

Sentiment Analysis Project Report

Team Members:

Ranjan Kumar, Pankaj Yadav, Ashutosh Kumar, Vidya Bharti

Introduction

In this report, we'll discuss our project aimed at creating a system to understand whether movie reviews are positive or negative. We used a dataset from Kaggle with 50,000 reviews for this purpose. Our task involved developing our own analysis system, checking its accuracy by comparing its predictions to the actual sentiments in the dataset, and testing it against two other analysis tools in Java—one based on machine learning and the other on deep learning.

Methodology

Data Collection

The dataset of movie reviews has been downloaded from Kaggle, where each row represents a single review with sentiment.

Token Scoring

After reading every row of the dataset, we tokenized the sentences. Then filtered the sentiment words manually. We distributed tokens to our team members. We added the sentiment score for each word manually. This is how we created the newScore.txt file.

Sentiment Analysis

First of all we created a dictionary from newScore.txt, In which tokens were taken as key and the scores were assigned as value. Now we tokenized each review in the dataset to extract individual words. Matched these words with tokens in the dictionary to calculate the cumulative score for each review.

If the cumulative score for a review was greater than zero, categorised the review as positive. Reviews with a score of zero or lower were classified as negative, based on the context and tone of the review.

This process was iteratively done for each row and thus we calculated the accuracy which is 67%.

Using Other Sentiment Analysis Tools

We used other sentiment analysis tools, first we compared with machine learning models and then we compared with deep learning to see how they compare to our own system. We

chose these tools because they're well-known and work well with language processing. Our goal was to check how accurate, fast, and easy they were to use compared to our system.

Results

Accuracy of Our Model

```
PS C:\Users\py962\OneDrive\Documents\Adv_Java\Assignment\Assignment3> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\py962\AppData\Roaming\Code\User\workspaceStorage\7419a6156e14188e408366be0ef50758\redhat.java\jdt_ws\Assignment3_db27e685\bin' 'Analysis'
Accuracy: 67%
PS C:\Users\py962\OneDrive\Documents\Adv_Java\Assignment\Assignment3>
```

Using Our model we got accuracy of around 67%.

Accuracy of Machine Learning

```
[[4209 752]
 [1627 3412]]
0.7621
```

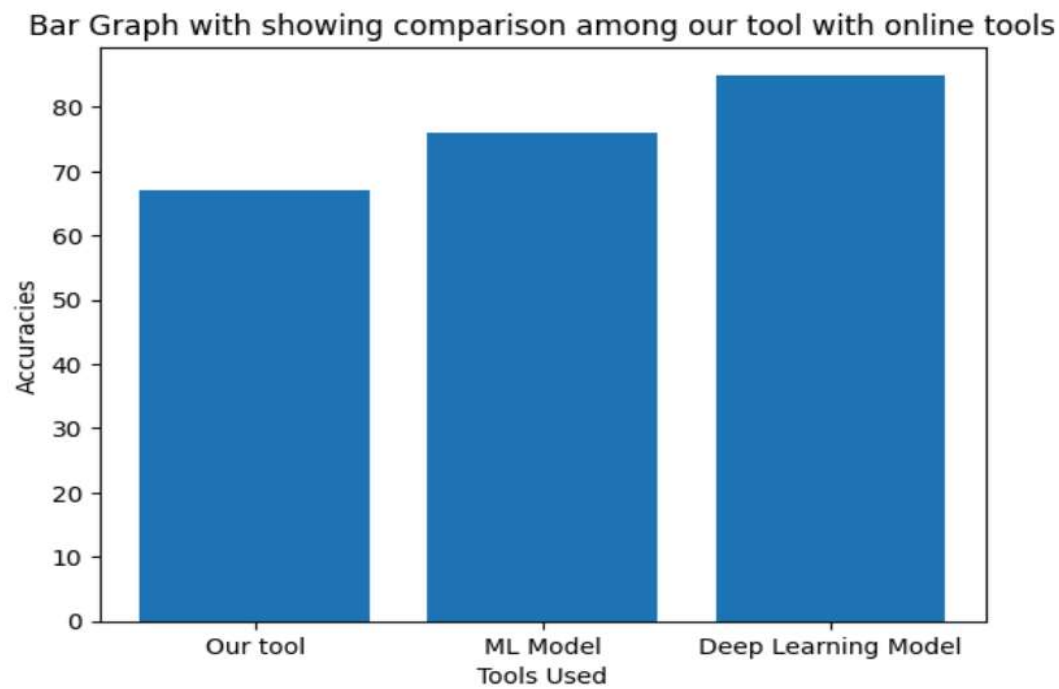
Using Machine Learning we got accuracy of around 76%.

Accuracy of Deep Learning

```
391/391 [=====] - 280s 699ms/step - loss: 0.4412 - accuracy: 0.7894 - val_loss: 0.3469 - val_accuracy: 0.8530
391/391 [=====] - 33s 84ms/step - loss: 0.3469 - accuracy: 0.8530
Test Accuracy: 0.8530399799346924
```

Using Deep Learning we got accuracy of around 85%.

Graphical Comparison



Reflections

Our sentiment analysis system achieved reasonable accuracy in predicting the sentiments of movie reviews. However, compared to the other tool our system's accuracy was slightly lower. This can be attributed to the simplicity of our model compared to the more complex models used in the other tools.

Conclusion

In conclusion, our sentiment analysis project successfully developed a system for predicting sentiments in movie reviews. While our system's accuracy was competitive, there is room for improvement, particularly by exploring more advanced machine learning and deep learning techniques. The comparison with existing tools provides valuable insights into the state-of-the-art in sentiment analysis.