Rating Reviews Based on Sentiment Analysis

Saeed Al-Qarni
School of Computing and Engineering
University of Missouri - Kansas City
saacfb@mail.umkc.edu

Kahlid Dhabbah
School of Computing and Engineering
University of Missouri - Kansas City
kmdk2t@mail.umkc.edu

Anshit Saxena
School of Computing and Engineering
University of Missouri - Kansas City
asm5y@mail.umkc.edu

***Abstract*—Nowadays, Social media has become popular places for sharing the best description of a movie through texts. Analyzing reviews or text helps in recognizing the sentiment stated by people. IMDb movie has a lot of reviews that have been expressed by reviewers towards a movie. A positive review and a negative review have taken place for many researchers to apply sentiment analysis to them. In the proposed work, we show how to give a rating based on reviews and compare them with IMDb rating. Getting positive and negative reviews based on our rating plays a significant role in text classification using sentiment analysis. As a result, we will see how the difference will be between our rating and IMDb. Moreover, our model will help people find the easiest way to read the top words for a movie instead of spending more time to just read a few completed reviews.**

I. INTRODUCTION

Text classification is a valuable task in Natural Language Processing (NLP), such as sentiment analysis, web search, and reviews classification. sentiment analysis is one of the hot topics in the area of Natural Language Processing (NLP). Nowadays, people watch a lot of movies after they can see the rating for that movie. However, people can only view 10-20 reviews at a time. Through sentiment analysis, people can know the words that are described for a movie and the review ratings from above 5000 reviews. It is an easy way to describe and find the best movie instead of reading a lot of text in a review. In this paper, we explore ways to scale these reviews and compare them with IMDB rating, in the context of text classification. We will grab the reviews for a movie and do the sentiment analysis. After that, we will give the rating based on that sentiment and compare it with IMDB rating. Once we get our rating, we can find the top words which describe a movie.

II. RELATED WORK

In [1], they have used IMDb reviews to do sentiment analysis. This paper is about enhancing the accuracy and improving the overfitting by utilizing two networks, CNN and LSTM. Also, by using the Dropout layer with max Pooling and Batch Normalization, it helped them overfitting issues. They achieved the best accuracy of 89.5

The author's objective in [2] is to improve the vanishing gradient issue. They applied sentiment analysis on IMDb reviews. Their model was a mix of CNN and residual connection, and they tested it on four different models. They achieved the best accuracy of 90.02

In [3], two models were proposed. The first model is a hybrid LSTM-CNN, which performs better than the second hybrid CNN-LSTM model. The accuracy achieved was 79%. They have used preprocessing techniques to enhance the model's performance. Not only that but also, they used an embedded layer to embeds the words into a vector.

[4] used (TF, TF-IDF) beside Lexicon features to result in a better output of the model. It is also used the IMDb dataset. The pr-eprocessing and feature extraction were applied to the dataset to give a better result. They have used different algorithms, such as SVM, KNN, Maximum Entropy, and Naïve Bayes.

[5] applied a new way of sentiment analysis using Bayesian Rough Decision Tree, which uses the Decision Tree and Bayesian Rough set combined. They have done the text pre-processing, such as tokenization and conversion to lowercase, as well as feature extraction, such as POS. They achieved around 95

III. ARCHITECTURE

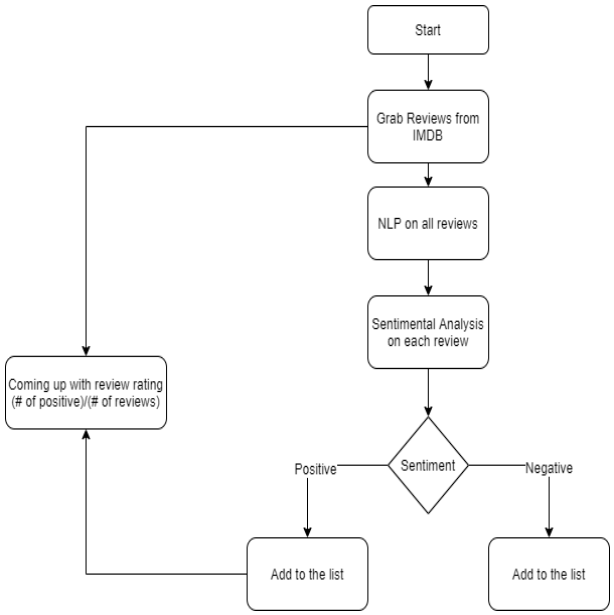

Figure 1: Flowchart for the proposed architecture

We have followed several steps to implement the sentiment analysis as shown in Figure 1.

1) Collecting the Dataset

For the purpose of collecting the data from the IMDB for a particular movie, we are using the Chrome Browser. We use a driver dedicated for chrome for opening up the IMDb reviews

page for that particular movie and wait for it to load all the reviews. After the page has loaded all the reviews we get all the reviews using BeautifulSoup and HTTP Parser.

Figure 2 shows our approach for collecting the IMDb Dataset.

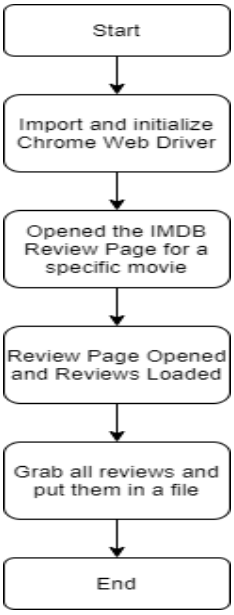

Figure 2: Flowchart for getting the IMDb dataset

2) Performing NLP

Removing stop words Encoding the data in proper format (UTF-8) Tokenization and Capitalization/De-Capitalization Breaking attached words Lemmatization/Stemming

3) Performing Sentiment Analysis

After running sentiment analysis on each review, we classify them as positive and negative and separate them.

4) Calculating the Rating

We calculate our rating using the total number of reviews and the number of positive reviews by:

Total number of Positive reviews / Total number of reviews

IV. Implementation and Evaluation

In order to get the dataset, we are using a WebDriver for Chrome browser, in which we will open the review site and load all the reviews on the page as shown in Figure 3 & 4.

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.common.exceptions import ElementNotVisibleException
```

Figure 3: Importing webdriver

Figure 5 shows that After loading all the reviews on the browser page, we use a python library 'BeautifulSoup' to parse all the content to 'utf-8' format and find out how many reviews are there on the page.

we write all the reviews in a file by grabbing the reviews one by one from the page as in Figure 6, and in 7 to see

```
driver = webdriver.Chrome("C:\Webdriver/chromedriver.exe")
driver.get("https://www.imdb.com/title/tt0468569/reviews?ref_=tt_ql_3")

button = driver.find_element_by_id('load-more-trigger')
key = True
while key:
    time.sleep(1)
    try:
        button.click()
        button = driver.find_element_by_id('load-more-trigger')
    except:
        key = False
```

Figure 4: Using webdriver to load the data

```
print(type(TheDarkKnight))
print(len(TheDarkKnight))
```

```
<class 'bs4.element.ResultSet'>
6597
```

Figure 5: Displaying the number of reviews

```
brothListReviewsOnly = []
brothList = soup.select(".review-container")
temp = 1
for item in range(0,len(brothList)):
    title = brothList[item].select(".title")[0].text
    review = brothList[item].select(".text")[0].text
    brothListReviewsOnly.append(review)
    file = open(r'C:\Users\Saeed\Desktop\KDM_Project\AllReviews_beforeCleaning\r' +str(temp) +'.txt', 'w',
encoding='utf-8')
    temp = temp+1
#     print(review +"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    file.write(str(review))
file.close()
```

Figure 6: Displaying the number of reviews

```
print(len(brothListReviewsOnly))
```

```
6597
```

Figure 7: Displaying the number of reviews

```
brothListReviewsOnly[1]
```

```
'I got to see The Dark Knight on Wednesday night the reason though why Im writing this movie comment this
late is because I didnt wanna just jump and say this movie was awesome I wanted to think it through still
today I cant stop thinking about this movie The Dark Knight lives up to its hype and goes beyond it this i
s the Batman movie that goes where no other Batman movie has gone before It gave us a real sold story we a
re finally told why the villains are the way they are how all the injustice in the world can get underneat
h Batmans skin how you have to believe that people still have faith in good on Earth Heath Ledger I absolu
tely loved Jack Nicolsons performance in the 1989 film when I first read that Heath was signed onto The Da
rk Knight I was like What are they kidding once you see The Dark Knight Heath was incredible Im not going
```

Figure 8: A sample of a review

```
stop = set(stopwords.words('english'))
temp = 1
for x in range(0,len(brothListReviewsOnly)):
    filtered_words = [i for i in word_tokenize(brothListReviewsOnly[x].lower()) if i not in stop]
    brothListReviewsOnly[x] = " ".join(filtered_words)
    file = open(r'C:\Users\Saeed\Desktop\KDM_Project\AllReviews_afterCleaning\r' +str(temp) +'.txt', 'w',
encoding='utf-8')
    temp = temp + 1
    file.write(str(brothListReviewsOnly[x]))
file.close()
```

Figure 9: Cleaning up the reviews

```
brothListReviewsOnly[1]
```

```
'got see dark knight wednesday night reason though im writing movie comment late didnt wan na jump say mov
ie awesome wanted think still today cant stop thinking movie dark knight lives hype goes beyond batman mov
ie goes batman movie gone gave us real sold story finally told villains way injustice world get underneath
batmans skin believe people still faith good earth heath ledger absolutely loved jack nicolsons performanc
e 1989 film first read heath signed onto dark knight like kidding see dark knight heath incredible im goin
```

Figure 10: A sample of a review after cleaning

if we got all the reviews. A sample of how the review is grabbed is illustrated in Figure 8. Then we start cleaning up the reviews by first removing the stop words as in Figure 9. Figure 10 is how the sample review looks after removal of stopwords. We use python core nlp from StanfordCoreNLP to do our sentimental analysis on the reviews and then segregate the reviews on the basis of the sentiment as in Figure 11. Segregating the reviews and counting them is shown in Figure 12. As a result, we are providing the negative rating and the positive rating for the movie in Figure 13.

To get the top words, we are just using the frequency for now to find most frequently used words in both positive and negative reviews. Then after getting the top words we are masking those top words onto images for both positive and

```
from pycorenlp import StanfordCoreNLP
import json
```

```
nlp = StanfordCoreNLP('http://localhost:9000')
#Sentiment_Reviews = reviews_from_file[7]
all_review_sentiment= []
for x in range (len(reviews_from_file)):
    results = nlp.annotate(reviews_from_file[x],properties={
        'annotators':'sentiment,ner, pos',
        'outputFormat': 'json',
        'timeout': 50000
        })
    #all_review_sentiment.append(results["sentences"]["sentiment"])
    for s in results["sentences"]:
        #print(s["sentiment"])
        all_review_sentiment.append(s["sentiment"])
```

Figure 11: Using CoreNLP

```
sent_dict = {}
for y in range (len(all_review_sentiment)):
    if all_review_sentiment[y] in sent_dict:
        sent_dict.update({all_review_sentiment[y]:sent_dict[all_review_sentiment[y]] + 1})
    else:
        sent_dict.update({all_review_sentiment[y]:1})
print(sent_dict)
```

```
{'Positive': 2165, 'Neutral': 477, 'Negative': 2994, 'Verynegative': 489, 'Verypositive': 471}
```

```
sent_negative = sent_dict['Verynegative'] + sent_dict['Negative']
print(sent_negative)
```

```
3483
```

```
sent_positive = sent_dict['Positive'] + sent_dict['Verypositive']
print(sent_positive)
```

```
2636
```

Figure 12: Segregating and counting the reviews

```
negative_rating = (sent_negative / len(reviews_from_file))* 10
print(int(negative_rating))
```

```
5
```

```
positive_rating = (sent_positive / len(reviews_from_file))* 10
print(positive_rating)
```

```
3.9963614311704063
```

Figure 13: Providing the negative rating and the positive rating

```
wordcloudNegative= WordCloud(max_words=50, mask=mask, background_color='white')
wordcloudNegative.generate(allNegative)
plt.figure(figsize=(10,10))
plt.axis("off")
plt.imshow(wordcloudNegative, interpolation = 'bilinear')
plt.savefig("Negative.png")
```

Figure 14: Providing the most frequently used words

negative reviews to make word clouds.

V. Conclusion

Our project provides ratings for movies on the basis of the sentiment of the reviews posted by the viewers on IMDB website. This project not only gives the positive rating but also negative ratings. Also, this project provides word clouds for both positive and negative sentiment to the users for them to post on social media.

VI. Future work

We will be extending this project and will develop a web app, on which users can go and type in any movie they would like to get the rating for and also get the word clouds to share on their social media.

References

[1] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, 2017, pp. 540-546.

[2] N. K. Thinh, C. H. Nga, Y. Lee, M. Wu, P. Chang and J. Wang, "Sentiment Analysis Using Residual Learning with Simplified CNN Extractor," 2019 IEEE International Symposium on Multimedia (ISM), San Diego, CA, USA, 2019, pp. 335-3353.

[3] A. Sajeevan and L. K. S, "An enhanced approach for movie review analysis using deep learning techniques," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1788-1794.

[4] Kumar, Himanshu, B. S. Harish and H. K. Darshan. “Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method.” IJIMAI 5 (2019): 109-114.

[5] İ.Tarımer, A. Çoban, A.E. Kocaman. "Sentiment Analysis on IMDB Movie Comments and Twitter Data by Machine Learning and Vector Space Techniques" 2019.