Sentiment Analysis

Progress Report
Minor Project

Submitted By:
Geetansh Chawla, 02220802716
Karan Rawlley, 03420802716
Kritarth Bisht, 03520802716
Mahak Sharma, 04120802716

Supervisor
Ms. Charu Gupta
Asst. Prof. (CSE)

Department of Computer Science & Engineering
Bhagwan Parshuram Institute of Technology
PSP-4, Sec-17, Rohini, Delhi-110089

Table of Contents

Chapter 1

Introduction

Sentiment analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. Sentiment analysis is the process of detecting a piece of writing for positive, negative, or neutral feelings bound to it. Humans have the innate ability to determine sentiment; however, this process is time consuming, inconsistent, and costly in a business context it's just not realistic to have people individually read tens of thousands of user customer reviews and score them for sentiment. Movie reviews are an important way to gauge the performance of a movie. While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie. A textual movie review tells us about the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer.

Sentiment Analysis is a major subject in natural language processing which aims to extract subjective information from the textual reviews. It can be used to determine the attitude of the reviewer with respect to various topics or the overall polarity of review.

Using sentiment analysis, cinematographer can find the state of mind of the reviewer while providing the review and understand if the person was "happy", "sad", "angry" and so on. This project aims to use sentiment analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie was, i.e. if they liked the movie or they hated it. This project aims to utilize the relationships of the words in the review to predict the overall polarity of the review.

Chapter 2

Problem Statement

One of the most important elements for businesses is being in touch with its customer base. It is vital for these firms to know exactly what consumers or clients think of new and established products or services, recent initiatives, and customer service offerings. In this study the framework uses to understand of sentiment analysis. As observed movies is a great source of entertainment to the people. The review (online/offline) matters in the overall profit of the movie. Word of mouth publicity is the best way to increase profit. Movies are reviewed using various mediums including social media, blogs, forums, imdb, movie viewing platforms. Any movie team need to continuously monitor the reviews given by the viewers on movies and use text classification for analysing them so that they could recognize the most trending genre. The problem is aimed at making use of natural language processing in interpreting Movie Review dataset.

2.1 Need of the project

This project determines the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed with an online mention. It can be extremely useful in social media monitoring as it allows to gain an overview of the wider public opinion behind certain topics. The applications can be broad and powerful. It can also be an essential part of market research and customer service approach. Every cinematographer needs to analyse viewer's review for knowing the movie's reception and for planning future projects. They can also find out how well the competitors are doing. The overall customer experience of the clients can be revealed quickly. The ability to understand customer attitudes and react accordingly is something that the business can take advantage of.

2.2 Features

The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. This project aims to classify whether a person liked the movie or not based on the review they give for the movie. This is particularly useful in cases when the creator of a movie wants to measure its overall performance using reviews that critics and viewers are providing for the movie. The outcome of this project can also be used to create a recommender by providing recommendation of movies to viewers on the basis of their previous reviews. Another application of this project would be to find a group of viewers with similar movie tastes (likes or dislikes). This project will aim to study several feature extraction techniques used in text mining e.g. keyword spotting, lexical affinity and statistical methods, and understand their relevance to our problem. In addition to feature extraction, we also look into different classification techniques and explore how well they perform for different kinds of feature representations. This project will finally draw a conclusion regarding which combination of feature representations and classification techniques are most accurate for the current predictive task.

2.3 Objective

Sentiment analysis is one way to solve the above mentioned problem. Sentiment Analysis is a field of Natural Language Processing (NLP) that builds models that try to identify and classify attributes of the expression e.g.:

- Polarity: if the speaker expresses a *positive* or *negative* opinion,
- Subject: the thing that is being talked about,
- Opinion holder: the person, or entity that expresses the opinion.

World generates 2.5 quintillion bytes of data every day, sentiment analysis has become a key tool for making sense of that data. This has allowed companies to get key insights and automate all kinds of processes.

Sentiment Analysis can help to automatically transform the unstructured information into structured data of public opinions about products, services, brands, politics or any other topic that people can express opinions about. This project will implement a Deep Learning model that can classify movie reviews as positive or negative. The model will take a whole review as an input (word after word) and provide label ratings for checking whether the review conveys a positive or negative sentiment. The dataset used contains roughly 25000 negative and 25000 positive reviews.

2.4 Settings in Training Phase

This dataset consists of a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided. The dataset used for this task was collected from Large Movie Review Dataset which was used by the AI department of Stanford University for the associated publication. The dataset contains 50,000 training examples collected from IMDB where each review is labelled with the rating of the movie on a scale of 1-10. As sentiments are usually bipolar like good/bad or happy/sad or like/dislike, ratings are categorized as either 1 (positive) or 0 (negative) based on the ratings. If the rating was above 5, it is marked 1, otherwise 0. A typical review text looks like this:

*"I'm a fan of TV movies in general and this was one of the good ones. The cast performances throughout were pretty solid and there were twists I didn't see coming before each commercial. To me it was kind of like Medium meets CSI.

Did anyone else think that in certain lights, the daughter looked like a young Nicole Kidman? Are they related in any way? I'd definitely watch it again or rent it if it ever comes to video.

Dedee was great. Haven't seen her in a lot of things and she did her job very convincingly.

If you're into to TV mystery movies, check this one out if you have a chance."*

2.5 Experimental Analysis

- Dataset is downloaded and extracted from Stanford website using the above written code.

```
In [11]: import os
         import tarfile
         from six.moves import urllib

In [12]: download_url = "http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz"
         data_path = "D:\Projects\Minor_Project\data"

In [14]: def fetch_data(d_url=download_url,d_path=data_path):

             tgz_path = os.path.join(d_path,"aclImdb_v1.tar.gz")
             urllib.request.urlretrieve(d_url,tgz_path)

             reviews_tgz = tarfile.open(tgz_path)
             reviews_tgz.extractall(path=d_path)
             reviews_tgz.close()


In [15]: fetch_data()
```

Figure 2.1: Downloading data

- Labeling data 1 and 0 and creating csv file.

```
labels = {'pos':1, 'neg':0}
pbar = pyprind.ProgBar(50000)
df = pd.DataFrame()
```

```
for s in ('test', 'train'):
    for l in ('pos', 'neg'):
        path = os.path.join(basepath,s,l)
        print(s,l)
        print(path)
        for file in os.listdir(path):
            with open(os.path.join(path, file),'r',encoding='utf-8') as infile:
                txt = infile.read()
            df = df.append([[txt,labels[l]]], ignore_index=True)
            pbar.update()
df.columns = ['reviews', 'sentiment']
```

Figure 2.2: Extraction and labelling

- Preprocessing each review by removing Html tags, punctuation marks and making it case-insensitive by lower casing.

```
In [4]: def preprocess(text):
          text = re.sub('<[^>]*>', '', text)
          emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)', text.lower())
          text = re.sub('[\W]+', ' ', text.lower()) +\
              ' '.join(emoticons).replace('-', '')
          return text
```

Figure 2.3: Method for Pre-processing

- Creating embedding matrix for converting text form data into machine readable form, a matrix consisting of each word of a review converting into a list of 1's and 0's.

```
# using GloVe for creating feature matrix
from numpy import array
from numpy import asarray
from numpy import zeros

embeddings_dictionary = dict()
glove_file = open('glove.6B.100d.txt', encoding="utf8")

for line in glove_file:
    records = line.split()
    word = records[0]
    vector_dimensions = asarray(records[1:], dtype='float32')
    embeddings_dictionary [word] = vector_dimensions
glove_file.close()
```

```
embedding_matrix = zeros((vocab_size, 100))
for word, index in tokenizer.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector
```

Figure 2.4: Embedding Matrix

- Creating a bidirectional recurrent neural network model for classification.

```
model = Sequential()
model.add(Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=maxlen))
model.add(Bidirectional(LSTM(200)))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['acc'])
```

Figure 2.5: Model Creation

- Training model using training dataset.

```
model.fit(X_train, y_train, epochs=10, batch_size=64)
```

```
Epoch 1/10
40000/40000 [==============================] - 332s 8ms/step - loss: 0.4632 - acc: 0.7772
Epoch 2/10
40000/40000 [==============================] - 293s 7ms/step - loss: 0.3080 - acc: 0.8713
Epoch 3/10
40000/40000 [==============================] - 301s 8ms/step - loss: 0.2605 - acc: 0.8936
Epoch 4/10
40000/40000 [==============================] - 298s 7ms/step - loss: 0.2318 - acc: 0.9082
Epoch 5/10
40000/40000 [==============================] - 299s 7ms/step - loss: 0.2236 - acc: 0.9105
Epoch 6/10
40000/40000 [==============================] - 299s 7ms/step - loss: 0.1814 - acc: 0.9297
Epoch 7/10
40000/40000 [==============================] - 297s 7ms/step - loss: 0.1490 - acc: 0.9438
Epoch 8/10
40000/40000 [==============================] - 303s 8ms/step - loss: 0.1129 - acc: 0.9600
Epoch 9/10
40000/40000 [==============================] - 318s 8ms/step - loss: 0.0851 - acc: 0.9706
Epoch 10/10
40000/40000 [==============================] - 300s 8ms/step - loss: 0.0618 - acc: 0.9797
```

Figure 2.6: Training model

Chapter 3

Design Phase

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into an architecture. The architecture defines the components, their interfaces and behaviors. The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the ``how" phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

- **Use Case Diagram**

  Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable results to the actors or other stakeholders of the system. The User looks into the camera. The camera captures frames of the User and according to User head movements determines the motion of the pointer to the left, right, up and down. Similarly, User also judges whether to select mode 1 or mode 2 according to the mouth open and close motion

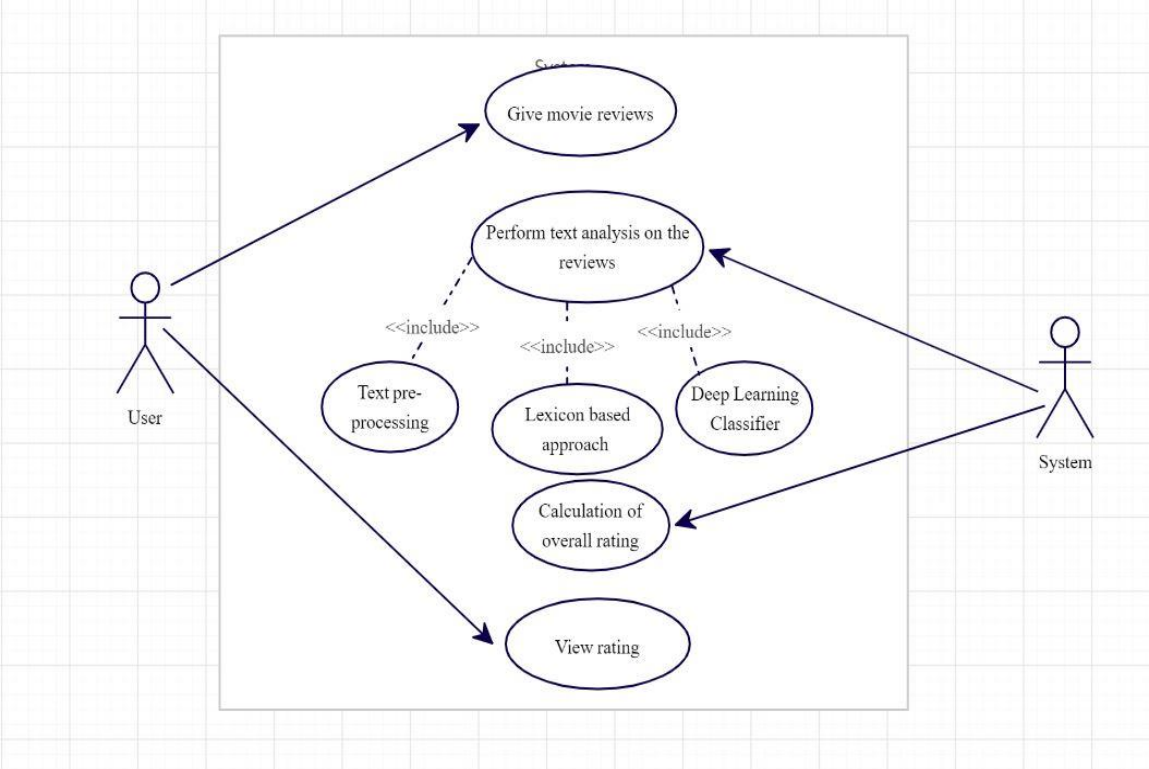

Figure 3.1: Use Case Diagram

- **Data Flow Diagram**

  Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and report generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes the flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

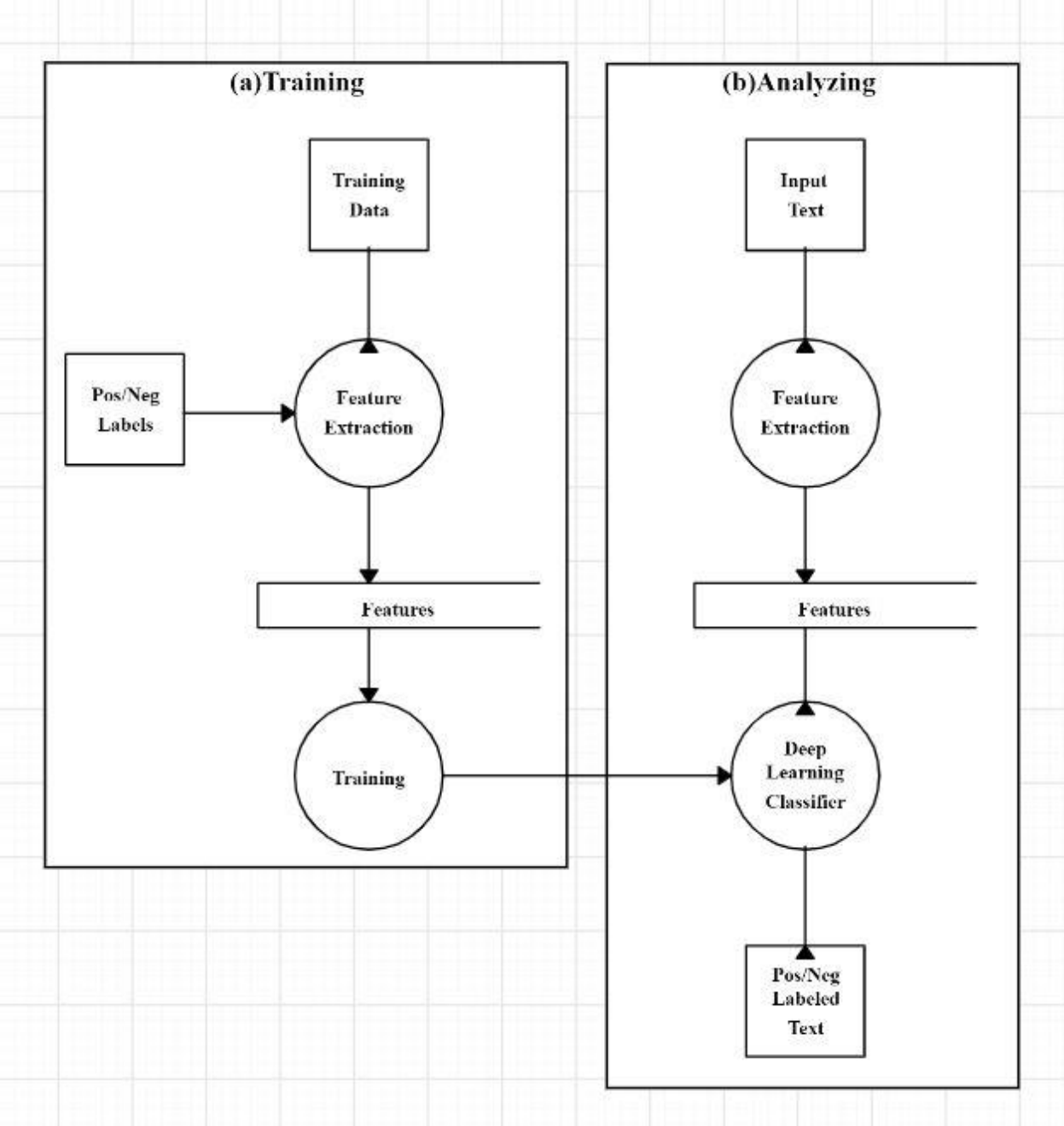

Figure 3.2: Level-1 Data Flow Diagram

References

1. Burscher, Bjorn, Rens Vliegenthart, and Claes H. de Vreese. "Frames Beyond words: applying cluster and sentiment analysis to news coverage of the nuclear power issue." *Social Science Computer Review* 34, no. 5 (2016): 530-545.

2. Sentiment Analysis - Wikipedia - https://en.wikipedia.org/wiki/Sentiment_analysis

3. Large Movie Review Dataset - https://ai.stanford.edu/~amaas/data/sentiment/

4. Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142-150. Association for Computational Linguistics, 2011.

5. Internet Movie Database - https://www.imdb.com/

6. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.

7. Tumasjan, Andranik, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Welpe. "Predicting elections with twitter: What 140 characters reveal about political sentiment." In *Fourth international AAAI conference on weblogs and social media*. 2010.

8. Bidirectional Recurrent Neural Network - https://en.wikipedia.org/wiki/Bidirectional_recurrent_neural_networks

9. Snyder, Benjamin, and Regina Barzilay. "Multiple aspect ranking using the good grief algorithm." In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 300-307. 2007.

10. Liu, Yang, Chengjie Sun, Lei Lin, and Xiaolong Wang. "Learning natural language inference using bidirectional LSTM model and inner-attention." *arXiv preprint arXiv:1605.09090* (2016).

11. BRNN using keras - https://keras.io/layers/wrappers/