

A Mini Project Report
On
**IMDB Movie Review Sentiment Analysis
using SentiWordNet**

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Natural Language Processing
In
Computer Engineering (VIII SEM)

Submitted By
Adarkar Amol Mangesh
Bharambe Rishikesh Vijay
Gladson Daniel Roy
Sharma Smriti Sushil Kumar

Subject Incharge
Prof. Dhiraj Amin

Lab Incharge
Prof. Dhiraj Amin



Department Of Computer Engineering
PILLAI COLLEGE OF ENGINEERING
New Panvel – 410 206
UNIVERSITY OF MUMBAI
Academic Year 2020 – 21



Department of Computer Engineering
Pillai College of Engineering
New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the project report entitled '**IMDB Movie Review Sentiment Analysis using SentiWordNet**' have been successfully completed by the following students:

Name	Roll No.
Adakar Amol Mangesh	A801
Bharambe Rishikesh Vijay	A811
Gladson Daniel Roy	A823
Sharma Smriti Sushil Kumar	A871

in partial fulfillment of the course Natural Language Processing in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2020 – 21.

(Prof. Dhiraj Amin)
Subject Incharge



Department of Computer Engineering
Pillai College of Engineering
New Panvel – 410 206

PROJECT APPROVAL

This project entitled “IMDB Movie Review Sentiment Analysis using SentiWordNet” by Adakar Amol Mangesh, Bharambe Rishikesh Vijay, Gladson Daniel Roy, Sharma Smriti Sushil Kumar are approved for the course Natural Language Processing in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Examiners:

1. _____

2. _____

Subject Incharge:

1. _____

Date:13/05/2021

Place: New Pavel



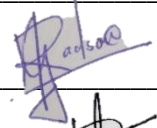
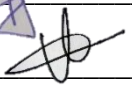


Department of Computer Engineering
Pillai College of Engineering
New Panvel – 410 206

DECLARATION

We declare that this written submission for Natural Language Processing mini project entitled “IMDB Movie Review Sentiment Analysis using SentiWordNet” represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group Members:

Adakar Amol Mangesh	:	
Bharambe Rishikesh Vijay	:	
Gladson Daniel Roy	:	
Sharma Smriti Sushil Kumar	:	

Date: 13/05/2021

Place: New Panvel

Table of Contents

Abstract.....		i
List of Figures.....		ii
List of Tables.....		iii
1 .	Introduction.....	1
	1.1 Fundamentals.....	1
	1.2 Objectives.....	2
	1.3 Scope.....	2
	1.4 Organization of the Project Report.....	2
2 .	Literature Survey.....	3
	2.1 Introduction.....	3
	2.2 Literature Review	3
	2.3 Summary of Literature Survey.....	5
3 .	Project Implementation.....	6
	3.1 Overview.....	6
	3.1.1 Existing Systems.....	7

		3.1.2	Proposed System.....	7
	3. 2		Implementation Details.....	9
		3.2.1	Methodology	9
		3.2.2	Details of packages, data set	12
4			Project Inputs and Outputs.....	16
	4. 1		Input Details Outputs/Screenshots.....	16
	4. 2		Evaluation Parameters Details.....	16
	4. 3		Output Details and Screenshots	16
5 .			Summary and Future Scope.....	19
	5. 1		Summary.....	19
	5. 2		Future Scope.....	19
			References.....	20
			Acknowledgement.....	21

Abstract

Sentiment analysis or opinion mining, is the process to analyze textual documents and predict their sentiment or opinion based on the content of the document. Sentiment analysis is perhaps one of the most popular applications of natural language processing. The aim of the project is to create a sentiment analyzer for determining the sentiment value of movie reviews. In this project, the sentiment which the opinions are delivering are predicted using a standard dataset from IMDB containing movie reviews which is taken through different steps of preprocessing which involve cleaning the text & then normalizing the reviews. Sentiment analysis is done using the SentiWordNet algorithm & based on that the sentiment polarity, whether positive or negative is predicted.

List of Figures

Fig 1.1	Example of Sentiment Analysis	1
Fig 3.1	Different Approaches of Sentiment Analysis	6
Fig 3.2	Sentiment Analyzer	7
Fig 4.1	Loading Data	15
Fig 4.2	Cleaning the Text - Expanding Contractions	15
Fig 4.3	Sentiment Analysis with SentiWordNet - Likelihood of a Word to make the Review Positive or Negative	16
Fig 4.4	Sentiment Analysis with SentiWordNet - Sample Review	16
Fig 4.5	Sentiment Analysis with SentiWordNet - Normalization of the Review	16
Fig 4.6	Sentiment Analysis with SentiWordNet - Sentiment Prediction for Reviews	17
Fig 4.7	Predict Sentiment for Complete Dataset	17
Fig 4.8	Evaluate Model Performance	17
Fig 4.9	Model Classification Report	18
Fig 4.10	Prediction Confusion Matrix	18

List of Tables

Table 2.1	Literature survey summary	5
Table 3.1	Dataset specification	12
Table 3.2	Hardware details	12
Table 3.3	Software details	12

Chapter 1

Introduction

1.1 Fundamentals

The Internet movie database (IMDb) is the most popular comprehensive database having information about movies, actors and production. Furthermore it allows users to post comments and provides the aggregated ratings about the movie. Given that massive amount of data and degree of interactions between the users, IMDB can be used as a good source for Data mining use cases. Sentiment analysis is a major topic in Natural Language processing (NLP) which aims at extracting insights from the textual reviews. It also helps in identifying the polarity of the textual content or mind-set of the reviewer with respect to multiple topics. Using sentiment analysis, we can find the reviewer's state of mind, while reviewing and understanding whether the person was “happy,” “sad,” “angry,” etc.



Fig 1.1: Example of Sentiment Analysis

Sentiment analysis is also popularly known as opinion analysis or opinion mining. The key idea is to use techniques from text analytics, NLP, Machine Learning, and linguistics to extract important information or data points from unstructured text. This in turn can help us derive qualitative outputs like the overall sentiment being on a positive, neutral, or negative scale and quantitative outputs like the sentiment polarity, subjectivity, and objectivity proportions.

1.2 Objectives

1. To study the existing solutions, their methodologies and techniques.
2. Sentiment Analysis to determine the attitude of the mass is positive, negative or neutral towards the subject of interest.
3. To extract from the opinions, the appraisals & emotions of people in regards to entities, events & attributes.

1.3 Scope

The SentiWordNet Model is designed to help in recognition of different sentiments of movie reviews. This system performs the various pre-processing and normalization techniques on unstructured IMDB movie reviews in order to get clean text and segregate the textual data based on the sentiment of the movie reviews.

1.4 Organization of the Report

Chapter 1: This chapter provides the background information needed in order to understand the task at hand.

Chapter 2: This chapter presents the literature review for the previous work done in the field of sentiment analysis. It describes the pros and cons of each technique.

Chapter 3: This chapter presents the Theory and proposed work. It also explains the pre-processing techniques that are used before feeding the data into the wordnet for sentiment analysis.

Chapter 4: This chapter provides the details of experimental setup that are used for the implementation and also describes the experimental results we have achieved and the discussion of the results.

Chapter 5: This chapter concludes the thesis by providing the summary of the work done in the thesis. Also, outline the potential work that could be done to our research in the future to achieve better desirable results.

Chapter 2

Literature Survey

2.1 Introduction

Users express their opinions about their experience with the applications. These opinions are valuable to enhance their User Experience. Opinions are being expressed in the form of reviews & provide an opportunity for new explorations to find collective likes and dislikes of the cyber community. The domain of movie reviews affects everyone from audience, film critics to the production company. The movie reviews being posted on the websites are not formal reviews but are rather very informal and are an unstructured form of grammar. Opinions expressed in movie reviews give a very true reflection of the emotion that is being conveyed. The presence of such a great use of sentiment words to express the review inspired us to devise an approach to classify the polarity of the movie using these sentiment words.

2.2 Literature Review

Corpus-Based Approaches :- The corpus-based approach (hereafter CBA) is a method that uses an underlying corpus as an inventory of language data. It is a method where the corpus is interrogated and data is used to confirm linguistic pre-set explanations and assumptions.

Dictionary-Based Approaches :- Dictionary-based sentiment analysis is a computational approach to measuring the feeling that a text conveys to the reader. This method relies heavily on a predefined list (or dictionary) of sentiment-laden words.

A pre-trained model called SentiLARE for sentiment analysis is created by authors, which introduces linguistic knowledge from SentiWordNet via context-aware sentiment attention, and adopts label-aware masked language model to deeply integrate knowledge into BERT-style models through pre-training tasks [1]. Experiments show that SentiLARE outperforms state-of-the-art language representation models on various sentiment analysis tasks, and thus facilitates sentiment understanding.

To identify the polarity of the tweets, various techniques were used [2]. The algorithms performed were Naïve Bayes, K-Nearest Neighbour, Random Forest. The best results were given by Naïve Bayes classifier. Finding the polarity of the reviews can help in various domains. Intelligent systems can be developed which can provide the users with comprehensive reviews of movies, products, services etc. without requiring the user to go through individual reviews, he can directly take decisions based on the results provided by the intelligent systems.

A fuzzy logic-based technique is applied to online reviews to compute the fuzzy sentiment score [3]. Two sentiment lexicons- SentiWordNet and AFINN are used to compute the sentiment score of words. The key highlights are: i) proposed an unsupervised approach based on fuzzy logic for sentiment analysis of textual reviews, ii) the proposed model uses fuzzy cardinality as the measure for the evaluation of word polarity scores, iii) it has two versions based on the sentiment lexicon deployed in the model, iv) the author's fuzzy cardinality approach is compared to non-fuzzy state-of-the-art methods.

Sentiment analysis is done on IMDB movie review database [4]. Here, sentiment expressions are framed to classify the polarity of the movie review on a scale of 0(highly disliked) to 4(highly liked) and perform feature extraction and ranking and use these features to train a multi-label classifier to classify the movie review into its correct label. The extracted new features had a strong impact on determining the polarity of the movie reviews. The authors also applied computation linguistic methods for the preprocessing of the data [4]. Feature impact analysis was performed by computing information gain for each feature in the feature set and used it to derive a reduced feature set. Among six classification techniques, it was found that the highest accuracy was given by Random Forest with an accuracy of 88.95%.

Unarguably, sentimental analysis techniques are among the utmost significant bases in the decision-making process [5]. A lot of people depend on sentimental analysis for achieving efficient results of services or products. The authors initially worked on a base model that was decent in producing IMDB movie reviews. So, the idea of applying a pre-trained language model to actually outperformed the cutting-edge research in academia as well. It is an undeniable fact that human languages are relatively complex to be understood by the machine, which leads to conditions

where a negatively said word has a positive association and vice versa. So, a sentimental analysis of movie reviews was a challenging task. In this study, neural networks were used, which lead the authors to achieve the task of opinion mining from movie reviews, which was trained on “Movie Review Database by Stanford University” in concurrence with two immense lists of negative and positive words. The trained system accomplished to accomplish an ultimately exceptional final precision. For the movie reviews, a neural network model has been made using the artificial neural network having six layers (four hidden, one input, and one output) having one neuron for binary classification (positive or negative review). The training accuracy of this model has reached 91.9% and validation accuracy which is 86.67%

2.3 Literature Summary

Table 2.1: Literature survey summary

SN	Techniques	Datasets	Accuracy	Author & Year of Publication
1.	SentiWordnet, SentiLARE	Movie Reviews (MR)	90.07%	Ke, Pei, et al. (2020)
2.	Random Forest	IMDB	78.65%	Baid, Palake et al, (2017)
3.	Fuzzy Cardinality SentiWordNet Approach	IMDB	64.13%	S. Vashishtha et al, (2020)
4.	SentiWordnet, Random Forest	IMDB	88.95%	T. P. Sahu et al (2016)
5.	Artificial Neural Network (with 6 layers)	IMDB	91.9%	Shaukat, Z. et al, (2020)

Chapter 3

Implementation Details

3.1 Overview

How to classify Sentiment?

Machine Learning: This approach employs a machine-learning technique and diverse features to construct a classifier that can identify text that expresses sentiment. Nowadays, deep-learning methods are popular because they fit on data learning representations.

Lexicon-Based: This method uses a variety of words annotated by polarity score, to decide the general assessment score of a given content. The strongest asset of this technique is that it does not require any training data, while its weakest point is that a large number of words and expressions are not included in sentiment lexicons.

Hybrid: The combination of machine learning and lexicon-based approaches to address Sentiment Analysis is called Hybrid. Though not commonly used, this method usually produces more promising results than the approaches mentioned above.

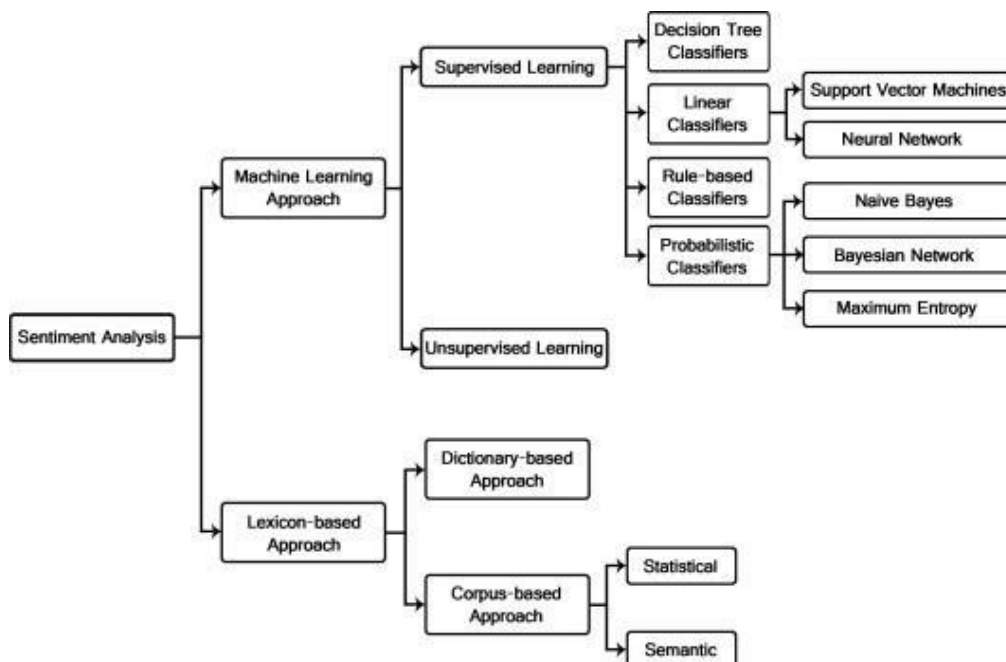


Fig 3.1: Different Approaches of Sentiment Analysis

3.1.1 Existing Methodology and Systems

There have been many researches in the field of Sentiment Analysis for Movie Reviews using SentiWordnet and by using various Machine Learning and Rule-based techniques to provide better results. The polarity extraction of conveyed opinions has been majorly done using the POS tagging, chunking, simple negation forms. Feature Extraction is done using different approaches such as bag-of-words model, using large movie reviews corpus, restricting to adjectives and adverbs, handling negations, bounding word frequencies by a threshold, and using WordNet synonyms knowledge. In order to improve classification, word semantic orientation is extracted from the lexical resources like SentiWordNet, ConceptNet, etc.

3.1.2 Proposed Methodology and System

The methodology comprises 5 steps which are depicted in the diagram below and are explained in detail further. Figure 3.2 shows the process of sentiment analysis. First the data is collected & stored in the dataset. Then the text in the dataset is preprocessed before giving it to the model. After preprocessing, the polarity of the sentiment words is calculated. In this process, wordnet is used. Based on that ‘Total Sentiment Score’ is calculated for the Input Text. Based on the sentiment scores, sentiment results are produced.

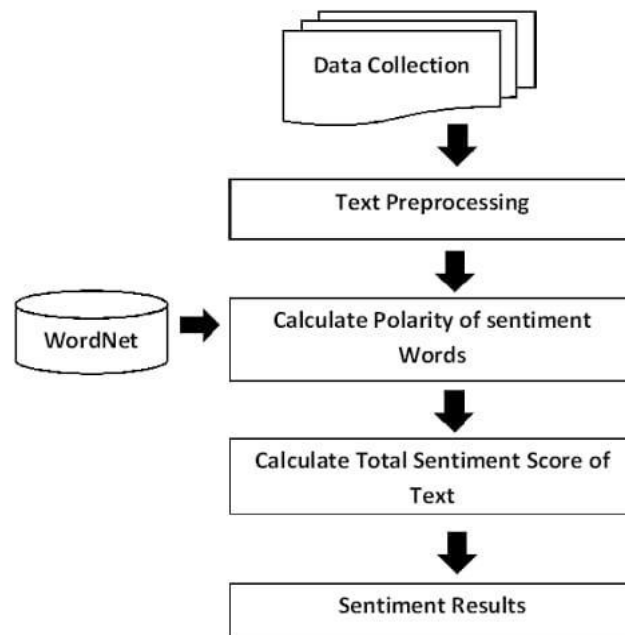


Fig 3.2: Sentiment Analyzer

Data Collection: The IMDB movies review dataset is utilized for this project. Any NLP system needs enough data for training and testing purposes. They can be split into two datasets: correct and incorrect (erroneous) data. Usually, it is not a problem to find and get a set of correct data because the correct texts are available from different sources, although they may also contain some mistakes. On the other hand, it is a hard task to clean data containing errors like typos, mistakes and misspellings in the process of data collection. This kind of data is usually obtained by a lengthy manual process and it requires annotation by humans.

Text Preprocessing: Preprocessing the raw text which is acquainted after data collection involves removing URLs, removing all irrelevant characters (Numbers and Punctuation), converting all characters into lowercase, tokenization, removing stopwords, stemming and lemmatization of resultant text, removal of words having length ≤ 2 , and then finally converting the list of tokens into back to the string, one which is processable for the next stages.

Calculating Polarity of Sentiment Words: The key aspect of sentiment analysis is to analyze a body of text for understanding the opinion expressed by it. Typically, we quantify this sentiment with a positive or negative value, called polarity. The overall sentiment is often inferred as positive, neutral or negative from the sign of the polarity score. In this project, the sentiment is calculated using synsets in SentiWordNet. Based on the polarity and subjectivity, you determine whether it is a positive text or negative or neutral.

Calculating Total Sentiment Score of Text: The number of occurrences of positive and negative words in each document is counted to determine the document's sentiment score. To calculate the document sentiment score, each positive word counts as + 1 and each negative word as - 1.

Sentiment Results: Sentiment is classified as per the net final score returned by the sentiment analyzer.

3.2 Implementation Details

3.2.1 Methodology

A) Text Pre-Processing and Normalization:

An initial step in text and sentiment classification is pre-processing. A significant amount of techniques is applied to data in order to improve classification effectiveness. This enables standardization across a document corpus, which helps build meaningful features, to reduce dimensionality and reduce noise that can be introduced due to many factors like irrelevant symbols, special characters, XML and HTML tags, and so on.

The main components in our text normalization pipeline are:

Cleaning Text — strip HTML

Our text often contains unnecessary content like HTML tags, which do not add much value when analyzing sentiment. Hence we need to make sure we remove them before extracting features. The BeautifulSoup library does an excellent job in providing necessary functions for this. Our `strip_html_tags(...)` function enables cleaning and stripping out HTML code.

Removing accented characters

In our dataset, we are dealing with reviews in the English language so we need to make sure that characters with any other format, especially accented characters are converted and standardized into ASCII characters. A simple example would be converting `é` to `e`. Our `remove_accented_chars(...)` function helps us in this respect.

Expanding Contractions

In the English language, contractions are basically shortened versions of words or syllables. Contractions pose a problem in text normalization because we have to deal with special characters like the apostrophe and we also have to convert each contraction to its expanded, original form. Our `expand_contractions(...)` function uses regular expressions and various contractions mapped to expand all contractions in our text corpus.

Removing Special Characters

Simple regexes can be used to achieve this. Our function `remove_special_characters(...)` helps us remove special characters. In our code, we have retained numbers but you can also remove numbers if you do not want them in your normalized corpus.

Lemmatizing text

Word stems are usually the base form of possible words that can be created by attaching affixes like prefixes and suffixes to the stem to create new words. This is known as inflexion. The reverse process of obtaining the base form of a word is known as stemming. The `nlk` package offers a wide range of stemmers like the `PorterStemmer` and `LancasterStemmer`. Lemmatization is very similar to stemming, where we remove word affixes to get to the base form of a word. However, the base form, in this case, is known as the root word but not the root stem. The difference is that the root word is always a lexicographically correct word, present in the dictionary, but the root stem may not be so. We will be using lemmatization only in our normalization pipeline to retain lexicographically correct words. The function `lemmatize_text(...)` helps us with this aspect.

Removing Stopwords

Words which have little or no significance especially when constructing meaningful features from the text are also known as stopwords or stop words. These are usually words that end up having the maximum frequency if you do a simple term or word frequency in a document corpus. Words like `a`, `an`, `the`, and so on are considered to be stopwords. There is no universal stopword list but we use a standard English language stopwords list from `nlk`. You can also add your own domain-specific stopwords if needed. The function `remove_stopwords(...)` helps us remove stopwords and retain words having the most significance and context in a corpus.

Normalize text corpus — tying it all together

We use all these components and tie them together in the following function called `normalize_corpus(...)`, which can be used to take a document corpus as input and return the same corpus with cleaned and normalized text documents.

B) Importing Necessary Libraries

This step mostly includes codes for the importing of different libraries and loading the data files and processing them into a proper format for manipulation.

C) Sentiment Analysis — Unsupervised Lexical

Even though we have labelled data, this section should give you a good idea of how lexicon-based models work and you can apply the same in your own datasets when you do not have labelled data. Unsupervised sentiment analysis models use well-curated knowledge bases, ontologies, lexicons, and databases that have detailed information pertaining to subjective words, phrases including sentiment, mood, polarity, objectivity, subjectivity, and so on. A lexicon model typically uses a lexicon, also known as a dictionary or vocabulary of words specifically aligned toward sentiment analysis. Usually, these lexicons contain a list of words associated with positive and negative sentiment, polarity (magnitude of negative or positive score), parts of speech (POS) tags, subjectivity classifiers (strong, weak, neutral), mood, modality, and so on. You can use these lexicons and compute sentiment of a text document by matching the presence of specific words from the lexicon, look at other additional factors like presence of negation parameters, surrounding words, overall context and phrases and aggregate overall sentiment polarity scores to decide the final sentiment score.

There are several popular lexicon models used for sentiment analysis. Some of them are mentioned as follows.

- Bing Liu's Lexicon
- MPQA Subjectivity Lexicon
- Pattern Lexicon
- AFINN Lexicon
- SentiWordNet Lexicon
- VADER Lexicon

Sentiment Analysis with SentiWordNet

The WordNet corpus is definitely one of the most popular corporas for the English language used extensively in natural language processing and semantic analysis. WordNet gave us the concept of synsets or synonym sets. The SentiWordNet lexicon is based on WordNet synsets and can be used for sentiment analysis and opinion mining. The SentiWordNet lexicon typically assigns three

sentiment scores for each WordNet synset. These include a positive polarity score, a negative polarity score and an objectivity score. We will be using the nltk library, which provides a Pythonic interface into SentiWordNet.

3.2.2 Details of packages, data set

Table 3.1: Dataset specification

Dataset	Size	Total No. of Reviews	Classes	Labeling
IMDB Dataset.csv	76 MB	50K	2	pre-labelled text reviews

IMDB dataset having 50K movie reviews for natural language processing or Text analytics.

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. A set of 25,000 highly polar movie reviews is provided for training and 25,000 for testing. Thus, predicting the number of positive and negative reviews can be done by using either classification or deep learning algorithms.

Table 3.2 Hardware details check for colab

Processor	2.3 GHz Dual-Core, Haswell, 12 GB NVIDIA Tesla K80 GPU
HDD	107.77 GB
RAM	12.69 GB

Table 3.3 Software details

Operating System	Windows
Programming Language	Python
Libraries Used	nltk, pandas, numpy, sklearn, seaborn, matplotlib, string, re, bs4, unicodedata

Natural Language Toolkit (NLTK)

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. ... NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

SentiWordNet

SentiWordNet is an opinion lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information. SentiWordNet is imported from nltk.corpus & is used as an important resource for sentiment classification tasks.

Pandas

Pandas is one of the tools in Machine Learning which is used for data cleaning and analysis. It has features which are used for exploring, cleaning, transforming and visualizing from data. It provides fast, flexible, and expressive data structures.

Numpy

NumPy stands for 'Numerical Python'. It is an open-source Python library used to perform various mathematical and scientific tasks. It contains multi-dimensional arrays and matrices, along with many high-level mathematical functions that operate on these arrays and matrices. The core of NumPy is a powerful optimized C Code.

Scikit-learn (sklearn)

Scikit-learn is a free machine learning library in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy .

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. SciPy makes use of Matplotlib.

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes.

BeautifulSoup 4 (bs4)

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

Unicode Database (unicodedata)

This module provides access to the Unicode Character Database (UCD) which defines character properties for all Unicode characters. Returns the decimal value assigned to the character chr as integer. If no such value is defined, default is returned, or, if not given, ValueError is raised.

String

The string module contains a number of functions to process standard Python strings

Regular Expression (RE)

This module provides regular expression matching operations similar to those found in Perl. Both patterns and strings to be searched can be Unicode strings (str) as well as 8-bit strings (bytes). Regular expressions use the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning.

Chapter 4

Project Inputs and Outputs

4.1 Inputs Details

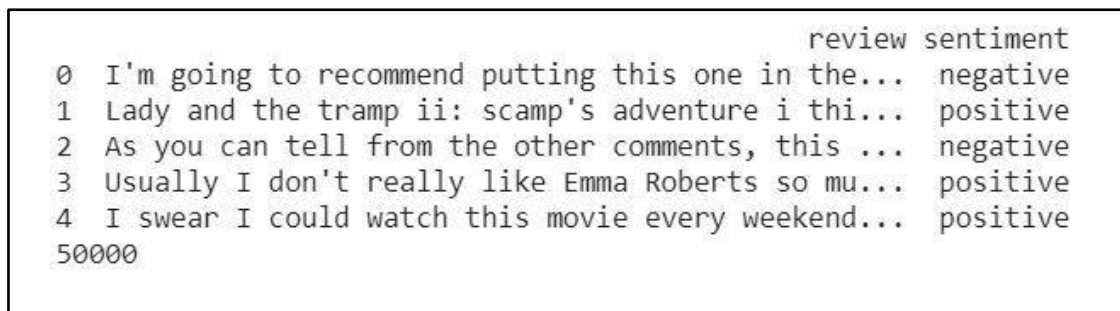
IMDB Dataset.csv is used for classification & is also used as input for training & testing purposes. Reviews are given as inputs to the trained model.

4.2 Evaluation Parameters Details

Systems implementing a SentiWordNet based Sentiment Analyzer takes a sample of reviews from the test dataset & determines it's sentiment polarity. Also, the model is evaluated using the parameters like precision, recall & F1-measure.

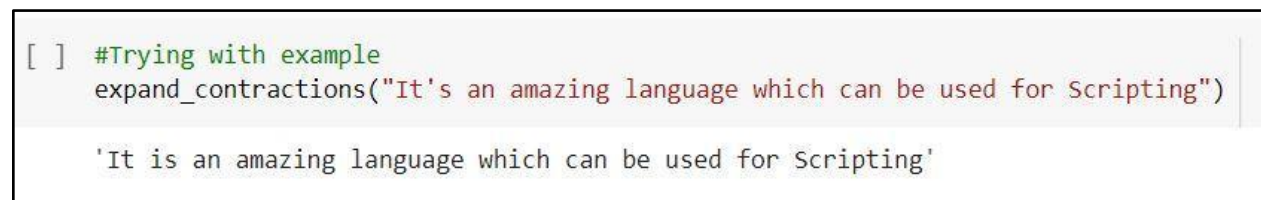
4.3 Output Details and Screenshots

The input reviews are processed & based on that the sentiment is predicted. Both, the actual sentiment & predicted sentiment are returned in the output. Also, the positive, negative & overall polarity of the predicted sentiment are given as output.



```
review sentiment
0 I'm going to recommend putting this one in the... negative
1 Lady and the tramp ii: scamp's adventure i thi... positive
2 As you can tell from the other comments, this ... negative
3 Usually I don't really like Emma Roberts so mu... positive
4 I swear I could watch this movie every weekend... positive
50000
```

Fig 4.1: Loading Data



```
[ ] #Trying with example
expand_contractions("It's an amazing language which can be used for Scripting")

'It is an amazing language which can be used for Scripting'
```

Fig 4.2: Cleaning the Text - Expanding Contractions

```

awesome = list(swn.senti_synsets('happy', 'a'))[0]
print('Positive Polarity Score:', awesome.pos_score())
print('Negative Polarity Score:', awesome.neg_score())
print('Objective Score:', awesome.obj_score())

```

```

Positive Polarity Score: 0.875
Negative Polarity Score: 0.0
Objective Score: 0.125

```

Fig 4.3: Sentiment Analysis with SentiWordNet - Likelihood of a Word to make the Review Positive or Negative

```
In [ ]:
```

```

sample_id= [4726,12103,25726,49255]
sample_reviews= reviews[sample_id]

```

```
In [ ]:
```

```
sample_reviews[2]
```

```
Out[ ]:
```

```

'What the (beep) is going wrong with Disney the last years? Are there totally run out of
good ideas? Where is the magic? Where are the good animators, the good songwriters, the g
ood directors, the good... Okay, i know, Walt himself and the famous "nine old man" can\'
t come back. But is this a reason to crank out countless of those cheap sequels and slowl
y but surely destroying the ideals of Walt Disney? I never rent or bought a Disney-sequel
of what movie however. Because i had read much enough about its (absence of) quality. But
"Atlantis: Milo\'s Return" was aired today on TV in Germany and so i watch it. It confirm
ed my doubts about sequels. It was absolutely boring. Flaw animation, primitive color-rot
ation, simple characters, some unsuccessful tries to simulate the famous Multiplane-Camer
a with CGI, mediocre music and a patchwork of different, simple stories. It looks absolut
ely not like Disney! Not like Disney i know! It looks like one of the countless, cheap an

```

Fig 4.4: Sentiment Analysis with SentiWordNet - Sample Review

```
norm_reviews=clean_text(sample_reviews)
```

```
In [ ]:
```

```
norm_reviews[2]
```

```
Out[ ]:
```

```

'beep going wrong disney last year totally run good idea magic good animator good songwri
ter good director good okay know walt famous `` nine old man not come back reason crank c
ountless cheap sequel slowly surely destroying ideal walt disney never rent bought disney
sequel movie however read much enough absence quality `` atlantis milo return wa aired to
day tv germany watch confirmed doubt sequel wa absolutely boring flaw animation primitive
colorrotation simple character unsuccessful try simulate famous multiplanecamera cgi medi
ocre music patchwork different simple story look absolutely not like disney not like disn
ey know look like one countless cheap simple animationseries like `` dragonballz `` beybl
ade etc aired every day tv child first reaction showing crap wa load `` bambi dvdplayer s
ee disney immortal magic depth spirit charm see disney climax see awesome art handmade an
imation `` bambi wa first today movie give 10 10 star `` atlantis milo return no magic no
depth no charm no spirit deserved 3 10'

```

Fig 4.5: Sentiment Analysis with SentiWordNet - Normalization of the Review

```

▶ for review, sentiment in zip(reviews[sample_id], sentiments[sample_id]):
    print('REVIEW:', review)
    print('Actual Sentiment:', sentiment)
    pred = analyze_sentiment_sentiwordnet_lexicon(review, verbose=True)
    print('*'*40)

REVIEW: Utter dreck. I got to the 16 minute/27 second point, and gave up. I'd
Actual Sentiment: negative
Predicted Sentiment Objectivity Positive Negative Overall
0 negative 0.86 0.06 0.07 -0.01
*****

REVIEW: This must be the worst thriller I have seen in a long long time. The
Actual Sentiment: negative
Predicted Sentiment Objectivity Positive Negative Overall
0 positive 0.85 0.08 0.07 0.01
*****

REVIEW: What the (beep) is going wrong with Disney the last years? Are there
Actual Sentiment: negative
Predicted Sentiment Objectivity Positive Negative Overall
0 positive 0.82 0.11 0.07 0.04
*****

REVIEW: Assassin Hauser's (John Cusak) mission is to whack a Mid-Eastern oil
Actual Sentiment: negative
Predicted Sentiment Objectivity Positive Negative Overall
0 positive 0.85 0.08 0.07 0.0
*****

```

Fig 4.6: Sentiment Analysis with SentiWordNet - Sentiment Prediction for Reviews

```

%%time
predicted_sentiments_swn = [analyze_sentiment_sentiwordnet_lexicon(review, verbose=False)
for review in norm_complete_reviews]

CPU times: user 16min 13s, sys: 5.09 s, total: 16min 18s
Wall time: 16min 19s

```

Fig 4.7: Predict Sentiment for Complete Dataset

```

Model Performance metrics:
*****

Accuracy: 0.6772
Precision: 0.6824
Recall: 0.6772
F1 Score: 0.6749

```

Fig 4.8: Evaluate Model Performance

Model Classification report:				

	precision	recall	f1-score	support
positive	0.65	0.76	0.70	25000
negative	0.71	0.59	0.65	25000
accuracy			0.68	50000
macro avg	0.68	0.68	0.67	50000
weighted avg	0.68	0.68	0.67	50000

Fig 4.9: Model Classification Report

Prediction Confusion Matrix:		

	predicted:	
	positive	negative
Actual: positive	19050	5950
negative	10190	14810

Fig 4.10: Prediction Confusion Matrix

Chapter 5

Summary and Future Scope

5.1 Summary

In this project, sentiment analysis is carried out on an unprocessed IMDB dataset which consists of movie reviews provided by the general population. Initially, the raw data is passed for preprocessing. While preprocessing, the text is cleaned through various steps like removing HTML tags & special characters, removing stopwords, lemmatization, expanding contractions, etc. The gathered cleaned text through each step is collectively acquired for text normalization. The normalized reviews are given as inputs for further processing. The SentiWordNet algorithm is used in order to determine the polarity of the sentiments. For finding the total Sentiment Score of the overall dataset, Scikit-Learn & other ML libraries are used. Performance of the sentiment classification model is also visualized.

5.2 Future Scope

Further study is to combine information from various sites such as twitter and YouTube comments in addition to IMDB user reviews. And also this work will continue to implement sentiment analysis techniques on other domains like product reviews, newspaper articles, political forums etc.

References

- [1] Ke, Pei, et al. "Sentilare: Linguistic knowledge enhanced language representation for sentiment analysis." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020.
- [2] Baid, Palak & Gupta, Apoorva & Chaplot, Neelam. (2017). Sentiment Analysis of Movie Reviews using Machine Learning Techniques. *International Journal of Computer Applications*. 179. 45-49. 10.5120/ijca2017916005.
- [3] S. Vashishtha and S. Susan, "Fuzzy Interpretation of Word Polarity Scores for Unsupervised Sentiment Analysis," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225646.
- [4] T. P. Sahu and S. Ahuja, "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms," 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), 2016, pp. 1-6, doi: 10.1109/MicroCom.2016.7522583.
- [5] Shaukat, Z., Zulfiqar, A.A., Xiao, C. et al. Sentiment analysis on IMDB using lexicon and neural networks. *SN Appl. Sci.* 2, 148 (2020). <https://doi.org/10.1007/s42452-019-1926-x>

Acknowledgement

We deeply express our sincere thanks to our guide Prof. Dhiraj Amin for guiding us throughout this mini project. He took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We would like to express our special thanks to our H.O.D. Dr. Sharvari Govilkar who gave us the opportunity to do this project work in Pillai College of Engineering, Panvel and gave us all support and guidance which made us complete the project duly.

We are extremely thankful to our Principal Dr. Sandeep M. Joshi, for his encouragement and more over for his timely support and guidance till the completion of our project work. We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Engineering Department which helped us in successfully completing our project work. Also, I would like to extend our sincere esteem to all staff in the laboratory for their timely support.

Adakar Amol Mangesh
Bharambe Rishikesh Vijay
Gladson Daniel Roy
Sharma Smriti Sushil Kumar