

Topics

- 1) Corpus — Paragraph ✓
- 2) Documents — Sentences ✓
- 3) Vocabulary — Unique Words
- 4) Words —

Corpus
=

{ " My name is KRISH and I have
a interest in teaching ML, NLP
and DL. I am also a Youtuber".
↓

Tokenization

Tokens { sentences }

Words
=

← Tokenization
Tokens ① My name is KRISH and I have
in teaching ML, NLP and

② I am also a Youtuber



[I like to drink Apple Juice. My friend likes Mango Juice]

↓ Tokenization

11 words

Token (sentences)

① I like to drink Apple Juice

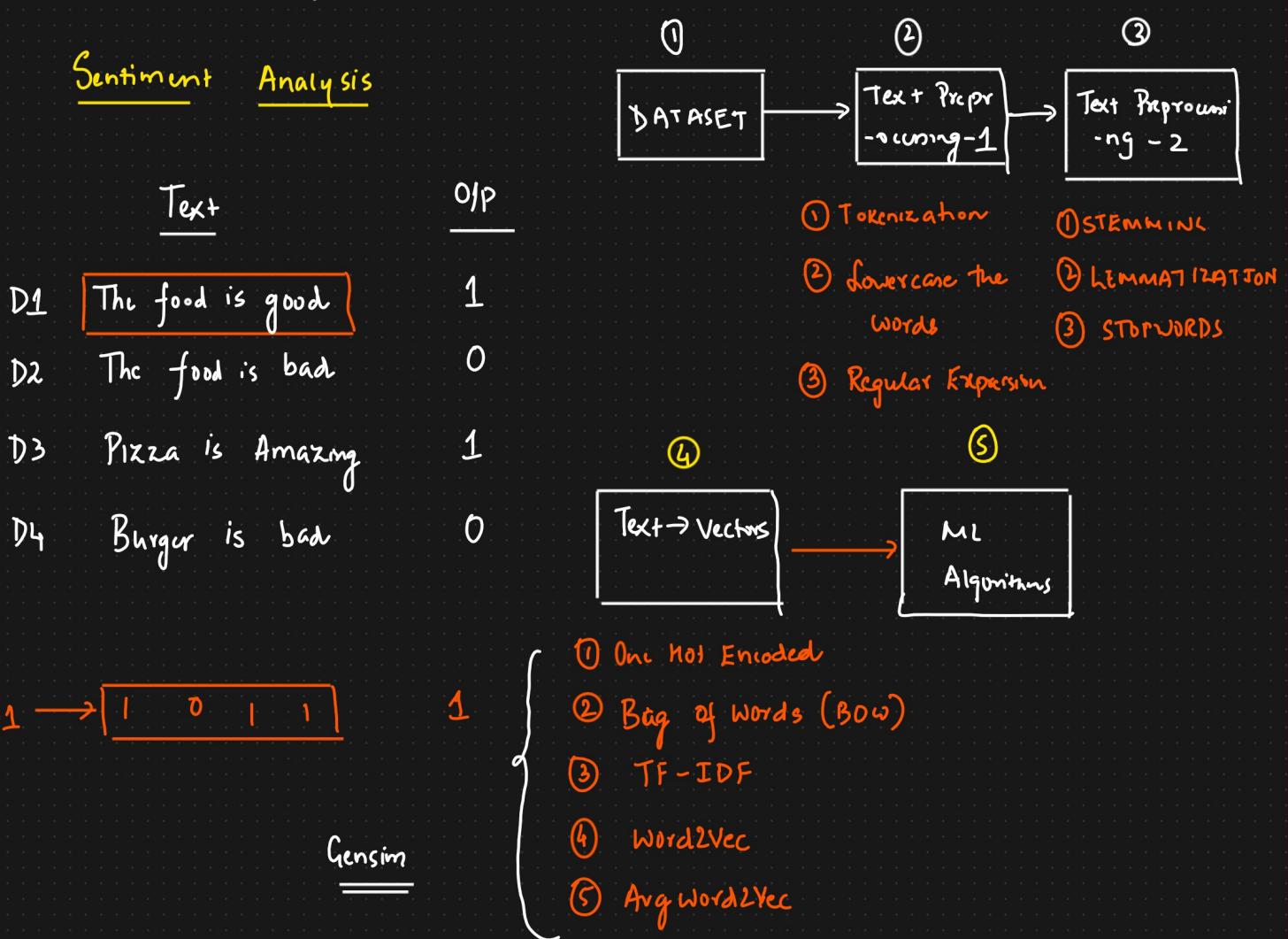
⇒ Unique Words \Rightarrow 10 words.

② My friend like Mango Juice.

Vocabulary



Text Preprocessing → What we have learnt?



① One Hot Encoding

Vocabulary size: 7

Vocabulary {unique words}

	<u>Text</u>	<u>O/P</u>	The food is good bad Pizza Amazing
D1	The food is good	1	1 0 0 0 0 0 0
D2	The food is bad	0	0 1 0 0 0 0 0
D3	Pizza is Amazing	1	0 0 1 0 0 0 0
Test	Burger is bad		
D1	[1 0 0 0 0 0 0],		D2 [1 0 0 0 0 0 0], D3 [0 0 0 0 1 0],
	[0 1 0 0 0 0 0],		[0 0 1 0 0 0 0], [0 0 1 0 0 0 0],
4x7	[0 0 1 0 0 0 0].	4x7.	3x7 [0 0 0 0 0 1]

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

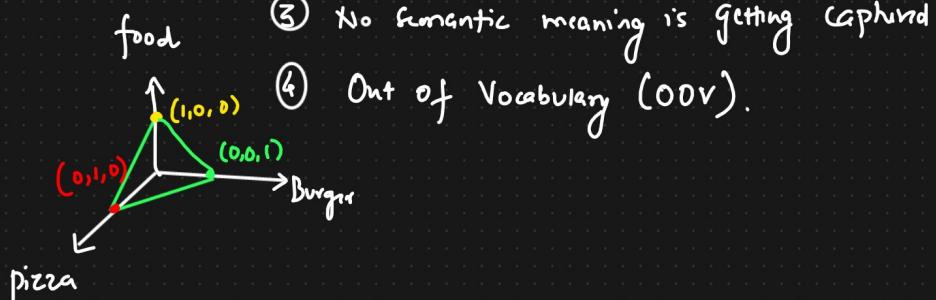
$\left\{ \text{50K vocabulary size} \right\}$ —
Disadvantages

Advantages

- ① Easy to implement with python

[sklearn OneHotEncoder, pd.get_dummies()]

food	pizza	burger
1	0	0
0	1	0
0	0	1



- ① Sparse matrix → Overfitting

- ② ML Algorithm → Fixed Size IIP

- ③ No semantic meaning is getting captured

- ④ Out of Vocabulary (OOV).

② Bag of Words

Dataset

Text	O/P		
He is a good boy	1	Counts all the words case	S1 → good boy
She is a good girl	1	⇒	S2 → good girl good
Boy and girl are good	1	Stopwords	S3 → Boy girl good [School] [Test]

Vocabulary	frequency		[good boy girl]	O/P
good	3	↓	S1 [1 1 0]	1
boy	2	↓	S2 [1 0 1]	1
girl	2	↓	S3 [1 1 1]	1

Binary Bow and BOW

{ 1 and 0 }

{ Count will get updated
based on frequency }

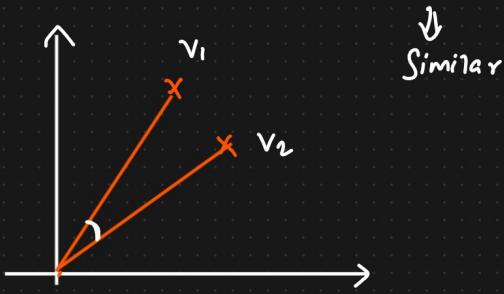
Advantages

- ① Simple and Intuitive
- ② Fixed Sized I/p \rightarrow ML Algorithms

Disadvantages

- ① Sparse matrix or array \rightarrow Overfitting
- ② Ordering of the word is getting changed
- ③ Out of Vocabulary (OOV).
- ④ Semantic meaning is still not captured.

{ The food is good $\rightarrow [1 \ 1 \ 1 \ 0 \ 1] \rightarrow v_1$
 The food is not good $\rightarrow [1 \ 1 \ 1 \ 1 \ 1] \rightarrow v_2$



N-grams Eg: bigrams, trigrams

S1 \rightarrow The food is good
 S2 \rightarrow The food is not good

Bigram

food not good

1 0 1

1 1 1

$[\text{food} \ \text{not} \ \text{good} \ \text{food good} \ \text{food not} \ \text{not good}]$

S1 1 0 1 1 0 0 }
 S2 1 1 1 0 1 1 }

Sklearn \rightarrow n-grams = (1,1) \rightarrow unigrams

= (1,2) \rightarrow unigram, bigram

= (1,3) \rightarrow unigram, bigram,
 trigram

= (2,3) \rightarrow Bigram, trigram.

④ TF-IDF [Term Frequency - Inverse Document Frequency]

S₁ → good boy

$$\text{Term Freq(TF)} = \frac{\text{No. of rep. of words in sentence}}{\text{No. of words in sentence}}$$

S₂ → good girl

S₃ → boy girl good

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

Term Frequency * IDF

	S ₁	S ₂	S ₃	Words	IDF
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	1/2	1/3	girl	$\log_e(3/2)$

Final TF-IDF

	good	boy	girl	<u>Op</u>	<u>Bow</u>
Sent 1	0	$\frac{1}{2} \times \log_e(3/2)$	0	1	1 0
Sent 2	0	0	$\frac{1}{2} \log_e(3/2)$	1	0 1
Sent 3	0	$\frac{1}{3} \log_e(3/2)$	$\frac{1}{3} \log_e(3/2)$	1	1 1

Advantages

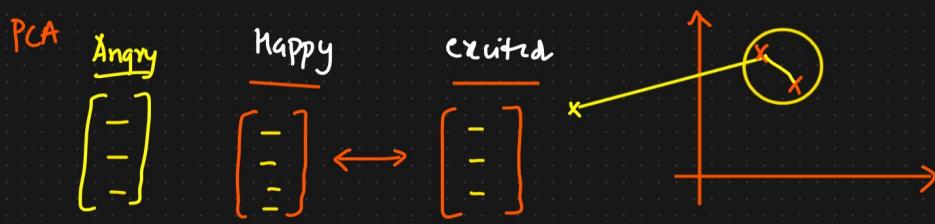
- ① Intuitive
- ② Fixed Size → Vocab size
- ③ Word Importance is getting captured

Disadvantages

- ① Sparsity still exists
- ② OOV

Word Embeddings [Wikipedia]

In natural language processing (NLP), word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.



Words → vectors

Sentence → vectors

Word Embeddings

[ANN]

Count or frequency

- {
- ① One
- ② Bow
- ③ TF-IDF

Deep Learning Trained Model

↓

Word2Vec

Cbow Skipgram

[Continuous BAG OF WORDS]

Word2Vec → Feature Representation

↑ Google

Word2vec is a technique for natural language processing published in 2013. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector.

Vocabulary → Unique Words → Corpus.

	Boy	Girl	KING	QUEEN	Apple	Mango
--	-----	------	------	-------	-------	-------

Gender	-1	1	-0.92	0.93	0.01	0.05
Feature Representation	0.01	0.02	0.95	0.96	-0.02	0.02
Royal	0.03	0.02	0.75	0.68	0.95	0.96
Age	-	-	-	-	0.91	0.92
Food	-	-	-	-	-	-

300 dimension !

$$\begin{bmatrix} - \\ - \\ - \\ - \\ - \end{bmatrix}$$

$$[\text{KING} - \text{BOY} + \text{QUEEN} = \text{GIRL}]$$

$$\text{KING} [0.95, 0.96]$$

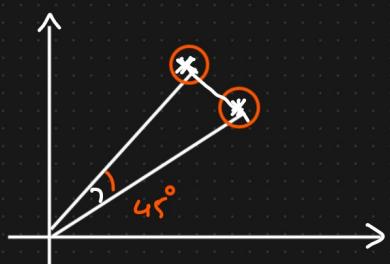
$$\text{Man} [0.95, 0.98]$$

$$\text{QUEEN} [-0.96, 0.95]$$

$$\text{Women} [-0.94, -0.96]$$

$$\text{KING} - \text{MAN} + \text{QUEEN} = \text{WOMEN}$$

Cosine Similarity



$$\text{Distance} = 1 - \text{Cosine Similarity}$$

$$\text{Cosine-Sim} = \cos 45^\circ = \frac{1}{\sqrt{2}} = 0.7071$$

$$\text{Distance} = 1 - 0.7071 \\ \hookrightarrow = 0.29$$

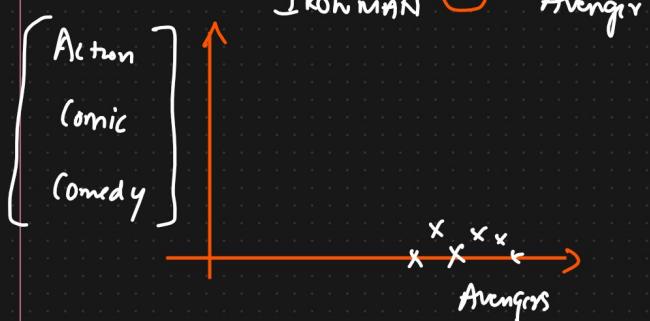
$$\boxed{\sqrt{1-1}=0}$$

$$\text{Distance} = 1 - 0 \\ = 1$$



$$\text{Distance} = 1 - \cos 0^\circ$$

$$= 1 - 1 \\ = 0\%$$



ANN, Loss, Optimizers



Pretained Train A

Model Model For Scratch

① CBOW [Continuous Bag of Words]

CORPUS / DATASET



300 dimension

300 ← Window size = 5 ⇒ Feature Representation

[---] ONE

I/P

O/P

iNeuron [1 0 0 0 0 0 0]

Company [0 1 0 0 0 0 0]

Related [0 0 0 1 0 0 0]

To [0 0 0 0 1 0 0]

→ [iNeuron, Company, Related, To]

→ Company, IS, To, DATA

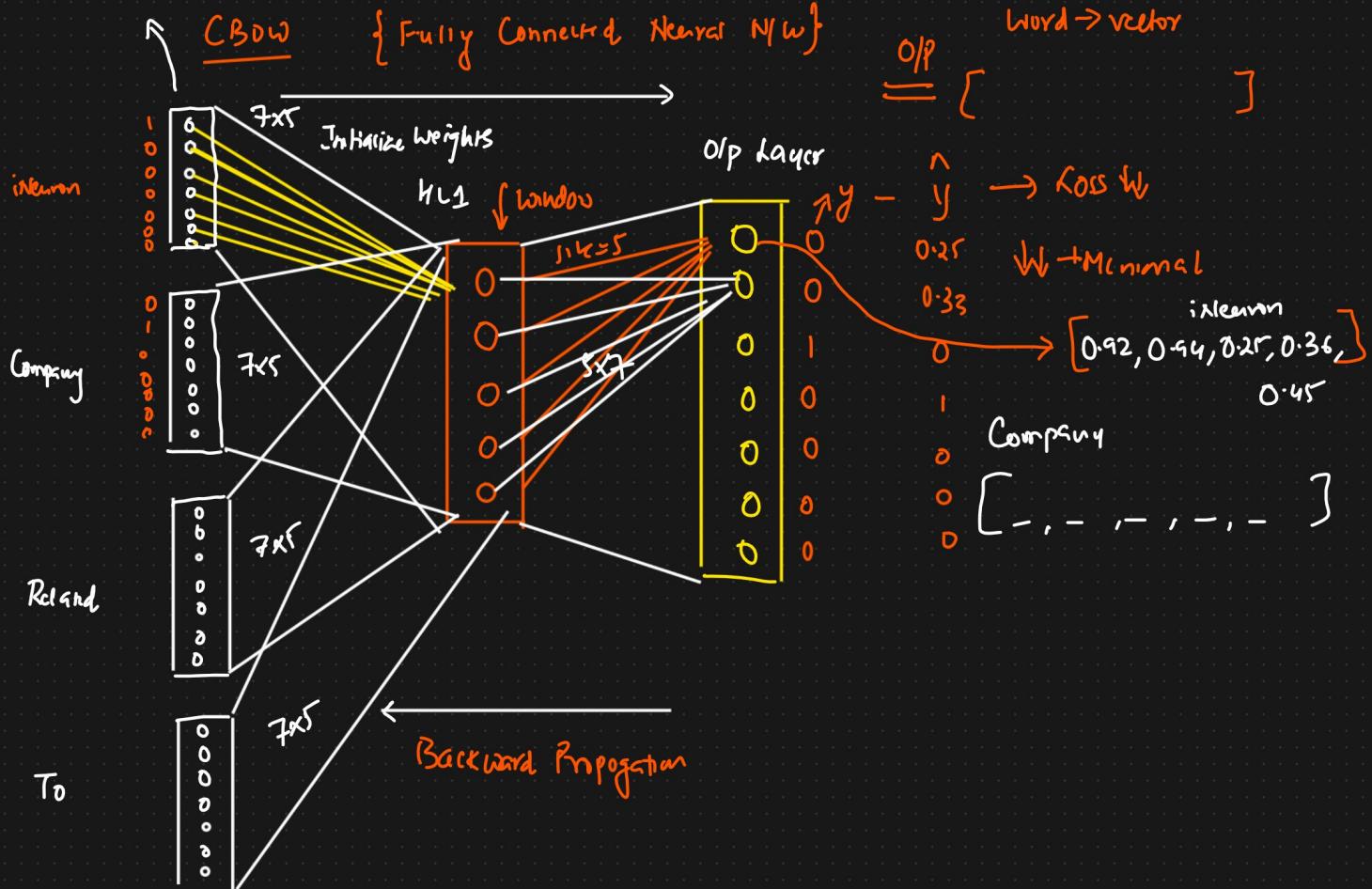
→ Is, Related, DATA, SCIENCE

Related

To

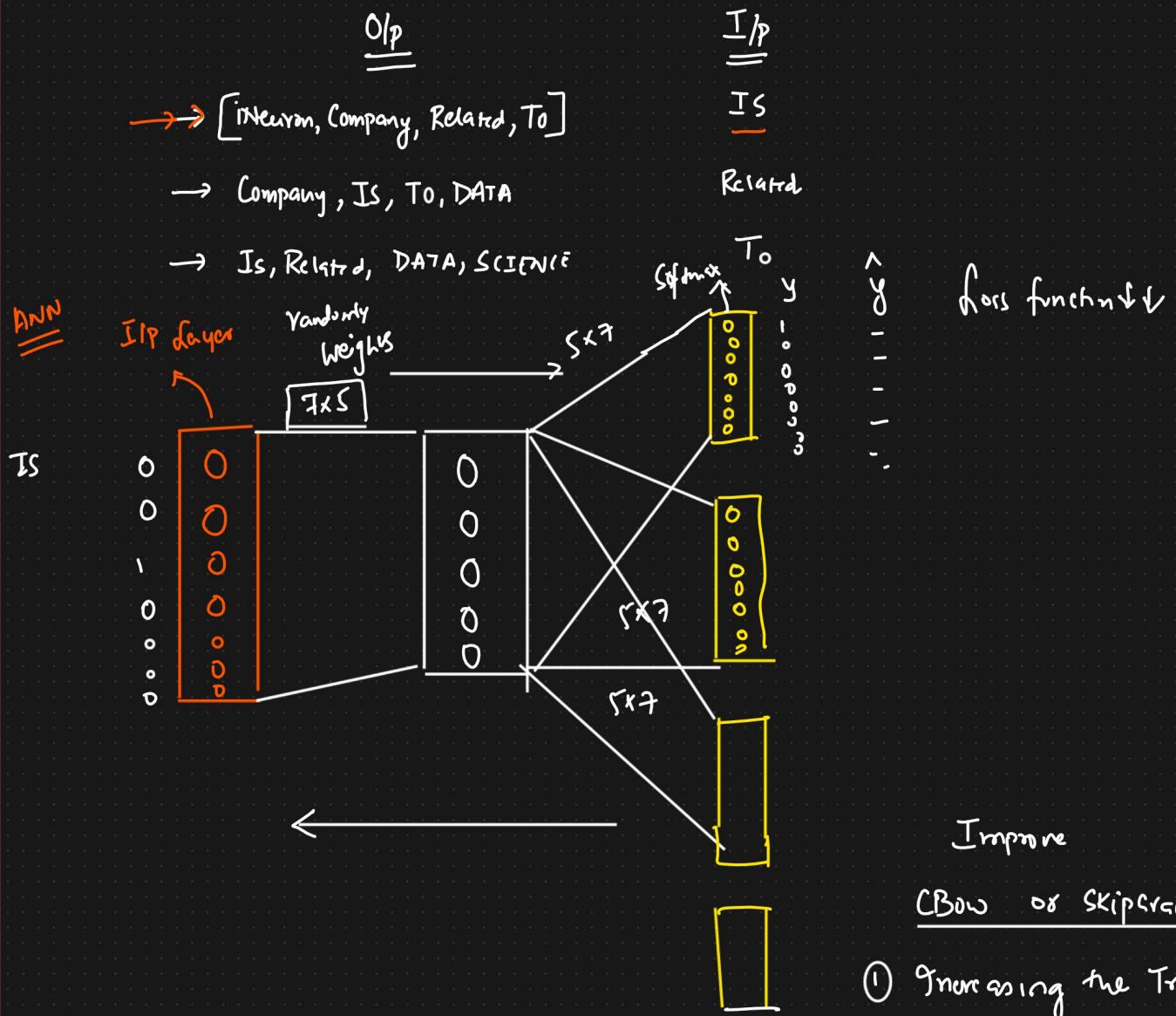
I/P Layer

ANN



② Skipgram - Word2Vec

Window Size = 5



- ① Increasing the Training Data

When Should we apply CBOW or SkipGram.

{ Small Dataset \rightarrow CBOW }
 Huge Dataset \rightarrow SkipGram

- ② Increase the window size
- vector dimension is also increasing.

Google Word2vec

Cnnism

3 billion words \rightarrow Google News

feature representation of 300 dimension vectors]

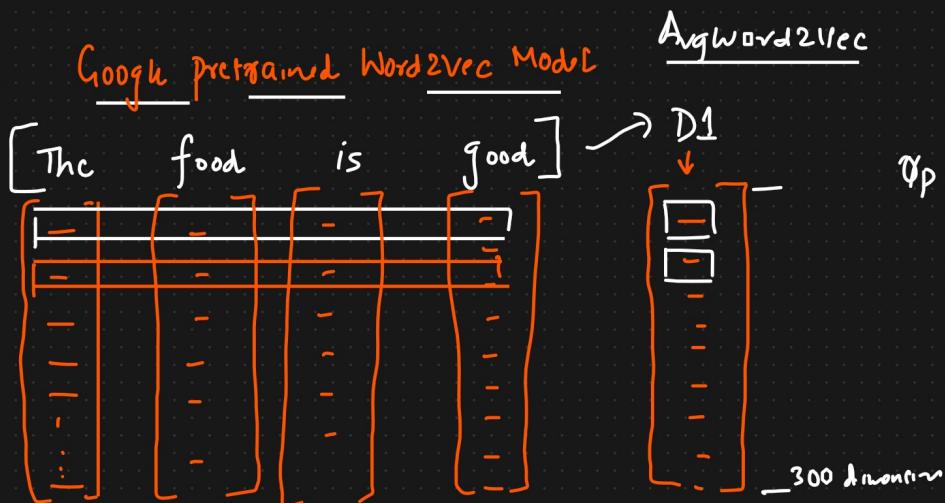
Cricket \rightarrow [- - - - - - - - - - - -].

Advantages of Word2Vec

- f) Sparse Matrix \longrightarrow Dense Matrix
- f) Semantic Info is getting captured $\begin{bmatrix} \text{Moust}, \text{good} \end{bmatrix}$
- f) Vocabulary Size \longrightarrow Fixed set of dimension vectors
Google Word2Vec $[300 \text{ dimension}]$
- f) OOV is also solved

Avg Word2Vec

<u>Text</u>	<u>O/P</u>	<u>Avg word2vec</u>	<u>O/P</u>
D1 The food is good	1	$\begin{bmatrix} - & - & - & - & - & - & - \end{bmatrix}$	$\frac{1}{1}$
D2 The food is bad	0	$\begin{bmatrix} \quad \quad \quad \quad \quad \quad \quad \quad \end{bmatrix}$	
D3 Pizza is Amazing	1	$\begin{bmatrix} \quad \quad \quad \quad \quad \quad \quad \quad \end{bmatrix}$	



Gensim, Glove

\hookrightarrow pretrained Google Word2Vec

