(MATRICES)

VECTORS — EMBEDDINGS

go-no-go beach problem

inputs
- crowd size (0-1000) — Scaled 0-1
- wave height (1-10 m) — 0-1

21-Mar-2021 ⇒ 239 people ⇒ 9.14m ht

(0.23989, 0.91423) = 2-D

(0.21, 0.85)

crowd size

wave height

Training dataset

0.23989	0.91423
0.102	0.385
0.654	0.718
⋮	⋮

wave ht	crowd
3	300
5	200
8	980
⋮	⋮

[(0.23...0.914), (0.102, 0.385), (0.654, 0.718), ⋮]

- crowd size
- wave height
- Temperature
- Humidity
- UV index
- Pollution index

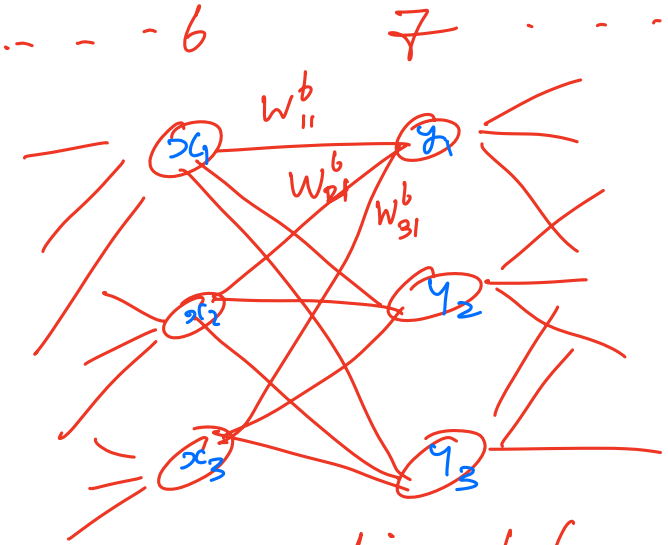
6-D

[(0.239, 0.118, 0.539, 0.1, 0.0009, 0.99832), ⋮ ( )]

(MATRICES)

use VECTORS in ML lifecycle

- input encoding
- calculations (training, prediction)



weight matrix between layers 6 & 7

6-7

$$\begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{21} & w_{22} & w_{32} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

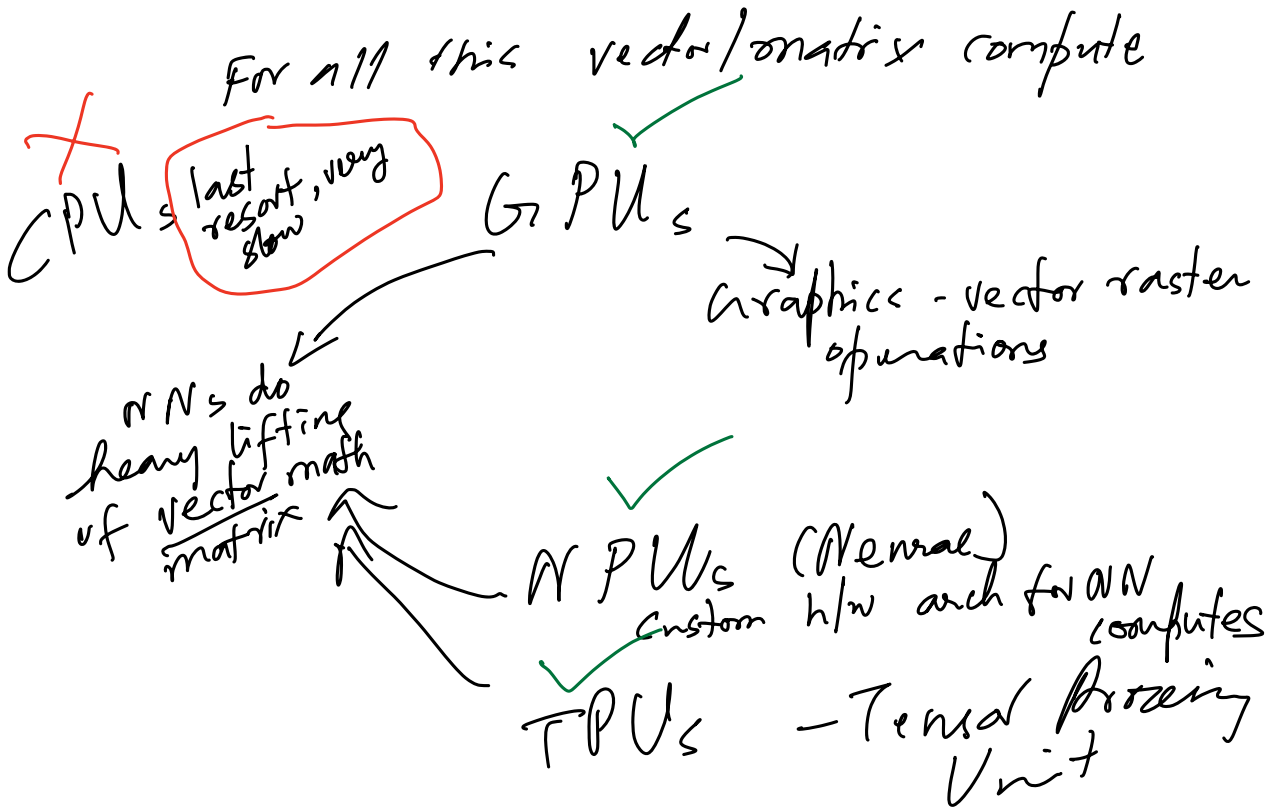
$$y_1 = x_1 \cdot W^6_{11} + x_2 \cdot W^6_{21} + x_3 \cdot W^6_{31}$$

$$y_2 = x_1 \cdot W^6_{12} + x_2 \cdot W^6_{22} + x_3 \cdot W^6_{32}$$

$$y_3 = x_1 \cdot W_{31} + x_2 \cdot W_{23} + x_3 \cdot W_{33}$$

$$\begin{bmatrix} w_{11}^6 & w_{21}^6 & w_{31}^6 \\ w_{12}^6 & w_{22}^6 & w_{32}^6 \\ w_{13}^6 & w_{23}^6 & w_{33}^6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

weight matrix layers 6-7 x layer 6 value matrix = layer 7 value matrix



INPUT ENCODINGS FOR TEXT

achieved by the use of encoding models 4d vector

"cat" = (0.3, 0.2, 0.1, 0.4)

"system" = (0.2, 0.4, 0.3, 0.5)

"what's up, doc?" = (0.3, 0.2, 0.1, 0.005) ← mathematically near

close meaning

"Hello, this is a paragraph of text. . . " = (0.29, 0.15, 0.11, 0.003)

"this is an entire document of text"

↑ Embeddings = Vectors = Encodings

= (0.5, 0.598, 0.1, 0.005)

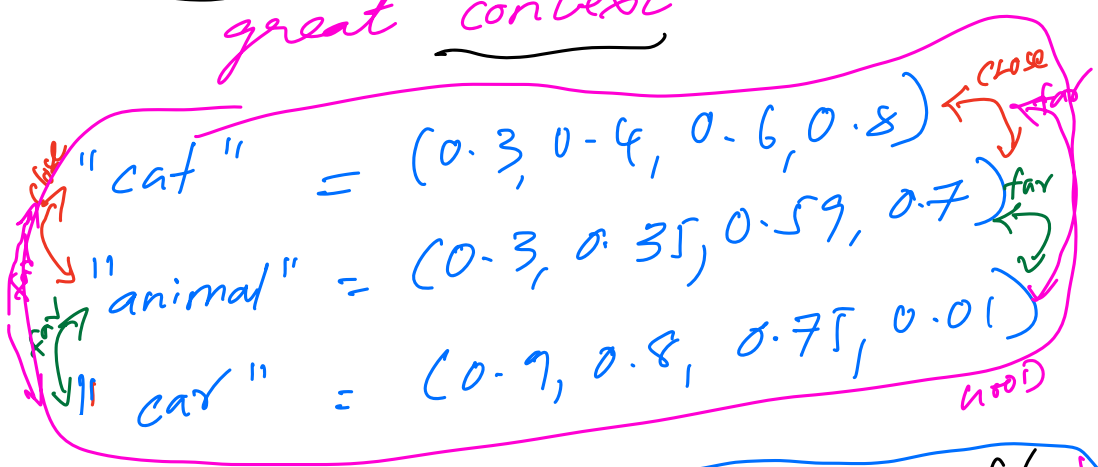
"a" = (0.1 . . .)

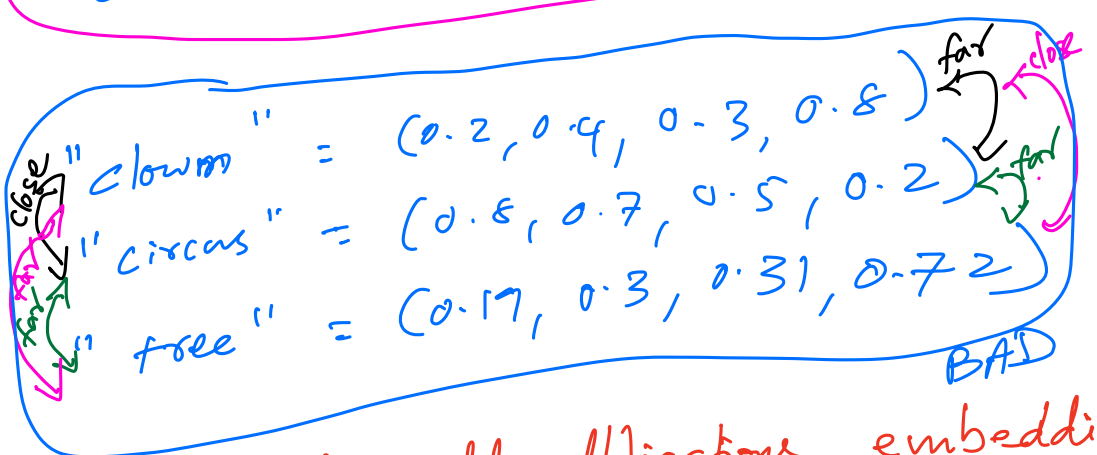
"≡" = (0.2 . . .)

When does an encoded vector become an "embedding"?

MEANING

semantically loaded vector with great context





In real world applications, embedding models often return 50-D embeddings to 1056-d embeddings

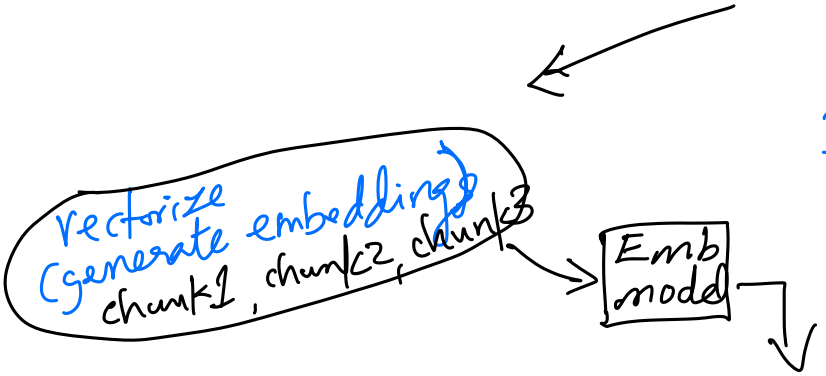
$(v_1, v_2, v_3 \ldots\ldots, v_{50})$

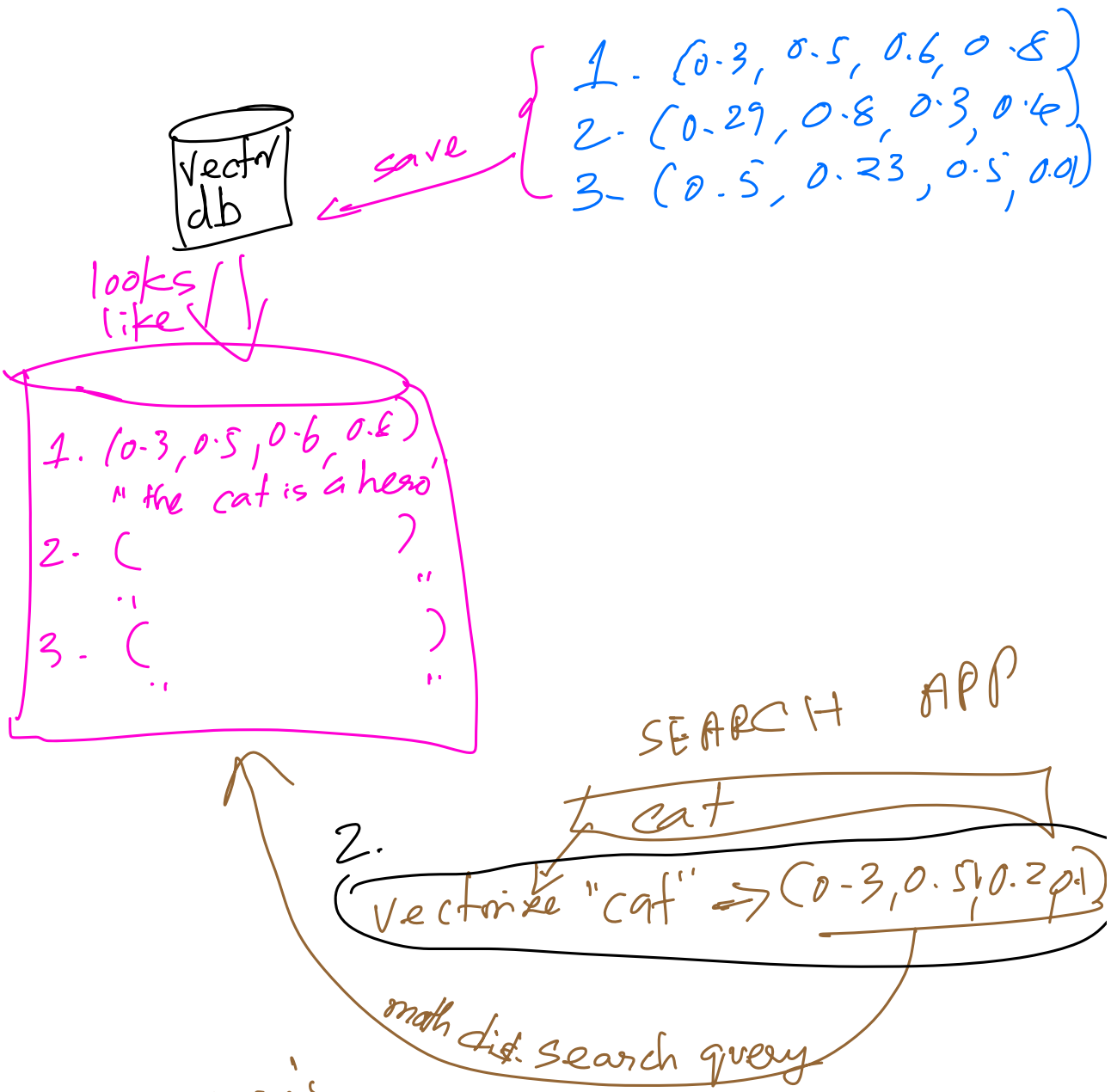
SEARCH (semantic)

For example

The cat is a hero. It can do amazing things. It was born in 1999 and had a good life.

chunks 1. The cat is a hero
2. It can do amazing things
3. It was born in 1999 and had a good life

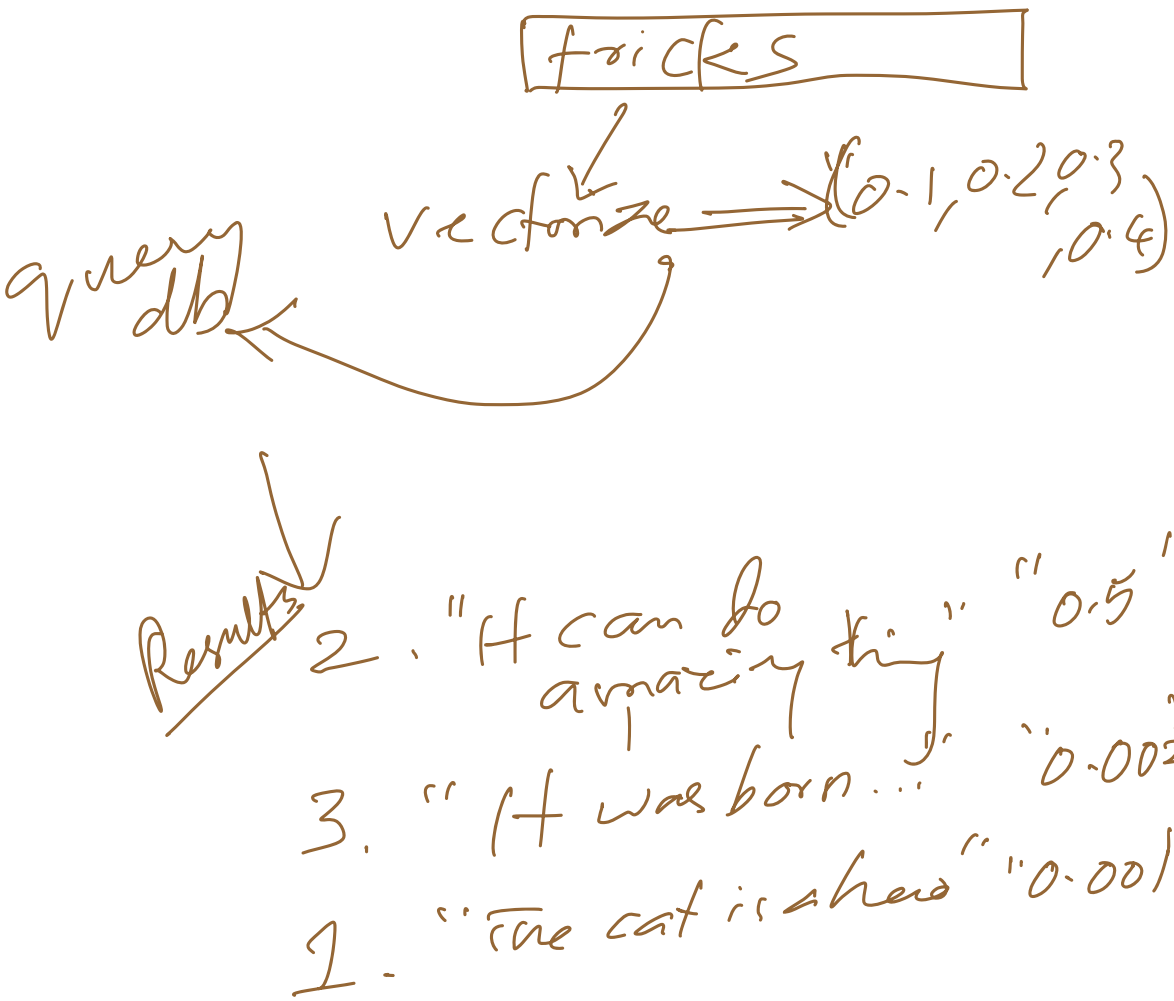




Vector DB Engine's Result:

Relevance score

1. "cat is a hero", "0.8"
2. "-----", "0.75"


tricks
vectorize
(0.1, 0.2, 0.3, 0.4)
query db
Results
2. "It can do anything" "0.5"
3. "It was born..." "0.002"
1. "The cat is a hero" "0.001"

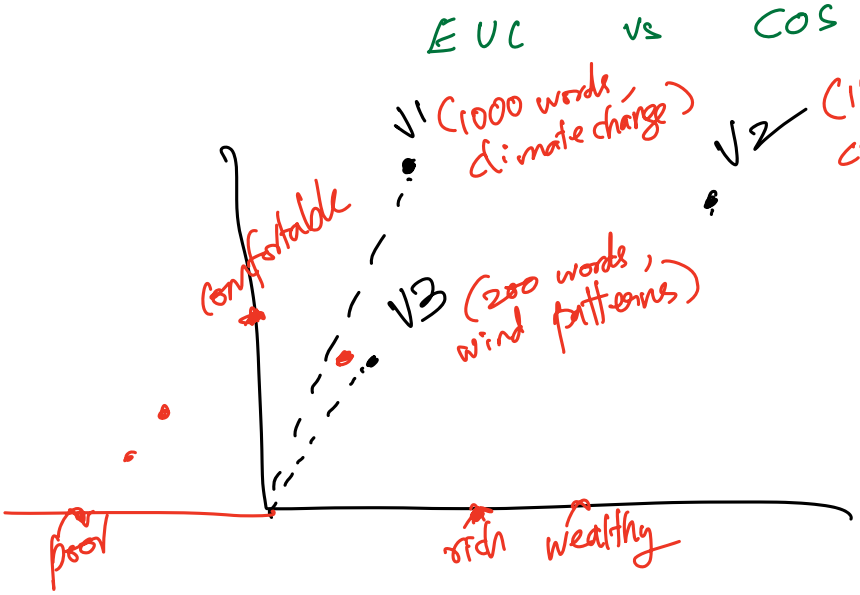
Vector database query

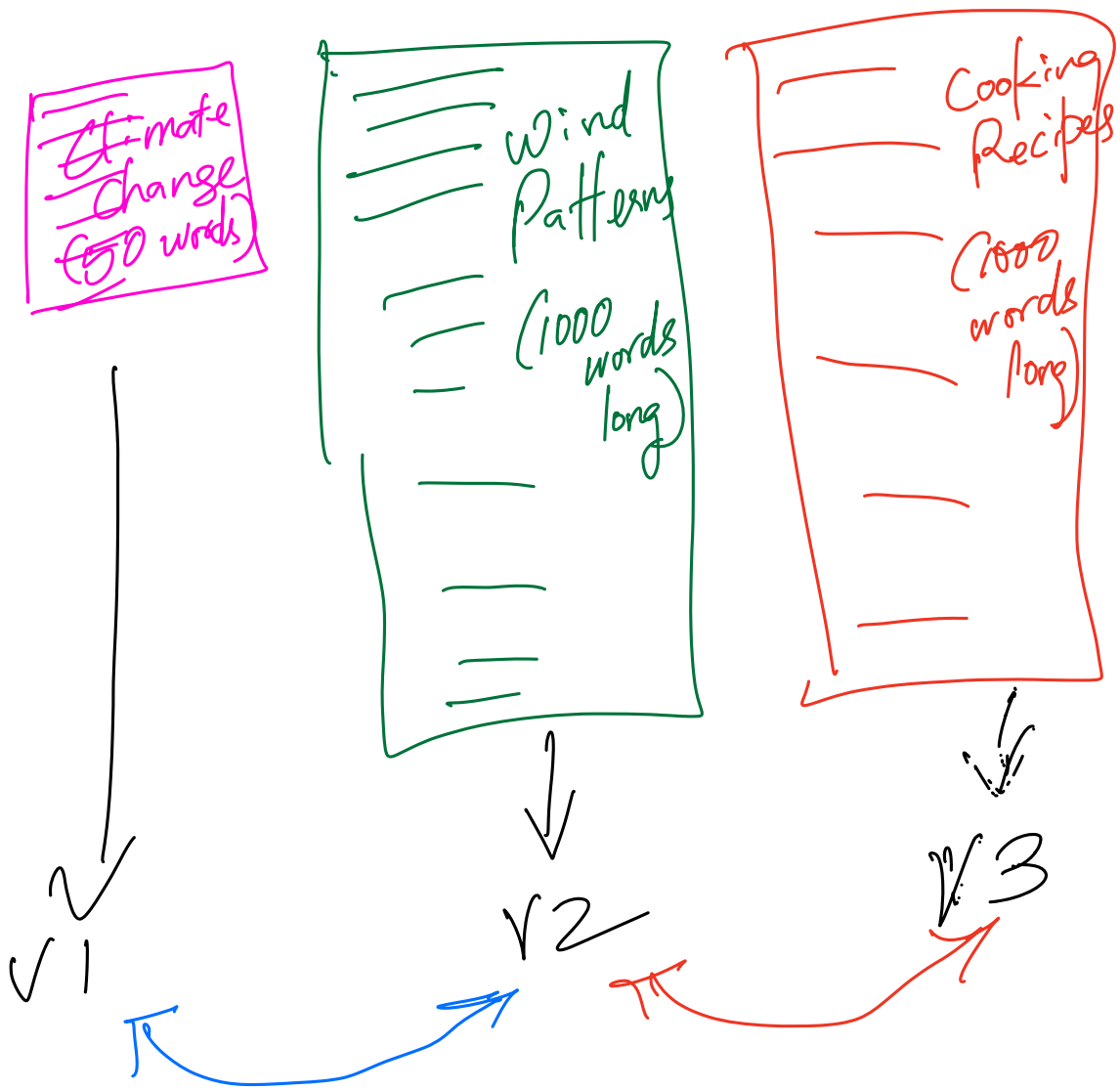
REQ ⟹ ("0.39891, 0.24132S, . . . ",
<input vector>
<search method>
SEARCH_METHOD_EUCLIDEAN/COSINE,
<# of results requested>
10
)

RSP ⟹ [("text1 blah blah . . ", <text blob/chunk>, 0.8234, <relationship score>)
(
⋮
(
]

opposites −1 to +1 same

EUC vs COS distances search




Climate Change (50 words)
Wind Patterns (1000 words long)
Cooking Recipes (1000 words long)
V1
V2
V3