* Learn Python *.

CIF

Assignment / Paper reading / Research paper —— 40%.

Mid term —— 20%.

End term —— 40%.

Book:- Christopher D. Manning -

Intro to IR.

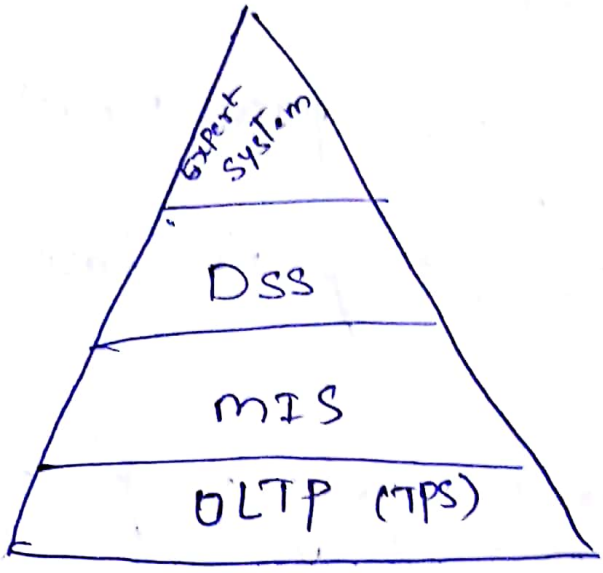
Student meeting hrs:— Tuesday :- 4 to 6.

Wednesday:- 4 to 6.

Friday :- after 3 o clock

Data — Processing — Information

→ collection
→ classification
→ coding
→ sorting
→ validation
→ verification
→ calculation
→ 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 & Syntantic analysis

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 :-

$d_1 = \{dt_1, dt_2, dt_3 \ldots dt_n\}$ → may be

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

$d_3 = \{ - - - - \}$.

- 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 :-~~

no. of docs = 1000

no. of relevant docs = 350

no. of retrieved docs = 500

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 docs

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

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

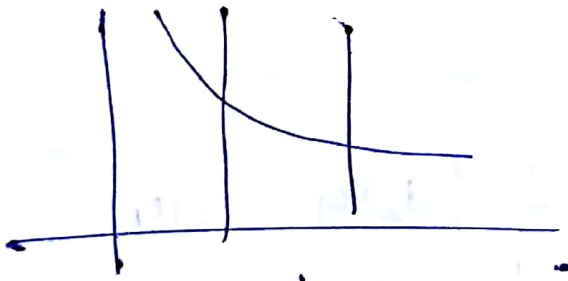
Ex:- play, played, plays | go, went, gone

Tokenization :- Seperate out all words

Normalization :- Need to normalize terms into same form.

Indexing

Assignment:- Take 500 docs on 5 diff topics, Tokenise all docs, find unique terms in all docs. How to extracts keywords from unique terms? Stemming ff - - ——— . 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.



Ziff's law. → freq.

Keywords ↓ DocNo ↓.

*National academy of Sciences [Intern opp]

IDF's also

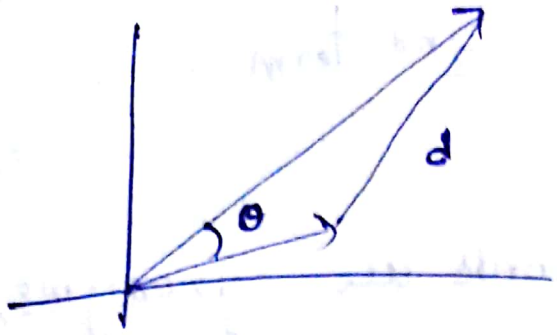
Similarity Measures / Metrics

How to find Similarity between two docs:

	t_1	t_2	t_3	----	t_n
d_1	ft_1	ft_2	ft_3	----	
d_2	ft_1	ft_2	----		
d_3					
:					
d_m					

Euclidean distance $= \sum_{i=1}^{m} (d_2 ft_1 - d_1 ft_1)^2$

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



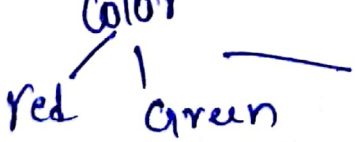
Cosine Similarity :- if angle is small, the docs are more similar.

dot product of two vectors.

Jaccard Coefficient :- frequency doesn't play significant role in similarity of docs

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

hyponym :-



Colour is hyponym of red, blue, Green.

Meronym:- 1 term is kind of 2nd term.

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

F-Score:-

$$F = \frac{2 \cdot p \cdot r}{p + r}$$

p - 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~~ .

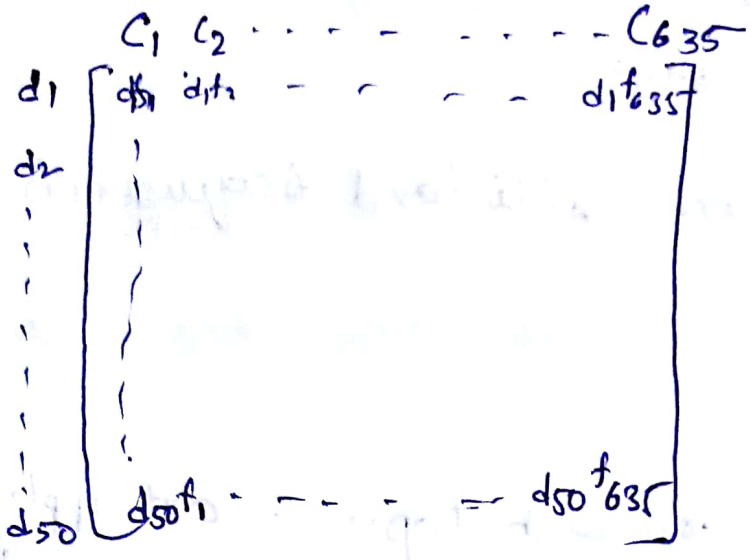
Query 1 - Anna Hazare anti land Acquisition Bill.

Download 100 docs on each topic . and apply Similarity measure.

Problem 1

Masquuade user ~~Problem~~ - Detection -

By detecting unusual commands by users and declaring masquerade.



Commands that are not used much → outliers.

How to generate signature of each user using above matrix.

Objective is to find similarity between commands, and generate similarity matrix.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_1		C_1C_2	C_1C_3	C_1C_4	C_1C_5	C_1C_6	C_1C_7
C_2	"		C_2C_3	C_2C_4	C_2C_5	C_2C_6	C_2C_7
C_3	"	"		C_3C_4	C_3C_5	C_3C_6	C_3C_7
C_4	"	"			C_4C_5	C_4C_6	C_4C_7
C_5						C_5C_6	C_5C_7
C_6							C_6C_7
C_7							

Use any similarity measure to generate matrix.

What should be size of Similarity matrix.

$$Y(C_i, C_j) = \frac{\sum_{k=1}^{50} f_{i,k} f_{j,k} - \frac{\sum_{k=1}^{50} f_{j,k} \sum_{k=1}^{50} f_{i,k}}{n}}{\sqrt{\left(\sum_{k=1}^{50} f_{ik}^2 - \frac{\left(\sum_{k=1}^{50} f_{ik}\right)^2}{n}\right)\left(\sum_{k=1}^{50} f_{jk}^2 - \frac{\left(\sum_{k=1}^{50} f_{jk}\right)^2}{n}\right)}}$$

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 $C_1C_3C_6$, $C_2C_3C_6$, $C_4C_3C_6$, $C_5C_3C_6$.

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

frequency of C_3C_6 can be Sum, avg, max or min of frequencies. [expt].

We also need to put a threshold to Score of Similarity.

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

$P_1, P_2 \cdots\cdots P_{40}$ → Patterns of commands obtained.

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

Now check 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 masquerade

	b_1	b_2	$\cdots$	b_{100}
d_1				
d_2				
⋮				
d_r				

0 → if block is normal

1 → if masquerade

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

Text Summarization:-

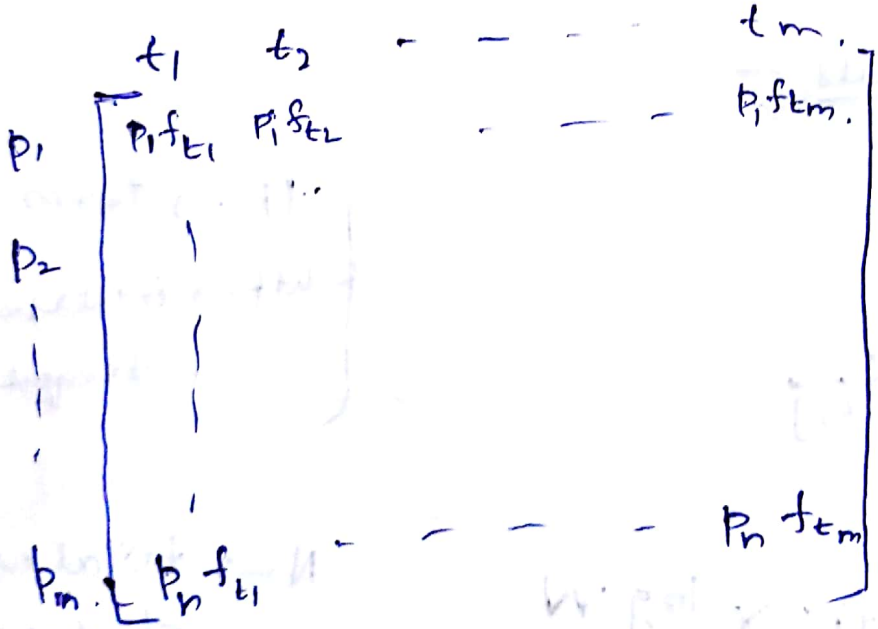
Why do we Summarise :- If a student of history is to be evaluated, then text summarisation is used.

for keywords finding.

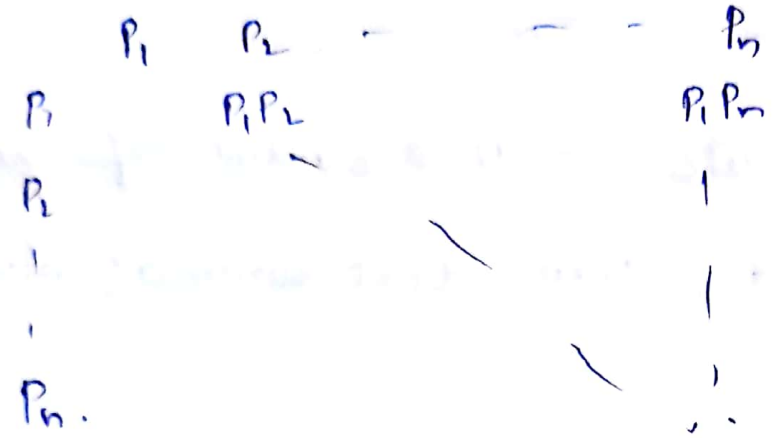
A Single document.

t → unique terms

p → Paragraphs.



Now we find Similarity between paragraphs, using Similarity measure.



In place of using paragraphs, Sentences are more appropriate.

Weighting Methods :-

tf - idf :-

tf → term frequency
idf → inverse document frequency

i,j ← doc no, → term.

$$t_{f_{i,j}} = t_{f_{i,j}}$$

N → total number of documents.

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

df_{t_j} → document frequency.

document frequency $\leq N$.

Ex1

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19241	1.65
best	25235	1.5

total no. of docs $(N) = 806,791$

Renters data set.

Ex2

term	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best.	14	0	17

Variant of tf-idf

$$rft = \frac{ft_{i,j}}{\max_x df_{i,j}}$$

$$rft_{i,j} = \begin{cases} 1 + \log ft_{i,j} & ft_{i,j} > 0. \\ 0, & \text{otherwise} \end{cases}$$

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

$$p(idf) = \max \left\{ 0, \frac{\log N - df_t}{df_t} \right\}$$

Discount

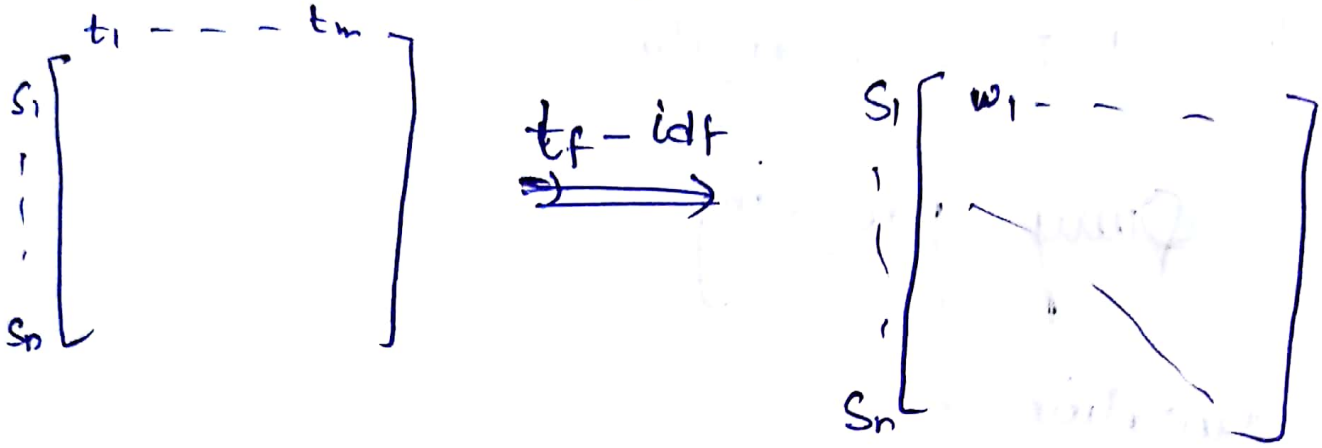
we need to summarize over document

① Unique terms in the whole document

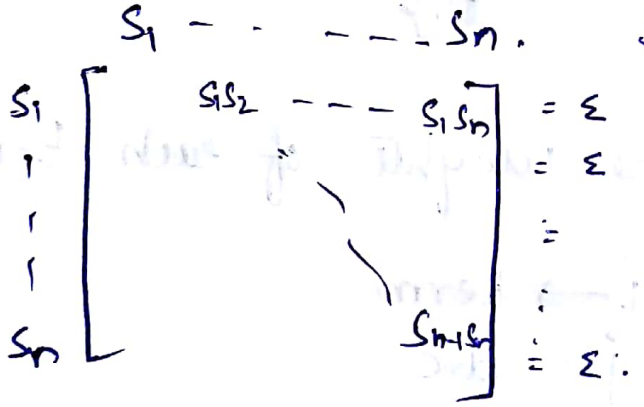
② Represent the document in terms of sentences.

③ Find the frequency matrix

④ Apply tf-idf ⇒ Weight Matrix



⑤ Apply any Similarity ~~Matrix~~ Metric between Sentences



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:-

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

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

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 = 1/3.

To increase precision we need to take care of synonyms of keywords of query. Hence query must be expanded.

Query Expansion :-

Approaches for ~~Every~~ query Expansions :-

Local methods : use docs that are retrieved by by original query

① Relevance Feedback → what user gives

② Pseudo relevance feedback → 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 reweighing 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

Rocchio formula :-

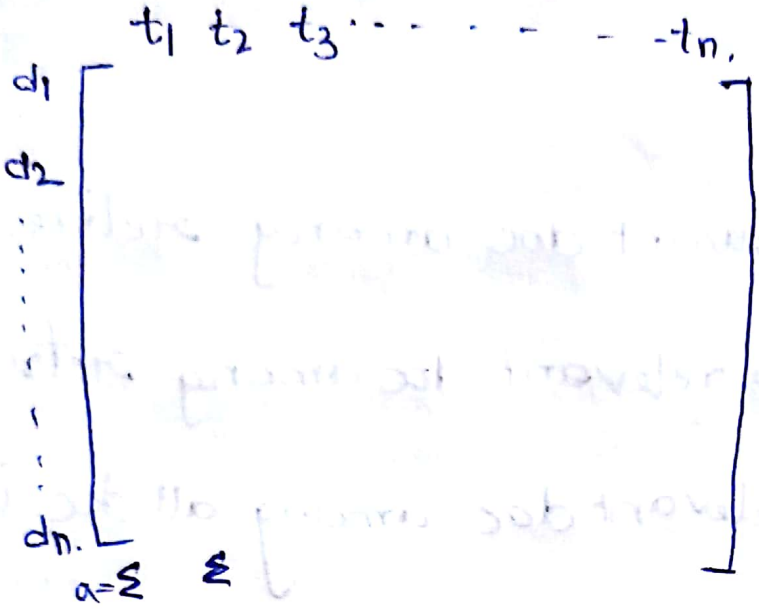
$$q_{i+1} = \alpha q_i + \frac{\beta}{|D_r|} \sum_{d_j \in D_r} d_j - \frac{\gamma}{|D_n|} \sum_{d_j \in D_n} d_j$$

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

$\gamma << \beta$.

Positive relevance feedback ($\gamma = 0$)

Generation of Concept cloud:-



$a = \Sigma$ ε

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

$$= \frac{H(A \cap B)}{H(A) + H(B) - H(A \cap B)}$$

$H(A)$ — count of ~~term A in all docs~~ all docs. which term A exi

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

Concept cloud

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.

~~Consn~~

Consesus

Performance Evaluation of Information Retrieval Systems :—

For a given set of docs (gold standard)

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

R-Precision :— At R^{th} position, we need to stop

F-Measure :— $\frac{2PR}{P+R}$

E-Measure :— $\frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{1+\beta^2}{\frac{\beta^2}{R} + \frac{1}{P}}$.

	relevant	not relevant
Retrieved	t_p	f_p
Not retrieved	f_n	t_n.

$$P = t_p / (t_p + f_p)$$

$$R = t_p / (f_p + t_n)$$

$$\text{Accuracy} = (t_p + t_n) / (t_p + f_p + f_n + t_n)$$

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

There are 50 emails & 49 are not useful and 1 mail is useful.

TP — we are saying yes, actually yes

TN — " " " NO, " NO

FP — " " " ~~Yes~~ NO " ~~No~~ Yes

FN — " " " ~~NO~~, " ~~Yes~~ NO. Yes

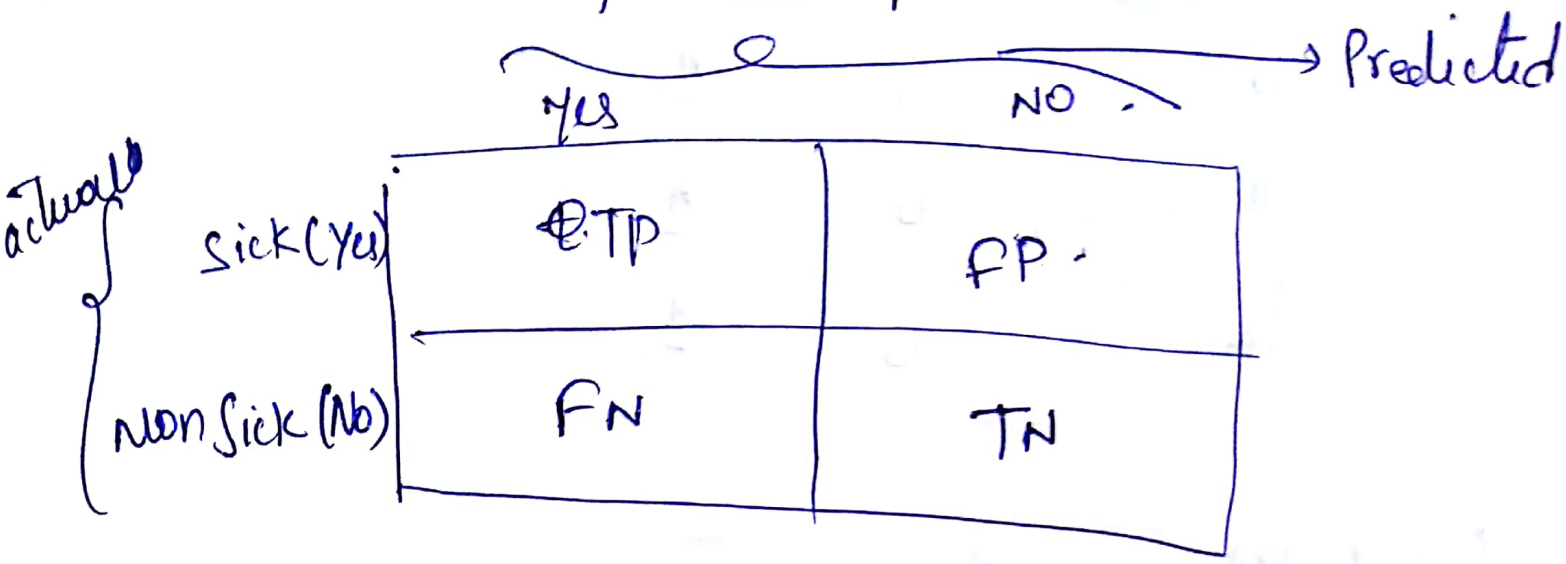
Accuracy :- It is not a dependable criteria for a System.

If f_p is Some, precision should be Considered.

If f_N is high, Recall should be considered.

$$P = \frac{t_p}{t_p + f_p}, \quad R = \frac{t_p}{t_p + f_n}$$

Draw the matrix for Sick & non Sick.



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].

accuracy = 8/12 = 2/3

Precision = 1/2 = .

Recall = 1/4

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

Case-2:

	relevant	non relevant
relevant	4 [TP]	6 [FP]
non relevant	0 [FN]	2 [TN]

accuracy = 6/12 = 1/2 Recall = 4/10 = 2/5

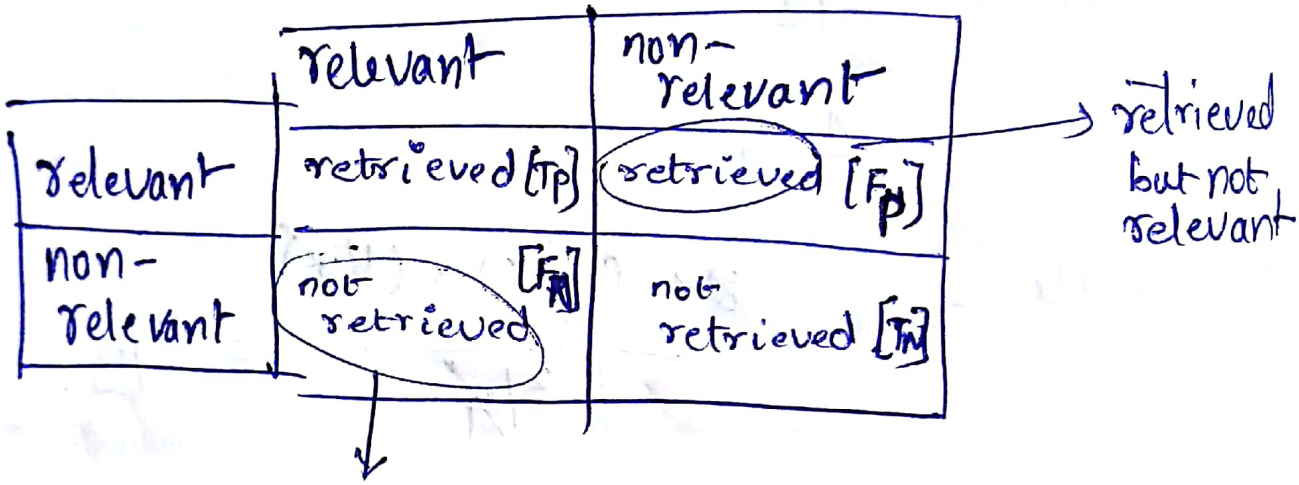
Precission = 4/4 = 1 F-Measure = $\frac{2 \times 2/5}{7/5} = 4/7$

F-measure, can we use it as Similarity measure. [Monday]

in document Text Summarization.

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

Ex:-



Ex: Sachin AND Pilot — Precission is high

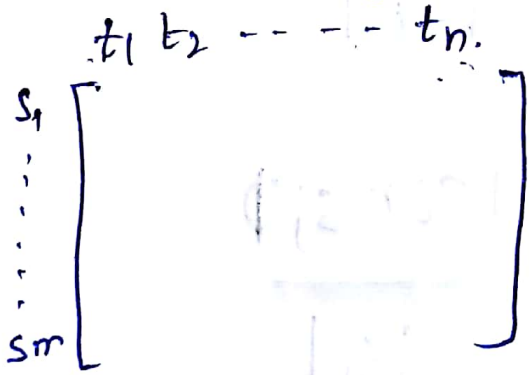
Sachin OR Pilot — Recall is high.

Ex: 100 docs., ~~Search query~~ IR Systems returns a particular paragraph of a doc relevant to Search query.

levels of index → paragraph indexing, Sentence indexing

this will increase Recall.

Ex: F-Measure as similarity measure.

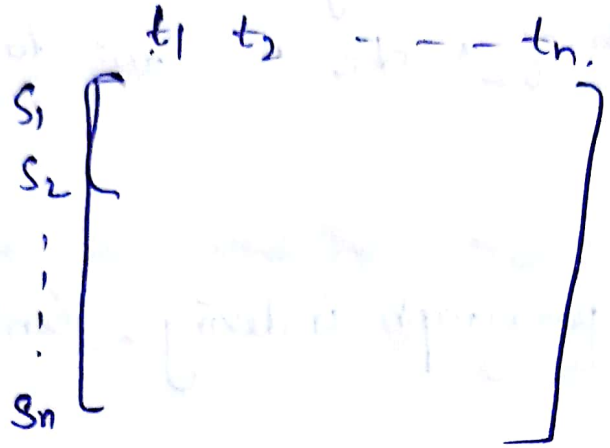


$\frac{S_1 \cap S_2}{S_1}$ = $\frac{S_1 \cap S_2}{S_2}$

$F(S_1, S_2)$ = → my formula.

$$P = \left[\frac{S_1 \cap S_2}{S_1 \cup S_2}\right], \quad R = \left[\frac{S_1 \cap S_2}{S_1} \cup \frac{S_1 \cap S_2}{S_2}\right]$$

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.

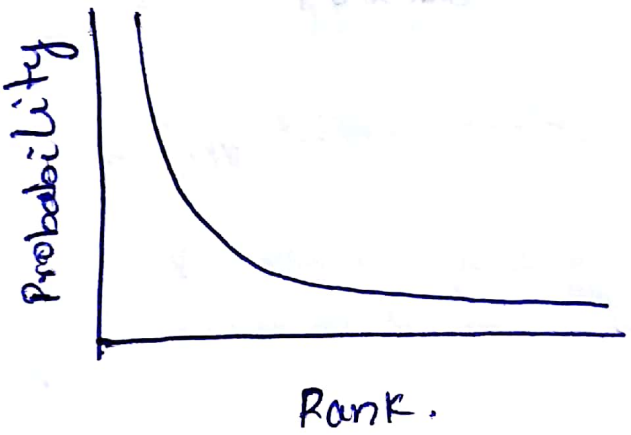
$$P = \frac{|S_{man} \cap S_{auto}|}{|S_{man}|}$$

$$R = \frac{|S_{man} \cap S_{auto}|}{|S_{auto}|}, \quad F = \frac{2PR}{P+R}.$$

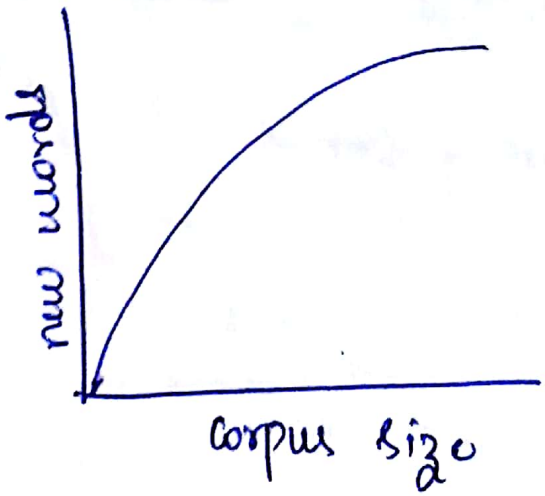
Text Statistics :-

Ziff's law :-

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



Heap's Law -



corpus size $= K \times n^{\beta}$

Can we apply Heap's law to text document.

Make a new data set needed at 3rd & 4th unit.

Query & question generation.

Mid term

① Intro to IR.

Boolean, Vector space → Probabilistic model.

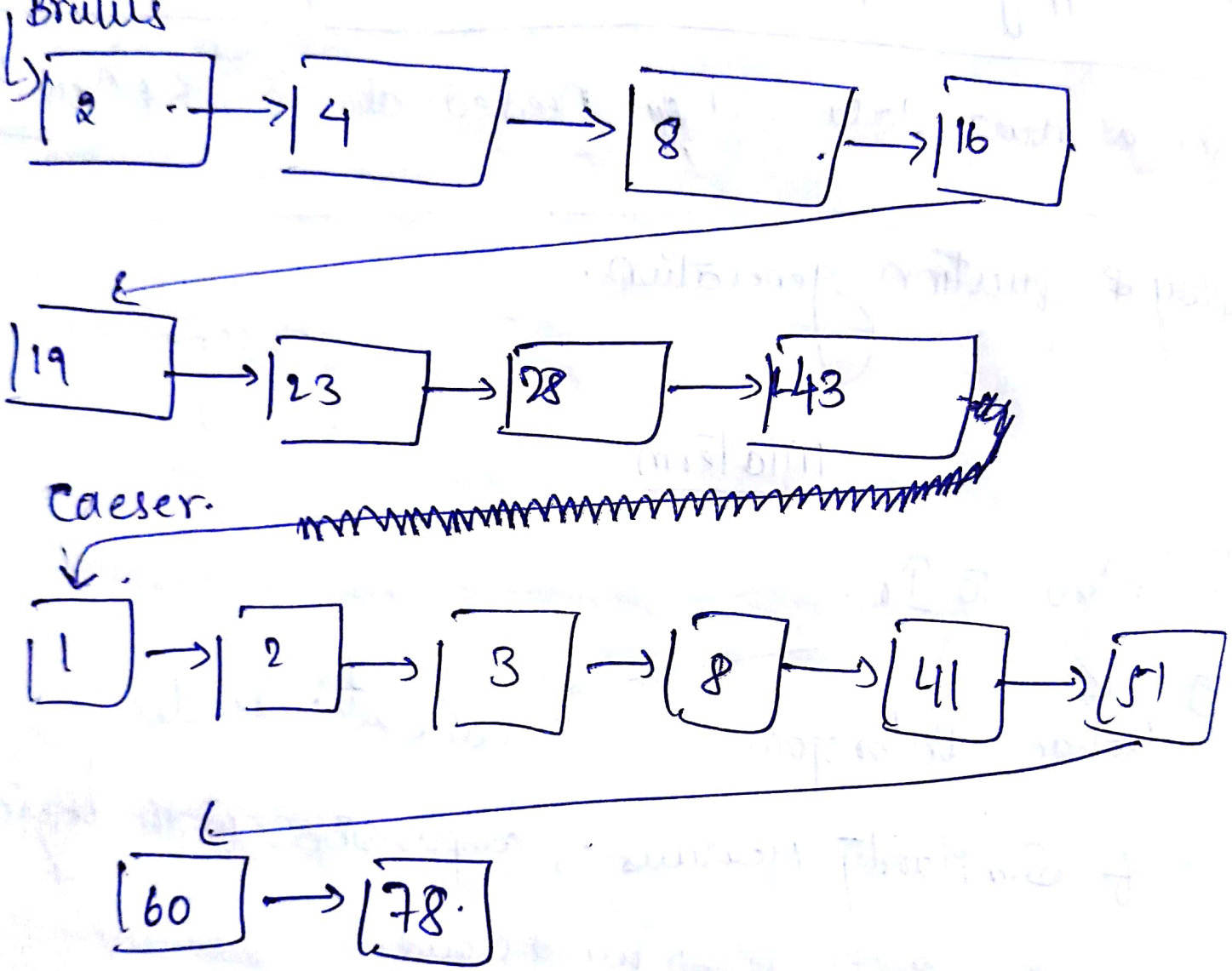
Type of Similarity Measures, comparision. with logic

Query expansion. → word cloud
→ word Net
→ relevance & Pseudo relevance feedback

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

Performance.

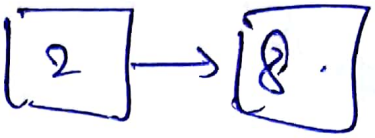
Indexing Numerical example-



Query Brutus AND CAESER

intersection of both ~~indexed~~ posting list.

Outcomes — 2, 8.



efficiency :- O(M+N)

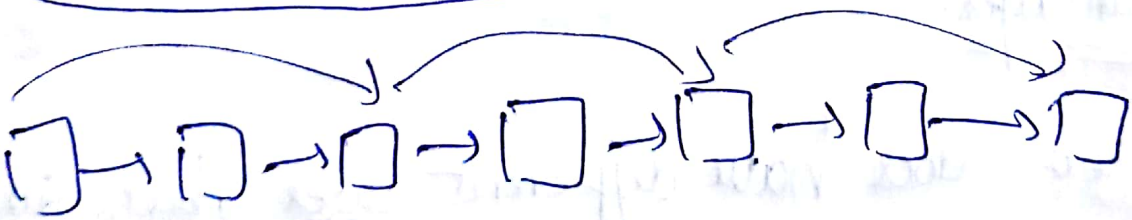
[Skippointer]

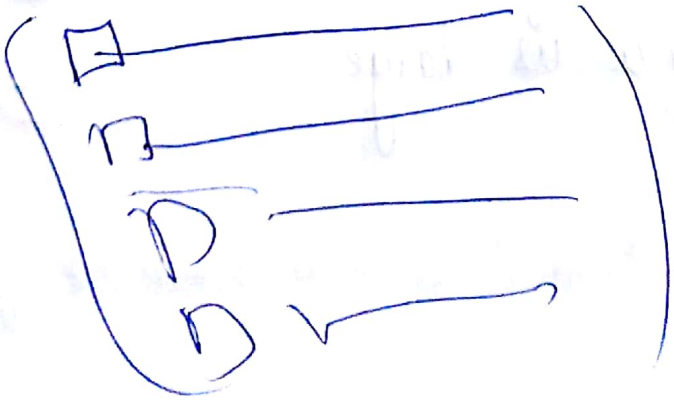
Improve efficiency

$SP = \sqrt{min(M,N)}$

SP / 2 (displacement)

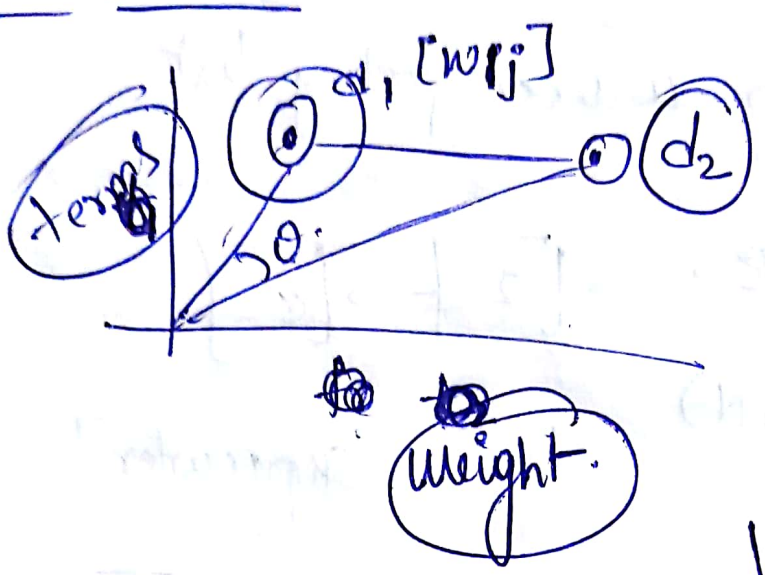
Sample questions practise





Euclidean distance :-

w_{ij} = weight of i^{th} term on j^{th} doc.



$$Ed(d_1, d_2) = \sqrt{\sum (t_i - t_j)^2}$$

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.

[Soln may be direction / Magnitude.]), my soln.

Cosine :-

Adj :- disadj of ED

disadj ignores Magnitudes.

lenght may be an issue if not normalized. if normalized then Δlenght will not effect.

Normalization process = /Max, log( ) -- etc.

Jaccard coefficient :-

interesection present, only works. — adv.

in case of related ness it fails — disadv.

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 ⟶ Pollution & weather forecast.

Based on computation power required :-

~~1. PCC~~

2. Cosine

3. ED.

4. PCC.

5. Jaccard.

↓ my

descending order

1. PCC

2. ED.

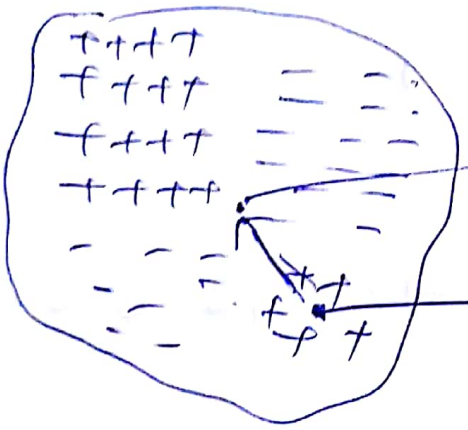
3. Cosine

4. Jaccard.

↓ actual

Query Expansion :-

- local
  - relevance
  - Pseudo relevance.
- Global
  - Thesaurus / wordnet
  - concept cloud.



to improve recall.

$$q_{opt} = \alpha q_i + \frac{\beta}{|D_r|} \Sigma q_i \cdot dw_i - \frac{\gamma}{|D_n|} \Sigma q_i \, dw_i .$$

query :- News about presendential camp (1, 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 presedential 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 ...]

D_3 : (1.5, 3.0, 2.0, 0, 0 ...)

D_4 : (1.5, 0, 4.0, 2.0, 0, 0 ...)

(+)((1.5+1.5)/2, (3.0+4.0)/2, (2.0+2.0)/2, 0, 0 ...)

= (1.5, 3.5, 2.0, 0 ... 0)

(÷) = [1.5, 0.067, 0, 2.6, 1.3, 0).

Centroid = [1, 1, 1, 1].

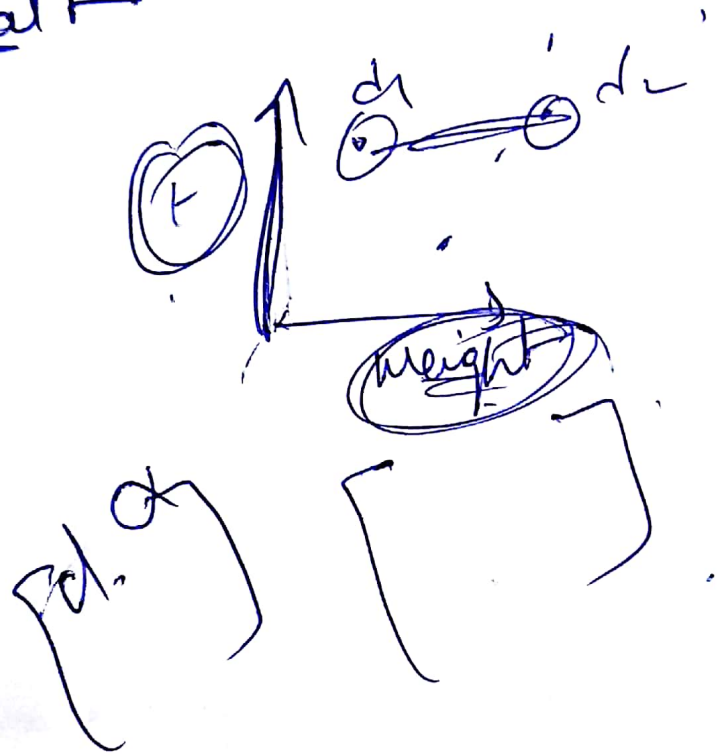
Q(initial) = [1, 1, 1, 1].

Q_{opt} = [$\alpha \times 1 + \beta \times 1.5 - \gamma \times 1.5$, $\alpha \times 1 + \beta \times 3.5$ $- \gamma \times 0.06$

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

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

Global ⊢



---J.