ALLIANCE UNIVERSITY
Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Celebrating 25 SILVER JUBILEE years of Alliance Education

COLLEGE OF ENGINEERING AND DESIGN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING/INFORMATION & TECHNOLOGY

MACHINE LEARNING (CS 603) PROJECT

MOVIE GENRE CLASSIFICATION USING SUMMARY

Abstract

I have developed a tool to classify movies into different genres or categories using only the plot summary. A Naïve Bayes classifier with a multinomial model was used in this effort. The classification is considered a multi-category and multi-label problem since there are more than two classes (genres) in total and each individual data can be categorized into multiple classes.

Introduction

Each movie has information like title, genre plot tec. These pieces of information can be obtained from the web with ease using online databases like IMDb. After learning different machine learning techniques and their possible applications, I came up with the interesting idea of testing one of these machine learning methods to see how accurate it can predict a movie genres given the plot summary. The chosen machine learning method was the Naïve Bayes classifier. The reliability of using this method was determined by calculating the accuracy of predictions on the test data set.

I used Python as the programming language and PyCharm as the IDE. The movie list containing titles and IDs was obtained from “MovieLens Datasets”.

Methodology

This section focuses on the major steps of the classification. The algorithm starts with building the required database. It continues with loading the movies from the database into the program to train the classifier. The final step is testing a new database of movies and calculating the accuracy of the classification. The following sub-sections, elaborate on these major steps in the classification process.

- Building the database
- Feature extraction and building the corpus
- Classification
- Classification
- Testing

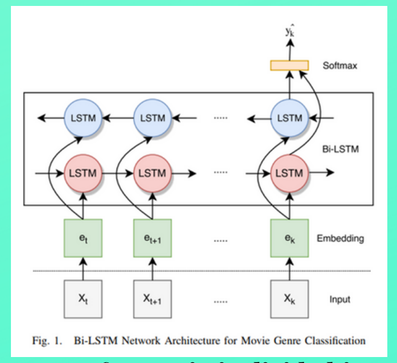
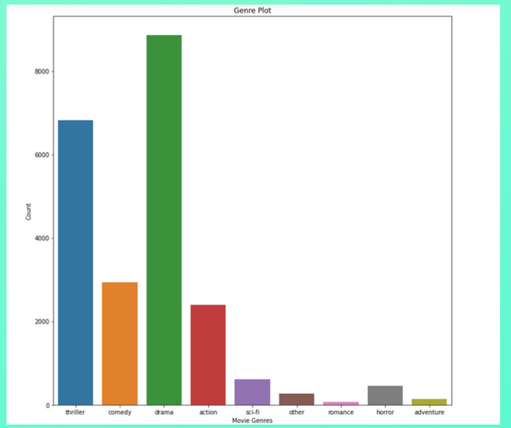
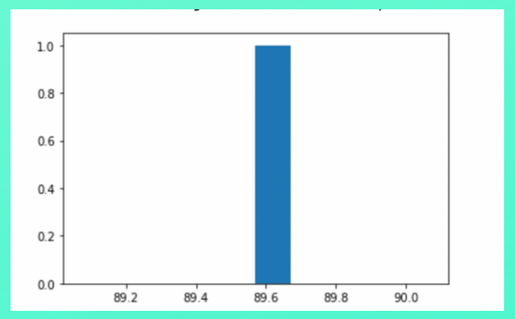


Fig. 1. Bi-LSTM Network Architecture for Movie Genre Classification

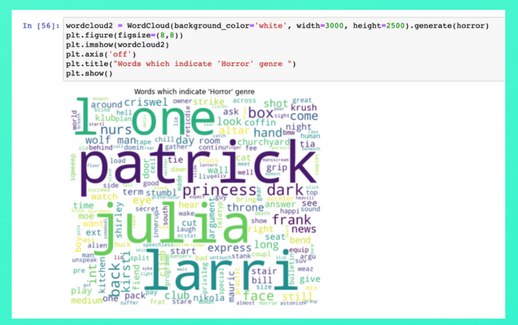
Each plot summary of a movie is divided into sentences and the genre of corresponding movie is assigned to each sentence. During training, each input (sentence) is represented as the words it includes and continuous word representations are obtained using [15]. It is useful when the limited data is used since semantic and syntactic relationship among the words are captured. We name this representation in the architecture in Fig. 1 as embedding layer. Then, the word representations are fed into the Bi-LSTM network.

Testing the system, software, tabulation, graphs





This result is plausible given the skewed dataset, which makes the Naive Bayes models to predict the popular genres more often, and to predict the less popular genres only when it has strong belief.



Results

The classification was performed on a training database consisting 26000 movie records. The test database includes 1200 movie records. The total number of genres among all the movies was 27.

Conclusion

This project explores several Machine Learning methods to predict movie genres based on plot summaries. This task is very challenging due to the ambiguity involved in the multi-label classification problem. Experiments with both Naive Bayes and GRU networks show that combining a probabilistic classifier with a probability threshold regressor works better than the k-binary transformation and the rank methods for the multi-label classification problem. Using a GRU network as the probabilistic classifier in this approach, the model achieves impressive performance with a Jaccard Index of 50.0%, F-score of 0.56 and hit rate of 80.5%. Finally, the data is highly skewed, making the model biased towards popular genres such as drama or comedy. Dealing with this issue would further improve the performance.

References

- [1] Imdb data. ftp://ftp.fu-berlin.de/pub/misc/movies/database/.
- [2] Xgboost python package. http://xgboost.readthedocs.io/en/latest/build.html.
- Pretrained google news vectors. https://code.google.com/archive/p/word2vec/, 2013.
- Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

Submitted by:

Payal (19030141CSE062)
Section – "A"
B. Tech-CSE (2019 Batch)