INDIAN STATISTICAL INSTITUTE, KOLKATA

"PREDICTING QUALITY OF MOVIE FROM METADATA AND PLOT SUMMARY"

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE

BY

Ashwani Patel (CS1916)

UNDER THE GUIDANCE OF
Dr. Debapriyo Majumdar
Assistant Professor
Computer Vision and Pattern Recognition Unit, Kolkata
Computer and Communication Sciences Division Kolkata Headquarters

CERTIFICATE

This is to certify that the dissertation entitled **"Predicting Quality of Movie from Metadata and Plot Summary"** submitted by **Ashwani Patel** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Debapriyo Majumdar
Assistant Professor,
Computer Vision and Pattern Recognition Unit, Kolkata,
Indian Statistical Institute,
Kolkata-700108, India.

Acknowledgements

Foremost, I would like to express my highest gratitude to my advisor, Debapriyo Majumdar, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

He also gave the theoretical and practical knowledge on the topics which was the basis of my dissertation. His guidance helped me in all the time of completion and writing of this dissertation.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.

Ashwani Patel
Indian Statistical Institute
Kolkata - 700108 , India.

ABSTRACT

The worldwide theatrical market had a box office of US $42.2 billion in 2019. In recent years it has been seen that it is growing even more and more, as a consequence urge to predict the success of the movie has increased. To inspect this issue various methodology has been proposed some of which rely on reviews and the trailer when most or all the budget of the movie has been enervated. To overcome this some recent papers have also used the plot summary of the movie to classify the movie as successful or not successful. In this work, we will try to predict the quality of the movie not only by using the plot summary but other metadata of the movie too. We have used the CMU corpus for the movie metadata and the IMDB database for the ratings. We have experimented with LSTM, ELMO, Sentiment analysis, and Transformer based architecture like BERT. We have experimented with all these and combined them to come up with a feature engineering architecture suitable for our task.

Contents

Introduction

Predictive models in the early stages of movie production is effective to minimize investments in flops. Forecasting the success of a movie has intrigued many scholars and industry leaders as a difficult and challenging problem.

As every year a substantial amount of money is being invested in the film industry worldwide. This huge investment in the movie business is a high-risk venture. These risks can be significantly reduced if we can predict the success of the movie at the early stage.

Several attempts have been made earlier for prediction. Researchers have used the data mining technique to develop a mathematical model based on several attributes, Analysis of the sentiments of the tweets, predicting the success from the plot of the movie, and much more.

This task is extremely challenging and to address this issue we will use CMU movie corpus[1] and IMDb dataset[2]. While CMU corpus have the plot summary of the movie and other metadata related to the movie, IMDb have the ratings of that movie which will be utilized as the scoring system of our model.

In this work, we have used deep learning models to classify a movie as a successful or non-successful movie using the movie's textual summary and other metadata of the movie.

In this work our major steps are:

- Collecting the data from CMU datasets and IMDb.
- Data cleaning and pre-processing.
- Using contextual embeddings like ELMO and BERT.
- Incorporating the sentiment score of the plot summary in the architecture.
- Using the other metadata of the movie such as actor/actress, release year and country of the origin.
- Evaluating and analysis of the different models such as CNN and Bi-Directional LSTM on the data.

[1] http://www.cs.cmu.edu/ ark/personas/
[2] https://www.imdb.com/interfaces/

Related Work

The prediction of the success of the movie has been actively researched. Some researchers predicted a movie's success on the basis of the success record of actors, release season, award history and cost of a movie, etc. Few works directly use sentiment analysis results on the data collected from social media platforms such as Twitter for prediction. Here are some of the related work which has contributed to this area.

2.1 Prediction of Movie Success using Sentiment Analysis of Tweets

One of the early works in this field was carried out with the help of sentiment analysis of the tweets. A research paper [Jain, 2013] in which the author attempted to predict the movie popularity from the tweets about the movie.

2.1.1 Methodology

They used sentiment analysis results of tweets sent during the movie release to predict the box office success of the movie.

Data

The authors download an existing twitter data set and retrieves recent tweets via Twitter API, which included Tweet Id, Username of the person who tweeted Tweet text, and Time of tweet. They have used the dataset of years 2009 and 2012.

In the 2009 dataset, they randomly choose 24 movies (8 hit, 8 flop, 8 average) as the training set (4800 tweets in total). The other 6 movies are used as the test set. All 2012 data (8 movies, 200 tweets each) are used as another test set.

Sentiment Analysis

To create the training set and data for evaluation, they label the tweets based on the sentiment they carry. Positive (positive review of the movie), Negative (negative review of the movie), Neutral (Mixed positive and negative reviews) and Irrelevant (Not on-topic e.g. spam).

2.1.2 Prediction

The authors used the statistics of tweets labels to classify the movies as hit/flop/average.

They have used a simple metric called PT(Positive tweets)-NT(Negative tweets) ratio to predict the movie categories of the success.
PT-NT Ratio (more than or equal to 5): Movie is hit

PT-NT Ratio (less than 5 but more than 1.5): Movie would do Average business
PT-NT Ratio (less than 1.5): Movie is Flop

The authors have used Lingpipe sentiment analyzer to estimate the sentiment of the tweets. They predicted 8 movies which just released using PT/NT ratio, and the results are shown in the table.

Movie	PT/NT Ratio	Prediction	Budget	Sales	Days since Release	Prediction Confidence
Wreck-it Ralph	24.5	Hit	$120 M	$158 M	32	Yes
Skyfall	4.08	Hit	$200 M	$246 M	25	Yes
Twilight Saga: BD II	47.53	Super hit	$120 M	$255 M	18	Yes
Rise of the Guardians	30.89	Hit	$145 M	$49 M	18	May be
Red Dawn	17.7	Hit	$65 M	$32 M	13	Yes
Miami Connection	20	Hit	$1 M	**	25	Can't say
Citadel	#####	N/A	N/A	***	25	Can't say
Nature calls	1.5	Avg	N/A	N/A	25	Can't say

Figure 2.1: Prediction of success

In prediction, it came out that 5 movies to be hit and one to be super hit, one to be average and one's success rate could not be determined due to its data unavailability. Comparing these prediction results with box office results to date it has been found that prediction to be exact for four cases, for a case it is on the border line between hit and average and for one data has not been found to check prediction confidence.

2.2 Movie success prediction using Data Mining

In this project [Ahmad et al., 2017], they developed a mathematical model to predict the success and failure of the upcoming movies based on several attributes. Some of the criteria in calculating movie success included budget, actors, director, producer, set locations, story writer, movie release day, competing for movie releases at the same time, music, release location, and target audience.

2.2.1 Methodology

The core idea was to developed a mathematical model to predict the success and failure of the upcoming movies based on several attributes, considering the factors movie name, year of release, genres, directors, producers, languages.

- Find x^2 (chi-square) analysis between movie actors, genres, rating.
- Find the correlations from the respective x^2 analyses above.
- Predict success rating from the correlations between various movie criteria.

2.3 Prediction of a Movie's Success From Plot Summaries Using Deep Learning Models

In this recent research, paper [Kim et al., 2019] deep-learning based approach has been used to classify movie popularity and quality labels using the movie textual summary data.

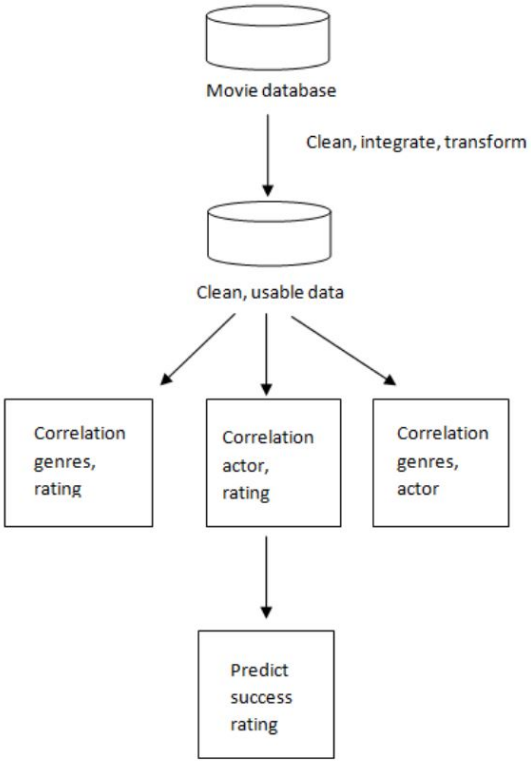

Figure 2.2: Correlation flow

They have used the CMU Movie Summary Corpus (Bamman et al., 2013) data which contains crowd-sourced summaries from the real users. The success of a movie is assessed with the review scores of Rotten Tomatoes, an American review-aggregation website.

2.3.1 Methodology

The scoring system utilizes two scores: the tomato-meter and the audience score. The tomato-meter score is estimated by hundreds of film and television critics, appraising the artistic quality of a movie. The audience score is computed by the collective scores from regular movie viewers.

The steps involved:

- Preprocessing the plot summary of the movies.
- Incorporate sentiment analysis in plot summary in predicting a movie's success.
- ELMO embedding in plot summary.
- To evaluate merged deep learning models (CNN and residual LSTM) in predicting a movie's success.

Predictions

Authors have used the F1 score as the primary scoring system for comparison as it is the harmonic mean of recall and precision. Their evaluation result of the critic score of CNN for predicting 'well-made' movies achieved the F1 score of 0.70. The LSTM model achieved the best performance in predicting 'not well-made' movies, with an F1 score of 0.65.

Data and Preprocessing

To evaluate our approach we will use CMU Movie Summary Corpus[1] which contains movie plot summaries and their metadata such as genre, release date, cast, character traits, etc. We will also use IMDb data for our scoring system.

3.1 Data

3.1.1 CMU

This dataset [Bamman et al., 2013] contains 42,306 movie plot summaries extracted from Wikipedia + aligned metadata extracted from Freebase, including:

- Movie box office revenue, genre, release date, runtime, and language.
- Character names and aligned information about the actors who portray them, including gender and estimated age at the time of the movie's release.

Movie metadata

This contains information about the movie like Wikipedia ID, Movie name, release date, country, genre, etc. Here is an example of metadata of a movie *Indiana Jones and the Raiders of the Lost Ark*.

Wikipedia movie ID	5416
Freebase movie ID	/m/0f4yh
Movie name	Indiana Jones and the Raiders of the Lost Ark
Movie release date	1981-06-12
Movie box office revenue	389925971
Movie runtime	115.0
Movie languages	Arabic Language, Nepali Language, Spanish Language, English Language, German Language
Movie countries	United States of America
Movie genres	Adventure, Costume Adventure, Action/Adventure, Action

Table 3.1: Movie metadata

Character metadata

This dataset contains the characters in the movies with name, gender, height, DOB, etc. Here is an example of character metadata of the same movie *Indiana Jones and the Raiders of the Lost Ark*.

[1] http://www.cs.cmu.edu/ ark/personas/

Wikipedia movie ID	Freebase Movie ID	Character Name	Actor DOB	Actor gender	Actor height	Actor Name
54166	/m/0f4yh	Dr Marcus Brody	1992-05-31	M	1.816	Debholm Elliott
54166	/m/0f4yh	Simon Katanga	1949-10-20	M	1.87	George Harris
54166	/m/0f4yh	Dr. René Belloq	1943-01-18	M	1.77	Paul Freeman
54166	/m/0f4yh	Major Arnold Toht	1935-09-28	M		Ronald Lacey
54166	/m/0f4yh	Indiana Jones	1942-07-13	M	1.85	Harrison Ford
54166	/m/0f4yh	Marion Ravenwood	1951-10-05	F	1.7	Karen Allen

Table 3.2: Character metadata of CMU

Plot Summary

This contains the plot summary of the movie. Here is the example of the plot summary of movie *Indiana Jones and the Raiders of the Lost Ark*.

”In 1936, archaeologist Indiana Jones braves an ancient Peruvian temple filled with booby traps to retrieve a golden idol. Upon fleeing the temple, Indiana is confronted by rival archaeologist René Belloq and the indigenous Hovitos.

(.....)

Back in Washington, D.C., the Army intelligence agents tell a suspicious Indiana and Brody that the Ark ”is someplace safe” to be studied by ”top men”. In reality, the Ark is sealed in a wooden crate labeled ”top secret” and stored in a giant government warehouse filled with countless similar crates.”

3.1.2 IMDb

Our other dataset has been taken from IMDb. It is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew, and personal biographies, plot summaries, trivia, ratings, and fan, and critical reviews. This data can be obtained from IMDb dataset page[2].

Ratings

In this dataset *title.ratings.tsv.gz* we have average rating and the number of votes casted by the user along with title Id as shown in Figure 3.1. This rating will be the scoring system of our task.

Release Year

For original movie title and release year of the movie we have used another file *title.basics.tsv.gz* from the imdb datasets as shown in Figure 3.2.

3.2 Preprocessing

After gathering all the required data of our interest it is time to preprocess and clean the data. We will drop some of the attributes and extract the data after the cleaning.

[2]https://www.imdb.com/interfaces/

	titleId	Rating	numVotes
0	tt0000001	5.7	1684
1	tt0000002	6.0	207
2	tt0000003	6.5	1421
3	tt0000004	6.1	121
4	tt0000005	6.1	2221

Figure 3.1: IMDb Ratings

	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes
0	tt0000001	short	Carmencita	Carmencita	0	1894	\N	1
1	tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5
2	tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4
3	tt0000004	short	Un bon bock	Un bon bock	0	1892	\N	12
4	tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	\N	1

Figure 3.2: IMDb movie release year

3.2.1 CMU

We have already shown the movie metadata of CMU in Table 3.1. We will drop Freebase movie ID from the dataframe.

For characters of the movie, we have character metadata of CMU which was shown in Table 3.2. In this, we have made the list of all the characters in the movie along with the Wikipedia movie ID for the merging with movie metadata.

We had a separate text file of the movie plot summary, one sample of the movie plot summary has already been shown in Section 3.1.1 under plot summary. This text file also have the Wikipedia movie ID.

Now we have merged all the dataframe of CMU with the help of Wikipedia movie ID which will serve as a primary key for all three datasets. After merging all the dataframes our final data contains the following information as shown in the figure.

	Wikipedia Movie ID	Movie Actor	Movie Name	Release Date	Revenue	Runtime
0	261236	[Gudrun Landgrebe, Mathieu Carrière, Hanns Zis...	A Woman in Flames	1983	NaN	106.0
1	2238856	[Hector Elias, John Hawkes, Miranda July, Mile...	Me and You and Everyone We Know	2005-01	8012838.0	91.0
2	18296435	[Müjde Ar]	Aaah Belinda	1986	NaN	NaN
3	32456683	[Erwin Geschonneck]	Die Fahne von Kriwoj Rog	1967	NaN	108.0
4	156558	[Taraji P. Henson, Tyrese Gibson, AlexSandra W...	Baby Boy	2001-06-27	29381649.0	123.0

Figure 3.3: CMU Dataframe (1)

Runtime	Languages	Countries	Genres	Plot Summary
106.0	German Language	Germany	[Drama]	Eva, an upper class housewife, becomes frustra...
91.0	English Language	United Kingdom	[Romantic comedy, Indie, Comedy of manners, Co...	The structure of the film consists of several ...
NaN	Turkish Language	Turkey	[Comedy]	Serap, a young actress with a strong, lively p...
108.0	German Language	German Democratic Republic	[]	Otto Brosowski, a communist miner, writes to t...
123.0	English Language	United States of America	[Crime Fiction, Drama, Coming of age]	A young 20-year-old named Jody lives with his...

Figure 3.4: CMU Dataframe (2)

3.2.2 IMDb

In IMDb, we have two dataframe one is of movie rating and the other of release year as shown in Figure 3.1 and Figure 3.2 respectively. In the rating dataframe we have title ID, average ratings, and a number of votes. In the year dataframe we will keep only title ID, Movie name, and release year and will drop the rest of the redundant field which is already in the CMU dataset.

Here title ID will serve as primary in merging the two dataframes. After merging all the dataframes our final data contains the following information as shown in figure

	titleId	Movie Name	Release Year	Rating	numVotes
0	tt0000001	carmencita	1894	5.7	1684
1	tt0000002	le clown et ses chiens	1892	6.0	207
2	tt0000003	pauvre pierrot	1892	6.5	1421
3	tt0000004	un bon bock	1892	6.1	121
4	tt0000005	blacksmith scene	1893	6.1	2221

Figure 3.5: IMDb Dataframe

3.2.3 Merging CMU and IMDb

Finally, we have merged the CMU and IMDb dataframe. Here we will merge both the dataframe on the movie title and release year.

After merging the dataframe we have the data of a total of 29339 movies along with their metadata which contains actor, language, genre, country, plot summary, rating, release year, runtime, language, etc.

	Movie Actor	Movie Name	Revenue	Runtime	Languages	Countries
0	[Gudrun Landgrebe, Mathieu Carrière, Hanns Zis...	a woman in flames	NaN	106.0	German Language	Germany
1	[Hector Elias, John Hawkes, Miranda July, Mile...	me and you and everyone we know	8012838.0	91.0	English Language	United Kingdom
2	[Erwin Geschonneck]	die fahne von kriwoj rog	NaN	108.0	German Language	German Democratic Republic
3	[Taraji P. Henson, Tyrese Gibson, AlexSandra W...	baby boy	29381649.0	123.0	English Language	United States of America
4	[Girija, Mohanlal, Jagadish, Sukumari, Mukesh,...	vandanam	NaN	168.0	Malayalam Language	India

Figure 3.6: Final Dataframe (1)

Countries	Genres	Plot Summary	Release Year	Rating	numVotes
Germany	[Drama]	Eva, an upper class housewife, becomes frustra...	1983	6.1	524
United Kingdom	[Romantic comedy, Indie, Comedy of manners, Co...	The structure of the film consists of several ...	2005	7.3	35215
German Democratic Republic	[]	Otto Brosowski, a communist miner, writes to t...	1967	8.1	14
United States of America	[Crime Fiction, Drama, Coming of age]	A young 20-year-old named Jody lives with his...	2001	6.4	13064
India	[Action]	Professor Kurian Fernandez , a convict escapes...	1989	7.9	966

Figure 3.7: Final Dataframe (2)

Approach

4.1 Our Approach

Now we have all the required data, it's time to use it with feature engineering architecture to predict the quality of the movies.

4.1.1 Embedding

Embedding is used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.

We have used word embedding for the representation of words for text analysis. When the list of sentences representing a movie summary is given as input, the module creates its corresponding word embedding vectors.

We have not used traditional word embeddings such as Glove [Pennington et al., 2014] and Word2Vec [Mikolov et al., 2013] which produces a fix vector for each word irrespective of their context. Instead, we have used contextualized word embedding techniques that can generate different word vectors depending on the context.

ELMO

ELMO [Peters et al., 2018] is a popular contextualized embedding method, which uses two bidirectional LSTM networks for constructing the vector. This biLSTM model has two layers stacked together and each layer has 2 passes, forward pass, and backward pass.

It uses a character-level CNN to represent words of a text string into raw word vectors. This raw word vector act as inputs to the first layer of biLSTM. The input to the architecture is computed from characters rather than words, it captures the internal arrangement of the word. For example, the biLSTM will be able to figure out that terms like beauty and beautiful are related at some level without even looking at the context they often appear in.

In this work, we utilized the TensorFlow Hub[1] implementation to represent the word vector.

BERT

BERT [Devlin et al., 2018], published by Google, is a new way to obtain pre-trained language model word representation.

[1] https://tfhub.dev/google/elmo/2

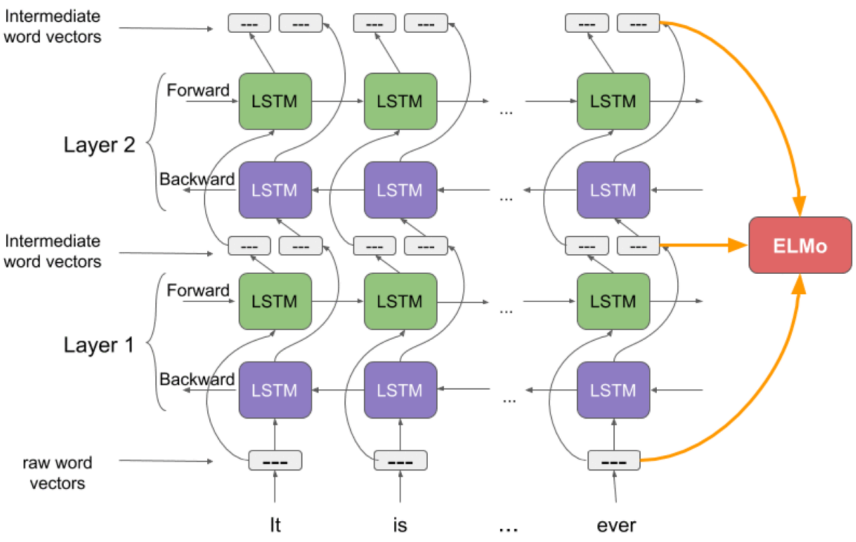

Figure 4.1: ELMO Architecture

BERT uses the blocks of the encoders of the Transformer in a novel way and does not use the decoder stack. The masked tokens (hiding the tokens to predict) are in the attention layers of the encoder. As such, it does not have a masked multi-head attention sub-layer. BERT goes further and states that a masked multi-head attention layer that masks the rest of the sequence impedes the attention process.

We have used TensorFlow's pre-trained BERT[2] which has 12 hidden layers, 12 attention heads, and a hidden size of 768. We then fine-tuned the weight for ELMO embedding to gain better performance for the classification task.

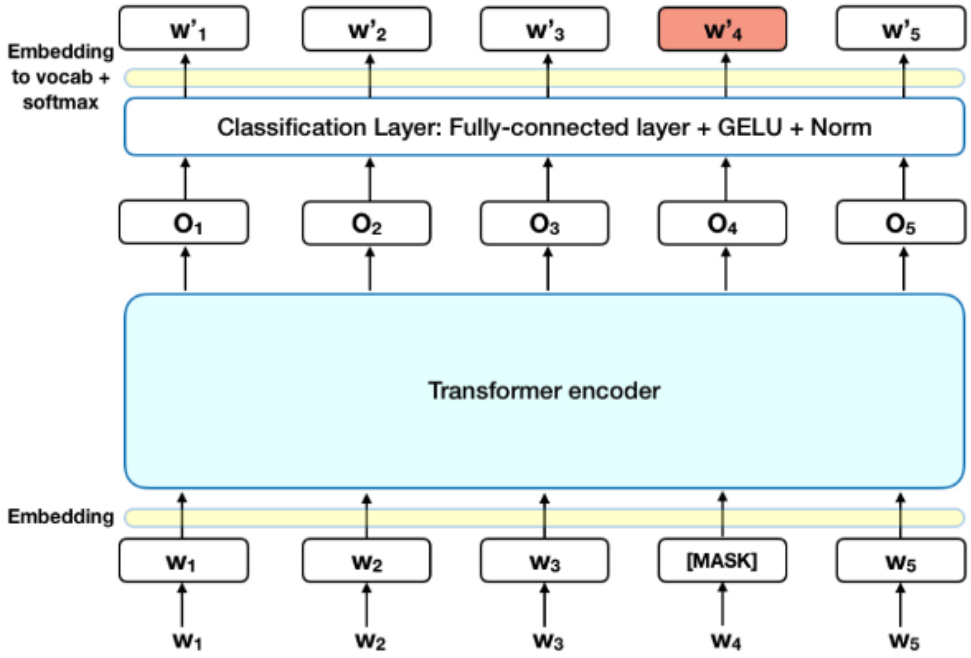

Figure 4.2: Bert Architecture

[2]https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

4.1.2 Sentiment Analysis

Sentiment Analysis (or opinion mining) which is a sub-field of Natural Language Processing is a technique used to determine whether the given statement is positive, negative, or neutral. This technique is actively used in many applications such as in business in which helps in understanding the overall opinion of the customer.

In this part, we have examined the sentiments of the plot summary of the movie. We have passed the list of sentences to the sentiment analyzer.

For extracting the sentiment score of each sentence, we have used the NLTK's Vader sentiment analyzer [Hutto and Gilbert, 2014] for each sentence. As the number of sentences in each plot summary of the movie was not the same we need to set a maximum number of sentences in each summary. By examining we found out the maximum and minimum number of sentences in the plot summary was 218 and 1 respectively.

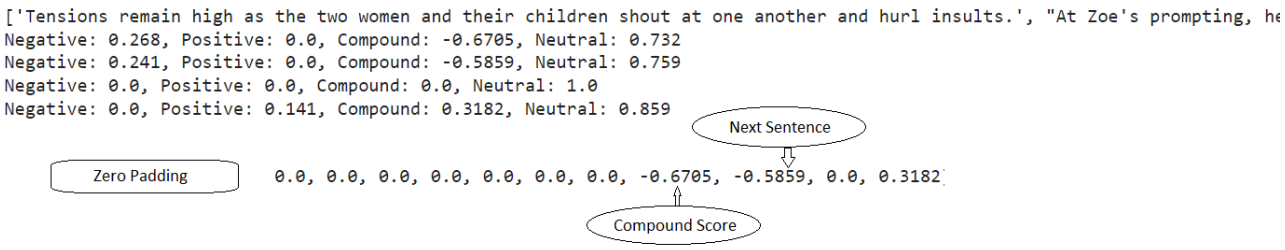

Figure 4.3: Sentiment Analysis

The VADER module computes 4 scores but as we can see in Figure 4.3 that we have used only compound score ranging from -1 (most negative) to 1 (most positive). Also, we can see that a plot summary shorter than 218 has been pre-padded with 0's as the conclusive statement of the review is usually located at the end of the story. This will help in feeding these scores to the LSTM deep learning model which better remembers the recent input.

4.1.3 Multilabel Binarizer

Unlike One-hot Encoder, Multilabelbinarizer allows you to encode multiple labels per instance. It does not only have lesser dimensions but is also computationally efficient. We have used sklearns Multilabelbinarizer to achieve this task.

We have used only 3 lead actors/actresses due to resource limitations. We have also included the release year and country of origin of that movie. We have passed this list to Multilabelbinarizer to achieve the encoding.

Here's an example of how multilable binarizer works:

In Figure 4.4 we can see that the 2D array passed to the multilabel binarizer transformed it into 0 and 1. If we pay heed to the common keyword in the array we can see that it is set in both rows of the transformed output.

4.1.4 Classification Models

Now we got all the inputs for our model architecture. We will experiment with these inputs one by one and gradually we will move to the architecture that is performing better.

```
from sklearn.preprocessing import MultiLabelBinarizer

y = [['Raj', 'Penny', 'Amit'],
     ['Amy', 'Raj', 'Sheldon'],
     ['Sheldon', 'Penny', 'Leonard']]

one_hot = MultiLabelBinarizer()
print(one_hot.fit_transform(y))
print(one_hot.classes_)
```

```
[[1 0 0 1 1 0]
 [0 1 0 0 1 1]
 [0 0 1 1 0 1]]
['Amit' 'Amy' 'Leonard' 'Penny' 'Raj' 'Sheldon']
```

Figure 4.4: Multilabel Binarizer

As this is the binary classification problem we have divided the movies into categories of Successful and Not-Successful classes. Since an IMDb rating indicates the quality of the movie, we define it as a successful movie having an IMDb rating greater than 7.5 and a non-successful movie having a rating less than 6. So every movie having a rating greater than 7.5 goes to the positive class and those who are below 6 go to the negative class.

By doing this we got a total of 14206 movies in total. There is two reasons that the number of movies is less than the total CMU movie corpus. Firstly for some movies in the CMU movie corpus, there is no IMDb rating, secondly, movies between ratings 6 and 7.5 have been filtered out.

Now out of 14206 movies, we took 2273 movies in the validation set and 2842 movies in the testing set, the rest of 9091 movies have been used for training.

ELMO with Bi-Directional LSTM

In this model, we have used only movie plot summary with ELMO embedding and Bi-Directional LSTM see Figure 4.6. As we can see in Figure 4.5 movie plot summary has been used in both embedding and sentiment score, Here we have not used the metadata of the movie, so there will be no concatenation of this. After concatenation, we have used the dense layer which is given to the last 1-dense classification layer. We employed the binary cross-entropy as the loss function, binary accuracy as our metric, and the Adam optimizer.

BERT with Bi-Directional LSTM

Here we replaced the ELMO embedding with the BERT embeddings rest of the thing will remain intact with the previous model. Here again, we have not used the metadata of the movie.

ELMO with CNN

Here we had used ELMO embedding for the plot summary with the CNN architecture for the sentiment score without using the metadata of the movie.

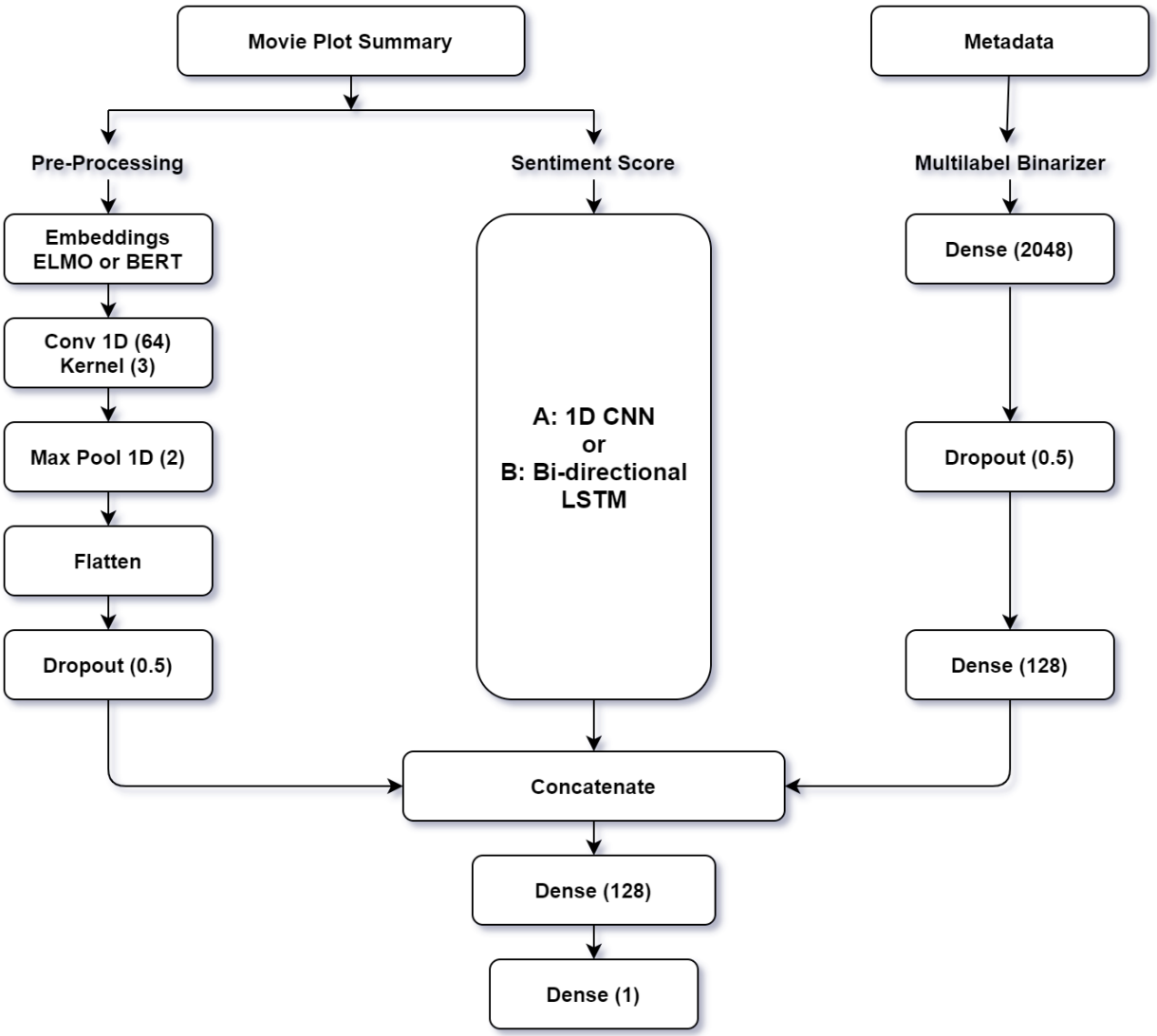

Figure 4.5: Model Architecture

BERT with CNN

Here we have replaced the ELMO embedding with the BERT embedding and used the CNN architecture for the sentiment score without using the metadata.

BERT with Bi-Directional LSTM and other metadata

After examining the results of the previous model we conclude that Bi-directional LSTM and BERT are performing better. So, in this final model, we took metadata of the movie too, after transforming it with the multilabel binarizer we concatenated it along with BERT and Bi-LSTM.

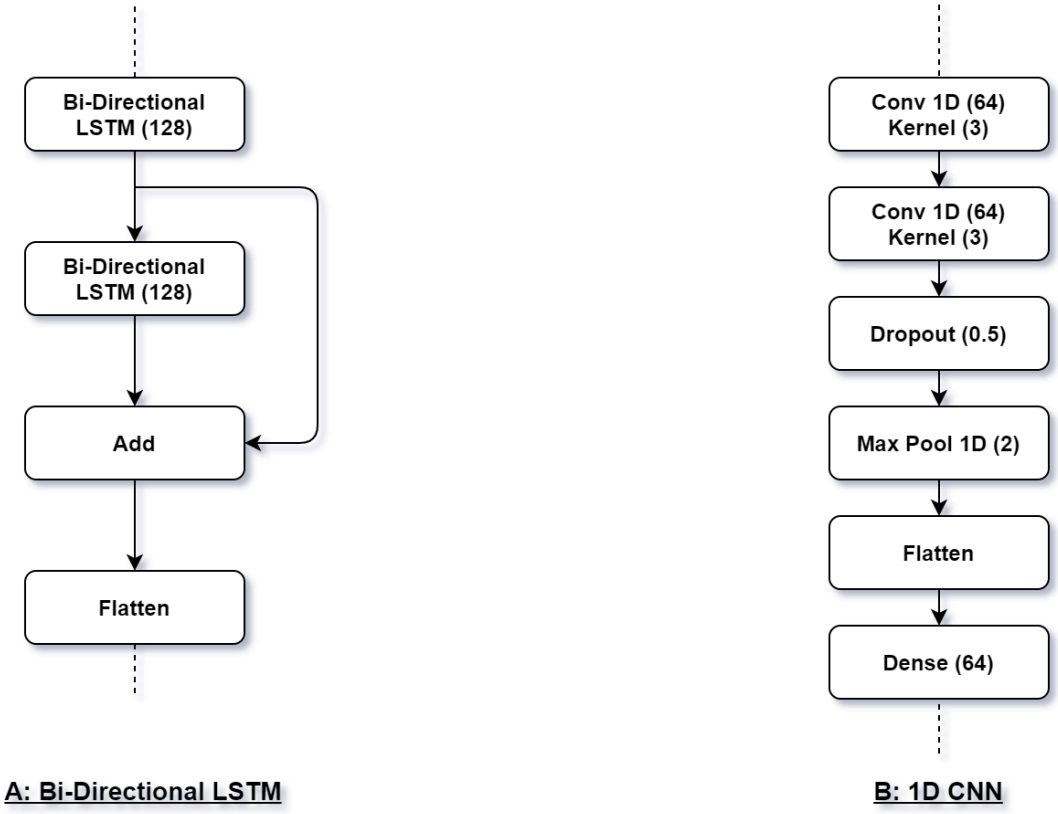

Figure 4.6: Sentiment Score module of Architecture

4.1.5 Performance Metric

We have used the F1 score as our primary metric for the performance measure. Before understanding the F1 score let us see what is the confusion matrix, precision, and recall.

Confusion Matrix

To compute the confusion matrix, we first need to have a set of predictions so that they can be compared to the actual targets. Each row in a confusion matrix represents an actual class, while each column represents a predicted class. A perfect classifier would have only true positives and true negatives, so its confusion matrix would have nonzero values only on its main diagonal (top left to bottom right).

	Negative	Positive
Negative	True Negative	False Positive
Positive	False Negative	True Positive

Precision

The confusion matrix gives you a lot of information, but sometimes we prefer a more concise metric. The accuracy of the positive predictions is called the precision of the classifier. The equation of precision is given by

$$Precision = \frac{TP}{TP + FP}$$

Recall

Precision is typically used along with another metric named recall, also called sensitivity or the true positive rate(TPR). This is the ratio of positive instances that are correctly detected by the classifier. The equation of recall is given by

$$Precision = \frac{TP}{TP + FN}$$

F1 Score

The F1 score is the harmonic mean of precision and recall. Whereas the regular mean treats all values equally, the harmonic mean gives much more weight to low values. As a result, the classifier will only get a high F1 score if both recall and precision are high. The equation of the F1 score is given by

$$\text{F1 Score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Evaluation Results

After experimenting with all the models we evaluated the classification performance of our approach. We reported the performance of each model in terms of precision, recall, and F1 score.

5.1 Results

As we can see in Table 5.1 Model 2 with ELMO embeddings and CNN with no metadata gave the F1 score of 0.54. If we change the sentiment model of the architecture with Bi-Directional LSTM as in model 1 we can see the marginal improvement in precision, recall, and F1 score.

Now in Model 3 and 4, we have replaced the embedding with BERT. We have achieved a significantly better score as compared to the first two models. We got an F1 score of 0.63 and 0.57 in models 3 and 4 respectively. Here again from the result it is evident that Bi-Directional LSTM is performing better than the CNN sentiment model.

Out of all the first four models, we can see that BERT with Bi-LSTM outperformed all the models whose F1 score is 0.63. So have picked this model for further experiment with the metadata. Now as we have discussed earlier that metadata will contain 3 lead actors of the movie along with release year and country of origin, Incorporating this information was proved to be the best model of all with an F1 score of 0.66.

	Embedding	Sentiment Model	Metadata	Precision	Recall	F1 Score
Model 1	ELMO	Bi-LSTM	No	0.66	0.48	0.55
Model 2	ELMO	CNN	No	0.63	0.47	0.54
Model 3	BERT	Bi-LSTM	No	**0.71**	**0.57**	**0.63**
Model 4	BERT	CNN	No	0.59	0.54	0.57
Model 5	**BERT**	**Bi-LSTM**	**Yes**	**0.76**	**0.59**	**0.66**

Figure 5.1: Model Results

We can see metadata is helping our model to achieve a better score. Here answer lies in the fact that some actors/actresses have a good track record of giving quality movies. We have included the country of origin, with this our model is able to extract the pattern in predicting the success rate of producing quality movies in a country.

If we compare this work with the previous related work where the scoring system was rotten tomatoes this result is quite good. Considering the fact that IMDb uses the complex rating system where the rating is crowdsourced on a scale of 1 to 10, Whereas rotten tomato score is the percentage of users who have rated the movie or show positively. The tomato meter score is estimated by hundreds of film and television critics, appraising the artistic quality of a movie.

Conclusion

In this work, we have used the CMU movie corpus and IMDb datasets to experiment with different deep learning model architectures. Since CMU's plot summaries were obtained from Wikipedia, which is crowd-sourced voluntarily data. Hence, some movie summaries may have been written by people who like or value the movie. This may complicate our task to predict the movie's success only from the summary, this is why we have included the other metadata too. We have included up to only 3 lead actors/actresses due to resource limitation and country of origin and release year of the movie.

We have used embeddings like ELMO and BERT, sentiment score from the movie plot summary, along with that we have also incorporated the metadata of the movies. We have used same architecture for the embedding line but two different lines for sentiment score which is Bi-Directional LSTM and CNN.

After analyzing the results of all stated models we gradually selected the best one which outperformed the previous one based on the F1 score. We saw that the BERT with Bi-Directional LSTM along with the metadata outperformed all with the F1 score of 0.66.

From the results, it is evident that predicting the non-successful movie performs better than predicting the successful movie. This can be very useful for the platform of OTT like (Amazon Prime, Netflix, etc.), where tons of content is available and only a small portion of it is consumed by each user.

References

[Ahmad et al., 2017] Ahmad, J., Duraisamy, P., Yousef, A., and Buckles, B. (2017). Movie success prediction using data mining. In *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–4. IEEE.

[Bamman et al., 2013] Bamman, D., O'Connor, B., and Smith, N. A. (2013). Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Hutto and Gilbert, 2014] Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8.

[Jain, 2013] Jain, V. (2013). Prediction of movie success using sentiment analysis of tweets. *The International Journal of Soft Computing and Software Engineering*, 3(3):308–313.

[Kim et al., 2019] Kim, Y. J., Cheong, Y. G., and Lee, J. H. (2019). Prediction of a movie's success from plot summaries using deep learning models. In *Proceedings of the Second Workshop on Storytelling*, pages 127–135.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.