

text
(abstract)
intn
 \equiv

IR approach (Bag-of-words)

ability
 $w_1 \quad w_2 \quad w_3 \quad \dots$
bag-of-words:

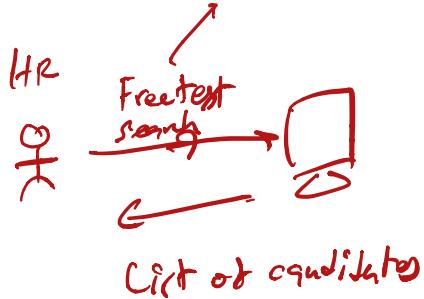
$$['\text{cat}', '\text{cat}', '\text{dog}'] \neq ['\text{cat}', '\text{dog}']$$

$$['\text{cat}', \text{dog}] = [\text{dog}, \text{cat}]$$

Feature Vector

	Age	Height	Weight	Disease
♂	30	1.60	60	Yes
♀	40	1.75	70	No

Scenario query vector



$[<30, 1.60, 60>, \text{Yes}]$

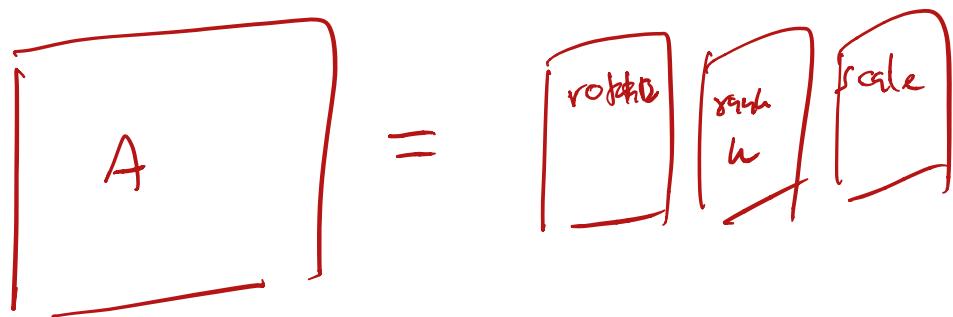
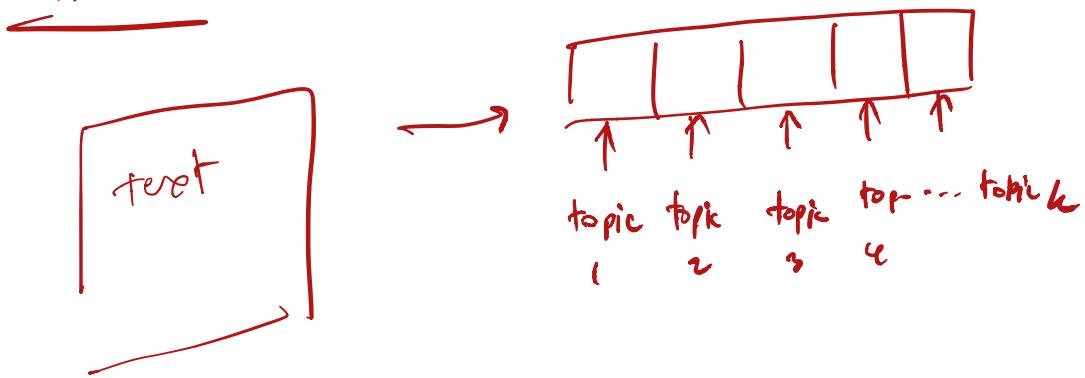
Feature Vector

class label

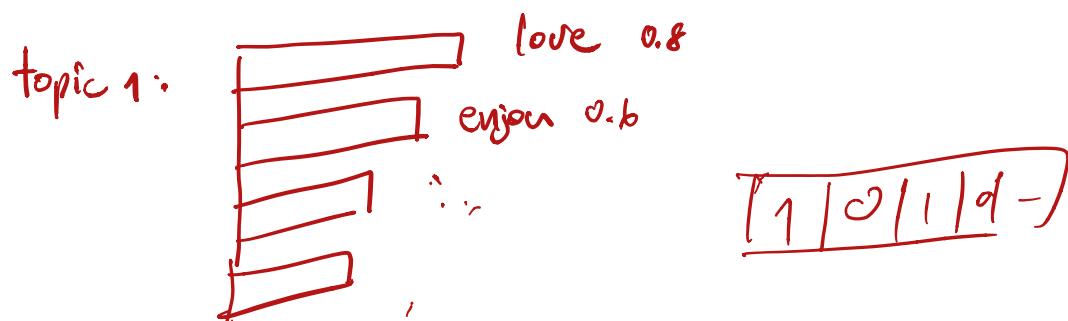
{:
:
}:

training sample

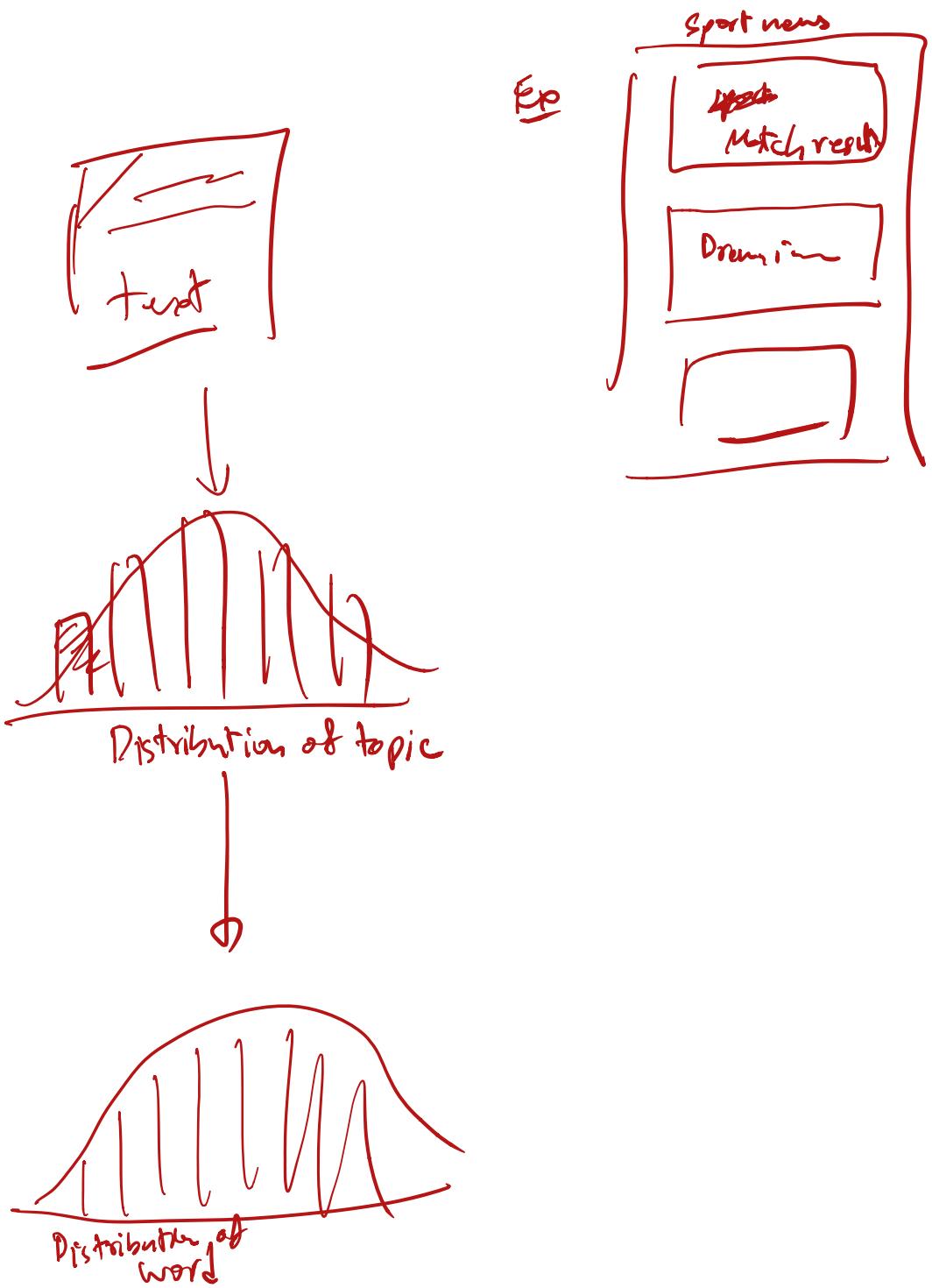
approach 2



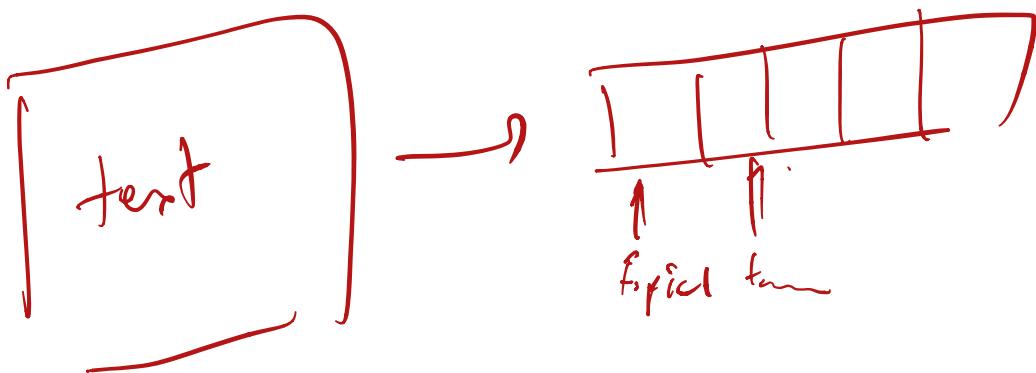
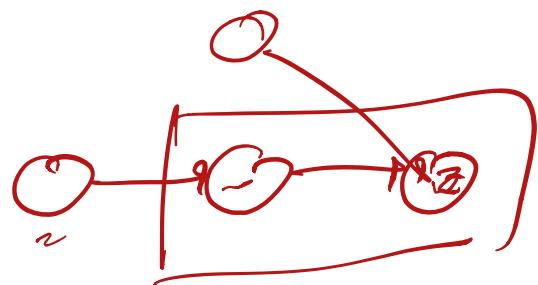
Read: SVD \rightarrow LSI (Latent Semantic Indexing)
↳ (Singular Value Decomposition)



Search keywords: SQL, AI, Machine learning



approach 3 LDA (Latent Dirichlet Allocation)



Choose Method 2

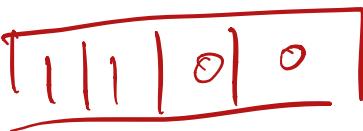
	t_1	t_2	t_3	t_4
1				
2				
3				
.				
:				

Query

Python, visualization

- $t_1 : [SQL, Python, R]$
- $t_2 : [computer language]$
- $t_3 : [presentation, visualization]$
- $t_4 : [communication, teamwork]$
- $t_5 : [Management]$

compute Distance matrix



Cosine

1 0.5

2 -0.2

3 0.2

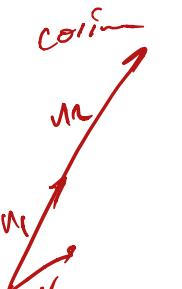
.

;

;

sort

Retrie top N



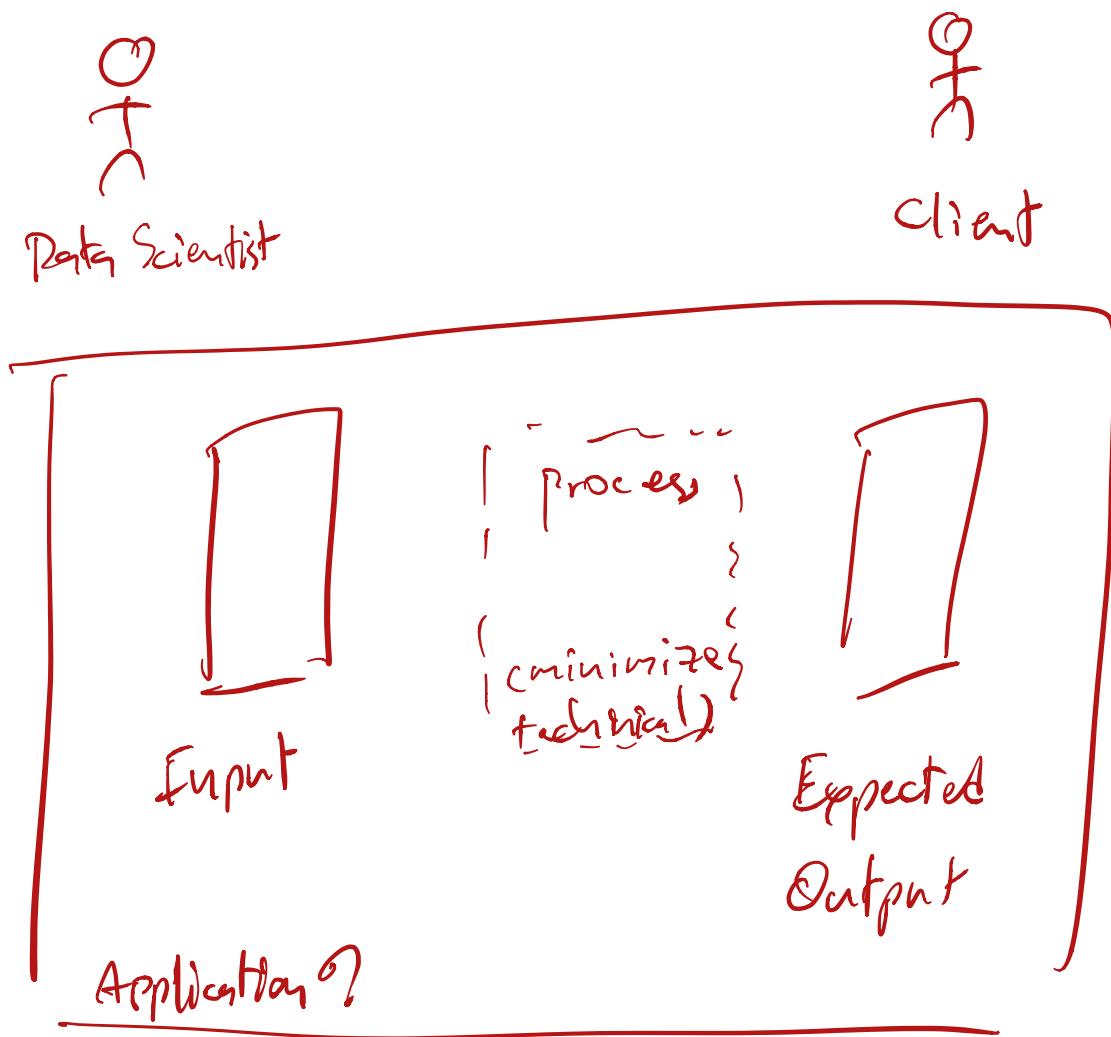
$$\text{sim}(u_1, v) = \text{sim}(u_n, v)$$



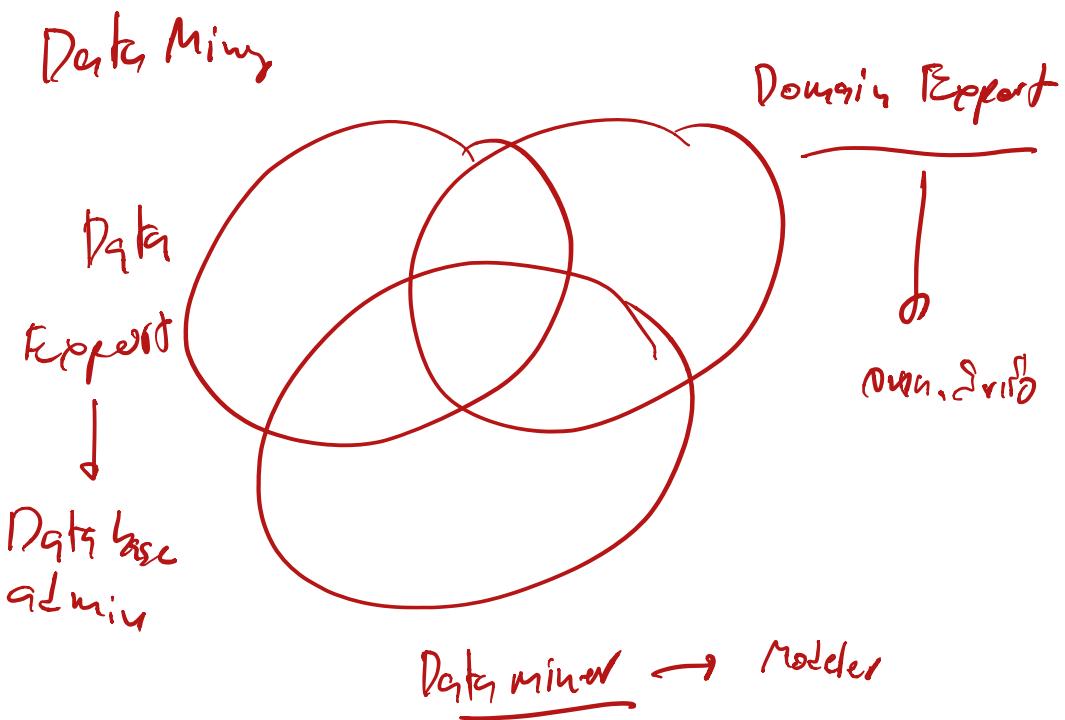
Story telling

- Application (what is it?)
- Assumptions
- Limitations
- Future work (How to improve)
 - └ technique
 - └ generalize to other application

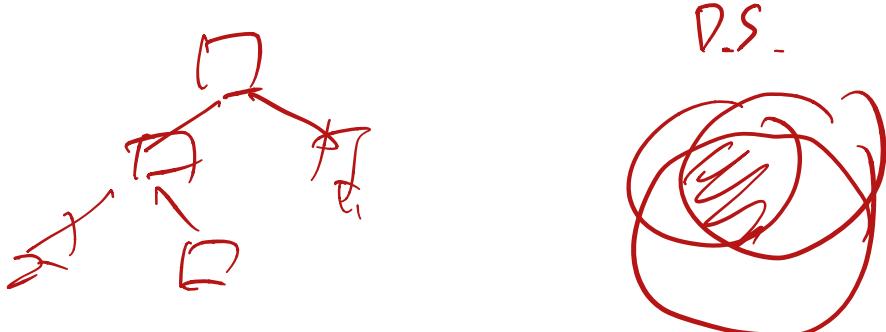
In practice



15 years ago



Eg Bank loan problem



Data Scientist

- Type A (Analyst) ← Stat - Data Analysis
Business Insight

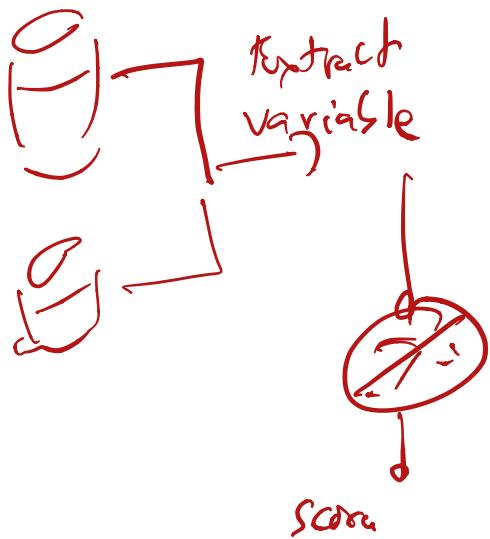
- Type B (Build) ← CS - Machine
technical team

Ex Given Budget

Find top 100 location for ATM installation

Type A

scoring problem



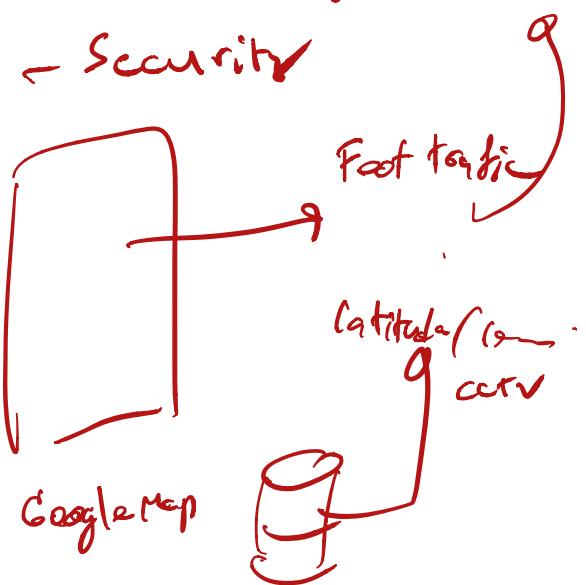
Type B

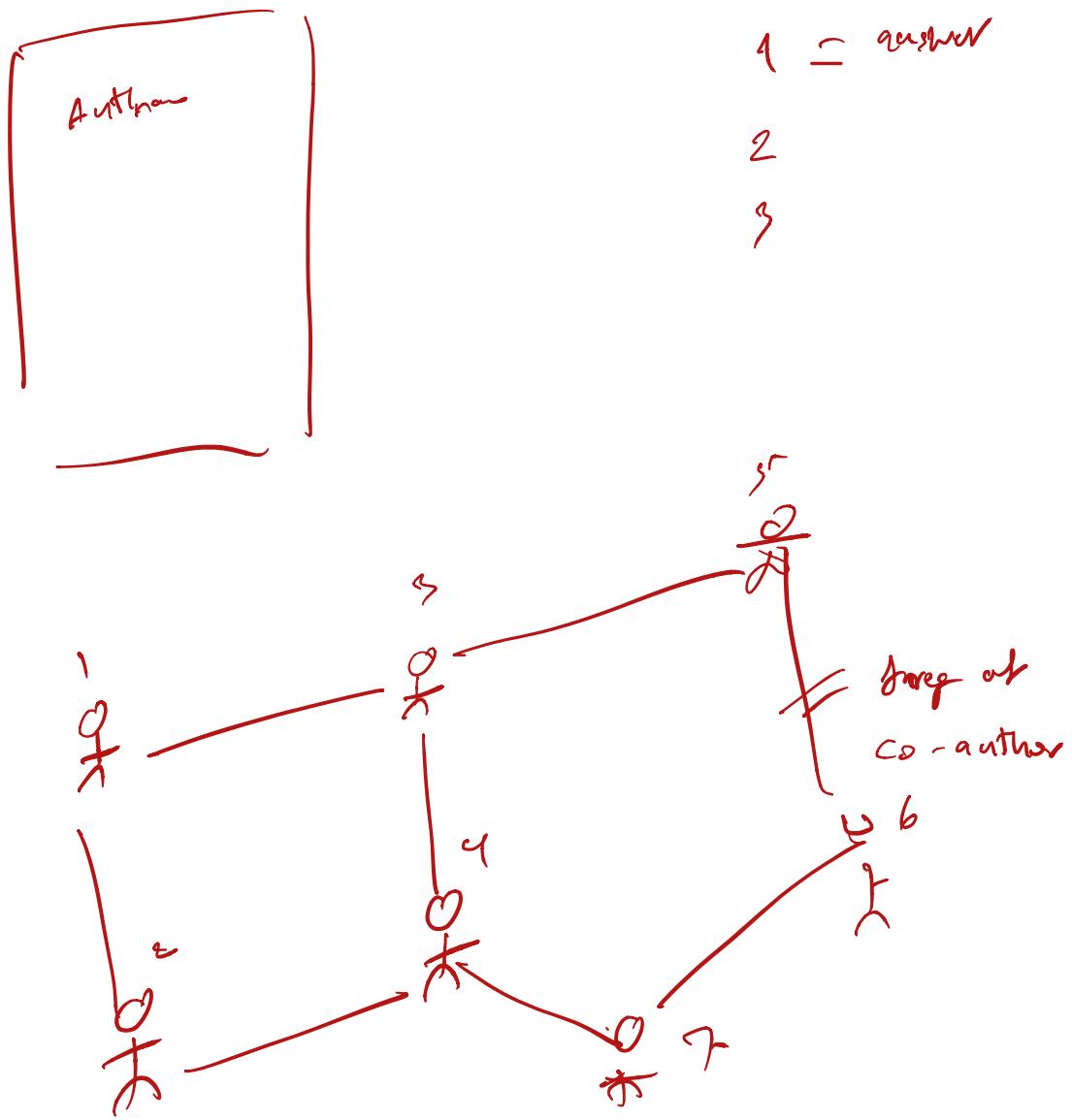
Scoring problem

- Accessibility

- Security

Score



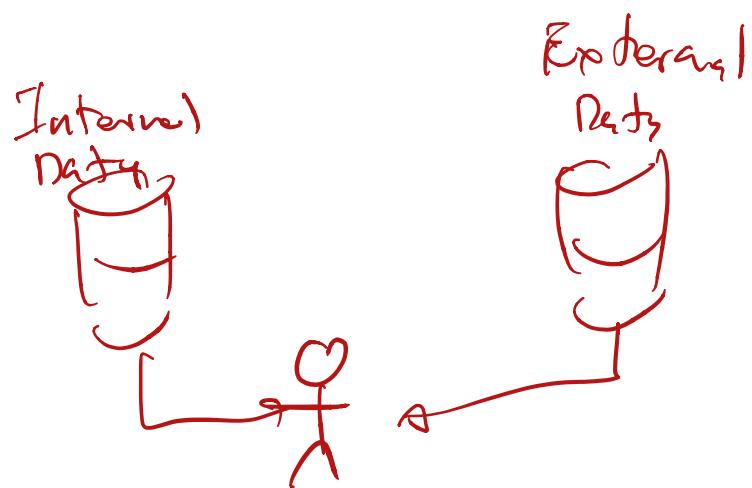


Expert Finding System in Researcher Network

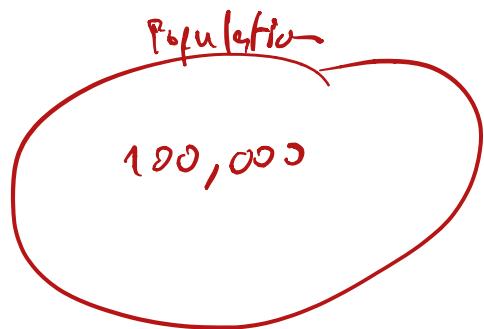
Journal Name

1) Computational in Medicine

Description



Filter Process



query



Author : keyword

1 : []

2 : []

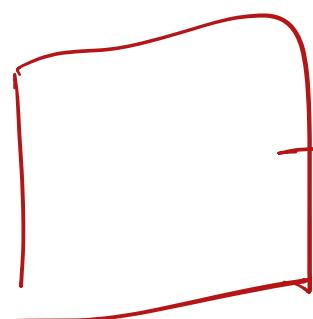
⋮

k

Ex

1 - 100

~~100,10~~



where
some condition

1 - 100



Python Data Structure

1) tuple (a, b, c)

2) list [a, b, c]

3) dictionary $x = \{ \text{key} : \text{value}, \text{key} : \text{value}, \dots \}$

key = user id

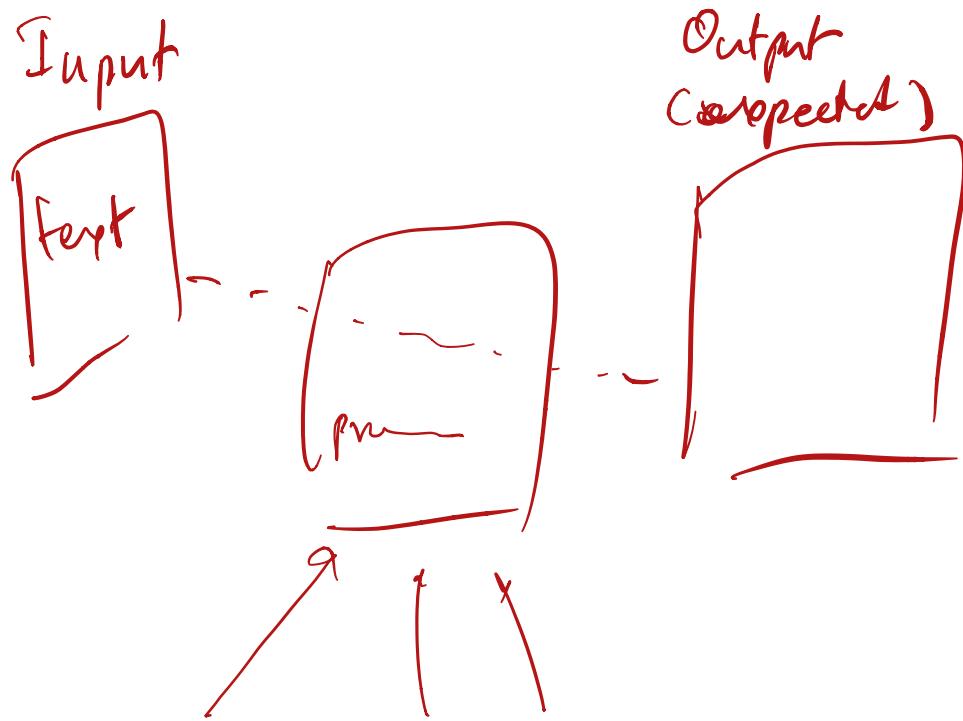
value = keyword

$df = pd.DataFrame(\text{filename}, \text{dtype}=\text{year: int})$

$x = \{ \}$

for i in df.id

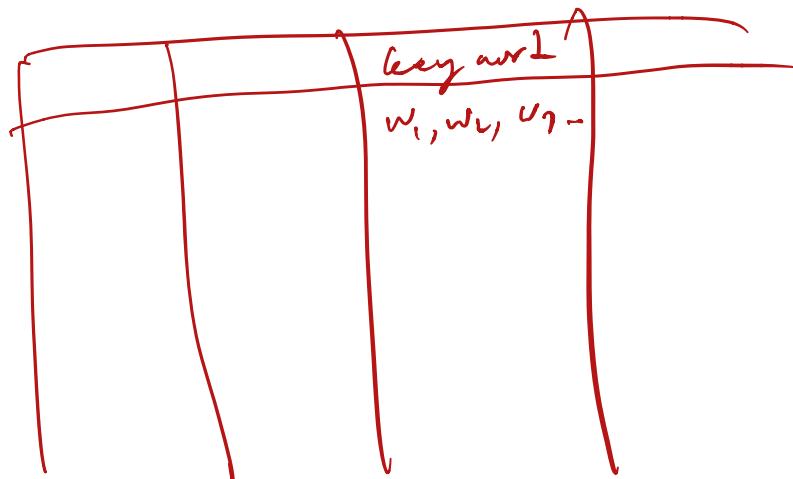
$x[i] = df.\text{keyword}$



Program Features

- 1) support keyword search
- 2) ~ Free text
- 3) ? wildcard s
- 4) u profile search

Pandas Dataframe



string-split(,)

optimization
→
Naive

Search crit.
(any method)

g|| m

$$\{ \begin{matrix} \overset{o}{x}_1 : [w_1 \ w_2 \ w_3] \\ \overset{o}{x}_2 : [w_2 \ w_5 \ w_6] \end{matrix} \}$$

$$\overset{o}{x}_3 : [w_1 \ w_3] \quad \} \quad \}$$

$$w_3 \rightarrow \overset{o}{x}_1 \ \overset{o}{x}_2$$