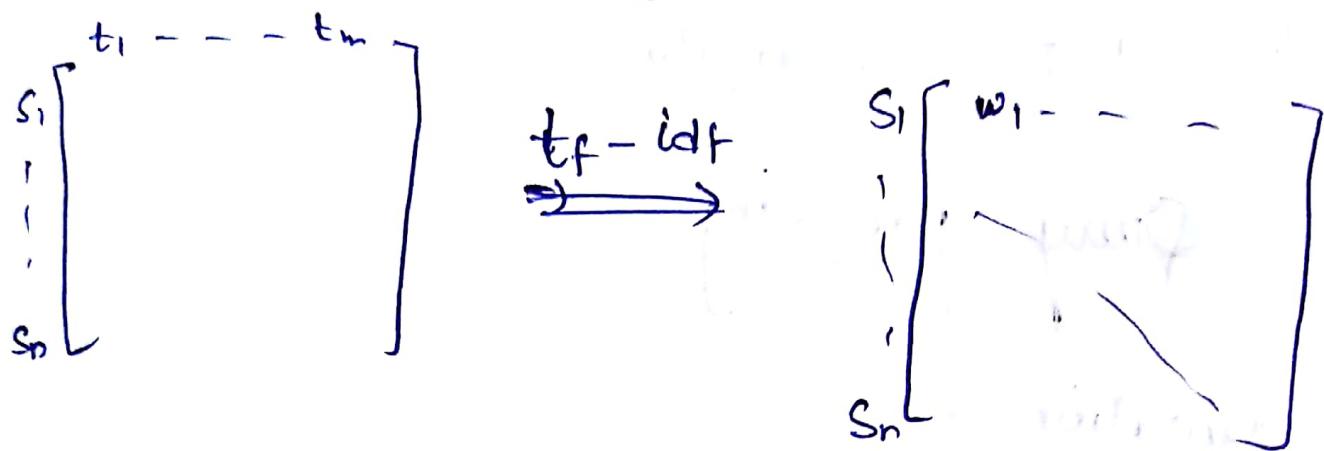
$$b(idf) = \max \left\{ 0, \frac{\log \frac{N-dft}{dft}}{1} \right\}$$

Process

we need to summarize our document

- ① Unique terms in the whole document
- ② Represent the document in terms of sentences.
- ③ Find the frequency matrix

④ Apply tf-idf  $\Rightarrow$  weight matrix



⑤ Apply any similarity ~~Metric~~ Metric between sentences

$$\begin{matrix} s_1 & \dots & s_n \\ \left[ \begin{matrix} s_1 \\ \vdots \\ s_n \end{matrix} \right] & \xrightarrow{\text{similarity metric}} & \text{Score} \\ \left[ \begin{matrix} s_1 s_2 & \dots & s_1 s_n \\ \vdots & \ddots & \vdots \\ s_n s_1 & \dots & s_n s_n \end{matrix} \right] & = \Sigma & \\ & = \Sigma & \\ & = \Sigma & \\ & = \Sigma & \end{matrix}$$

⑥ Find the similarity score with all other sentences.

⑦ Arrange all the sentences in descending order of Si

### Assignment - 3

① Above text summarization.

### Query processing

Basic notation:-

$$D = \{d_1, d_2, d_3, \dots\}$$

$$T = \{t_1, t_2, \dots\}$$

We have respective weights of each term.

$$W = \{w_{ij}\} \quad i \rightarrow \text{term} \\ j \rightarrow \text{doc.}$$

$$d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$$

Weight of each term in a doc.

in this case! -

Recall = 1

precision =  $\frac{1}{3}$ .

To increase precision we need to take care of Synonyms & Keywords of Query. Hence query must be expanded.

Query Expansion:-

~~Approaches~~ for Every Query Expansion:-

Local methods : use docs that are retrieved by original query

- ① Relevance feedback  $\rightarrow$  what user gives
- ② Pseudo relevance feedback  $\rightarrow$  In place of taking user feedback , what are top 10 docs are retrieved we take those as feedback.

Global methods independent of query & results from it

① Thesaurus or WordNet →

② concept clouds.

for every word find a SynSet and expand the

query

concept cloud → we get similar words for a specific word.

Relevance feedback (with user)

Query Expansion

Term reweighting

## Query Expansion & Term reweighting in Vector space.

### Model ↴

For a query  $Q$ .

- $D_r$ : Set of relevant doc among retrieved doc.
- $D_n$ : Set of non relevant doc among retrieved doc.
- $C_r$ : Set of relevant doc among all doc in collection

$$q_{opt} = \frac{1}{|C_r|} \sum_{d_j \in C_r} d_j - \frac{1}{N - |C_r|} \sum_{d_j \notin C_r} d_j$$

$q_i \rightarrow$  initial weight of query

### Rochio formula ↴

$$q_{i+1} = \alpha q_i + \frac{\beta}{|D_r|} \sum_{d_j \in D_r} d_j - \frac{\gamma}{|D_n|} \sum_{d_j \in D_n} d_j$$

hence  $\alpha \geq 1 \rightarrow$  initial formulation

$$\gamma < < \beta .$$

Positive relevance feedback ( $\gamma = 0$ )

## Generation of Concept cloud—

$$d_1 t_1 + d_2 t_2 + \dots + d_n t_n = 0$$

The diagram illustrates the equation above with terms  $t_1, t_2, t_3, \dots, t_n$  arranged horizontally. Each term is multiplied by a coefficient  $d_1, d_2, \dots, d_n$  respectively, shown as vertical arrows pointing from the coefficients to their respective terms. The entire sum is set equal to zero.

$$J(A, B) = \frac{H(A \cup B) - H(A \cap B)}{|A \cup B|}$$

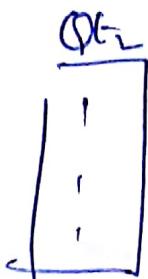
HIA) — count of ~~term A~~ in ~~all docs~~, which term A exist  
in all docs

$A \cap B$  — co-occurrence of A & B in all docs. (count of docs)

Concept cloud: each node is a synonym of the term



m terms



n terms

for every  $i$  in  $Qt_1$  calculate Jaccard similarity with every  $j$  in  $Qt_2$

Take average of all the Jaccard scores and assign it to  $i^{th}$  word.

~~consensus~~

Consensus

# Performance Evaluation of Information Retrieval Systems

Systems :-

For a given set of docs (gold standard)

Threshold values of Precision & recall  
would be precision = recall.

R-Precision ← At  $R^{\text{th}}$  position, we need to stop

$$\text{F-Measure} := \frac{2PR}{P+R}$$

$$\text{F-Measure} = \frac{(1+\beta^2) PR}{\beta^2 P + R} = \frac{1+\beta^2}{\frac{\beta^2}{P} + \frac{1}{R}}$$

	relevant	not relevant
Retrieved	tp	fp
Not retrieved	fn	fn.

$$P = tp / (tp + fp)$$

$$R = tp / (tp + fn)$$

$$\text{Accuracy} = (tp + tn) / (tp + fp + fn + tn)$$

	No	Yes
Negative (No)	48 (TN)	1 (FP)
Positive (Yes)	(FN) ①	1 (TP)

There are 50 emails of which 49 are <sup>not</sup> useful and 1 mail is useful.

TP — we are saying yes, actually yes

TN — " " " NO, " NO

FP — " " " ~~yes~~ ~~NO~~ " also ~~yes~~ NO

FN — " " " ~~NO~~, " ~~yes~~ ~~NO~~ " yes

Accuracy is not a dependable criterion for a System.

If  $f_p$  is high, precision should be considered.

If  $f_N$  is high, Recall should be considered.

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{tp + fn}$$

Draw the matrix for Sick & non Sick.

		Predicted	
		Yes	No
Actual	Sick (Yes)	TP	FP
	Non Sick (No)	FN	TN

If Precision & Recall are both imp. for any case we go for F-Measure.

doc ID	Judge 1	Judge 2
1	0	0
2	0	0
3	1	1
4	1	1
5	1	0
6	1	0
7	1	0
8	1	0

9	0	1
10	0	1
11	0	1
12	0	1

0 → Not relevant

1 → relevant.

Calculate Accuracy, Precision, Recall, F-measure

Case-1— when we consider a doc is relevant if both the judges are saying relevant

Case-2 If either of judges saying relevant consider relevant.

Actual relevant - 4, 5, 8, 9.

Case - 1

relevant

Non-relevant

relevant

1 [TP]

3. [Fp]

Non-relevant

1 [FN]

7 [TN]

$$\text{accuracy} = \frac{8}{12} = 2/3$$

$$\text{Precision} = \frac{1}{2} = .$$

$$\text{Recall} = \frac{1}{4}$$

$$\text{F-Measure} = \frac{2 \times (\frac{1}{2}) \times (\frac{1}{4})}{\frac{3}{4}} = \frac{1}{3}.$$

Case - 2

relevant

non relevant

relevant

4 [TP]

5 [Fp]

non relevant

0 [FN]

2 [TN]

$$\text{accuracy} = \frac{6}{12} = 1/2$$

$$\text{Recall} = \frac{4}{10} = 2/5$$

$$\text{Precision} = \frac{4}{4} = 1$$

$$\text{F-Measure} =$$

$$\frac{2 \times \frac{2}{5}}{\frac{7}{5}} = \frac{4}{7}$$

F-measure, can we use it as Similarity [Monday] measure.  
in document Text Summarization.

Ex:- declaring a valid bank transaction as fraud bank transaction ??? - How severe it is.

Ex:-

		relevant	non-relevant
relevant	retrieved $[T_p]$	retrieved $[F_p]$	
non-relevant	not retrieved $[F_N]$	not retrieved $[T_N]$	

→ retrieved but not relevant  
not retrieved.  
but relevant

Ex:-

Sachin AND Pilot — Precision is high

Sachin OR Pilot — Recall is high.

Exr 100 docs., ~~Search query~~ IR Systems returns  
o Particular paragraph of a doc relevant to Search  
query.

levels of index  $\rightarrow$  paragraph indexing, Sentence  
indexing  
this will increase Recall.

ExL F-measure as similarity measure.

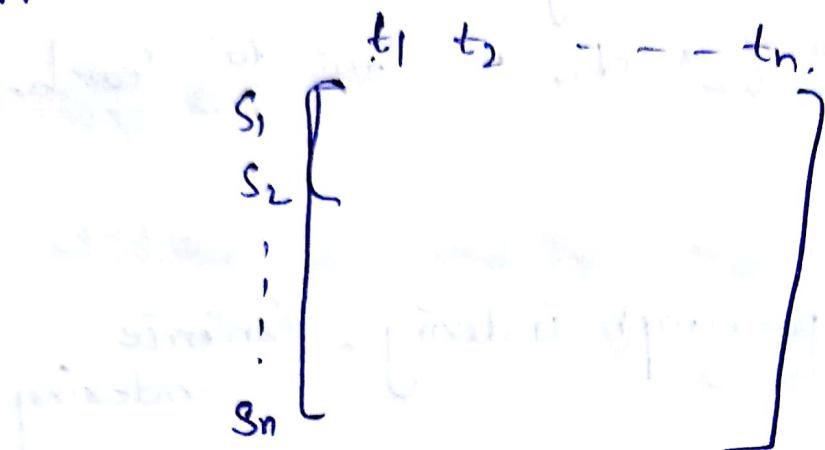
$$t_1 \ t_2 \ \dots \ t_n$$
$$\begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix}$$
$$\left[ \begin{array}{c} f_{12} \\ f_{13} \\ \vdots \\ f_{ij} \\ \vdots \\ f_{n1} \end{array} \right]$$
$$\frac{s_1 \cap s_2}{s_1} = \frac{s_1 \cap s_2}{s_1}$$
$$\frac{s_1 \cap s_2}{s_2} = \frac{s_1 \cap s_2}{s_2}$$

$$F(s_1, s_2) = \text{my formula.}$$

$$P = \left[ \frac{s_1 \cap s_2}{s_1 \cup s_2} \right]$$

$$R = \left[ \frac{s_1 \cap s_2}{s_1} \cup \frac{s_1 \cap s_2}{s_2} \right]$$

Official formula :-



$$P(s_i, s_j) = \frac{|S_i \cap S_j|}{|S_i|}$$

$$R(s_i, s_j) = \frac{|S_i \cap S_j|}{|S_j|}$$

$$F(s_i, s_j) = \frac{2|S_i \cap S_j|}{|S_i| + |S_j|}$$

Similarity between human generated summary & automatic  
Summary.

$$P = \frac{|S_{\text{man}} \cap S_{\text{auto}}|}{|S_{\text{man}}|}$$

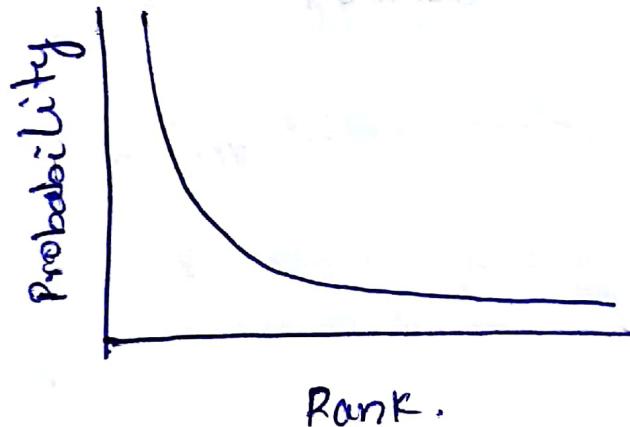
$$R = \frac{|S_{\text{man}} \cap S_{\text{auto}}|}{|S_{\text{auto}}|}$$

$$F = \frac{2PR}{P+R}$$

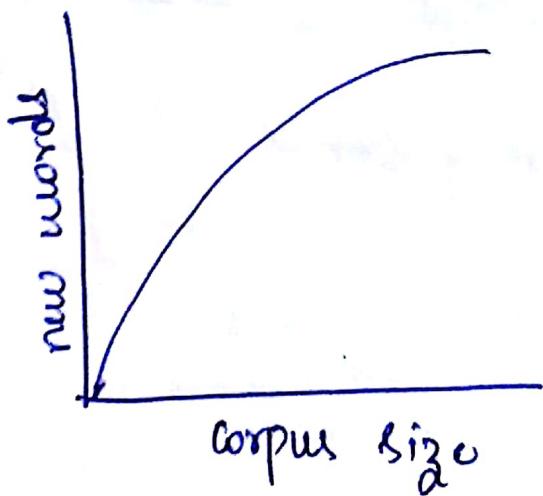
Text statistics:-

Ziff's law!:-

Rank all the words frequency and then  
 $r^{th}$  common term is inversely proportional to



Heap's Law:-



$$\text{corpus size} = K \times n^B$$

Can we apply Heaps law to text document.

Make a new data set. Needed at 3<sup>rd</sup> & 4<sup>th</sup> unit.

Query & question generation.

### Mid term

① Intro to Ir.

② Boolean      Vector space      Probabilistic model.

Type of Similarity Measures, comparison. with logic

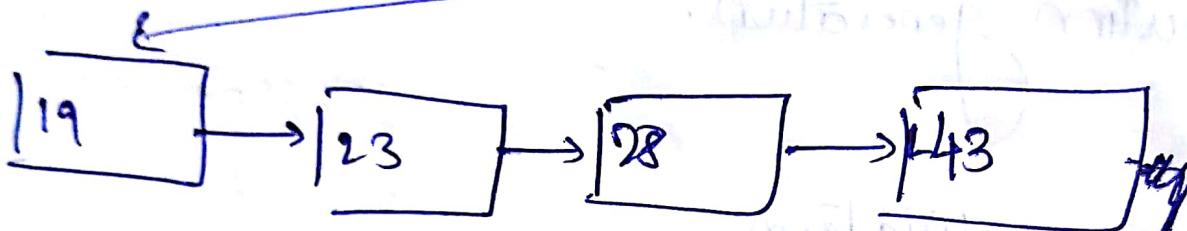
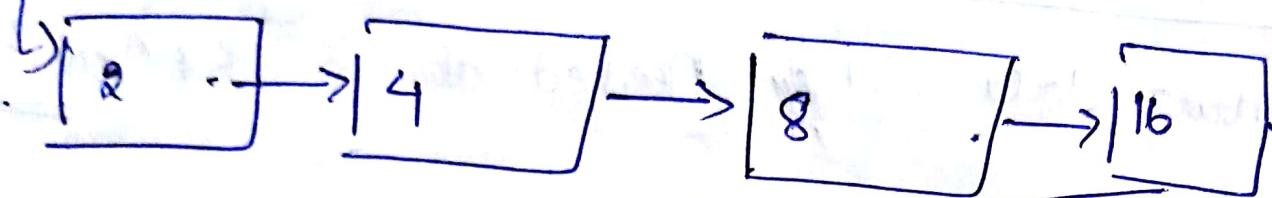
Query expansion.      word cloud  
                          word net  
                          relevance of pseudo  
                          relevance feedback

Index.      why  
                 what types. [Y-word index, inverted index]

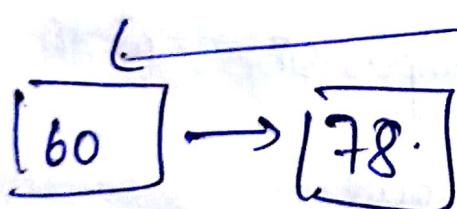
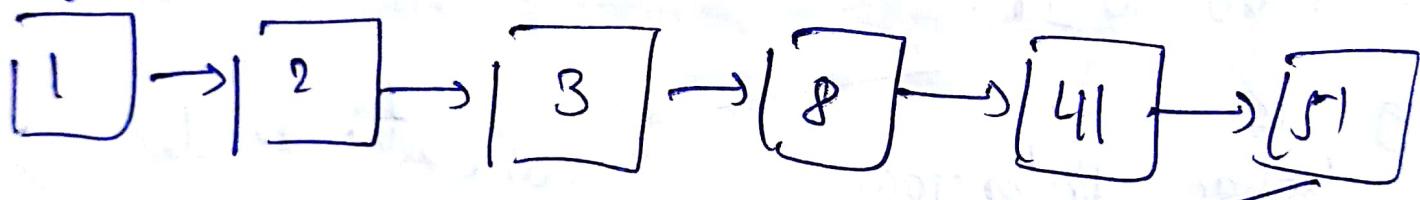
# Performance

## Indexing Numerical examples

Brute



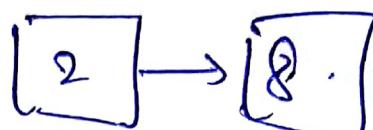
Caesar.



# Query Brutus AND CAESER

intersection of both ~~index~~. posting list.

outcomes — Q, S.



8

efficiency :  $O(M+N)$

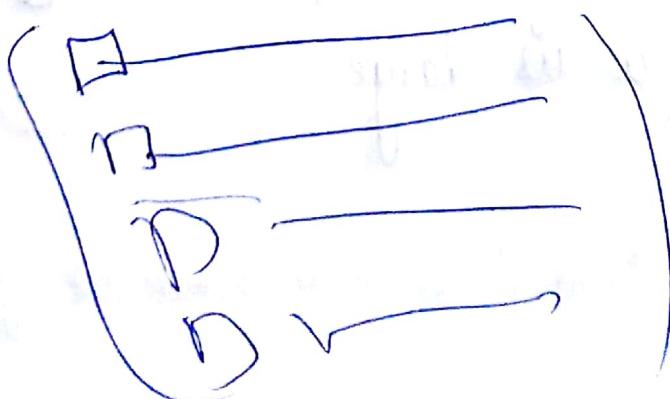
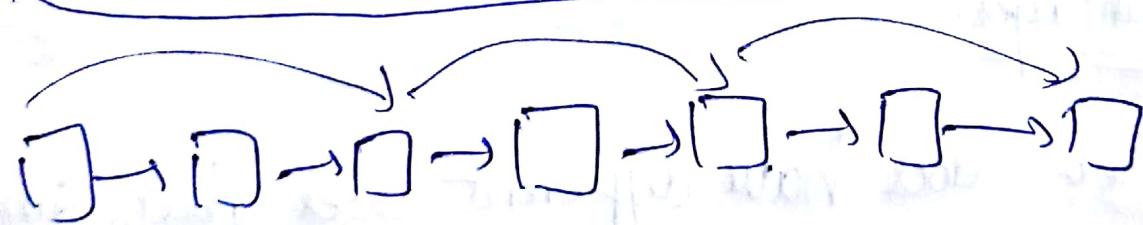
[skip pointer].

4  
2

Improve efficiency

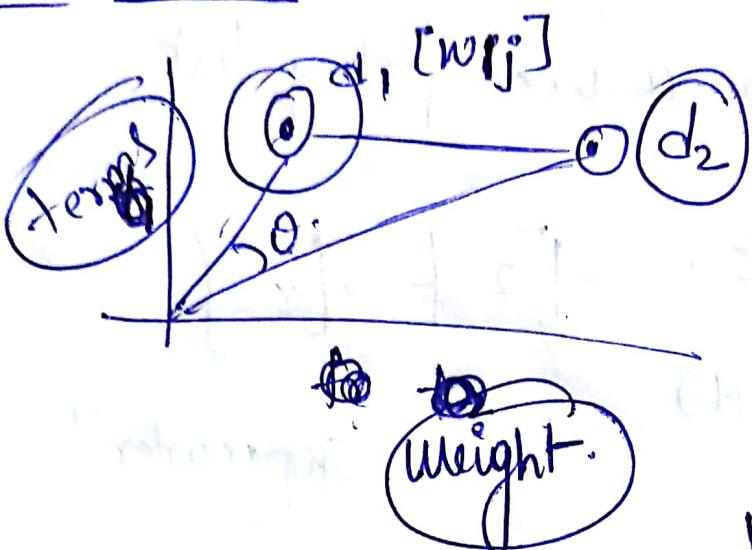
$$SP = \begin{cases} \min(M, N) & M < N \\ \frac{M}{2} & \text{(displacement)} \end{cases}$$

Sample questions Practise



Euclidean distance :-

$t_{i,j}^o$  = weight of  
i<sup>th</sup> term  
on j<sup>th</sup> doc.



$$\Sigma d(d_1, d_2) =$$

$$\sqrt{\sum (t_i - t_j)^2}$$

advantages :-

more dist = more diff [not efficient].

efficient in case if more vectors to domain, it won't affect the relation.

disadvantages :-

If two docs have different sizes, but similar hence distance is large.

We will apply ED, Cosine combinely and then decide docs are similar or not.

Sol<sup>n</sup> may be direction / magnitude (at) my sol<sup>n</sup>.

Cosine :-

Adv :- disadvantage of ED

disadv :- ignores Magnitudes.

length may be an issue if not normalized. if normalized then length will not effect.

Normalization process =  $/\text{Max}$ ,  $\log()$  - etc.

Jaccard coefficient:-

intersection present, only works. — adv.

in case of relatedness it fails — dis adv.

PCC :- [ -1 to 1 ] relative distance  
in recommendation system

it doesn't work on Categorical data.

works only when linear dependence  
between two does.

linear  $\longrightarrow$  Pollution & weather forecast.

Based on computation & power required :-

(\*) PCC

① cosine

② ED.

③ Pcc.

④ Jaccard.

④ Jaccard.

↓  
my

descending  
order



① PCC

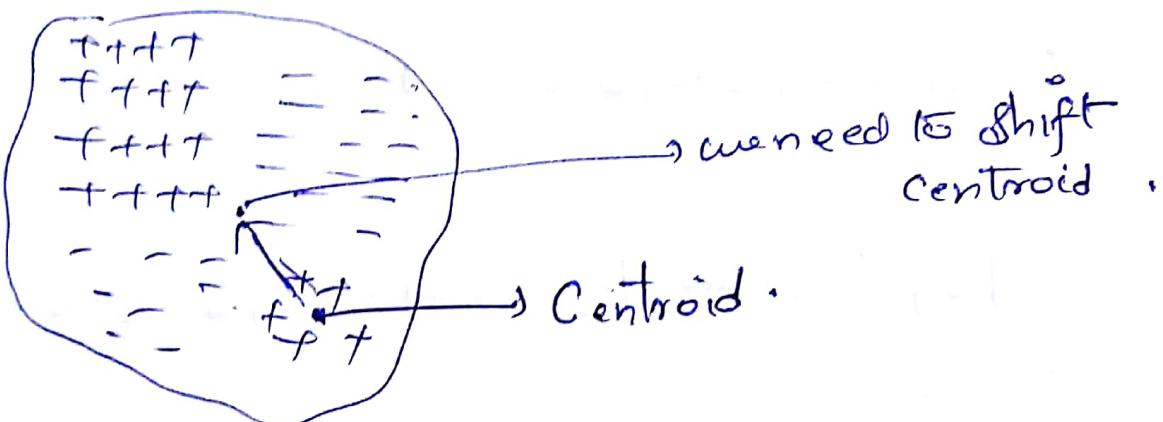
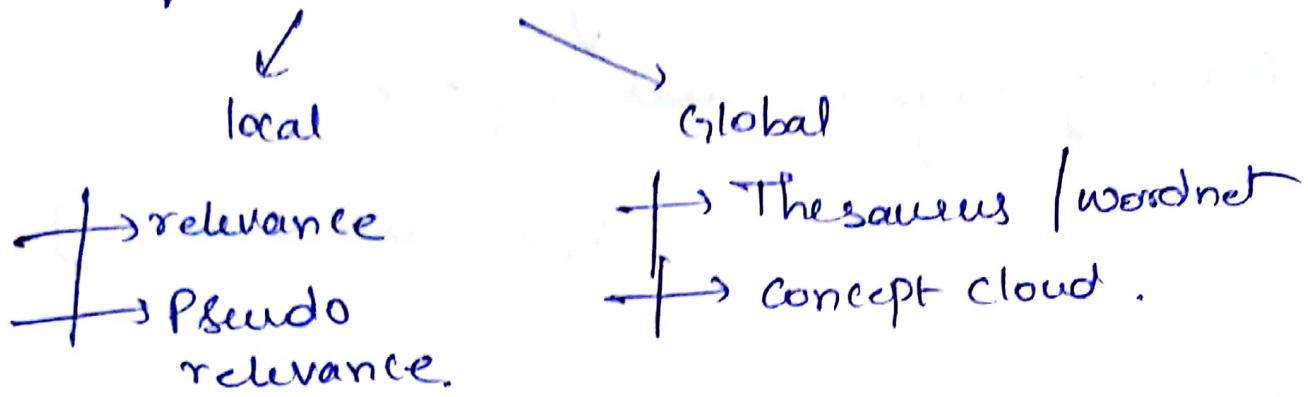
② BD.

③ Cosine

④ Jaccard

↓  
actual

## Query Expansion:-



to improve recall.

$$q_{opt} = \alpha q_i^o + \frac{\beta}{|D_r|} \sum q_i^o d_{wi} - \frac{\gamma}{|D_m|} \sum q_i^o d_{wi}$$

query:- news about presidential camp (1,1,1)

- $D_1$  = (news about - - -) (1.5, 0.1, 0, 0, 0 - -)
- $D_2$  = News about organic food campaign (1.5, 0.1, 2.0, 2.0, 0 - -)
- +  $D_3$  = News of presidential candidate (1.5, 0, 3.0, 2.0, 0, 0 - -)

+  $D_4$  = News of presidential campaign  $[1.5, 0, 3.0, 2.0, 0, 0]$

-  $D_5$  = News of organic food campaign  $[1.5, 0.0, 6.0, 2.0, \dots]$

$D_3$ :  $(1.5, 3.0, 2.0, 0, 0, \dots)$

$D_4$ :  $(1.5, 0, 4.0, 2.0, 0, 0, \dots)$

$(+)(1.5 + 1.5)/2, (3.0 + 4.0)/2, (2.0 + 2.0)/2, 0, 0, \dots)$

=  $(1.5, 3.5, 2.0, 0, \dots)$

$(\ominus) = [1.5, 0.67, 0, 2.6, 1.3, 0].$

Antroid =  $[1, 1, 1, 1]$ .

$\Phi_{\text{initial}} = [1, 1, 1, 1]$ .

$\theta_{\text{opt}} = [\alpha x_1 + \beta x_1 \cdot 1.5 - \gamma x_1 \cdot 1.5, \alpha x_1 + \frac{\beta \cdot 3.5}{2 \times 0.67} - \gamma x_1 \cdot 1.5]$

google search :- Adv's & disadu of Jaccard Similarity

in Pseudo relevance feedback, we can't increase recall.

Global  $\rightarrow$

