Rakesh Reddy Pallepati
Class ID : 26

1) N-grams:-

a) Bi-gram Probability:-

$$P(w_i/w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

(or)

Prob. that $word_{i-1}$ is followed by $word_i$ = [No. of times we saw $word_{i-1}$ followed by $word_i$] / [No. of times we saw $word_{i-1}$]

S = Beg. of sentence /S = End of Sentence

$P(I/S) = \frac{2}{3}$

$P(like/I) = \frac{1}{3}$

$P(green/like) = 1/3$

$P(eggs/green) = 1/3$

$P(and/eggs) = 1/3$

$P(ham/and) = 1/3$

$P(/S/ham) = 1/3$

$P(am/S) = 1/3$

$P(sam/S) = 1/3$

1.b) P(I / s / like) = 1/3 = 0.33

P(like / I / green) = 1/3 = 0.33

P(green. / like / eggs) = 1/3 = 0.33

P(eggs / green / and) = 1/3 = 0.33

P(and / eggs / ham) = 1/3 = 0.33

P(ham / eggs / is) = 1/3 = 0.33

P(is / and / ham) = 1/3 = 0.33

P(ham / <is> / and) = 1/3 = 0.33

2.

a) Word2Vec Model:-

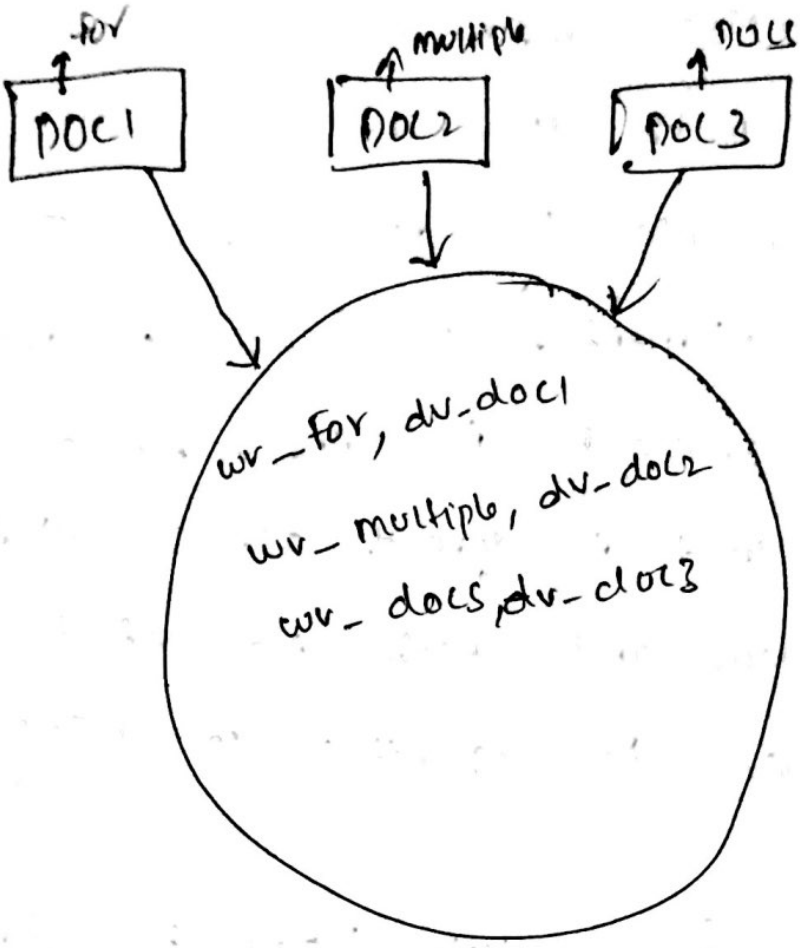
It is a two-layer neural network. It process the text. I/p is text corpus & O/p is set of vectors. feature vectors for words in that corpus. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to another in space.

⇒ Similarity is not expected as 90° angle

Total similarity of 1 is 0° angle

For Example, In word2Vec representation a word is used. Take a vector with 1000 as vector size. Each word is represented by distribution of weights across those elements. So, Instead of one-to-one mapping between an element in the vector and word the representation of word is spread across all the elements in the vector. and each element in the vector contributes to definition of many words.

b) Extension of word2Vec for multiple Documents:- An Extension of word2vec for multiple documents is doc2vec. It is an unsupervised algorithm to generate vectors of sentence / paragraphs & documents. Distributed representation of statements and documents The vectors generated by doc2vec can be used for tasks like finding similarity between sentences /paragraphs & Documents. Unlikely to sequence models like RNN, where word Sentence is captured. in generated sentence vectors.



Most Similard "word"

	Cmt Score
word1	0.916
word2	0.572
word3	0.369

Vectorspace consists of word2vec for each word & additional Doc. vectors.

In Word2Vec Can utilize either of two models architectures to produce a distributed representation of words.

a) Continuous Bag of words (CBOW)

b) Continuous Skip-gram.

Continuous Bag of words:- It is represented using multiple words for given target words. for example we can use "Man" and "animal" as context for "living" as target word. This will call modification.

of neural network architecture. The model rep

I/p to hidden layer. Connections "c" times, the

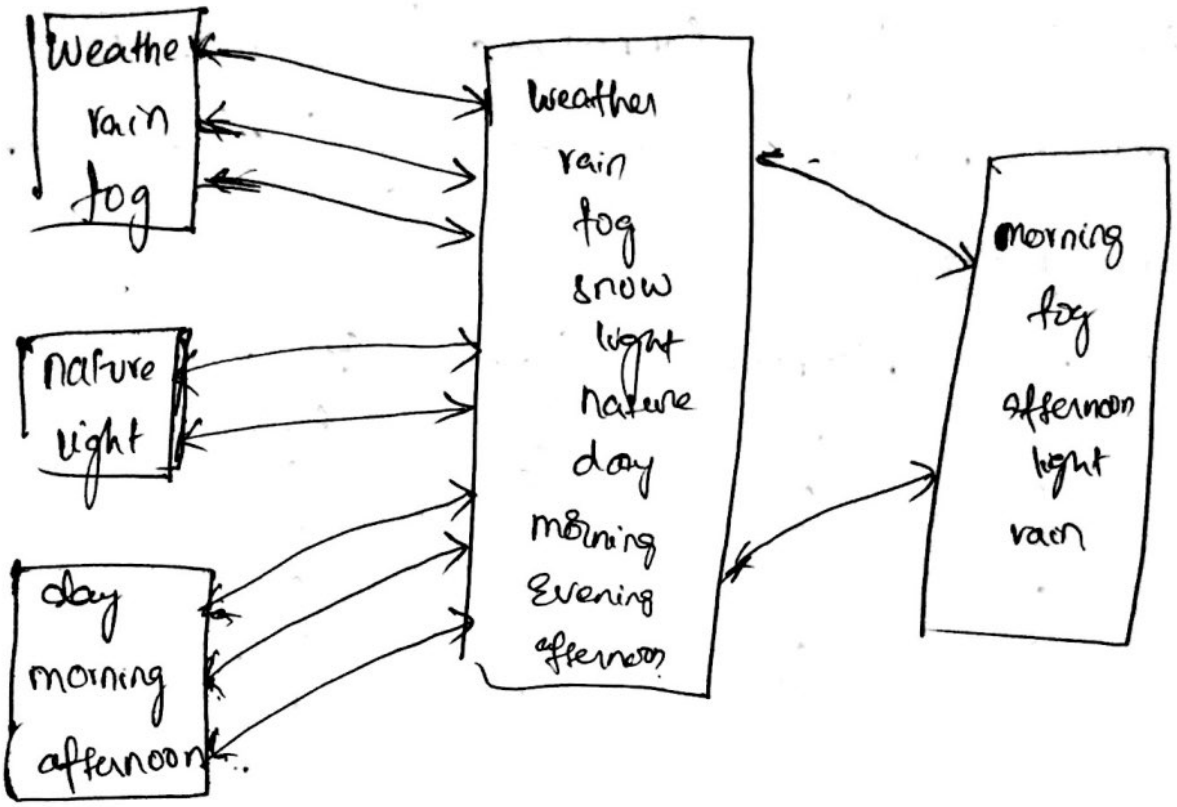
Context words, adding a divide by "c" operat

CBOW model

Morning :-	fog	Afternoon	light	Rain :-
w1	w2	w3	w4	w5
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

W2V

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1



Continuous Skip-Gram model.

It reverses the use of target and context words The target is fed at I/p, the hidden layer remains same. O/p layer of neural n/w is replicated multiple times to accomidate the chosen no. of context words. we can define window size Parameter to configure maximum window size. It works well with small amount of training data. represents even rear words or phrases.

Skip-gram:-

word2vec model