Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 6

Aim: Implementation of Sentiment Analysis

Objective:- To understand the use of various sentiment Analysis techniques by implementing them

Description:

Sentiment Analysis: Sentiment Analysis is a text analysis technique that allows companies to make sense of qualitative data. By detecting positive and negative sentiment in text data, such as tweets, product reviews, and support tickets, you can understand how customers feel about your brand, product, or service, and gain insights that lead to data-driven decisions

Sentiment Analysis deals with analyzing emotions and the perspective of a speaker or an author from a given piece of text. "Sentiment analysis or opinion mining refers to the appliance of language process, linguistics, and text analyticsto spot and extract subjective information in supply
materials". This field of technology deals with analyzing and predicting the hidden information keep within the text. This hidden information gives valuable insights regarding user's intentions, style and odds. Sentiment Analysis specializes in categorizing the text at the extent of subjective and objective nature. Judgement indicates that the text bears opinion content where's perspicacity indicates that the text is while not opinion content

Some examples-
1. Subjective- This motion picture by tom cruise and Angelina jolie is great. (This sentence has an opinion, it talks regarding the motion picture and also the writer's emotions regarding same "great" and thence its subjective

2. Objective- This motion picture stars tom cruise and Angelina.
(This sentence may be a reality, general information instead of an opinion or a read of some individual and thence its objective)
The subjective text may be additional categorized into three broad classes supported the emotions expressed within the text.

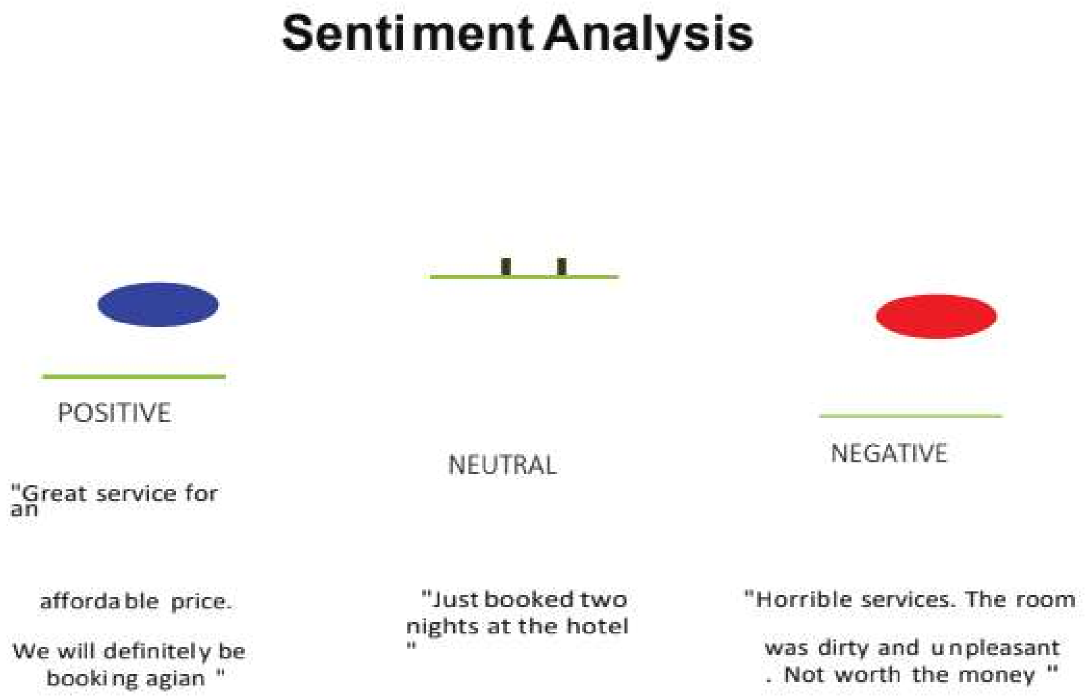
1. Positive- I like to look at Star series.

2. Negative- The movie was awful.

3.Neutral- I typically get hungry by evening. (This sentence has users views, emotions hence it's

subjective however because it doesn't have any positive or negative polarity therefore it's neutral).

Example:



"Great service for an

afforda ble price.

We will definitely be booki ng agian "

"Just booked two nights at the hotel "

"Horrible services. The room

was dirty and u npleasant . Not worth the money "

Method1:UsingPositiveandNegativeWordCount-WithNormalization forCalculating Sentiment Score
In this method, we will calculate the Sentiment Scores by classifying and counting the Negative and Positive words from the given text and taking the ratio of the difference of Positive and Negative Word Counts and Total Word Count. We will be using the Amazon Cell Phone Reviews
dataset from Ka le.

Code:

```
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize from nltk.stem import WordNetLemmatizer import re
df = pd.read_csv('20191226-reviews.csv' , usecols=['body']) lemma = WordNetLemmatizer()

stop_words = stopwords.words('english') def text_prep(x: str) -> list:
corp = str(x).lower()
corp = re.sub('[/\a-zA-Z]+',' ', corp).strip() tokens = word_tokenize( corp)
words = [t for t in tokens if t not in stop_words] lemmatize = [lemma.lemmatize(w) for w in words] return lemmatize
preprocess_tag = [text_prep(i) for i in dfl'body']] dfl"preprocess_txt"]  = preprocess_tag
file = open('negative-words.txt' , 'r') neg_words = file.read().split()
file = open('positive-words.txt' , 'r') pos_words = file.read().split()
num_pos = dfl'preprocess_txt'].map(lambda  x: len([i for i in x if i in pos_words]))
dfl'pos_count'] = num_pos
num_neg = dfl'preprocess_txt'].map(lambda  x: len([i for i in x if i in neg_words]))
dfl'neg_count'] = num_neg
```

Next, we will apply the formula and add the formulated value into the main DataFrame.

```
dfl'sentiment'] = round(dfl'pos_count'] I (dfl'neg_count']+ 1), 2)
```

Putting it all together.

Method 2: Using Positive and Negative Word Counts-With Semi Normalization to calculate Sentiment Score

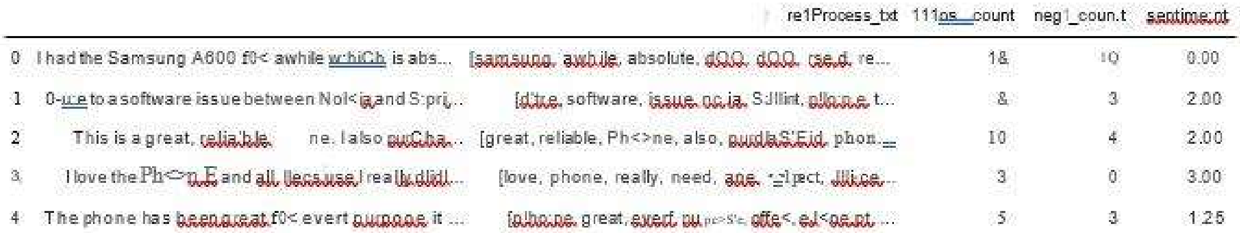
In this method, we calculate the sentiment score by evaluating the ratio of Count of Positive Words and Count of Negative Words + 1. Since there is no difference of values involved, the sentiment value will always be more than 0. Also, adding 1 in the denominator would save from Zero Division Error. Let's start with the implementation. The implementation code will remain the same till the Negative and Positive Word Count from the previous part with a difference that this time we don't need the total word count, thus will be omitting that part.

Code:

```
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize from nltk.stem import WordNetLemmatizer import re
df = pd.read_csv('20191226-reviews.csv' , usecols=['body']) lemma = WordNetLemmatizer()
stop_words = stopwords.words('english') def text_prep(x: str) -> list:
corp = str(x).lower()
corp = re.sub('[/\a-zA-Z]+',' ', corp).strip() tokens = word_tokenize( corp)
words = [t for t in tokens if t not in stop_words] lemmatize = [lemma.lemmatize(w) for w in
words] return lemmatize
preprocess_tag = [text_prep(i) for i in dfl'body']] dfl"preprocess_txt"]  = preprocess_tag

file = open('negative-words.txt' , 'r') neg_words = file.read().split()
file = open('positive-words.txt' , 'r') pos_words = file.read().split()
num_pos = dfl'preprocess_txt'].map(lambda  x: len([i for i in x if i in pos_words]))
dfl'pos_count'] = num_pos
num_neg = dfl'preprocess_txt'].map(lambda  x: len([i for i in x if i in neg_words]))
dfl'neg_count'] = num_neg
dfl'sentiment'] = round(dfl'pos_count'] I (dfl'neg_count']+ 1), 2) df.head()
```

Output:



```
                                                                           re1Process_txt  111os__count  neg1_coun.t  sentime:nt
0  I had the Samsung A600 f0< awhile w:hiCh is abs...  [samsung, awhile, absolute, dOO, dOO, rse.d, re...   1&     IQ   0.00
1  0-u:e to a software issue between Nol<ia and S:pri,..  [d:t:e, software, issue, nc.ia, S:lllint, p!ho:n.e, t...   &    3   2.00
2  This is a great, relia'ble     ne. I also puCha...  [great, reliable, Ph<>ne, also, purdlaS'Eid, phon...   10    4   2.00
3  I love the Ph<>n.E and all, llecsuse I really dlidl...  [love, phone, really, need, ane, '-:lpct, Jlli:ce...   3    0   3.00
4  The phone has been great f0< evert purpose it ...  [p!ho:ne, great, everf, pu...se, offe<, eJ<oe.nt, ...   5    3   1.25
```

Conclusion-

Techniques used for sentimental analysis are:

Sentiment analysis techniques involve analyzing text data to determine the sentiment expressed, typically using methods like:
1. Bag-of-Words (BoW): Counts occurrences of words to represent text as numerical features.
2. TF-IDF (Term Frequency-Inverse Document Frequency): Weighs the importance of words based on their frequency in a document relative to the entire corpus.
3. Machine Learning Models: Utilize algorithms like Support Vector Machines (SVM), Naive Bayes, or Recurrent Neural Networks (RNNs) for sentiment classification.
4. Lexicon-based Approaches: Assign sentiment scores to words and calculate an overall sentiment score for the text based on the words it contains.
5. Deep Learning: Techniques like Word Embeddings (e.g., Word2Vec, GloVe) and Transformers (e.g., BERT) capture semantic relationships for more nuanced sentiment analysis.