Monday, 12 July 2021 6:39 PM

Let's understand how a sentence is represented in numeric (vector) form using bag of words (BOW) technique.

Say you want to represent following sentences in numeric form

- → Dog bites man.
- → Man bites dog
- → Dog eats meat.
- → Man eats food.
- → Dog Dog friends.

→ clean the text eg lower all words, remove punctuations, ... using regular expressions / string manipulation

→ then remove stop words from these sentences

→ then apply stemming / lemmatization on these sentences

→ vocabulary set = {dog, bites, man, eats, meat, food}

(lastly do word tokenize)

this sequence could be as you wish.

	Dog	bites	man	eats	meat	food
Dog bites man	1	1	1	0	0	0
Man bites dog	1	1	1	0	0	0
Dog eats meat	1	0	0	1	1	0
Man eats food.	0	0	1	1	0	1

Dog bites man	$\xrightarrow[\text{representation}]{\text{vector}}$	[1 1 1 0 0 0]
Man bites dog		[1 1 1 0 0 0]
Dog eats meat		[1 0 0 1 1 0]
Man eats food.		[0 0 1 1 0 1]

← not a good method to represent in numeric form cause you can see sent1 and sent 2 are diff but has same numeric representation.

also for a large corpus this will be a highly sparse matrix which is a disadvantage.

Let's understand how a paragraph is represented in numeric (vector) form using bag of words (BOW) technique.

Paragraph is nothing but a set of sentemes. Hence stack the numeric form of each sentence to get numeric form of paragraph

Dog bites man.
Man bites dog
Dog eats meat.
Man eats food.
Dog Dog friends.

paragraph

$$\xrightarrow[\text{representation}]{\text{numeric}} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

matrix.