Feature Engineering in NLP

Common Terms

1) Corpus(c) = Sabda Sagar (pura dataset ma vayeko words lai Joddha gaune Combination lai corpus vaninxa) - Total combination of words in a dataset

eg -

my name is mira	pos
I am happy	neg

= { my name is mira, I a happy }

corpus = 7

2) Vocabulary(v) = Jati pani unique words corpus bany o teslai vocabulary vanxa. (no of unique words in the corpus)

3) Document(D) = Each individual row of the dataset is Document

4) word(w) = Each individual word in a document is word

One Hot Encoding

D1	People watch campusx
D2	campusx watch campusx
D3	People write comment
D4	campusx write comment

Corpus =

People watch campusx campusx watch campusx People write comment Campusx write comment

vocabulary = people watch campusx write comment

hami vocabulary bata ~~ma~~ vector form garxam

People	watch	campusx	write	comment
•	•	•	•	•
0	1	0	0	0

→ One hot encoding ma hami harek word lai in vdim vector ko represent ma convert garxam by above vocabulary table:

let's take example of document (D_1)

D_1 = people watch campusx

representation of people =

people	watch	campusx	write	comment
1	0	0	0	0

representation of watch =

people	watch	campusx	write	comment
0	1	0	0	0

representation of campusx =

people	watch	campusx	write	comment
0	0	1	0	0

∴ The vectorical representation of D_1 will be

$$D_1 = [[1,0,0,0,0],[0,1,0,0,0],[0,0,1,0,0]]$$

Similarly for D_2 from vocabulary table

$$D_2 = [\underset{\text{campusx}}{[0,0,1,0,0]},\underset{\text{watch}}{[0,1,0,0,0]},\underset{\text{campusx}}{[0,0,1,0,0]}]$$

And Similar for other also #

① People watch campus - D_1 = [[1,0,000] [00,100] = (3,5) [1,0,000]

② People watch campus watch = D_2 = [[1,0,000] [0,0,100] [1,0000] [00,10000]] = (4,5) can't be used in ml algorithm

Date

pros

→ Intutive

→ easy to implement

flaws

→ sparsity [1,00, 0000000]

→ ~~No~~ It needs same dimension input i.e words in document must be same

→ OOV(It can't deal with word which is not in vocabulary sinle word vector representation is not possible

→ NO capturing of Semantic meaning

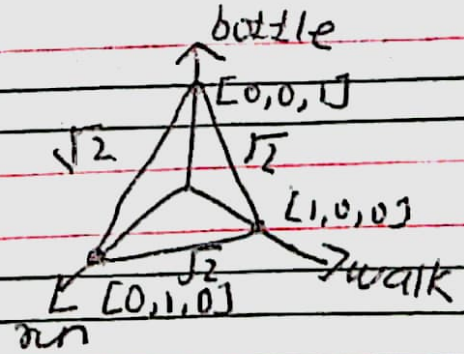
eg

In our vocab we have

walk	run	bottle

walk vector form = [1, 0, 0]
run vector form = [0, 1, 0]
bottle vector form = [0, 0, 1]

If we plot in 3d spale



All the distanles are same i.e equi distanle sinle the relation of walk, bottle and run bottle is not prevelf in real word Scenerio but our algorith shows ~~so~~ it as a relation so, it can't capture semantic meaning

Bag of words:

D_1	People watch miraj	1
D_2	miraj watch miraj	1
D_3	People comment miraj	0
D_4	miraj comment miraj	0

Corpus:
People watch miraj miraj watch miraj people comment miraj miraj comment miraj

Vocabulary: people watch miraj write comment
V = 5

people	watch	miraj	write	comment

Bag of words = yesma hami document ma vayeko harek word kati choti aayo (repeat) teslai herera ^ vector ma convert garma vdim vector ma.

Eg - Taking document D_1
People watch miraj

Vector representation of D_1 =

people	watch	miraj	write	cmnt
1	1	1	0	0

(miraj watch miraj)
Taking document D_2
Vector vepsetation of D_2 =

people	watch	miraj	write	cmnt
0	1	2	0	0

Similary
D_3 =

People	watch	miraj	write	comment
1	0	1	0	1

D_4 =

PEOPLE	watch	miraj	write	comment
0	0	2	0	1

∴ $D_1 = [1, 1, 1, 0, 0]$

$D_2 = [0, 1, 2, 0, 0]$

$D_3 = [1, 0, 1, 0, 1]$

$D_4 = [0, 0, 2, 0, 1]$

→ Bag of words ma dui ota same class denote garni document ko word frequency similar hunxa.

eg.

D_3 People write comment	0
D_4 campus write comment	0

Here, write = 1, comment = 1 times in document and this document denote 0 both.

TO use Bag of words we have count vectorizer in Sci-kit learn.

It has disadvantage i.e

I am miraj	1
I am not miraj	0

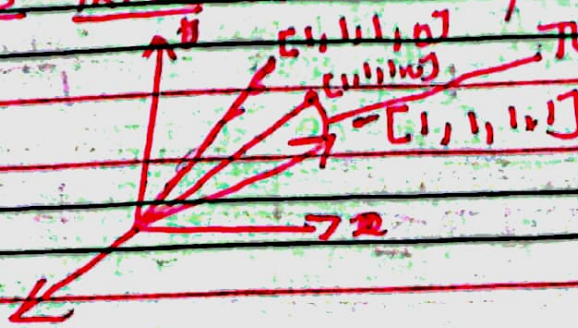
vocab: [I | am | miraj | not]

$D_1 = [1, 1, 1, 0]$

$D_2 = [1, 1, 1, 1]$

ota dimension ma word ko frequency same xa so, ML model will be confused and they both a same

The three dimension of vector are same



2) OOV - kunai word vocab vanda out bata aayo vane tesai vector ma we nai gardena

like - we have vocab → won | game | wow

Now, we have sentence lottery! you won iphone
Now, in vector this sentence will be [1, 0, 0]
wow game wow

∴ lottery!, iphone is not used since it can be useful information like in spam mail.

3) Sparse - It also create the sparse matrix which may lead to overfit

PTO

N-Grams: (Bag of n-grams)

D_1	People watch campusx	1
D_2	Campusx watch campusx	0
D_3	People write comment	0
D_4	Campusx write comment	0

Vocabulary:- (Uni-gram default) = Bag of words

People	watch	campusx	write	comment

N-gram ma hami multiple words ko combination lai vocabulary ma rakxam to prevent ~~word~~ sentence meaning.

∴ Bi-gram: we use two words to make vocabulary:

b_1	People watch campusx	1
b_2	campusx watch campusx	0
b_3	people write comment	0
b_4	Campusx write comment	0

Vocab Candidates: (words must be serial)

People watch, watch campusx, campusx watch, watch campusx, people write, write comment, campusx write, write comment

Vocab:

people watch	watch campusx	campusx watch	people write
write comment	campusx write	Write campusx	

Now, converting documents into vector using Bi-gram.

people watch	watch campusx	campusx watch	people write	write mirat	campusx write	write campusx
A	B	C	D	E	F	G

A B C D E F G

D_1 = 1, 0, 0, 0, 0, 0, 0

D_2 = 0, 1, 1, 0, 0, 0, 0

D_3 = 0, 0, 0, 1, 1, 0, 0

D_4 = 0, 0, 0, 0, 1, 1, 0

$\therefore$ $D_1 = [1, 0, 0, 0, 0, 0, 0]$
$D_2 = [0, 1, 1, 0, 0, 0, 0]$
$D_3 = [0, 0, 0, 1, 1, 0, 0]$
$D_4 = [0, 0, 0, 0, 1, 1, 0]$

$\therefore$ Tri-grams: we use three words to make vocab

People watch campusx Hello	1
mirat People watch campusx	0

Vocab Candidate:
people watch campusx, watch campusx Hello,
mirat people watch, people watch campusx

Vocab:

people watch campusx	watch campusx Hello	mirat people watch

D_1 = 1, 1, 0

D_2 = 1, 0, 1

$\therefore$ $D_1 = [1, 1, 0]$
$D_2 = [1, 0, 1]$

Tf - Idf: (Term frequency - Inverse document frequency)

D_1	People watch campusx	1
D_2	Campusx watch campusx	1
D_3	People write comment	0
D_4	Campusx write comment	0

Vocab:

people	watch	campusx	write	comment

∴ In Tf-Idf each token is vectorized by using Tf-Idf ~~metho~~ formula.

$$\Rightarrow Tf(t,d) = \frac{\text{(Number of ocurrenles of term } t \text{ in document } d)}{\text{(Total no of terms in document } d)}$$

∴ It's like probability Tf value lies between 0 and 1 i.e $0 \leq Tf \leq 1$

$$\Rightarrow Idf(t) = \left(\log_e \frac{\text{(Total no of documents)}}{\text{(No of documents with terms in them)}} \right) + 1$$

Tf-Idf ma kunai particular word euta document ma dherai aako xa ani arv document ma-ne frequent aako xa vane tyo particular word ko valve high assign garxa. (i.e means the word must be valuable to the document as well as whole corpus)

now, let's vectorize the documents for that we need to calculate Idf valve of volab words:

~~Pee~~

	IDF	
people	$\log(\frac{4}{2})+1$	$\Rightarrow 1.69$
watch	$\log(\frac{4}{2})+1$	$\Rightarrow 1.69$
campusx	$\log(\frac{4}{3})+1$	$\Rightarrow 1.28$
write	$\log(\frac{4}{2})+1$	$\Rightarrow 1.69$
comment	$\log(\frac{4}{2})+1$	$\Rightarrow 1.69$

$$Idf = \log\left(\frac{\text{Total no of documents}}{\text{no of document where term t is present}}\right)+1$$

now, let's give valve to each volab words on the basis of (Tf x Idf):
let's convert all the documents into vectors

P.T.O

	People	watch	campusx	write	comment
D1	1/3 × 1.69	$\frac{1}{3}$ × 1.69	$\frac{1}{3}$ × 1.28	$\frac{0}{3}$ × 1.69	$\frac{0}{3}$ × 1.69
D2	$\frac{0}{3}$ × 1.69	$\frac{1}{3}$ × 1.69	$\frac{2}{3}$ × 1.28	$\frac{0}{3}$ × 1.69	$\frac{0}{3}$ × 1.69
D3	$\frac{1}{3}$ × 1.69	$\frac{0}{3}$ × 1.69	$\frac{0}{3}$ × 1.28	$\frac{1}{3}$ × 1.69	$\frac{1}{3}$ × 1.69
D4	$\frac{0}{3}$ × 1.69	$\frac{0}{3}$ × 1.69	$\frac{1}{3}$ × 1.28	$\frac{1}{3}$ × 1.69	$\frac{1}{3}$ × 1.69

D1	0.56	0.56	0.42	0	0
D2	0	0.56	0.85	0	0
D3	0.56	0	0	0.56	0.56
D4	0	0	0.42	0.56	0.56

D1, D2 → 3 dim milt

D2, D3 → 0 dim milti

D3, D4 → 3 dim milt

Same class kaxhi chinta yes bata ne tezma milta.

∴ D_1 = [0.56, 0.56, 0.42, 0, 0]

D_2 = [0, 0.56, 0.85, 0, 0]

D_3 = [0.56, 0, 0, 0.56, 0.56]

D_4 = [0, 0, 0.42, 0.56, 0.56]