**Sentiment Classification**

Assignment Report

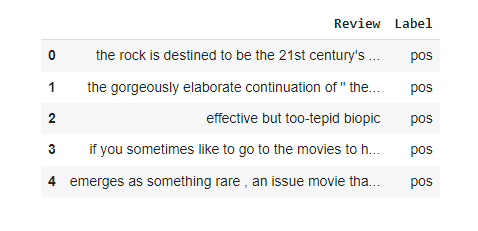
1 - Executive Summary

The aim of this assignment aims to provide a glimpse of the process of text preprocessing and the procedure of building a fully connected feedforward neural network to understand the text. I will also build another conventional supervised learning model for performance comparison. To obtain a relatively optimal model of neural networks, and tune hyperparameters associated with the network

2 - Model Building

2.1 - Data Description

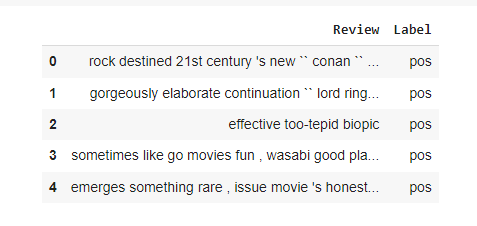
Training data contains total of 10662 Review and 2 Labels which is Pos and another is Neg.

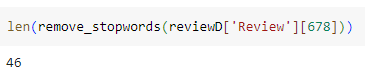


(10662, 2)

### 2.2 – Tokenization and stopwords removal

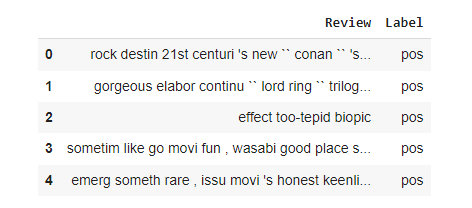
* First I Install the tokenizer this will impact on the overall model-building process.
* Then I Load the stop words from the nltk stopwords corpus and store it in a list *stopword\_list*
* This function will take a piece of text, tokenizes it, and remove all the stopwords using the *stopword\_list* Input a piece of text in (str) format and the output the same piece of text, tokenized and with the stopwords removed.
* And then I apply the function on a few rows from the data set and observe the result

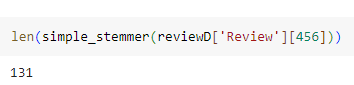




### 2.3 - Stemming

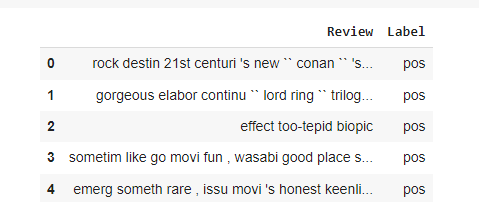
* In this first I Install the stemmer that I was going to use was PorterStemmer from nltk.porter module
* Then I create a function simple\_stemmer that takes a given piece of text and outputs the stemmed version of the individual words using the stemmer initialized earlier input a piece of text (str) output stemmed version of the text

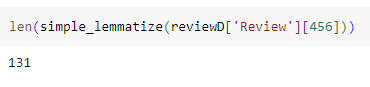




### 2.4 - Lemmatization

* In this first I Install the Lemmatized that I was going to use was 'WordNetLemmatizer'
* Then I create a function simple\_lemmatize that takes a given piece of text and outputs the lemmatized version of the individual words using the lemmatizer that we initialized earlier input a piece of text in the form of (str) output lemmatized version of the text





### 2.5 - Tf-Idf Data Preparation

* I use the TfidfVectorizer object to create an n-gram model for both the train and validation reviews
* For this data, I use max\_features as 500, though it can tweak the parameters to create your final data set and in n-gram\_range I set it to (1,2)
* After the TfidfVectorizer method has been initialized, use I fit and transform the train data. Use the same tf-idf model to transform the validation data as well.
* Then I check the final shape of train and validation set

(8000, 500)

(2000, 500)

### - Label Encoding sentiments

* In this dataset the sentiments are encoded as 'pos' and 'neg' so I use a label encoder to convert these into 1 and 0 respectively

