NLP Techniques

### **Introduction**

Natural Language Processing [NLP] is a subdomain of Deep Learning which allows machines to understand and process un-organised data like human languages while also being able to use the same to generate text/voice.

### **Text Representation/Word Embeddings:**

Machines do not understand common language hence these words and their meanings need to be converted into numerical formats.

#### **Traditional/Count Based:**

**One-Hot encoding:**

* It represents each word as a vector with values 0 and 1 with a dimension equal to the size of the vocabulary. 1 will only be present at the index where the word is present in the vocabulary
* Advantages:
  1. Eliminates Ordinality i.e eliminates the risk of model being biased towards a category which has no inherent order
  2. Improves Model Performance since models can capture complex relations better while being compatible with multiple algorithms
* Limitations:
  1. Doesn’t capture Semantic Relationships
  2. Restricted to seen vocabulary
  3. High computation and Memory Requirements
* Eg]
  1. Vocabulary = [“cat”, “dog”, “cow”]
  2. “Dog” = [0,1,0]
  3. Cat = [1,0,0]

**Bag Of Words:**

* This creates a vocabulary list from all the documents present and based on the frequency of each word in that document a vector/row is assigned with the index having numbers representing the frequency of the word at that index in the vocabulary.
* Eg]
  1. Doc1: “The cat sat on a mat with another cat”
  2. Doc2: “The dog slept on the floor”
  3. Vocabulary: [‘the’, ‘cat’, ‘dog’, ‘on’, ‘a’, ‘mat’, ‘with’, ‘another’, ‘slept’, ‘floor’, ‘sat’]
  4. Bag-of-Words Matrix:

[ [1 2 0 1 1 1 1 1 0 0 1],

[2 0 1 1 0 0 0 0 1 1 0] ]

* Advantages:
  1. Simple and easy to implement
  2. Easy to interpret
* Limitations:
  1. Doesn’t Capture Semantics
  2. Computational Inefficiencies
  3. Memory Intensive

**TF-IDF:**

* Formulas:
* TF shows the frequency of a word in that document while IDF gives weights to rarer words across documents. Both combined gives a balanced term which can be used for tasks like text classification and keyword extraction
* Limitations:
  1. Inability to capture semantics
  2. Sensitive to document length

#### **Context based Words Embedding (Neural Approach):**

**Word2Vec:**

All of these algorithms use neural networks to help predict words based on the context in one way or other

* CBOW:
  1. It is a model which uses a context window around the word to be predicted to help better predict the word with the context and semantics.
  2. Example:
     + “The cat sat on the \_\_\_”
     + CBOW will use the words around the missing word to predict the word mat
  3. Strength:
     + Fast with small datasets
  4. Limitations:
     + Less effective with rare words
* Skip-gram:
  1. At a very high level skip-gram takes in a word and a window length. It then uses a neural network to predict what would be around that word based on the window provided.
  2. This is different from CBOW since here we are giving the center word and asking it to predict the words around it.
  3. Example:
     + Word = “sat”
     + Predicts → [“the”, “cat”, ”on”, “the”]
  4. Strengths:
     + Handles rare words better
  5. Limitations:
     + More training time compared to CBOW

* GloVe:
  1. GloVe creates a global co-occurrence statistic of words and has similar vectors for words which are closer to each other.
  2. Example:
     + king - man + woman ≈ queen
* Fasttext:
  1. Fasttext doesn’t treat words as a single unit rather breaks them down into smaller pieces like running → run, run, .. and hence is better at generating embeddings for words not seen before if it’s broken down versions have been present

**Bert:**

* It is based on transformers which allows the model to read all the words in a sentence all at once and assign relevance to each word
* Bert is a collection of Transformers which helps provide contextual meaning of a word in this text.