**AIM:** Exploratory analysis on Twitter text data

**SOFTWARE/TOOLS USED:**

* Python
* Nltk
* Jupyter

**THEORY:**

Text Pre-processing -

To prepare the text data for the model building we perform text pre-processing. It is the very first step of NLP projects. Some of the pre-processing steps are:

* Removing punctuations like . , ! $( ) \* % @
* Removing URLs
* Removing Stop words
* Lower casing
* Tokenization
* Stemming
* Lemmatization

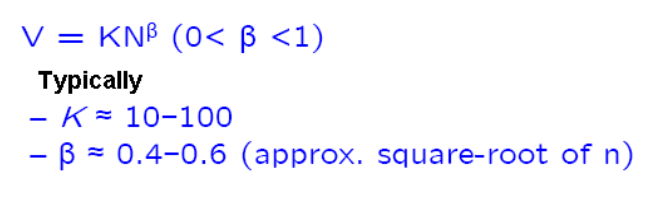
We need to use the required steps based on our dataset.

**Zips and Heaps law**

Zipf's law is a law about the frequency distribution of words in a language (or in a collection that is large enough so that it is representative of the language). To illustrate Zipf's law let us suppose we have a collection and let there be V unique words in the collection (the vocabulary).

For each word in the collection, we need to compute the freq(word) = how many times word occurs in the collection. Then we rank the words in descending by their frequency (most frequent word has rank 1, next frequent word has rank 2, ...)

Heap's law states that the number of unique words V in a collection with N words is approximately Sqrt[N].



The more general form of this law is Alpha and beta and usually found by fitting the data.

**Identifying topics**

Topic analysis (also called topic detection, topic modelling, or topic extraction) is a machine learning technique that organizes and understands large collections of text data, by assigning “tags” or categories according to each individual text’s topic or theme.

Topic analysis uses natural language processing (NLP) to break down human language so that you can find patterns and unlock semantic structures within texts to extract insights and help make data-driven decisions. The two most common approaches for topic analysis with machine learning are NLP topic modelling and NLP topic classification. Topic modelling is an unsupervised machine learning technique. This means it can infer patterns and cluster similar expressions without needing to define topic tags or train data beforehand. This type of algorithm can be applied quickly and easily, but there’s a downside – they are rather inaccurate.

Text classification or topic extraction from text, on the other hand, needs to know the topics of a text before starting the analysis, because you need to tag data in order to train a topic classifier. Although there’s an extra step involved, topic classifiers pay off in the long run, and they’re much more precise than clustering techniques**.**

**CONCLUSION:** Text pre-processing has multiple steps associated with it, available libraries and laid out techniques like zefs and heafs laws are helpful in this context. Tools like topic analysis help us get more information from text at a glance.

**RESULT:** Text pre-processing was done on Twitter text data set in python.