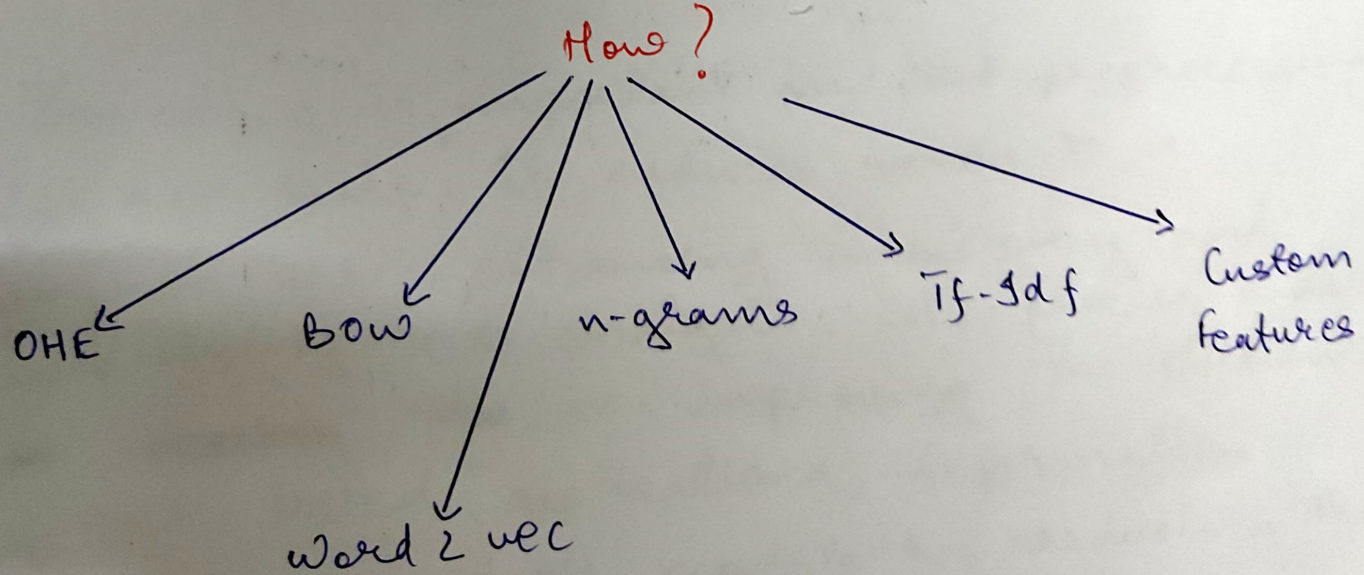
TEXT REPRESENTATION LECTURE-4

Feature Extraction

↳ Feature extraction from text involves transforming raw textual data into a set of relevant features that can be used for analysis or as i/p to machine learning models. In the context of NLP, features are numerical representations derived from the text, capturing its meaningful characteristics.

↳ Extracting meaningful features requires capturing the semantic nuances of language. Words can have multiple meanings depending on context, making it challenging to represent them accurately.



① ONE HOT ENCODING

COMMON TERMS

① **Corpus** – It is a collection of text documents or pieces used for linguistic analysis, language modelling, and NLP research.

② **Vocabulary** – In the context of NLP, it refers to the set of unique words present in a corpus.

③ **Document** – A document is a unit of text, which can be as short as a sentence or as long as a book.

④ **Word** – A word is a unit of language that carries meaning. In NLP, words are fundamental building blocks for analysis, and understanding their frequency, context, and semantics is crucial for various NLP tasks.

(A) ONE HOT ENCODING

D1 – People watch campusx
D2 – Campusx watch campusx
D3 – people write comment
D4 – campusx write comment

people	watch	campusx	write	comment
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

D1 = [[1,0,0,0,0], [0,1,0,0,0], [0,0,1,0,0]]

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

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

D4 = [[0,0,1,0,0], [0,0,0,1,0], [0,0,0,0,1]]

pros – intuitive, easy to implement.

cons – sparse array, irregular i/p size can prevent the ML algorithm from getting trained, out of vocabulary (OOV), no capturing of semantic meaning.

(B) BAG OF WORDS

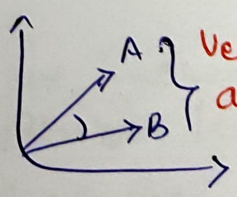
Text	O/P
D1 – people watch campusx	1
D2 – campusx watch campusx	1
D3 – people write comment	0
D4 – campusx write comment	0

	people	watch	campusx	write	comment
D1	1	1	1	0	0
D2	0	1	2	0	0
D3	1	0	0	1	1
D4	0	0	1	1	1

Pros – simple, intuitive, better semantic understanding than OHE, O/P array size will always be fixed even if in I/P we provide new words,

Cons – Sparsity (causes overfitting), ignores OOV words, not considering the order becomes an issue, BOW can't capture huge difference in meaning b/w 2 sentences caused by few words. For eg:-

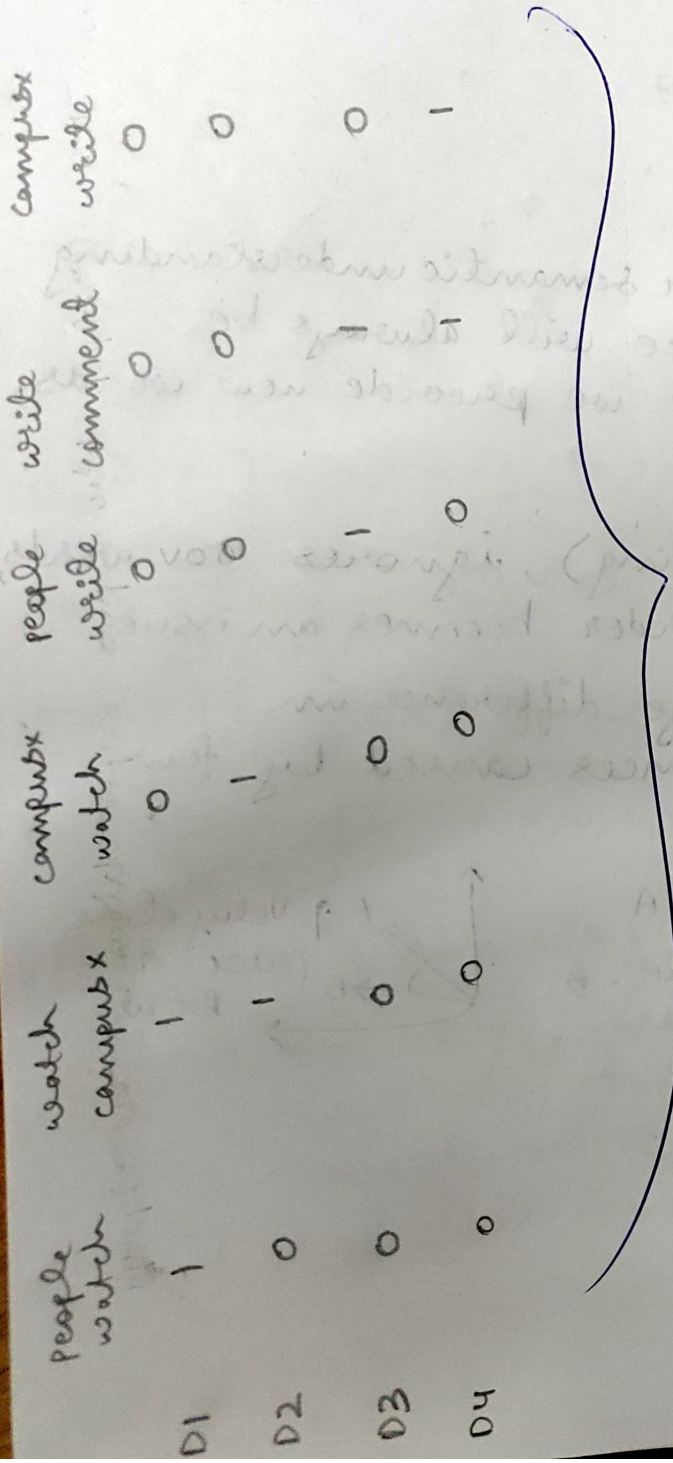
→ This is a good movie. A
→ This is not a good movie. B



N-GRAMS

D1 - people watch campusx
D2 - campusx watch campusx
D3 - people write comment
D4 - campusx write comment

* Bag of Bi-Grams



* Bag of Tri-Grams.

	people watch campusx	S2	S3	S4
D1	1	0	0	0
D2	0	1	0	0
D3	0	0	1	0
D4	0	0	0	1

NOTE:- If you try to form Quad-Grams from the above dataset, then it'll throw an error. This is because no document has more than 3 words.

→ Also Bag of words is a type of Uni-Gram, hence it is a special case of N-Grams.

Pros - Sentences like:-

(i) This is a very good movie.

(ii) This is not a good movie.

can be differentiated from each other if used Bi-Gram or tri-gram or n-gram.

- Able to capture semantic meaning of the sentence.
- Intuitive, easy implementation

Cons - Slows down the algorithm as we increase the value of N because the no. of features increases, no solution to OOV problem all we can do is ignore.

⑤ Tf-Idf

Tf : Term Frequency

Idf : Inverse Document Frequency

↳ The idea behind Tf-Idf is to quantify the importance of a term in a document relative to its importance across a collection of documents.

* $TF(t, d) = \frac{\text{No. of occurences of term } t \text{ in doc. } d}{\text{Total no. of terms in the doc. } d}$

* $IDF(t) = \log_e\left(\frac{\text{Total no. of docs. in corpus}}{\text{No. of documents with term } t \text{ in them}}\right)$

* why log is used in Idf ?

↳ Helps balance and stabalize the importance scores of words, preventing extreme values and ensuring fair representation across different documents.

Pros- Information Retreival

Cons- Sparsity, OOV, ↑ dimension, semantic