What is Bag of Words (BoW) in NLP

1. Widely used technique in NLP to represent text data into vectors
2. In this model document is represented as a collection (or bag) of its word without considering grammar or word order but keeping track of frequency of each word
3. The core idea is to create a vector that counts the number of occurrences of each word in the document

1. cat eats fish
2. dog eats fish
3. dog chases cat

→ Remove stopwords

1. Create Vocabulary

[cat, eats, fish, dog, chases] → unique words

2. Create frequency vector for each sentence

Sentence 1 → "cat eats fish"

cat → 1, eats → 1, fish → 1, dog → 0, chase → 0

Vector :- [1, 1, 1, 0, 0]

Sentence 2 → dog eats fish

Vector → [0, 1, 1, 1, 0]

Sentence 3 → dog chases cat

Vector → [1, 0, 0, 1, 1]

Binary BOW

{1 and 0}

BOW

{ update based on frequency }

Method

~~X~~ ~~X~~ ~~X~~ good boy

~~X~~ ~~X~~ ~~X~~ good girl

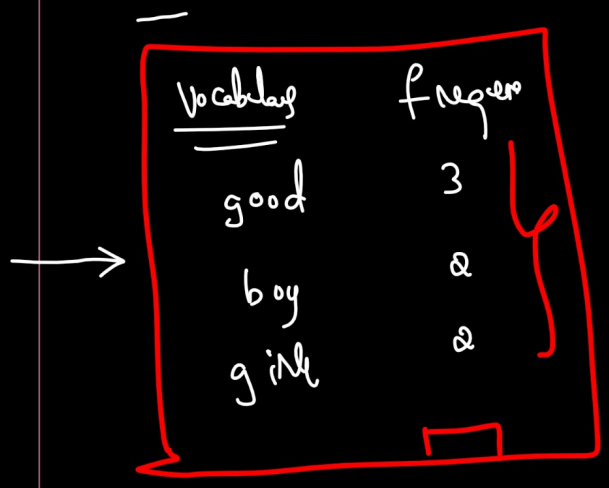
Boy ~~X~~ girl ~~X~~ good

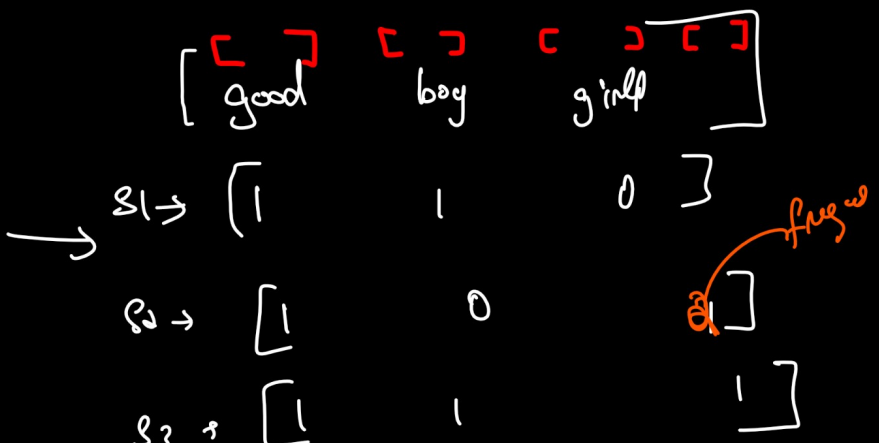
lower all sentence → ~~X~~ stopwords

S1 → good boy

S2 → good girl boy girl

S3 → Boy girl good





Advantages

1. simple and Intuitive
2. Fixed sized I/p

Disadvantages

1. sparse matrix or array
   → overfitting
2. ordering of the word is getting changed
3. Semantic meaning is still not captured
4. out of Vocabulary (OOV)

N-grams in Bag of Words

→ An n-gram is a contiguous sequence of n items (words, characters) from given sample of text

→ n grams are typically sequence of words

Unigram (n=1) : each individual word is treated as feature

Bigram (n=2) pair of consecutive words is treated as feature

Trigram (n=3) Triplets of consecutive words are treated as features

① → Cats ⊙ eats fish.

② → dog eats fish

③ → dog chase Cat

Unigram ["Cat", "eats, fish]

Bigram

["cat eats", "eats fish"]