Text preprocessing techniques- Twitter Data

Text files contain enormous amounts of information. Language data analysis is the most difficult task for a computer to perform since a computer cannot understand the semantics of text. In order to accomplish this, we convert text data into a machine-readable format.
Data in text format is converted to numerical values (or vectors) by text processing, so that these vectors may be given to the machine as input and analysed with the algebraic principles. However, there's a chance of data loss if we go through with the transition. The idea is to strike a balance between data conversion and data retention.

Preprocessing of text is necessary before processing the text. Text processing techniques will be discussed in this article. Before we begin, it's important to define a few concepts.

- A Document is the name given to each piece of text data.
- Corpus is the name given to the collection of documents as a whole.

The methods listed below can be used for text processing:

- Bag of Words
- TF-IDF
- Word2Vec

Let's get right in and find out everything there is to know about each one.

1. Bag of Words

Bag of Terms uses a dictionary of unique words to perform a simple vectorization of the document. There are only two phases involved in this process.

Step 1: Construction of Dictionary

To do this, create a vector dictionary containing all of the data's unique words. There should be d unique words in the corpus. As a result, each word has its own dimension, making this a d-dimensional dictionary vector.

Step 2: Vector construction

We establish a vector, say, vi, for every document, say, ri. There are two approaches to build a vi with d-dimensions now:
If the vi is built according to the dictionary, then each word is copied exactly how many times it appears in a certain document.

Each document's vi is built using the dictionary as a guide, so that every word in the dictionary appears in the document.

- 1 if the word is present in the document or
- If the word doesn't appear in the document, the result will be 0.

Binary Bag of Words is the term used to describe this kind of document.

A dictionary containing the unique terms from the data corpus has been created with vectors for each page. These vectors can be examined by performing an analysis

- plotting in d-dimension space or
- calculating distance between vectors to get the similarity (the closer the vectors are, more similar they are)

2. Term Frequency — Inverse Document Frequency

Word, document, and corpus are all present in this instance. If you want to convert text into vectors, you can utilise Term Frequency—Inverse Document Frequency (TF-IDF).

The association between a term and a document is described by Term Frequency. Inverse Document Frequency, on the other hand, discusses the connection between a word and its context.

Step 1: Calculating Term Frequency

The probability that the word wj will appear in the text ri is known as term frequency. And it's calculated in the following way:

$$Term\ Frequency = \frac{Number\ of\ times\ a\ word\ is\ present\ in\ the\ review}{Total\ number\ of\ words\ in\ that\ review}$$

If a word's Term Frequency is high in a review, it indicates that the word is frequently used in that review. If a word's Term Frequency is low in a review, it means the word is uncommon there.

Step 2: Calculating IDF

According to Inverse Document Frequency, a word appears in the corpus more often than it does in the original document. And it's calculated in the following way:

$$Inverse\ Document\ Frequency = Log(\frac{Total\ number\ of\ documents\ in\ the\ corpus}{Number\ of\ documents\ which\ contain\ that\ word})$$

If the Inverse Document Frequency is low, it indicates that the word is not frequently encountered in the corpus.

If the Inverse Document Frequency is high, it indicates that the word is relatively uncommon in the corpus.

In this case, scaling is the primary rationale for using log instead of the simple inverse ratio. There is a probability range of 0 to 1 for the term "Frequency." The total TF-IDF number will be skewed towards IDF if we merely take the inverse ratio. Logging in IDF is commonly used for a number of straightforward and well acknowledged reasons.

TF-IDF of a word in review is TF(word, review) *IDF(word, document corpus).

We now have the TF-IDF for the word in the vector form for each document. TF-IDF vectorization refers to employing TF-IDF values to turn a document into a vector.

When using TF-IDF vectorization, words that are

- frequent in a document (from TF)
- rare in the corpus (from IDF)

3. Word2Vec

Sentences are turned into vectors in Bag of Words and TF-IDF. Word2Vec, on the other hand, turns a text file into a graphical representation of the text. As a result, it's called word2vec!

As an input, Word2Vec uses a large corpus of text and outputs a vector space with hundreds of dimensions, with each unique word in the corpus given its own vector in the space. There is a close relationship between words in the corpus because of the way vectors are positioned in vector space.

The best time to utilise each method?

- This is a difficult issue to answer because it is so context-dependent.
- Document classification applications frequently employ the Bag of Words as a feature to train a classifier on the frequency of each word.
- Many search engines, including Google, utilise the TF-IDF score to determine how relevant a piece of content is.
- Word2vec comes into play when an app has to grasp the context of words, discover word similarity, or translate provided texts into another language, all of which need a significant amount of document information.