NLP

1. Numerical columns → Distribution analysis [Boxplot, Histogram]
2. Categorical columns → Frequency analysis [Bar chart, Treemaps]
3. Date column → Trend analysis [Line charts, Calendar]
4. Location columns → Geographical analysis [Maps]
5. Text data → Bag of word analysis [Word cloud]

Bag of word analysis

Tokens → Frequency

Text

T1 [— — — —] D1
T2 [— — — —] D2
T3000 [— — — —] D3000

Corpus – Collection of processed documents

Document – Collection of tokens

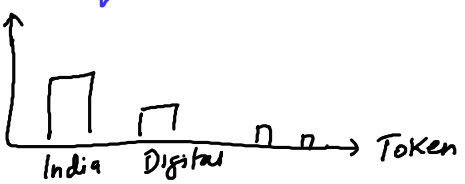
Tokens – Collection of words
- Unigram: One word /Token ✓
- Bigrams: Two words /Token
- Trigrams: Three words /Token
- Ngrams: N words /Token

Word → Tokens → Document → (Processed) → Corpus

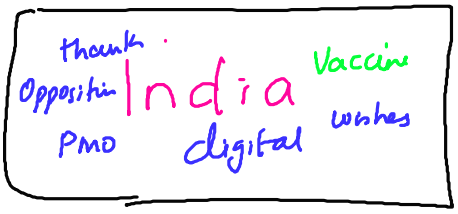
Bag of words → Frequency analysis of Token

	Token	Frequency
T1	India	50
T2	digital	25
⋮	⋮	⋮
T_N		

Barchart →



Word cloud →



Font size ≈ Frequency

Word cloud

D1 [— — —]
D2 [— — —]
⋮ [— — —]
D3320 [— — —]

[— — —
— — —
— — —
— — —
— — —]

[This product is not good.]

[product good]

Text cleaning

- Lower case conversion
- Retained only alphabets & few characters
- Removed common & custom stop words

IMDB Sentiment

Text classification → Sentiment analysis — Rule based
                                          \ Text classification
                    ↳ Ticket system
                    ↳ Intent identification

→ Feature Extraction

→ Stemming
→ Curse of high dimensionality ~ Thousands
→ Vectorization Text → Numbers → Document Term matrix
                                → TFIDF

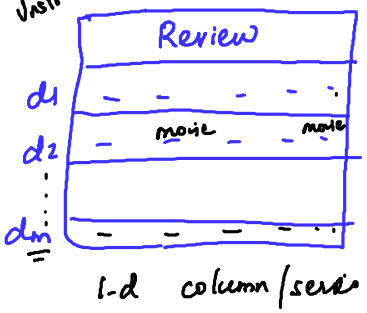
Vectorization

Text data → Structured numerical features

- ✓ Document Term Matrix
- Word Embeddings (word2vec)

Document Term Matrix

Unstructured



Vectorization →

Term ≈ Token

Freq. of a term in the corresponding doc

	T1	T2	T3	. .	Tn
	India	movie	good	. . . . .	awesome
d1	0	1	3	0 0 0	1
d2	0	2	0		0
⋮	0	0	1	0 0 0	0
dm	1	1	0	0 0 0	1

m x n matrix

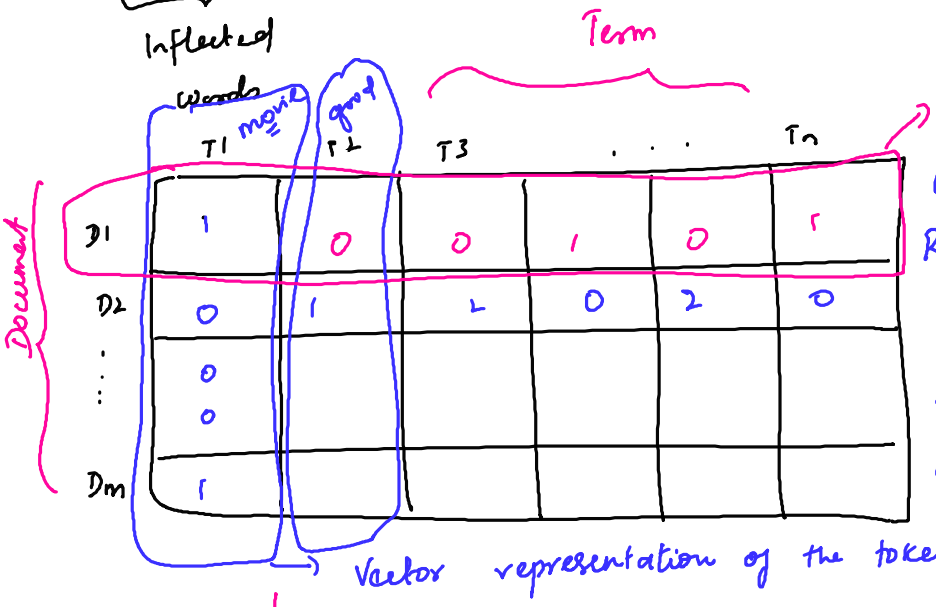
n → Total no. of unique tokens

m → Total no. of documents

Looking
Look
Looks
} Look → Root word

Stemming ✓
Lemmatisation

Inflected words



Document Term Matrix – DTM – Doc classification / cluster.

$(DTM)^T \approx$ Term Document Matrix (TDM) → Word clustering

TF - IDF – Term Frequency - Inverse Document Frequency

	T1	T2	T3	T4
D1	1	0	0	1
D2	0	0	1	1
D3	0	1	1	2

DTM

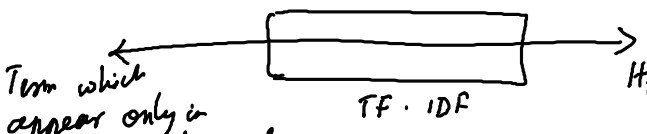
$$T.F(T_1, D_1) = \frac{\text{Frequency of the Term in the document}}{\text{Total no. of tokens in the document}}$$

$$T.F(T_1, D_1) = \frac{1}{2} = 0.5$$

$$\text{Inverse Document Frequency } (T_i) = \log\left(\frac{\text{No. of document in the corpus}}{\text{No. of documents in which the term appears}}\right)$$

$$IDF(T_1) = \log\left(\frac{3}{1}\right) = \log(3)$$

$$IDF(T_4) = \log\left(\frac{3}{3}\right) = \log(1) = 0$$



Term which appear only in fewer documents
TF (low)
IDF (high)

High frequent appearance in all documents
TF (high)
IDF (low)

$$TF.IDF\ (T_i, D_i) = \frac{\text{Freq of } T_i \text{ in } D_i}{\text{Total freq. in } D_i} * \log\left(\frac{\text{Total no. of docs in corpus}}{\text{Total no. of docs in which the } T_i \text{ appears}}\right)$$

This movie is good

Unigrams	Bigrams	Trigram
This	This movie	This movie is
movie	movie is	movie is good
is	is good	
good		
