TEXT PROCESSING

(1) Tokenization: In our order to get our computer to understand any text, we need to break the word down in a way that our machine can understand. That's where tokenization comes in.

It's a fundamental step in both traditional NLP methods like Count Vectorizer and advanced deep learning-based architectures like Transformers

Broadly classified into → Word
→ Character
→ Subword (n-gram characters)

eg:- I love NLP

Assuming space as a delimiter, tokenization results in 3 tokens → I, love, NLP

Each token is a word ∴ it is an eg of word tokenization.

Now consider word → "Smarter"

Character tokens: s, m, a, r, t, e, r

Subword tokens: smart, er

(•) Tokenization is performed on the corpus to obtain tokens. The tokens are used to prepare a vocabulary. Vocabulary refers to the set of unique tokens. ~~Vocabulary refers to the set of~~

in the corpus. & Vocabulary can be constructed by considering each unique token in the corpus or by considering the top K frequently occurring words.

Need of vocabulary in traditional NLP?

→ Count vectorizer and TF-IDF use to treats each word in vocabulary as a unique feature.

Pre-trained Word embeddings such as Word2Vec and GloVe comes under word tokenization.

Drawbacks of word tokenization

(i.) Out of Vocabulary (OOV) words: New words which are encountered at testing and is doesnot exist in vocabulary. Hence, this methods fails in handling the words

8/2020 We can rescue word tokenizers from OOV words by forming the vocabulary with top K frequent words and replace the rare words in training data with unknown token (UNK). This helps the model to learn the representation of OOV words in terms of UNK tokens.

- During test time, if any word is not present in vocabulary, it will be mapped with UNK tokens.
- Problem with this approach → Entire information of the word is lost when we map OOV with UNK. And, every OOV word gets the same representation.

Most popular Subword tokenization algorithm :-

Byte Pair Encoding (BPE)

↳ Widely use tokenization methods among transformer based-model

↳ BPE addresses the issues of word and character tokenizat

BPE tackles OOV by ~~representing~~ segmenting OOV as subwords and represents the words in terms of these subwords.

It is a word segmentation algorithm that merges most frequently occuring character or character sequences iteratively