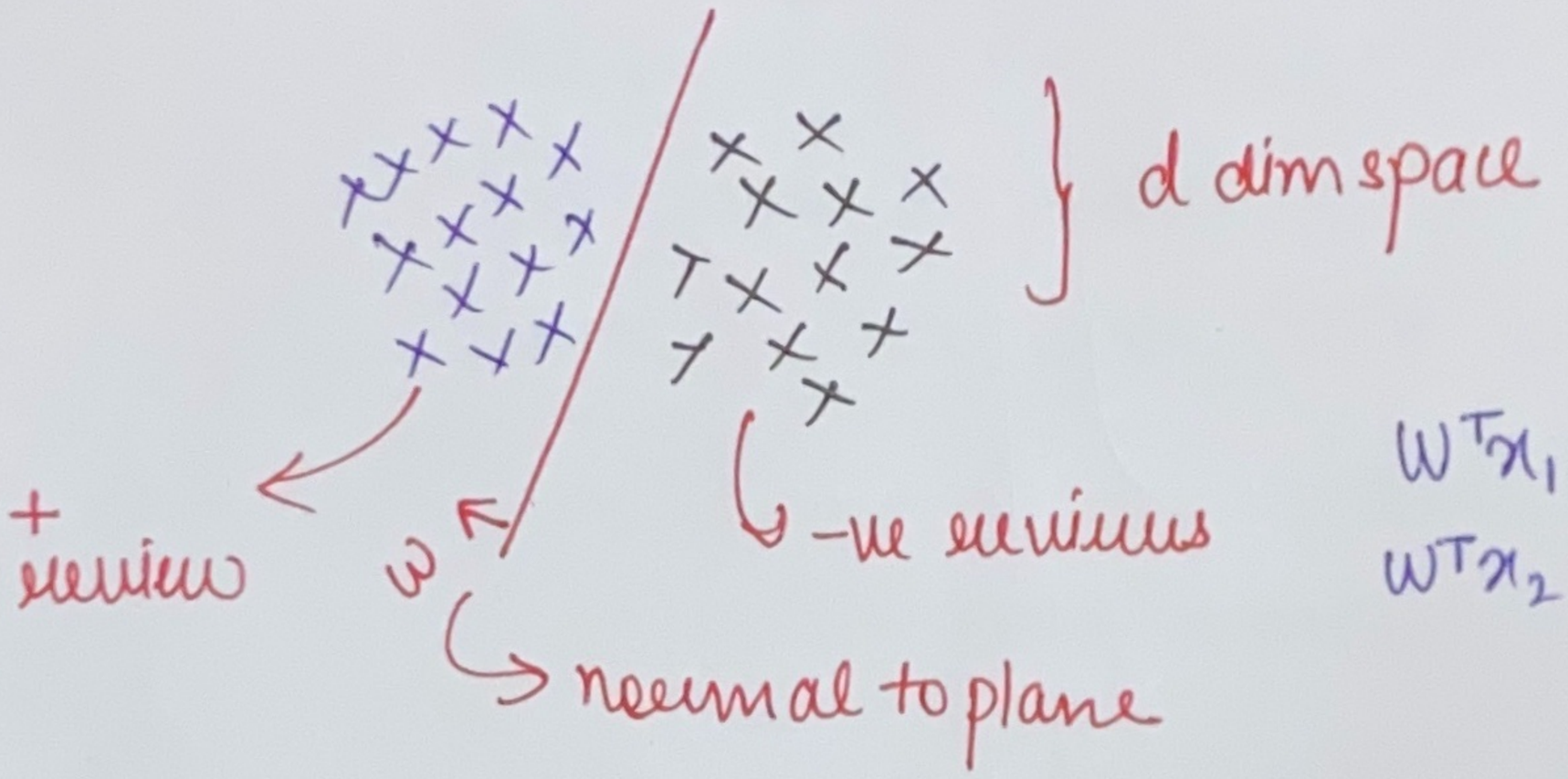
Text to Vector

(Q) how do you convert text into numerical vectors

Review text → d dim vector



$w^T x_1 \rightarrow$ +ve

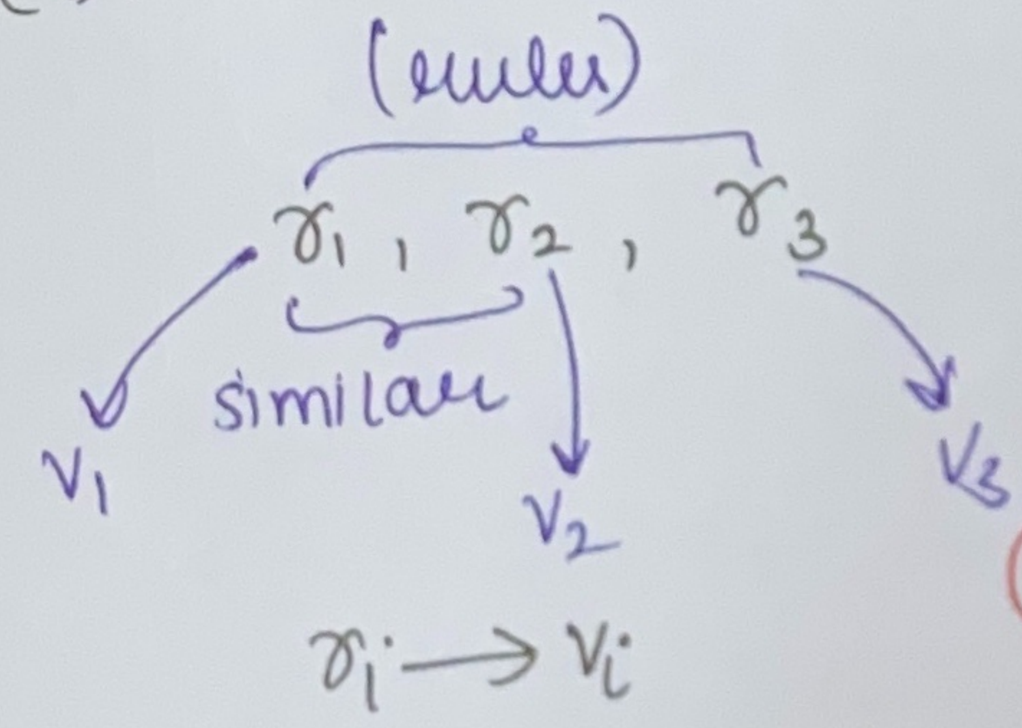
$w^T x_2 \rightarrow$ -ve

if $w^T x_1 > 0$ then x_i is +ve

else x_i is -ve

review text → d dim → finding a plane to separate

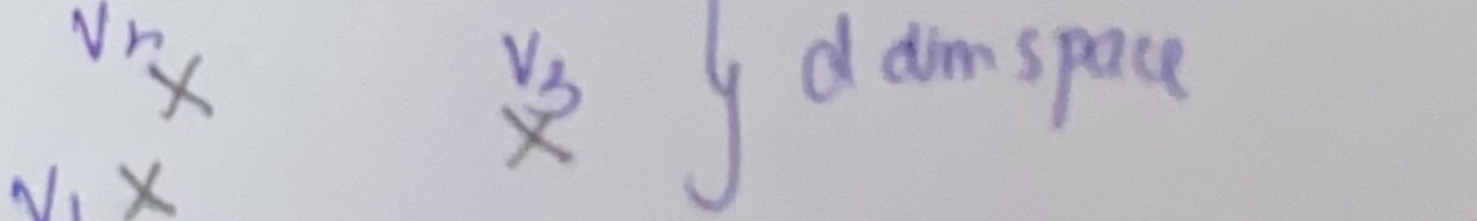
(Q) text → d-dim vector



$Sim(x_1, x_2) > Sim(x_1, x_3)$

$dist(V_1, V_2) < dist(V_1, V_3)$

(*) x_1 & x_2 are more similar then V_1 & V_2 must be close



(*) Similar text must be closer

$Eng\ Sim(V_1, V_2) > Eng\ Sim(V_1, V_3)$

$\Downarrow$

$length(V_1 - V_2) < length(V_1 - V_3)$

Bag of Words (BoW)

Corpus

r_1: This pasta is very tasty and affordable

r_2: This pasta is not tasty and is affordable

r_3: This pasta is delicious and cheap

r_4: Pasta is tasty and tastes good.

→ document

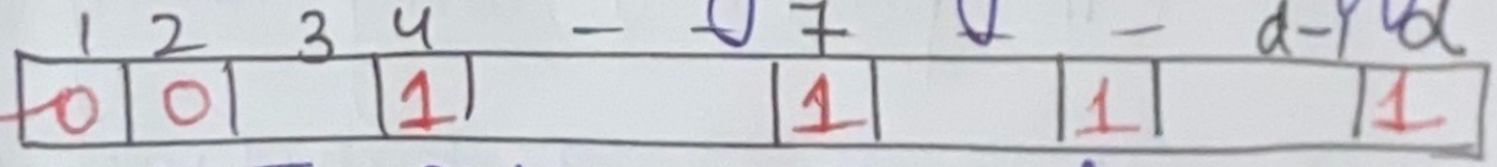
BoW

① Constructing a dictionary: Set of all unique words in your reviews.

(d unique words) → [This, pasta, is, very ----]

② r_1: This pasta is very tasty and affordable

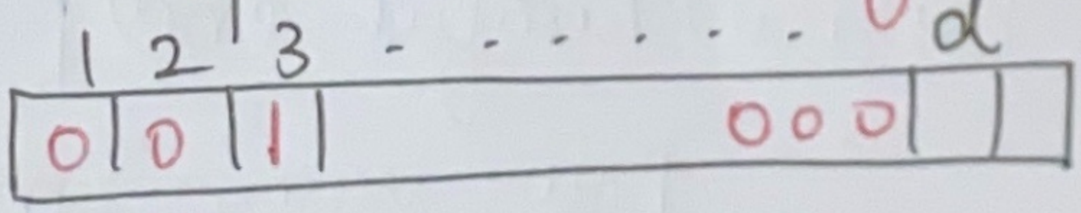
$v_1 \in R^d$



no of times word occurs

✳ Each word is different dimension

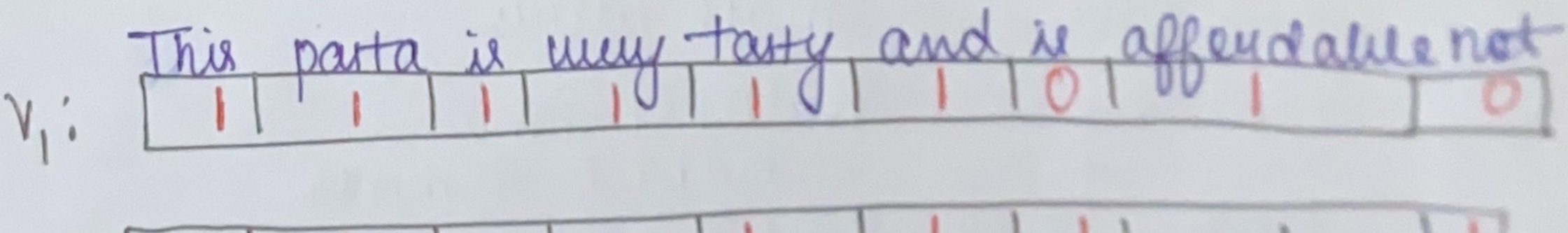
v_1 → sparse → most of elements are zero



d is very

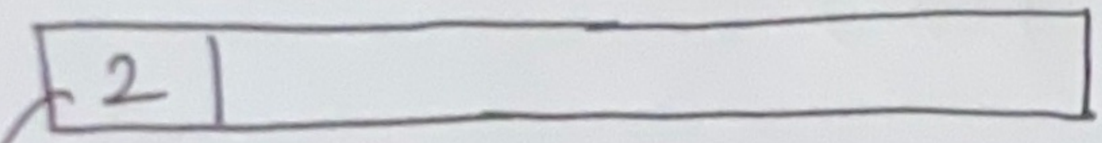
BoW: text → Vector

Similar text must result in closer vector



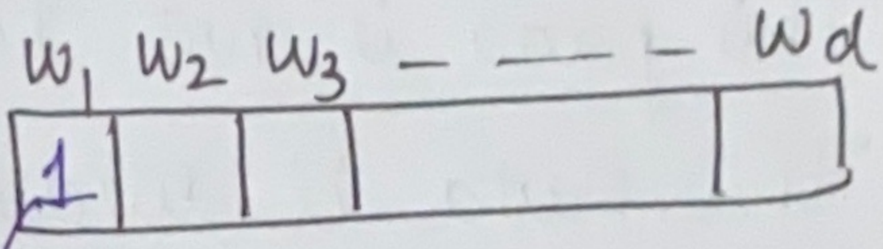
$$\text{length}(v_1 - v_2) = \underbrace{\|v_1 - v_2\|}_{\text{Norm}} = \sqrt{1^2+1^2+1^2} = \sqrt{3}$$

BoW:



#occurences

binary Bow / boolean Bow :-



1: If w_1 occurs atleast once

0: otherwise

Binary BoW

v_1: ——

v_2: ——

$||v_1 - v_2|| \propto \#\sqrt{\text{no of diff words}}$

BoW

Stop Words

r_1: This pasta is very tasty and affordable

r_2: This pasta is not tasty and is affordable

r_3: This pasta is very delicious and cheap

r_4: Pasta is tasteful and pasta tastes good.

BoW: vector: Smaller: meaningful

① → removing Stop Words → (Text Pre-processing)

② → Lower Case → Small letters

③ → Stemming → Porter Stemmer
Snow ball Stemmer

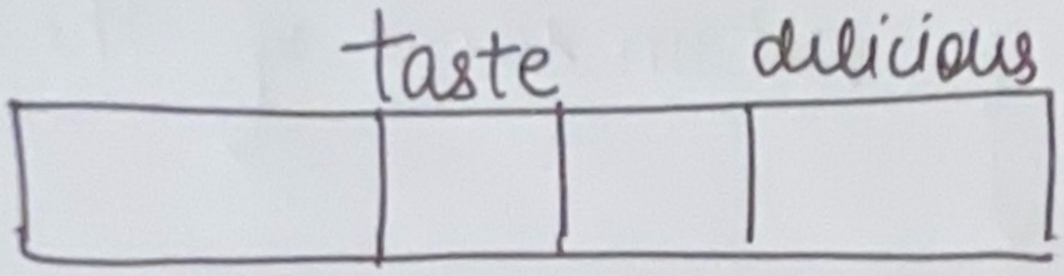
tastes, tasty, tasteful → taste

beautiful, beauty → beauty

④ Lemmitization (breaking sentence into words)

This pasta is very tasty, This is best in NewYork.

BOW:



Semantic meaning of words { tasty ⟷ dilicious

↳ **Word2Vec**

↳ BOW + Text Preprocessing

text → d-dim Vector

Unigram / Bi-gram / n-gram

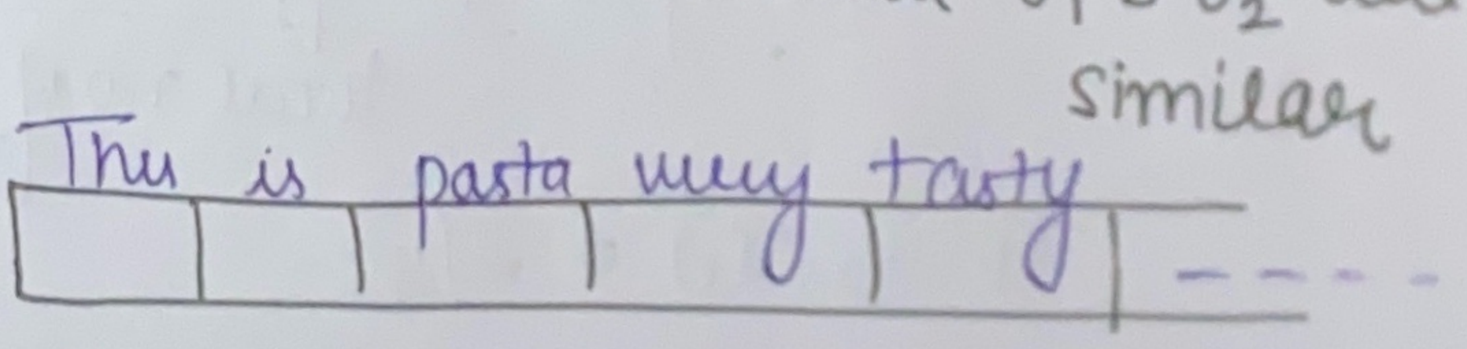
r_1: This pasta is very tasty and affordable

r_2: This pasta is not tasty and is affordable

removing stop words; V_1 & V_2 are exactly same

⟹ conclude r_1 & r_2 are very similar

Unigram:



↳ each word considered as a dim

bigram

↳ pair of words

trigram

↳ 3 consecutive words → word dimension

Uni gram

BOW: discards sequence information

bi gram
tri gram
n gram
} : retain some of the sequence info

bigram ≥ # unigrams

n-grams (n>1) → dimensionality 'd' increases

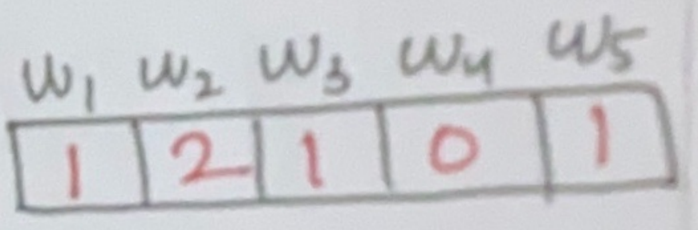
TF-IDF

Term frequency

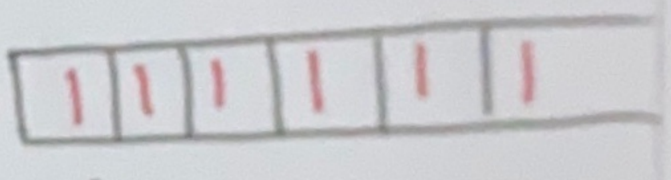
Inverse doc. frequency

N docs/ sentences

d_1 : $w_1\ w_2\ w_3\ w_4\ w_5$ — ⑤



d_2 : $w_1\ w_3\ w_4\ w_5\ w_6\ w_2$ — ⑥



d_3 :

d_N :

$$TF(w_i, d_j) = \frac{\#\text{ of times } w_i \text{ occurs in } d_j}{\text{total no of words in } d_j}$$

$TF(w_2, d_1) = 2/5$

$0 \leq TF(w_i, d_j) \leq 1$ ← probability

TF-IDF → Information Retrieval (NLP)

IDF: Inverse document freq

$\mathcal{D} =$ N docs / corpus

r_1 w_1

r_2 —

r_3 — w_1 —

⋮

r_N — w_1 —

$IDF(w_i, \mathcal{D})$

$\mathcal{D}_c = \{r_1, r_2 \cdots r_N\}$

$$IDF(w_i, \mathcal{D}_c) = \log\left(\frac{N}{n_i}\right)$$

N → # of docs

n_i → no of docs containing w_i

$n_i \leq N \Rightarrow \frac{N}{n_i} \geq 1$

$\log\left(\frac{N}{n_i}\right) \geq 0$

$\log\left(\frac{N}{n_i}\right)$ ① $IDF \geq 0$

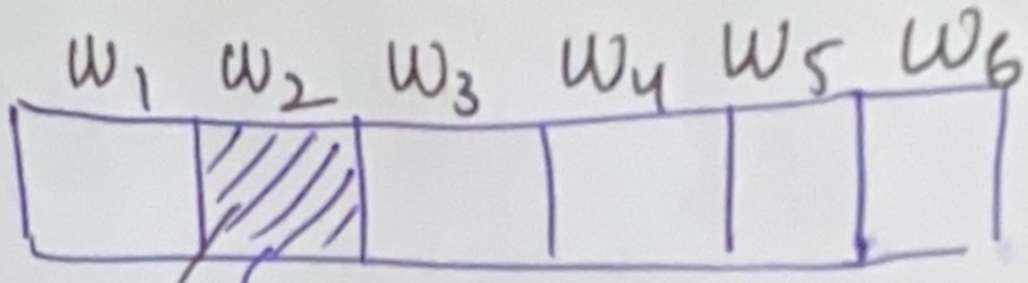
[if n_i increases; $\frac{N}{n_i} \downarrow$] ② $\log\left(\frac{N}{n_i}\right) \downarrow$

$IDF \downarrow$ $n_i \uparrow$
$n_i \downarrow$ $IDF \uparrow$

w_i is more frequent → IDF will be low

w_i is rare word → high IDF

D_c $\begin{cases} r_1: & w_1\ w_2\ w_3\ w_2\ w_4 \\ r_2: & w_1\ w_3\ w_5\ w_6\ w_2 \\ r_3: \\ \vdots \\ r_D: \end{cases}$

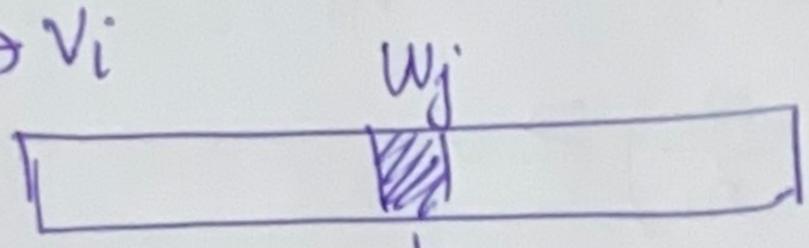


BoW: # occurence of w_2 in r_1

$TF(w_i, r_j) * IDF(w_i, D_c)$

$(r_1, r_2 \text{ --- } r_n) = D_c$

$r_i \rightarrow v_i$



$TF(w_j, r_i) * IDF(w_j, D_c)$

w_j frequent — w_j is rare in D_c

TF-IDF
- more importance to rare words in D_c
- more importance if a word is frequent in doc.

limitation
ignores semantic meaning
(tasty ↔ delicious)
(cheap ↔ affordable)

Why do we use $\log\left(\frac{N}{n_i}\right)$ for IDF ?

$$IDF(w_i, d_c) = \log\left(\frac{N}{n_i}\right)$$

$\frac{N}{n_i}$ → # docs
↳ # docs which contain w_i

↳ Zipf's law

freq ← natural property of eng

← histogram of word occurences

Power law dis

The is — words — civilization

Xv power law
↳ log normal

↓

Gaussian (box cox transform) → $\log(x)$

freq

Power law

words

log(freq)

words

	$\frac{N}{n_i}$	$\log_e\left(\frac{N}{n_i}\right)$	
The	1	0	
is	1	0	1000
⋮			
civilization (1 in 1000)	1000	~6.9	7

TF * IDF

due to log domination of IDF decreases