11) One Hot encoding :-

	Text	O/P
D1	The food is good	1
D2	The food is bad	0
D3	~~The~~ Pizza is Amazing	1

<2> → Vocabolury :-

The, food, is, good, bad, Pizza, Amazing

<3> D1 [[1 0 0 0 0 0 0 0],
[0 1 0 0 0 0 0 0],
[0 0 1 0 0 0 0 0],
[0 0 0 1 0 0 0 0]]

vector representation for D1.

D2 [[1 0 0 0 0 0 0 0],
[0 1 0 0 0 0 0 0],
[0 0 1 0 0 0 0 0],
[0 0 0 0 1 0 0 0]]

D_2

D3 [[0 0 0 0 0 1 0 0],
[0 0 0 0 0 0 0 0],
[0 0 0 0 0 0 1 0],
[0 0 0 0 0 0 0 0]]

D_3

or 3X7

OHE advantages :-

→ Easy to implement.

OHE Disadvantages :-

→ Sparse matrix. → Overfitting.
→ For ML Algorithum fix size of IP but we don't get it
→ No semantic meaning is getting capture.
→ Out of vocabulary. (~~out~~ OOV)

12) Bags of word

Dataset

Text	O/P
He is a good boy	1
She is a good girl	1
Boy & girl are good	1

Step 1 → lower the text
Step 2 → remove stopwords

11/

Vocabulary	Freq
good	3
boy	2
girl	2

T2

	good	boy	girl
S1 →	1	1	0
S2	1	0	1
S3	1	1	1

1) Binary Bag of words contain only 1 & 0
2) normal Bag of words contain counts the freq

Advantages of BOW

→ Simple & intuitive
→ fixed size I/P. for ML Algo

Disadvantage of BOW

→ Sparce matrix problem NLP → Overfitting
→ ordering of words getting changed.
→ Out of vocabolory (OOV)
→ Semantic meaning is still not caching.

S1 The food is good

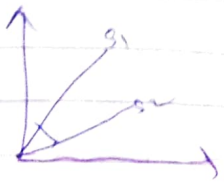
S2 The food is not good.

vectors

	The	food	is	~~not~~ good	not
S1	1	1	1	1	0
S2	1	1	1	1	1

Semantic meaning or cosine similarity show near

Using PCA we convert this in 2D plot



showing near to each other.

3) n-grams

S1 → The food is good

S2 → The food is not good

Step 1: vocabolury & remove stopwords

	food	not	good
S1 →	1	0	1
S2 →	1	1	1

Step 2 → Bigram using

	Food	not	good	foodgood	foodnot	notgood
S1	1	0	1	1	0	0
S2	1	1	1	0	1	1

Sklearn → n-gram → (1,1) → unigram

(1,2) → uni, bigram, ~~tri~~

(1,3) → uni, bi, tri

(2,3) → bi, trigram

b) TF-IDF (Term freq^n - inverse term freq^n)

S1 → good boy
S2 → good girl
S3 → boy girl good

$$\text{Term freq}^n = \frac{\text{no. of rep. of words in sent}}{\text{no. of words in sentences}}$$

$$\text{IDF} = \log_e\left(\frac{\text{no. of sentences}}{\text{no. of sentences containing the word}}\right)$$

Step 1

	Term freq^n			IDF	
	S1	S2	S3		S1 S2 S3
good	1/2	1/2	1/3	good	$\log_e(3/3) = 0$
boy	1/2	0	1/3	boy	$\log_e(3/2) =$
girl	0/2	1/2	1/3	girl	$\log_e(3/2) =$

Step 2

final TF-IDF

	good	boy	girl	girl
S1	0	$\frac{1}{2} \times (\log_e(3/2)$	0	
S2	0	0	$\frac{1}{2} \times \log_e(3/2)$	
S3	0	$\frac{1}{3} \times \log_e(3/2)$	$1/3 \times \log_e(3/2)$	

Note: If a word present in every document the importance is less.

Advantages of TF IDF

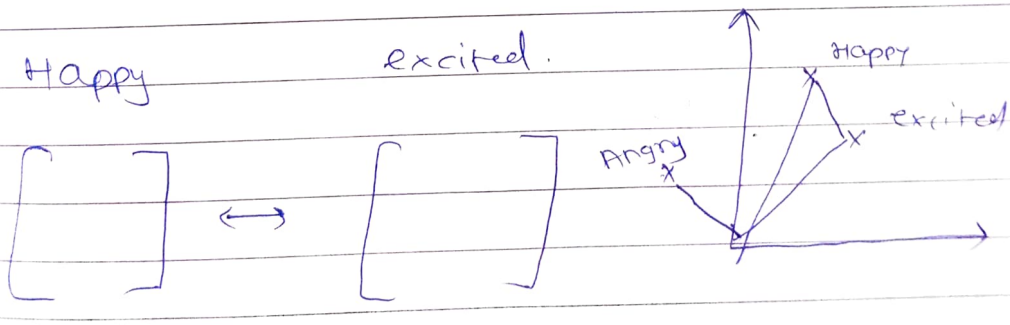
- Simple implementation
- fixed size → vocab size
- word importance is captured.

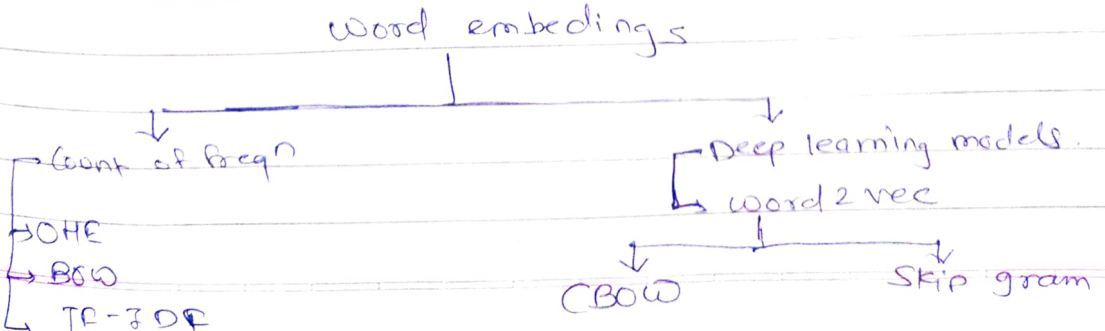
disadvantages

- sparse still exist - overfitting
- OOV (out of vocabolory).

15) Word embeddings

Word embedings are like "numbered maps" that represent the meaning of words. They turn words into simple number so that computers can understand and work with them.





Note:- If a word [illegible] in every document the importance is less.

Advantages of TF IDF

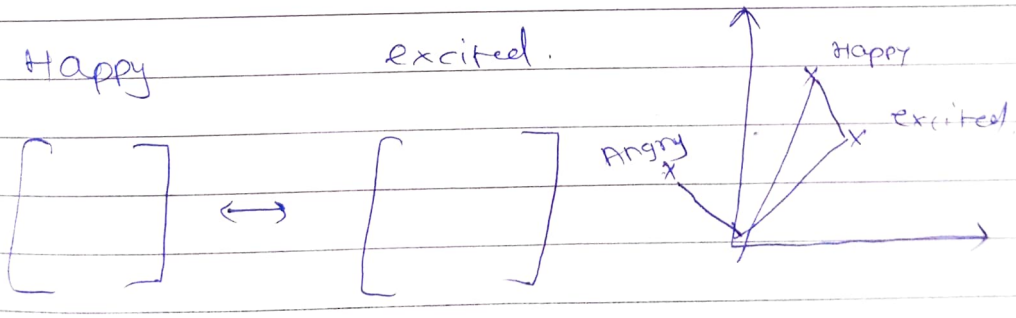
- Simple implementation
- fixed size → vocab size
- word importance is captured.

disadvantages

- sparse still exist - overfitting
- OOV (out of vocabulary).

15) Word embeddings

Word embedings are like "numbered maps" that represent the meaning of words. They turn words into simple number so that computers can understand and work with them.



Word embedings

- Count of freq
  - OHE
  - BOW
  - TF-IDF
- Deep learning models.
  - word2vec
    - CBOW
    - Skip gram

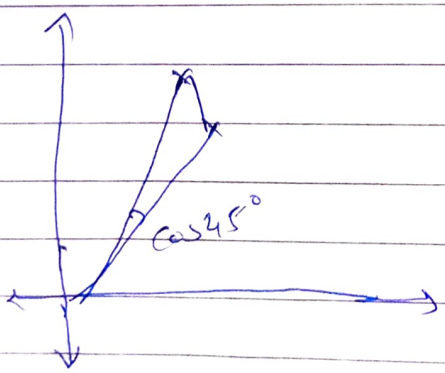
Vocabulary → unique words → corpus

Feature representation

↓	Boy	girl	King	queen	Apple
Gender	-1	1	-0.93	0.92	0.1
Royal	0.01	0.02	0.94	0.95	0.1
Age	0.03	0.04	0.75	0.65	0.95
food	0.01	0.01	0.2	0.1	0.99
.	.				
.	.				
.	.				

[King - Boy + Queen] = girl

Cosine Similarity :



Distance = 1 - cosine sim

Cosine = 1 - cos45°

= 1 - 0.70

= 0.29

Movie recommendation