<https://towardsdatascience.com/text-classification-in-python-dd95d264c802>

5.1. Text representation

Recall that, in order to represent our text, every row of the dataset will be a single document of the corpus. The columns (features) will be different depending of which feature creation method we choose:

With **Word count vectors**, every column is a term from the corpus, and every cell represents the frequency count of each term in each document.

***TF-IDF*** is a score that represents the relative importance of a term in the document and the entire corpus. *TF*stands for *Term Frequency*, and *IDF* stands for *Inverse Document Frequency*:

The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general**.**

It also takes into account the fact that some documents may be larger than others by normalizing the TF term (expressing instead relative term frequencies).

These two methods (Word Count Vectors and TF-IDF Vectors) are often named Bag of Words methods, since the order of the words in a sentence is ignored. The following methods are more advanced as they somehow preserve the order of the words and their lexical considerations.

* **Word Embeddings**

The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used. Word embeddings can be used with pre-trained models applying transfer learning.