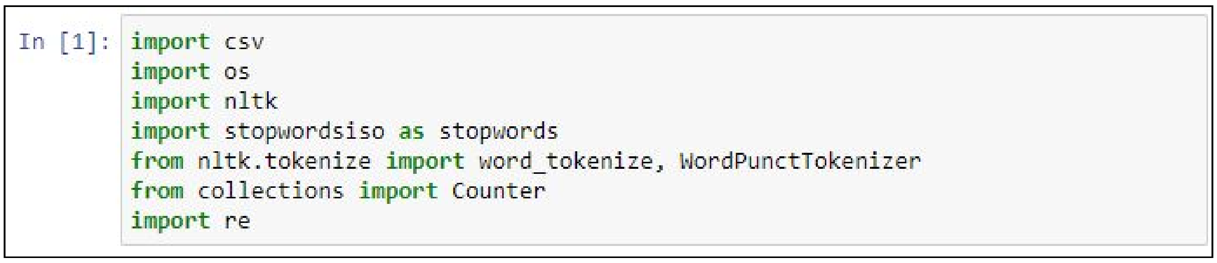
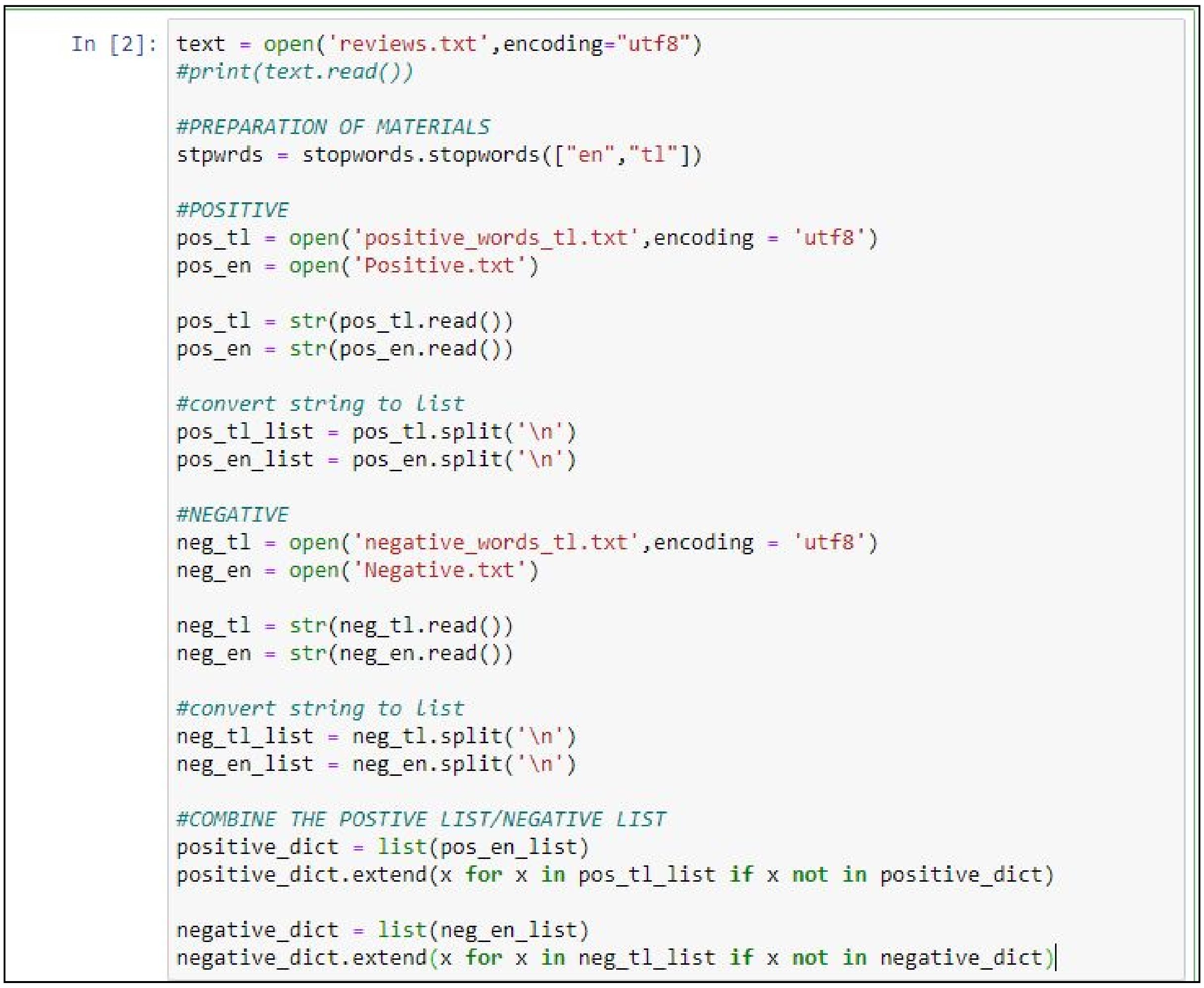
Sentiment Analysis in Filipino corpora using filipino soptwords

Importing the modules needed for the algorithm.Stopwordiso is a collection of stopwords for multiple languages. We use this to use the stopwords for tagalog language.



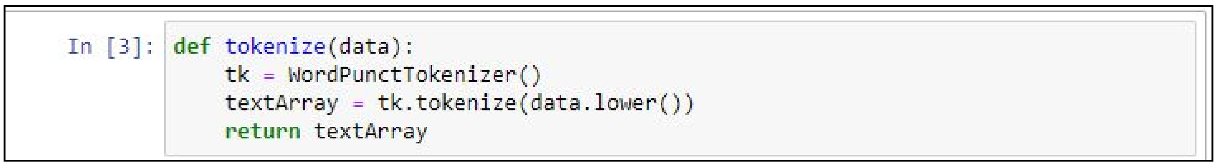
```
In [1]: import csv
        import os
        import nltk
        import stopwordsiso as stopwords
        from nltk.tokenize import word_tokenize, WordPunctTokenizer
        from collections import Counter
        import re
```

Preparing the materials that we need like list of english, tagalog stop words and dictionary for sentiment lexicon, the positive and negative list of words. This is already mixed of english and tagalog words.



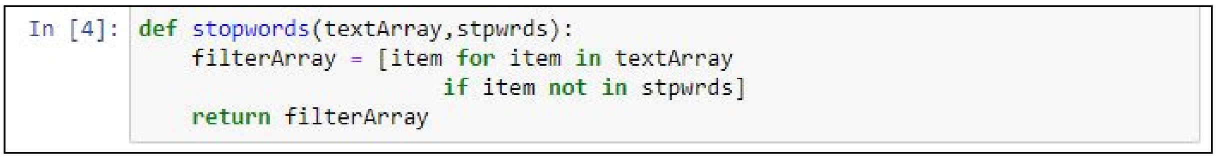
```
In [2]: text = open('reviews.txt',encoding="utf8")
        #print(text.read())

        #PREPARATION OF MATERIALS
        stpwrds = stopwords.stopwords(["en","tl"])

        #POSITIVE
        pos_tl = open('positive_words_tl.txt',encoding = 'utf8')
        pos_en = open('Positive.txt')

        pos_tl = str(pos_tl.read())
        pos_en = str(pos_en.read())

        #convert string to list
        pos_tl_list = pos_tl.split('\n')
        pos_en_list = pos_en.split('\n')

        #NEGATIVE
        neg_tl = open('negative_words_tl.txt',encoding = 'utf8')
        neg_en = open('Negative.txt')

        neg_tl = str(neg_tl.read())
        neg_en = str(neg_en.read())

        #convert string to list
        neg_tl_list = neg_tl.split('\n')
        neg_en_list = neg_en.split('\n')

        #COMBINE THE POSTIVE LIST/NEGATIVE LIST
        positive_dict = list(pos_en_list)
        positive_dict.extend(x for x in pos_tl_list if x not in positive_dict)

        negative_dict = list(neg_en_list)
        negative_dict.extend(x for x in neg_tl_list if x not in negative_dict)
```

Method for tokenization



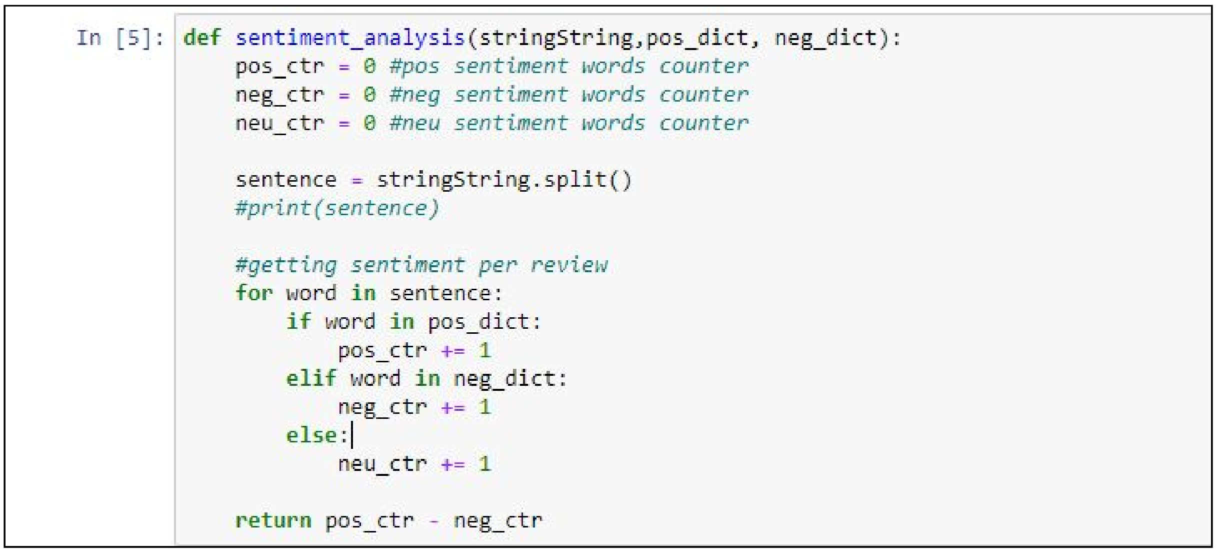
```
In [3]: def tokenize(data):
            tk = WordPunctTokenizer()
            textArray = tk.tokenize(data.lower())
            return textArray
```

Method for stop words removal



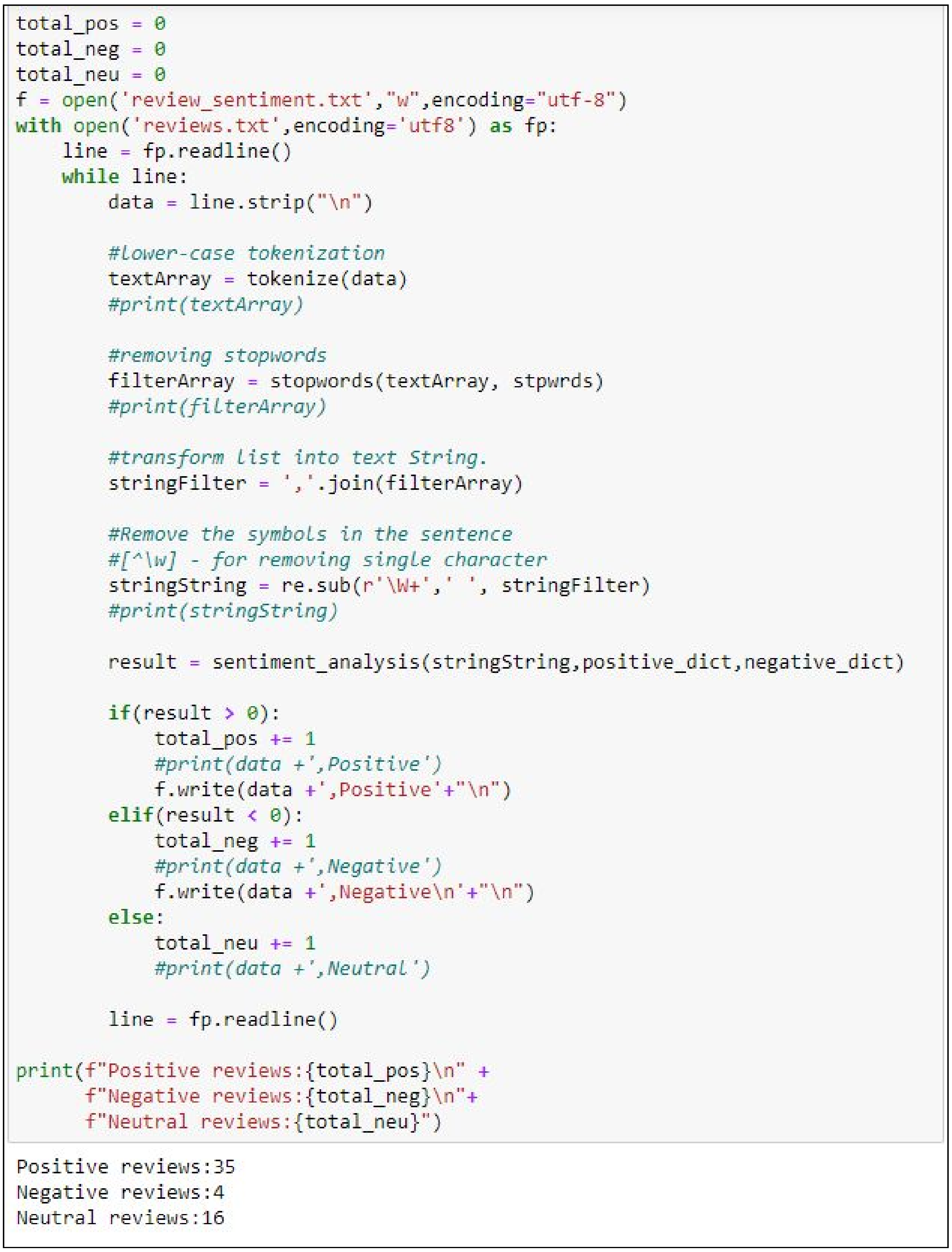
```
In [4]: def stopwords(textArray,stpwrds):
            filterArray = [item for item in textArray
                                if item not in stpwrds]
            return filterArray
```

Method for sentiment analysis, here it is simply a dictionary-based sentiment analysis where we read the dataset by reviews and identify its sentiment.



```
In [5]: def sentiment_analysis(stringString,pos_dict, neg_dict):
            pos_ctr = 0 #pos sentiment words counter
            neg_ctr = 0 #neg sentiment words counter
            neu_ctr = 0 #neu sentiment words counter

            sentence = stringString.split()
            #print(sentence)

            #getting sentiment per review
            for word in sentence:
                if word in pos_dict:
                    pos_ctr += 1
                elif word in neg_dict:
                    neg_ctr += 1
                else:
                    neu_ctr += 1

            return pos_ctr - neg_ctr
```

The process of whole sentiment analysis



```python
total_pos = 0
total_neg = 0
total_neu = 0
f = open('review_sentiment.txt',"w",encoding="utf-8")
with open('reviews.txt',encoding='utf8') as fp:
    line = fp.readline()
    while line:
        data = line.strip("\n")

        #lower-case tokenization
        textArray = tokenize(data)
        #print(textArray)

        #removing stopwords
        filterArray = stopwords(textArray, stpwrds)
        #print(filterArray)

        #transform list into text String.
        stringFilter = ','.join(filterArray)

        #Remove the symbols in the sentence
        #[^\w] - for removing single character
        stringString = re.sub(r'\W+',' ', stringFilter)
        #print(stringString)

        result = sentiment_analysis(stringString,positive_dict,negative_dict)

        if(result > 0):
            total_pos += 1
            #print(data +',Positive')
            f.write(data +',Positive'+"\n")
        elif(result < 0):
            total_neg += 1
            #print(data +',Negative')
            f.write(data +',Negative\n'+"\n")
        else:
            total_neu += 1
            #print(data +',Neutral')

        line = fp.readline()

print(f"Positive reviews:{total_pos}\n" +
      f"Negative reviews:{total_neg}\n"+
      f"Neutral reviews:{total_neu}")
```

```
Positive reviews:35
Negative reviews:4
Neutral reviews:16
```