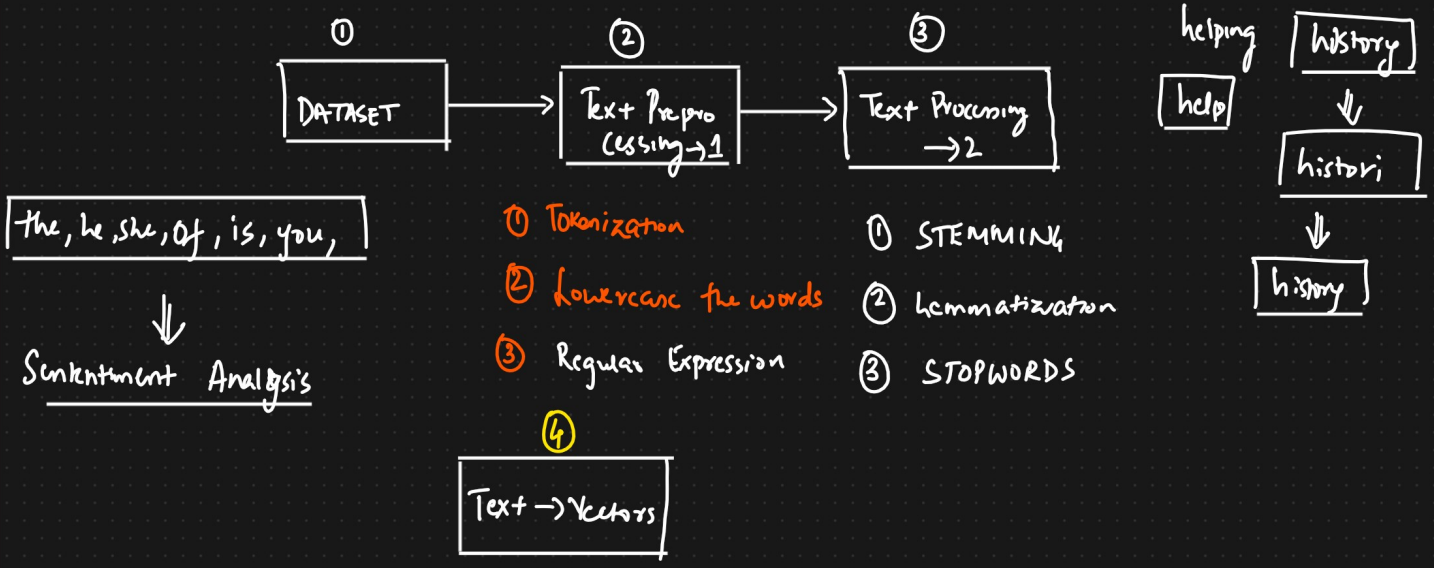
NLP In Machine Learning



Topics

1) Corpus ⟶ Paragraph
2) Documents ⟶ Sentences
3) Vocabulary ⟶ unique words
4) Words

Vocabulary size = 18 words.

Tokenization

Corpus:

"My name is KRISH and I have a interest in teaching ML, NLP and DL. I am also a good human".

⇓ Tokens

① My name is KRISH and I have an interest in " " ' ' '

② I am also a good human.

① One hot Encoding

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

Sentiment Analysis Problem

vocabulary ⇗

the	food	is	good	bad	pizza	amazing
1	0	0	0	0	0	0
0	1	0	0	0	0	0

Test [Burger is Bad] X

D1 [[1 0 0 0 0 0 0],
[0 1 0 0 0 0 0],
[0 0 1 0 0 0 0],
[0 0 0 1 0 0 0]]
4x7

D2 [[1 0 0 0 0 0 0],
[0 1 0 0 0 0 0],
[0 0 1 0 0 0 0],
[0 0 0 0 1 0 0]]
4x7

D3 [[0 0 0 0 0 1 0],
[0 0 1 0 0 0 0],
[0 0 0 0 0 0 1]]
3x7

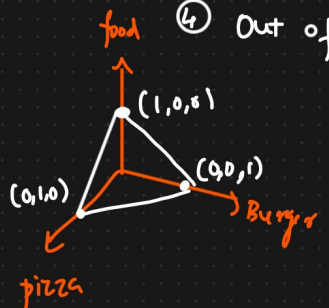
Advantages

① Easy to Implement with python
[Sklearn OneHotEncoder]

Disadvantages

① Sparse Matrix → Overfitting
② ML Algorithms → Fixed Size I/p.
③ No semantic is getting captured
④ Out of Vocabulary (OOV)

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



No Relation

② Bag of Words (BOW)

Dataset

Text	O/P
① ② ③ good boy	1
She is a good girl	1
Boy And girl are good	1

Lowercase all the words
⟹
Stopwords

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

Vocabulary	frequency
good	3
boy	2
girl	2

	f1 [good	f2 boy	f3 girl]	O/p
S1	1	1	0	1
S2	1	0	1	1
S3	1	1	1	1

Binary Bow

Advantages

1. Simple and Intuitive
2. Fixed I/p Size ⇒ ML Algorithms.

Disadvantages

1. Sparse Matrix → overfitting
2. Out of vocabulary (OOV Problem)
3. Semantic Meaning not there. ✗

The food is good [1 1 1 0 1] → v1

The food is not good [1 1 1 1 1] → v2

N-grams

Eg: unigram, bigram, trigram

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

Bigram (1,2) Trigram (1,3)

	food	not	good	food not	not good	food good
S1	1	0	1	0	0	1
S2	1	1	1	1	1	0

Sklearn → n-gram = (1,1) → unigrams

(1,2) → unigrams, bigrams

(1,3) → unigrams, bigrams, trigrams

(2,3) → bigram, trigram

(2,2) → Bigram.

③ TF-IDF [Term frequency – Inverse Document Frequency].

S1 → good boy

S2 → good girl

S3 → boy girl good

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

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

Term frequency

	S1	S2	S3
good	1/2	1/2	1/3
boy	1/2	0	1/3
girl	0	1/2	1/3.

IDF

Words	
good	$\log_e(3/3) = 0$
boy	$\log_e(3/2)$
girl	$\log_e(3/2)$

Final TF-IDF

	f1 good	f2 boy	f3 girl	O/P
Sent 1	0	$\frac{1}{2}\log_e(3/2)$	0	1
Sent 2	0	0	$\frac{1}{2}\log_e(3/2)$	1
Sent 3	0	$\frac{1}{3}\log_e(3/2)$	$\frac{1}{3}\log_e(3/2)$	1

Advantages

1. Intuitive
2. Fixed Sized I/p → Vocab Size
3. Word Importance is getting Captured

Disadvantages

1. OOV
2. Sparsity is still exists

⇓

Word2Vec