Movie Genre Prediction

from plot description using BERT, RNN and Naive Bayes

[redacted]

Character count: 32,004 (14 standard pages)

Page count: **15**

Abstract

This study explores the application of natural language processing (NLP) techniques to automate genre classification using plot summaries from the IMDb dataset. We investigate the performance of three models: BERT (Bidirectional Encoder Representations from Transformers), a Recurrent Neural Network (RNN) using Long Short-Term Memory (LSTM), and Multinomial Naive Bayes. The dataset underwent extensive preprocessing, including deduplication, language detection, and tokenization. In addition, The Synthetic Minority Over-sampling Technique (SMOTE) was employed with the Multinomial Naive Bayes model to address class imbalances.

From the test results, the pretrained BertForSequenceClassification model demonstrated the highest accuracy at 61%. The second-best model was the RNN, achieving a test accuracy of 53.51%. The Multinomial Naive Bayes model followed closely with a test accuracy of 53.38%. Despite BERT's superior performance, it is also the most computationally intensive. Therefore, if one seeks to balance computational complexity with accuracy, the RNN and Multinomial Naive Bayes models present viable alternatives.

1. Introduction

The Internet Movie Database (IMDB) is one of the largest and most comprehensive archives of movie and tv series related information, including user reviews, plot summaries and genre classifications. Utilizing this data for NLP tasks presents several use cases. As genre tagging provides essential context about a movie or tv-series, helping audiences to make informed choices and producers target specific audiences.

Predicting movie and tv-series genres on plot summaries, can also help producers get an insight of genre trends.Traditionally, this process has relied on human annotation, which is slow, inefficient, and can be subjective (IMDB, 2024). Automating genre classification through NLP techniques can significantly enhance efficiency and consistency. This project will therefore utilize the *Genre Classification Dataset IMDb* and aims to explore different NLP methods to develop a model for accurately predicting movie or tv-series genres from plot summaries(Kaggle, 2021) .

1.1 Motivation & research question

Accurately classifying movies and tv-series by genre is fundamental to various applications within the entertainment industry, including recommendation systems and genre trends.

As stated in the intro, genre classification has relied on human annotation, but it also relies on keyword matching, where predefined keyword lists identified genres from specific words or phrases in plot summaries (IMDB, 2024).

This method fails as a limited set of keywords can’t capture the ambiguities of natural language, lack contextual understanding between words and it relies on the presence of certain words or phrases that are associated with specific genres. For instance, the word "love" might be common in romantic films, but it could also appear in dramas or comedies, without implying the same genre. Also, with a growing trend of genre-blending films, like Rom-coms, presents a significant challenge for accurate movie genre classification (Lawson,2013). This approach is therefore not reliant and results in less accurate and insightful classifications, due to the complicated nature of the natural language and subjectivity due to manual tagging.

To address these limitations and inaccuracies, this paper explores the use of various NLP techniques. Specifically, this paper investigates the effectiveness of models such as BERT (a transformer model), RNN, and Naïve Bayes in enhancing genre identification. These methods offer a scalable approach to genre classification by providing a deeper understanding of language, capturing semantic relationships, and contextual ambiguities. Also automating this process enhances efficiency, reduces bias, and adapts

to evolving language and emerging genres.
This research project therefore leverages the power of different NLP techniques to explore a more nuanced approach to movie and tv-series genre prediction using plot summaries found on the IMDb dataset.

In order to do this the project aims to answer the following research question:

Research Question:
How can movie genres be predicted from plot summaries using NLP techniques & what model is best suited to accurately predict the genre of the movie or tv-series?

2. Related Work

Naive Bayesian classification is popular due to its simplicity and effectiveness. Saritas proposed two systems, 1BC and 1BC2, to extend naive Bayesian classification to structured data. The 1BC system uses first-order features derived from structural predicates, assuming feature independence. The 1BC2 system estimates probability distributions over structured objects, making it suitable for complex data like molecules (Saritas, et al., 2019).

These approaches address the limitations of traditional naive Bayesian classifiers on structured data, demonstrating effectiveness in domains like bioinformatics and relational data mining by balancing computational efficiency and accuracy (Saritas, et al., 2019).

Another noteworthy study in this field is detailed in the paper "Movie Genre Classification from Plot Summaries Using Bidirectional LSTM." It provides a foundational comparison with logistic regression, highlighting its baseline performance for genre classification. The study emphasizes that logistic regression, when combined with TF-IDF feature extraction, can serve as an effective baseline method. The logistic regression model's straightforward implementation and the ability to handle high-dimensional data make it suitable for initial genre prediction tasks (Akbar, et al., 2022).

In the paper "Fine-tuning BERT with Bidirectional LSTM for Fine-grained Movie Reviews Sentiment Analysis," Gibson et al., explore an enhanced sentiment analysis approach by integrating a Bidirectional LSTM (BiLSTM) with the pre-trained BERT model. Their objective is to improve both binary and fine-grained sentiment analysis (SA) of movie reviews. The authors utilize benchmark datasets for their analysis and employ two techniques, SMOTE and NLPAUG, to address class imbalance and enhance model generalization in fine-grained classification tasks.

Their approach shows competitive performance, achieving 59.48% accuracy in five-class classification on the SST-5 dataset, surpassing the performance of several leading models (Nikhata, et al., 2023). This

work demonstrates the effectiveness of combining BERT with BiLSTM for enhanced sentiment analysis, providing a robust framework for both binary and fine-grained sentiment classification in the context of movie reviews.

3. Conceptual Framework

To conduct this study, three models have been chosen. Bidirectional Encoder Representations from Transformers (BERT) has been selected due to its strong performance in natural language understanding. It is particularly effective in capturing contextual relationships within text, making it a powerful tool for processing and understanding movie plots descriptions, which we used to generate genre prediction. Secondly, Naive Bayes is included for its simplicity and effectiveness in text classification tasks. Despite its straightforward nature, it often performs well with text data by making probabilistic predictions based on the frequency of words. Lastly, a Recurrent Neural Network with Long Short Term Memory features (RNN-LSTM) has been included due to its strength in handling sequential data and capturing temporal dependencies. RNN-LSTMs are particularly well-suited for tasks where the order of words significantly impacts the meaning, such as movie plot descriptions. By maintaining a form of memory through its hidden states, an RNN-LSTM can understand and predict sequences in text data, making it an excellent addition to our study for generating genre predictions based on movie plots

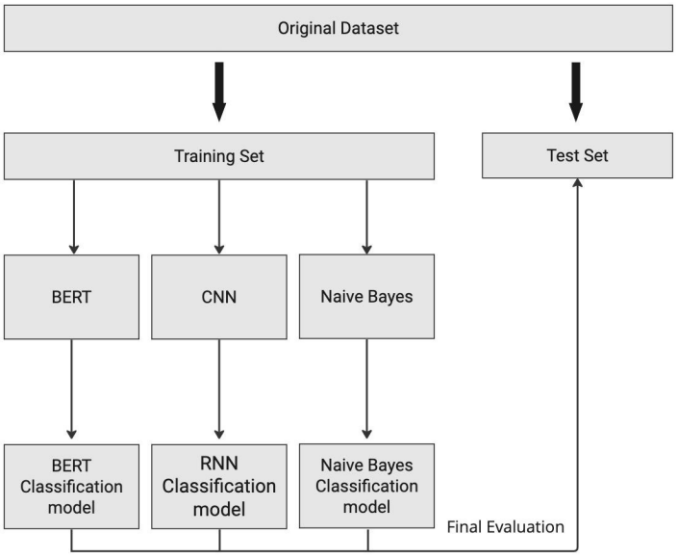


Figure 1: Conceptual Framework

4. Methodology

4.1 Dataset description

This paper utilizes the ’Genre Classification Dataset IMDb’ dataset. This open dataset was obtained from Kaggle.com (Kaggle, 2021). Kaggle is an online platform facilitating data science exploration and advancement. It offers a comprehensive repository of publicly accessible datasets encompassing diverse domains. The ’Genre Classification Dataset IMDb’ dataset contains information about films. We utilized both ‘train_data.txt’ and ‘test_data_solution.txt’ accordingly to train and test our models. The training data for this project is stored in the text file 'train_data.txt'. The files adhere to a tabular format with three columns: Title, Genre, and Description. The training dataset contains 54,214 rows, while the test one has 54,200 rows. While most of the values in the ‘Description’ column contain a summary of the movie plot, some contain short notes about movie production details, time stamps and names of chapters/episodes which is not ideal for our aim.

	Title	Genre	Description	Length
1	Oscar et la dame rose (2009)	drama	Listening in to a conversation between his do...	546
2	Cupid (1997)	thriller	A brother and sister with a past incestuous r...	184
3	Young, Wild and Wonderful (1980)	adult	As the bus empties the students for their fie...	650
4	The Secret Sin (1915)	drama	To help their unemployed father make ends mee...	1082
5	The Unrecovered (2007)	drama	The film's title refers not only to the un-re...	625

Figure 2: Dataset Description

4.2 Data Preprocessing

In preparation for deploying our model, we will execute a comprehensive preprocessing procedure. Within this phase we will carry out data filtering, cleaning, tokenization, and utilize SMOTE to balance feature quantity .Through this pre-processing, we aim to optimize the quality and structure of our dataset, ensuring that it is well-suited for subsequent model training and prediction.

4.2.1 Data Filtering

Filtering data is a crucial preprocessing step in developing machine learning models, as it greatly affects their accuracy and performance (Tarun, Batth, & Kaur, 2021). As shown in Figure 2**,** movie genres datasets don’t have equal distribution among genres, which suggests potential for bias issues due to imbalanced data.

Movie Genre	Train Dataset	Test Dataset	Total	Total %
Drama	13,569	13,555	27,124	25,1%
Documentary	13,053	13,072	26,122	24,2%
Comedy	7,424	7,426	14,850	13,7 %
Short	5,054	5,058	10,112	9,4 %
Horror	2,202	2,199	4,401	4,1 %
Thriller	1,590	1,589	3,179	2,9 %
Action	1,312	1,313	2,625	2,4 %
Western	1,032	1,029	2,061	1,9 %
Reality-TV	881	880	1,761	1,6 %
Adventure	775	772	1,547	1,4 %
Family	773	770	1,543	1,4 %
Music	711	712	1,423	1,3 %
Romance	670	669	1,339	1,2 %
Sci-fi	647	645	1,292	1,2 %
Adult	589	589	1,178	1,1 %
Crime	504	505	1,009	0,9 %
Animation	496	496	992	0,9 %
Sport	429	430	859	0,8 %
Talk-Show	387	386	773	0,7 %
Fantasy	321	321	642	0,6 %
Mystery	319	317	636	0,6 %
Musical	275	273	548	0,5 %
Biography	265	263	526	0,5 %
History	243	243	486	0,4 %
Game-Show	193	191	384	0,4 %
News	180	180	360	0,3 %
War	132	132	264	0,2 %

Figure 3: Genre Distribution

The datasets were subjected to three key filtering steps, as shown in Figure 4: deduplication, Non-English Words and Short Description Detection.

Over-representation of data points by duplicate entries in the dataset might distort the model's learning process and result in overfitting (Han, Kamber, & Pei, 2011). In order to eliminate this possibility, we detected and removed all the duplicate descriptions using the drop_duplicates function from the pandas library.

Furthermore, removing descriptions written in languages other than English was essential because our objective is to identify film genres from English plot summaries. The langdetect library was employed to determine the primary language of a text through the use of probabilistic models. We developed a function to calculate the ratio of English words in each description and established a threshold of at least 60% English content.

During our dataset exploration, we discovered that many of these non-english words were just simply names or location mentioned in valuable plots. Removing all non-English words was not beneficial. Therefore, we set a threshold of 60% to balance these two factors.

Lastly, Short Description Detection is used to clean the training dataset by removing descriptions that are shorter than 130 characters. This step is used to eliminate outliers that may not provide enough information for effective machine learning model training. A total of **3,128 movie titles,** constituting **5.7%** of the dataset, were removed.

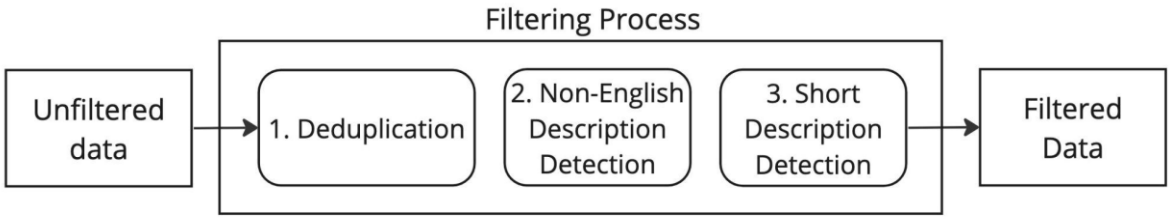


Figure 3: Filtering Process

4.2.2 Data Cleaning & Tokenization

To perform NLP tasks effectively, a robust vocabulary is needed. In order to build this vocabulary, text needs to be split into distinct tokens of meaning (Egger, R., & Gokce, E. 2022). In this section, we detail data cleaning and tokenization steps applied to our dataset using the clean_text function. This function is designed to prepare raw text data for further natural language processing (NLP) tasks by normalizing, cleaning, and tokenizing text. Firstly, we import the necessary libraries. We then create an instance of the Lancaster stemmer and load the set of English stop words. The clean_text function is defined to take a text string as input and process it through several steps. Initially, the function converts all characters in the text to lowercase to ensure uniformity. Next, hyphens are replaced with spaces to separate words that might be connected by hyphens. All numeric characters are removed from the text, followed by the removal of Twitter handles (and URLs starting with "http" or "pic."). Subsequently, the function removes all characters that are not letters or spaces. Single alphabetic characters surrounded by spaces are also removed. The text is then tokenized into individual words. During this process, stop words are removed, and the remaining words are stemmed to their base form if they have more than two characters. Finally, any repeated, leading, or trailing spaces are removed to clean up the text.

4.2.3 Smote

Developing a robust machine learning model requires addressing the class imbalance, shown in Figure 3. Class imbalance occurs when certain classes are underrepresented in the dataset, leading to biased

models that favor the majority classes and perform poorly on minority classes. To mitigate this issue, we employed the Synthetic Minority Over-sampling Technique (SMOTE).

SMOTE is an advanced oversampling method that, instead of just copying existing samples, creates synthetic samples for the minority class by interpolating between them. Furthermore, it enhances the ability of machine learning models to learn effectively from imbalanced data, improving their generalization to new, unseen data (Fernández et al., 2018).

4.3 Modelling Framework

This project will deploy and compare the performance of three distinct models, to predict movie genres from plot summaries. Firstly will we utilize a pre-trained BERT model, finetuning it on our dataset to leverage its deep contextual understanding. Secondly, will we design an RNN with LSTM architecture, training it from scratch to capture sequential dependencies in the plots. Lastly will we implement a Naive Bayes model as a baseline for comparison. By benchmarking these models against each other, we aim to identify the most effective approach for genre prediction.

4.3.1 BERT - BertForSequenceClassification

In this subchapter, we explore how BERT (Bidirectional Encoder Representations from Transformers) is used to classify movie genres based on their plot descriptions. BERT is highly effective for natural language processing tasks due to its ability to understand the context of text using a transformer-based architecture. Here, we outline the steps involved in preparing the dataset, training the model, and evaluating its performance.

To start, we created a custom dataset class to manage our text data. This class takes in movie plot descriptions, their corresponding genre labels, a tokenizer, and a maximum sequence length. The tokenizer, an integral part of the BERT model, converts textual plot descriptions into token IDs that the model can process. This setup ensures that each plot description is padded to a consistent length, making it suitable for input into the BERT model.

Model Initialization

We used a pre-trained BERT model specifically adapted for sequence classification tasks. This model, known as BertForSequenceClassification, extends the standard BERT model by adding a classification layer on top. It is initialized with pre-trained weights from the 'bert-base-uncased' variant and configured to classify texts into a number of classes equal to the unique genres in our training dataset.

To fine-tune the model on our dataset, we defined a set of training parameters. These parameters control various aspects of the training process, including the number of training epochs, batch sizes for both training and evaluation, and the frequency of logging and evaluation steps. These settings help manage the learning process, such as controlling learning rates and logging important metrics, which are crucial for monitoring the model's performance and preventing overfitting.

We employed the Trainer class from the Hugging Face transformers library to handle the training process. This class simplifies the training pipeline by managing tasks such as forward passes, gradient updates, and model evaluation. During training, the model learns to associate specific patterns in plot descriptions with the corresponding genres. The evaluation strategy we used ensured that the model's performance was periodically assessed on a validation set, providing checkpoints to avoid overfitting and to fine-tune hyperparameters effectively.

4.3.2 Multinomial Naive Bayes

The Naive Bayes classifier is a probabilistic model that utilizes Bayes' theorem to make predictions based on the likelihood of different features occurring within each class. The "naivety" approach assumes that features are independent of each other, thereby simplifying the computational demand. This algorithm has been frequently used for classification tasks, particularly in natural language processing applications.

We used the Multinomial Naive Bayes (MNB) variation for this project because it is effective for text classification. The MNB classifier works by calculating the probabilities of words (features) belonging to different genres and then predicting the genre that maximizes this probability for a given movie description.This approach is a solid option for genre prediction since it is effective and capable of handling high-dimensional data (Aggarwal & Zhai, 2012).

To preprocess the textual data specifically for our Naive Bayes model, we employed the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. Using TF-IDF, text is converted into numerical feature vectors that represent the importance of each word in the plot description as well as throughout the whole dataset. This technique improves the model's comprehension of context and semantics by emphasizing important words and minimizing the impact of common but uninformative ones (Ramos, 2003). In order to capture a wide variety of textual patterns and improve the model's ability to understand context, we vectorized the data using TF-IDF employing unigrams, bigrams, and trigrams.

Given the class imbalance in our dataset, where some genres are underrepresented, we applied the Synthetic Minority Over-sampling Technique (SMOTE), a technique we previously covered.

After training the initial Naive Bayes model, we found its accuracy unsatisfactory and took several steps to improve its performance. First, we performed hyperparameter tuning using GridSearchCV. This method systematically explores multiple parameter combinations, continuously validating to find the best settings. For this model, we focused on the "alpha" parameter, which controls smoothing, in order to enhance the model's predictive capability.

Furthermore, we employed cross-validation to assess the model's robustness. To ensure that the improvements were not only the result of specific data splits, we employed 5-fold cross-validation to assess reliability and consistency of the model.

4.3.3 Recurrent Neural Network

A third method this project tests for obtaining the best genre prediction is a Recurrent Neural Network (RNN). The RNN model utilized in this project is an LSTM (Long Short-Term Memory), which was introduced by Hochreiter and Schmidhuber in 1997 as an improvement to the standard RNN. Unlike traditional RNNs, which suffer from the limitation of only having a single hidden state and thus struggle to capture long-term dependencies while facing issues like the vanishing or exploding gradient problem. The LSTM network seeks to solve this by adding a context layer controlled by weighted gates (Jurafsky & Martin, 2023).

This allows the network through its LSTMs layers to maintain crucial context and dependencies in processing sequential text data, which is crucial for accurate classification, while producing a single output per sequence, aligning perfectly with genre labeling needs for this project. Its ability to grasp order dependence is crucial for solving intricate problems and is the reason this project hypothesizes that it will be a good model for movie genre prediction. Though its advantages in language modeling, LSTM can struggle with large volumes due to computational constraints compared to simpler models(Jurafsky & Martin, 2023).

The model employs a streamlined approach to pre-processing, beginning with tokenization using a Tokenizer configured with a vocabulary limit of 50,000 words and excluding punctuation. Padding ensures uniformity by fixing the sequence length to 250 tokens.

The model architecture employs an embedding layer which converts sequences into dense vectors, followed by a SpatialDropout1D layer to promote feature independence and robustness. The core LSTM layer, with 100 units, employs gates to manage information flow, crucial for capturing long-term dependencies. Followed by a dense layer utilizing softmax activation to make genre probability

distributions for each genre. Then another dropout layer is employed to prevent overfitting and ensure the model's ability to generalize to new data.

The model's training utilizes categorical cross-entropy loss and the Adam optimizer across six epochs, with a batch size of 32. During training, the model aims at minimizing the categorical cross-entropy loss between the predicted probability distributions and the true genre labels. Furthermore, Callback and EarlyStopping are employed to prevent overfitting.

The optimization algorithm Adam is an extension to the classic stochastic gradient descent and is chosen in this training for its ability to compute individual learning rates for its gradients by utilizing two extensions to stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) which improves performance on problems with sparse gradients and Root Mean Square Propagation (RMSProp) which adapts learning rates based on recent gradient magnitudes, making it suitable for handling noisy data(Kingma & Ba ,2015).

4.4 Evaluation metrics

To evaluate the performance of our genre prediction models, we utilize the evaluation metrics Accuracy, F1-score and its intrinsic measures Precision and Recall.

Accuracy measures the proportion of correctly predicted genres to the total number of predictions. Precision and Recall measures the models' ability to identify positive instances, with Precision measuring the ratio of true positive predictions to all predicted positives, and recall measuring the ratio of true positive predictions to actual positives in a class.
Precision and recall are interconnected, where improving one often comes at the expense of the other. Maximizing the F1 score aims to strike a balance between them and serves as a harmonic means to optimize both measures (Jurafsky & Martin, 2023).

In addition, we also looked at the *Loss* measure for the RNN model, which looks at the discrepancy between the predicted probabilities and the actual genre labels in the dataset.
So it evaluates the alignment of predictions with actual outcomes, with the aim of minimizing this measure through model training to improve accuracy and performance in predicting class labels.

5 Results

The three models' effectiveness in predicting movie genres from plot summaries provides important insights into their strengths and weaknesses. Below is a summary of the test accuracy for all the models,

with precision, recall, and F1-score based on the weighted average, taking into consideration the large number of genres:

Model	Precision	Recall	F1-Score	Accuracy
Multinomial Naive Bayes	0.52	0.53	0.53	0.53
Recurrent Neural Network	0.51	0.54	0.51	0.54
BERT	0.58	0.61	0.59	0.61

Figure 5: Model's Test Accuracy Comparison

From the test results of our models, it is evident that the pretrained BertForSequenceClassification performed the best, achieving an accuracy of 61%. The second-best model was the RNN, with a test accuracy of 53.51%. Lastly, the Multinomial Naive Bayes model achieved slightly worst test accuracy of 53.38%.

	Movie Genre	Train Dataset	Test Dataset	Total	Total %	F1 Score (BERT)	F1 Score (NB)	F1 Score (RNN)
Top 2	Drama	13,569	13,555	27,124	25,1%	66%	59%	61%
	Documentary	13,053	13,072	26,122	24,2%	78%	72%	73%
Bottom 2	News	180	180	360	0,3 %	0%	8%	0%
	War	132	132	264	0,2 %	24%	16%	6%

Figure 6: Impact of sample size in prediction

As shown in Figure 6, it is evident that genres with higher support inputs have better classification predictions. For example, in the BertForSequenceClassification model, the drama and documentary genres have F1 scores of 0.78 and 0.66, respectively. In contrast, genres with the lowest support inputs have some of the lowest scores, such as the news and war genres, which have F1 scores of 0.00 and 0.23, respectively, in the BertForSequenceClassification model.

Additionally, we identified potential overfitting issues, which we attempted to handle using different approaches. For the Naive Bayes model, we utilized smoothing and cross-validation. For the BERT and RNN models, we applied regularization and fine-tuning.

6. Discussion

6.1 Comparison and model complexity

As stated in the last section, the transformer-based model BERT outperformed the other models on all metrics: accuracy, precision, recall and F1-score - with scores of 0.61, 0.58, 0.61 and 0.59 respectively (Figure 5).

The fact that BERT is a better model is no surprise as the pre-training allows it to capture complex contextual understanding in plot descriptions. Yet this strength also introduces some limitations. Another factor to consider before deciding which model is best for the purpose is the overall performance and complexity of the models. Naive Bayes offers a simple, interpretable approach but struggles with real-world text dependencies. LSTMs excel at capturing sequential information within the text, improving accuracy but demanding more training data and computing than simpler models. Lastly, BERT is a transformer that possesses the advantage of being pre-trained. This allows this model to understand language patterns and relationships between words, allowing it to understand complex relationships within movie descriptions but necessitates significant computational power and potential fine-tuning.

So even though the BERT model outperforms the other model, thanks to its inherent features and pre-training, its running time of approx. 1,5 (Appendix 14) is presenting a challenge. While BERT has the best output, its lengthy runtime may hinder real-time applications. Naive Bayes and RNN-LSTM, although potentially less accurate, offer faster execution, making them more suitable for time-sensitive tasks. Therefore, the choice among these models depends on the trade-off between runtime and performance requirements.

6.2 Error Analysis

Despite the rigorous methods and techniques employed, the accuracy achieved by our models is not perfect. The purpose of this part is to investigate the causes of inaccuracies in our genre prediction models and comprehend the difficulties in reaching more accurate predictions.

Our models, including BERT, Naive Bayes, and RNN, faced several challenges that limited their performance. The nature and quality of the dataset proved to be a significant factor influencing our models' performance. In the course of our analysis, we discovered that certain plot summaries contained information beyond just the movie's storyline, such as production details, episode titles, and timestamps. This unnecessary information has the potential to confuse models and produce inaccurate genre

predictions. We attempted to identify and eliminate irrelevant information by detecting patterns, but it proved to be highly unpredictable.

Moreover, the subjective nature of genre classification presents a challenge. Genres are not always mutually exclusive, and many movies fall into multiple categories. For example, a movie could be both a comedy and a romance, making it challenging for models to correctly categorize it into one genre. The obstacle of overlapping genres is well-documented in genre classification literature, where genre-blending movies are recognized as a significant challenge (Nikhata et al., 2023).

Our efforts to improve model accuracy through hyperparameter tuning and cross-validation provided marginal improvements. Nevertheless, these modifications proved insufficient to overcome the obstacles presented by the dataset. In our research, we discovered that many other researchers have encountered similar difficulties when performing movie genre prediction from plot summaries. Ertugrul and Karagoz (2018) demonstrated that even advanced models like BiLSTM combined with BERT could only achieve moderate improvements, suggesting that the fundamental complexity of genre prediction tasks limits model performance. This result is consistent with other studies, suggesting that the problems we encountered are not exclusive.

7. Limitations

This study has a few potential limitations, mostly centered around the datasets and data selection. The first limitation concerns the significant class imbalance in the number of movies per genre. The dataset includes 27 genres, but two genres (drama and documentary) have significantly more movies (both exceeding 13,000) compared to others. Genres like ‘game- show’, ‘news’, and ‘war’ have fewer than 200 movies each. This imbalance allows models to train more effectively on the well-represented genres while neglecting the less frequent ones. This is apparent from the overfitting on all of our models, where the models exhibit high accuracy during training and effectively capture the patterns in the data, but perform poorly on unseen test data, as illustrated in Appendix 9, Appendix 10 and Appendix 11, where high accuracy on the training set contrasts with low accuracy on the test set.

Another limitation potentially arises from the inclusion of the "short" genre. Although it ranks fourth in popularity with 5,054 movies in the training set, our analysis of descriptions suggests it reflects film length rather than plot or other characteristics used by the "Description" metric. This makes it difficult for models to classify "short" movies accurately, as a short film about war might be labeled as "short" instead of "war," even if the description suggests a war theme. This issue further highlights the challenge of subjective genre classification, as discussed in section 6.2.

Another limitation which we faced was computational complexity. The models utilized for this project, especially BERT and RNN-LSTM required substantial computational resources and training time. Even with the utilization of cloud computing, these models took a long time to train (Appendix 15) which in comparison to BERT requires training. This limited our ability to test and make adjustments to the models. Moreover, the computational boundary limited us from implementing techniques such as SMOTE to even the genre distribution and/or add an extra layer in the RNN-LSTM model.

8. Conclusion & Future Work

This study investigated the application of NLP techniques for automating genre classification of movie plot summaries from the IMDb dataset, comparing BERT, RNN using LSTM, and Multinomial Naive Bayes models. The BERT model outperformed the others with a 61% accuracy, highlighting its ability to understand contextual relationships within text. However, its high computational demands present a challenge for widespread application. The RNN and Multinomial Naive Bayes models, achieving accuracies of 53.51% and 53.38% respectively, offer more computationally efficient alternatives despite their slightly lower performance.

Our findings emphasize the trade-off between accuracy and computational efficiency in NLP-based genre classification. While BERT's pre-training enables it to capture complex language patterns effectively, it requires significant computational resources, which may limit its practical application. On the other hand, the RNN and Naive Bayes models, though less accurate, are more suitable for scenarios with limited computational power.

Several limitations were identified, including the dataset's class imbalance and the presence of non-informative content in plot summaries. These factors contributed to the models' suboptimal performance in underrepresented genres and necessitated extensive preprocessing. Future research should focus on developing more sophisticated preprocessing techniques, employing advanced oversampling methods like SMOTE, and exploring additional model architectures to improve genre classification accuracy.

9. References

-Aggarwal, C. C., & Zhai, C. (2012). Mining Text Data. Springer Science & Business Media.

-Akbar, J., Utami, E., & Yaqin, A. (2022, December). Multi-Label Classification of Film Genres Based on Synopsis Using Support Vector Machine, Logistic Regression and Naïve Bayes Algorithms. In *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)* (pp. 250-255). IEEE.

-Brownlee, J. (2018). Train-Test Split and Cross-Validation in Python. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/train-test-split-and-cross-validation-in-python/

-Egger, R., & Gokce, E. (2022). Natural language processing (NLP): An introduction: Making sense of textual data. In Applied data science in tourism: Interdisciplinary approaches, methodologies, and applications (pp. 307-334). Cham: Springer International Publishing.

-Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Learning from Imbalanced Data Sets. Springer.

-Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

-Hochreiter, S. and J. Schmidhuber.1997. Long short-term memory. Neural Computation, 9(8):1735–1780.

-IMDb.com. (n.d.). Genre. IMDb Help Center - Genre. https://help.imdb.com/article/contribution/titles/genres/GZDRMS6R742JRGAG?ref_=helpart_nav_31#howto

-Jurafsky, D., & Martin, J. H. (2023). Speech and language processing (3rd ed.). Pearson.

-Kaggle. (2021, June 18). Genre classification dataset imdb. Kaggle.com, User: RADMIRKAZ. https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb/code?datasetId=1417162&searchQuery=c

-Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

-Lawson, M. (2013, June 4). *The Americans, Love & marriage and the rise of hybrid TV genres | Mark Lawson*. The Guardian.com. https://www.theguardian.com/tv-and-radio/tvandradioblog/2013/jun/04/americans-love-marriage-hybrid-genres

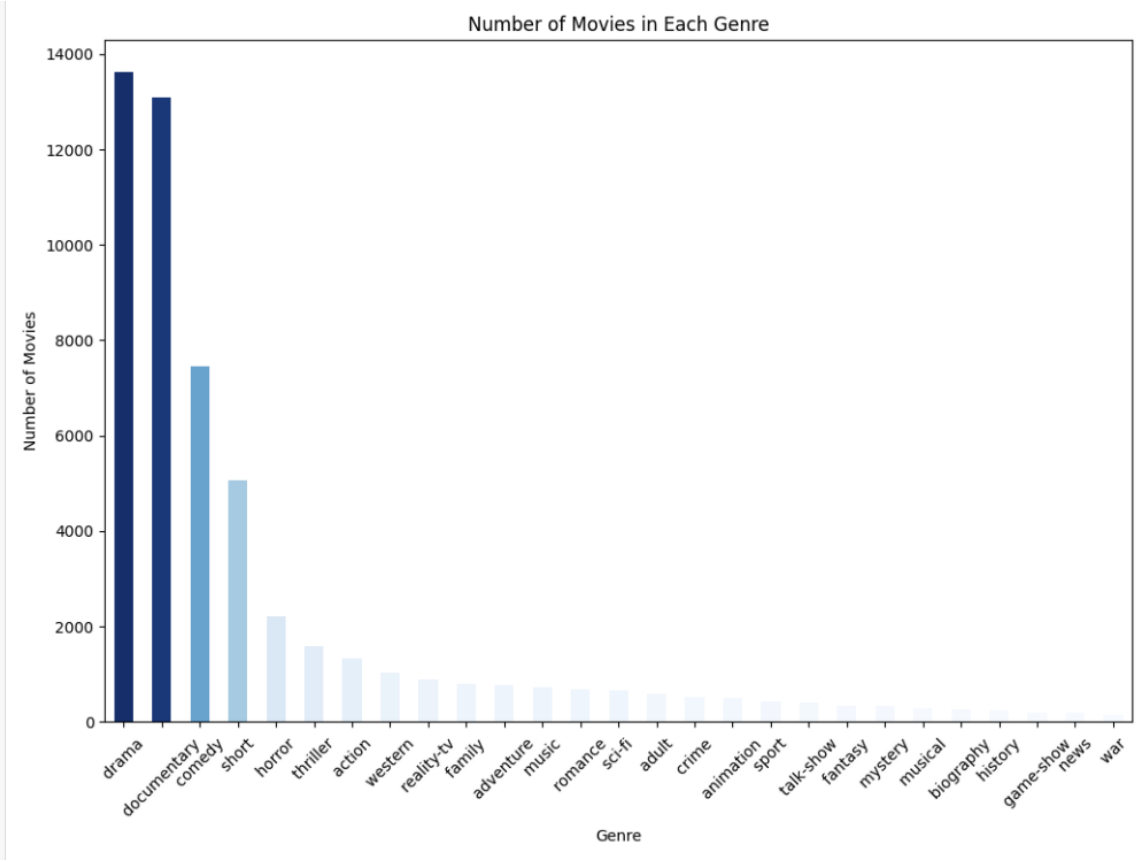
-Nikhata, G., & Gauch, S. (2023). Fine-tuning BERT with Bidirectional LSTM for Find-Grained Movie Reviews Sentiment Analysis. *International Journal On Advances in Systems and Measurements*.

-Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. Proceedings of the First Instructional Conference on Machine Learning.

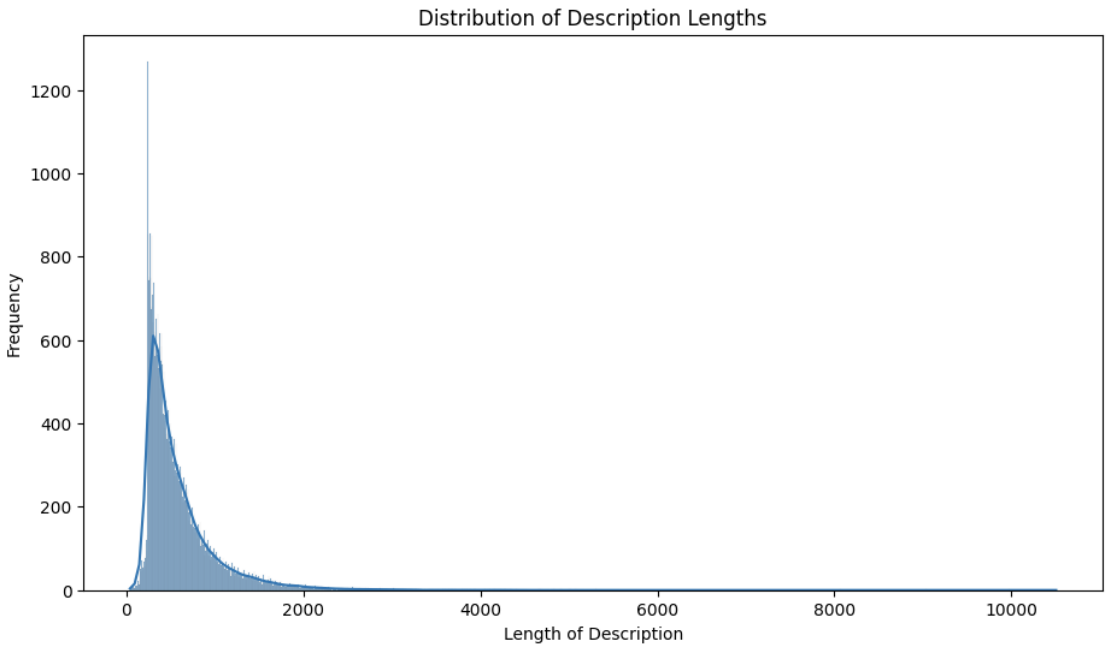
-Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. *International journal of intelligent systems and applications in engineering*, *7*(2), 88-91.

-Tarun, S., Batth, R. S., & Kaur, S. (2021). A Scheme for Data Deduplication Using Advance Machine Learning Architecture in Distributed Systems. 2021 International Conference on Computing Sciences (ICCS), 53-60.

10. Appendix



Appendix 1: Bar chart of genre distribution



Appendix 2: Histogram of Distribution of Description Lengths

Test Accuracy: 0.5339998148662408
Classification Report on Test Set:

	precision	recall	f1-score	support
0	0.34	0.41	0.37	1313
1	0.48	0.31	0.38	589
2	0.32	0.21	0.25	772
3	0.25	0.13	0.17	496
4	0.05	0.02	0.03	263
5	0.51	0.56	0.53	7426
6	0.12	0.09	0.10	505
7	0.70	0.74	0.72	13072
8	0.60	0.58	0.59	13555
9	0.25	0.17	0.20	770
10	0.17	0.07	0.10	321
11	0.92	0.54	0.68	191
12	0.09	0.05	0.07	243
13	0.53	0.58	0.56	2199
14	0.45	0.60	0.52	712
15	0.14	0.04	0.06	273
16	0.10	0.04	0.05	317
17	0.18	0.05	0.08	180
18	0.33	0.37	0.35	880
19	0.17	0.20	0.19	669
20	0.34	0.30	0.32	645
21	0.38	0.36	0.37	5058
22	0.51	0.46	0.48	430
23	0.36	0.22	0.27	386
24	0.23	0.30	0.26	1589
25	0.30	0.11	0.16	132
26	0.82	0.82	0.82	1029
accuracy			0.53	54015
macro avg	0.36	0.31	0.32	54015
weighted avg	0.53	0.53	0.53	54015

Appendix 3: Multinomial Naive Bayes Accuracy Report on Test Set

Test Accuracy: 0.5497361843932241
Classification Report on Test Set:

	precision	recall	f1-score	support
0	0.33	0.35	0.34	1313
1	0.42	0.34	0.38	589
2	0.37	0.13	0.20	772
3	0.19	0.08	0.11	496
4	0.00	0.00	0.00	263
5	0.54	0.50	0.52	7426
6	0.24	0.02	0.03	505
7	0.65	0.83	0.73	13072
8	0.56	0.67	0.61	13555
9	0.28	0.06	0.10	770
10	0.33	0.00	0.01	321
11	0.87	0.38	0.53	191
12	0.00	0.00	0.00	243
13	0.56	0.63	0.59	2199
14	0.50	0.43	0.46	712
15	0.14	0.05	0.07	273
16	0.00	0.00	0.00	317
17	0.00	0.00	0.00	180
18	0.28	0.22	0.25	880
19	0.16	0.10	0.13	669
20	0.35	0.29	0.32	645
21	0.43	0.30	0.36	5058
22	0.38	0.36	0.37	430
23	0.31	0.27	0.29	386
24	0.26	0.22	0.24	1589
25	1.00	0.03	0.06	132
26	0.84	0.80	0.82	1029
accuracy			0.55	54015
macro avg	0.37	0.26	0.28	54015
weighted avg	0.51	0.55	0.52	54015

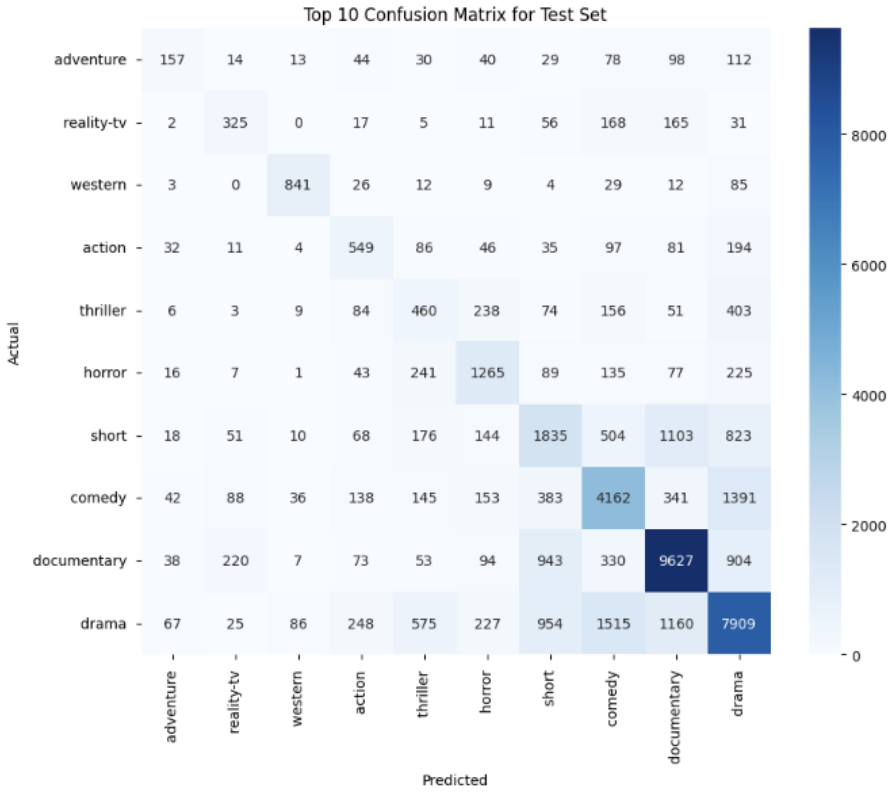
Appendix 4: RNN Accuracy Report on Test Set

Accuracy: 0.6108488382856614

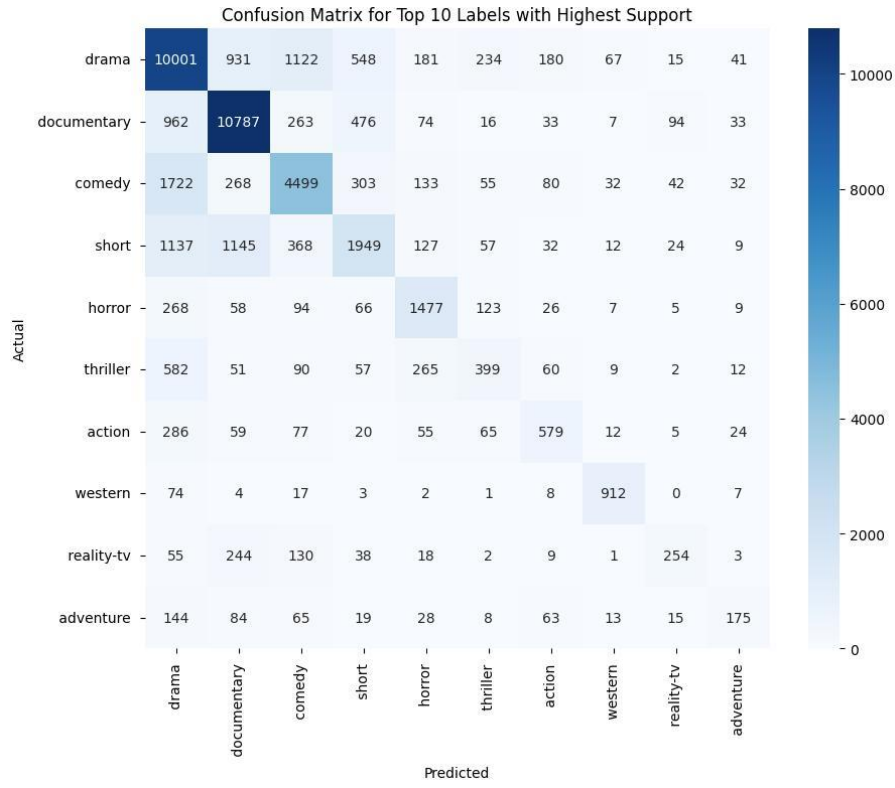
	precision	recall	f1-score	support
action	0.44	0.44	0.44	1313
adult	0.54	0.49	0.52	589
adventure	0.39	0.23	0.29	772
animation	0.31	0.22	0.26	496
biography	0.00	0.00	0.00	263
comedy	0.60	0.61	0.60	7426
crime	0.29	0.04	0.07	505
documentary	0.73	0.83	0.78	13072
drama	0.60	0.74	0.66	13555
family	0.37	0.21	0.26	770
fantasy	0.30	0.05	0.09	321
game-show	0.80	0.65	0.72	191
history	0.25	0.01	0.02	243
horror	0.58	0.67	0.62	2199
music	0.60	0.64	0.62	712
musical	0.24	0.07	0.11	273
mystery	0.00	0.00	0.00	317
news	0.00	0.00	0.00	180
reality-tv	0.41	0.29	0.34	880
romance	0.47	0.09	0.15	669
sci-fi	0.43	0.45	0.44	645
short	0.51	0.39	0.44	5058
sport	0.58	0.51	0.54	430
talk-show	0.45	0.45	0.45	386
thriller	0.34	0.25	0.29	1589
war	0.51	0.16	0.24	132
western	0.84	0.89	0.86	1029
accuracy			0.61	54015
macro avg	0.43	0.35	0.36	54015
weighted avg	0.58	0.61	0.59	54015

Appendix 5: BERT Accuracy Report on Test Set

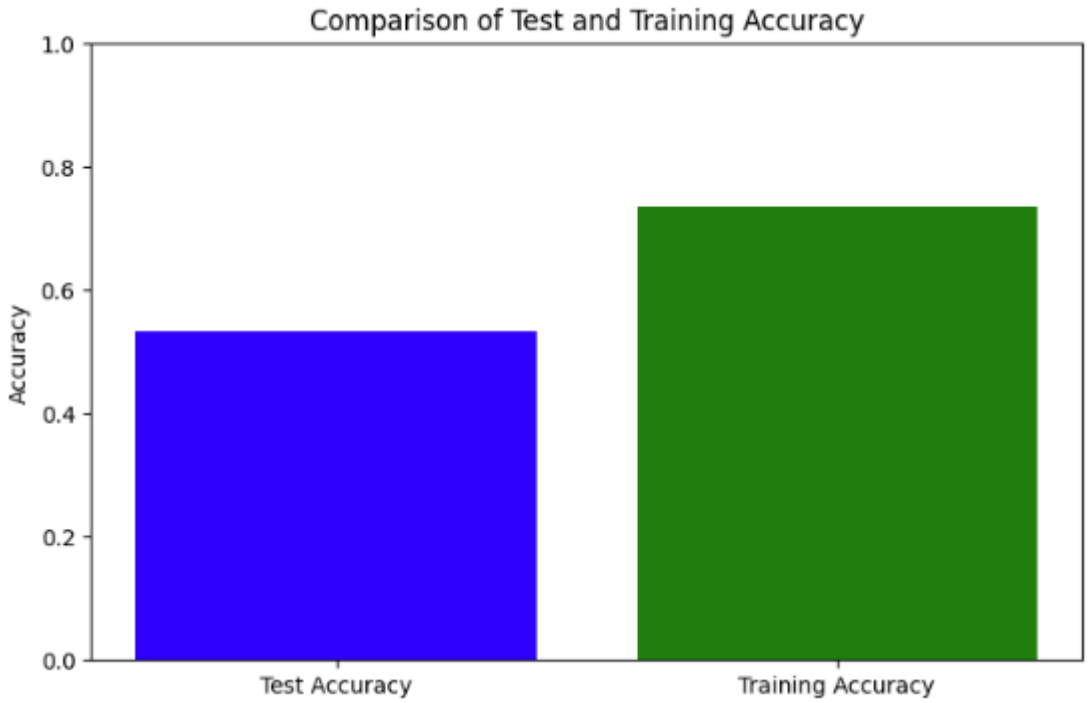
Appendix 6: Multinomial Naive Bayes Top 10 Confusion Matrix for Test Set



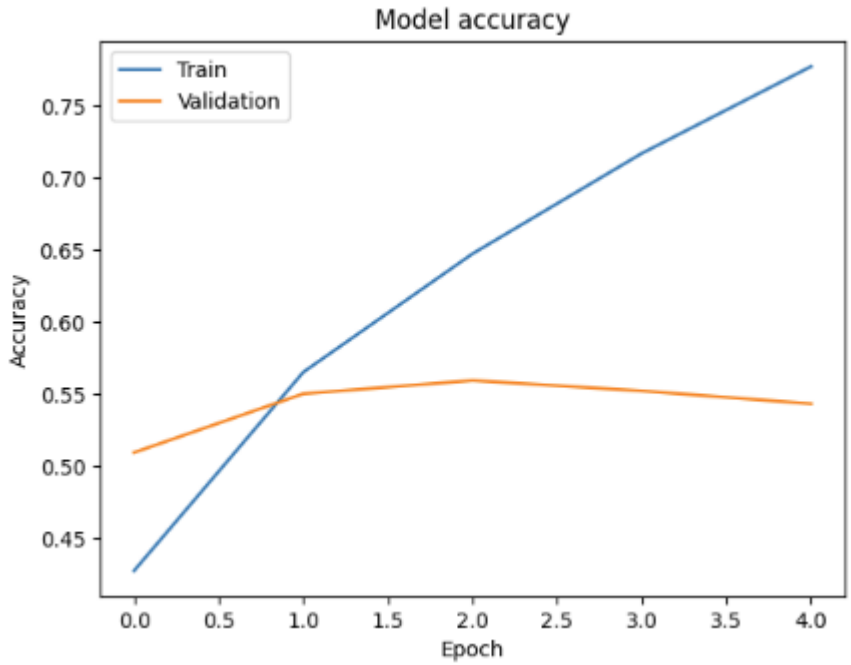
Appendix 7: RNN Top 10 Confusion Matrix for Test Set



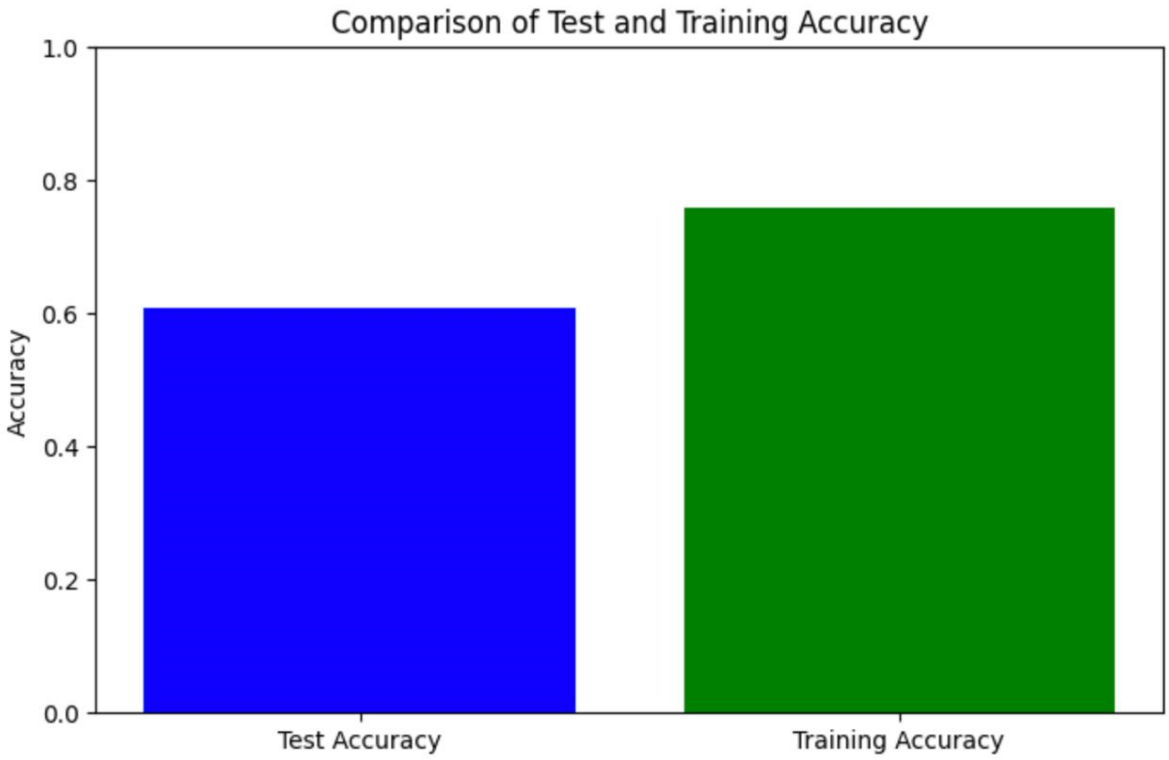
Appendix 8: BERT Top 10 Confusion Matrix for Test Set



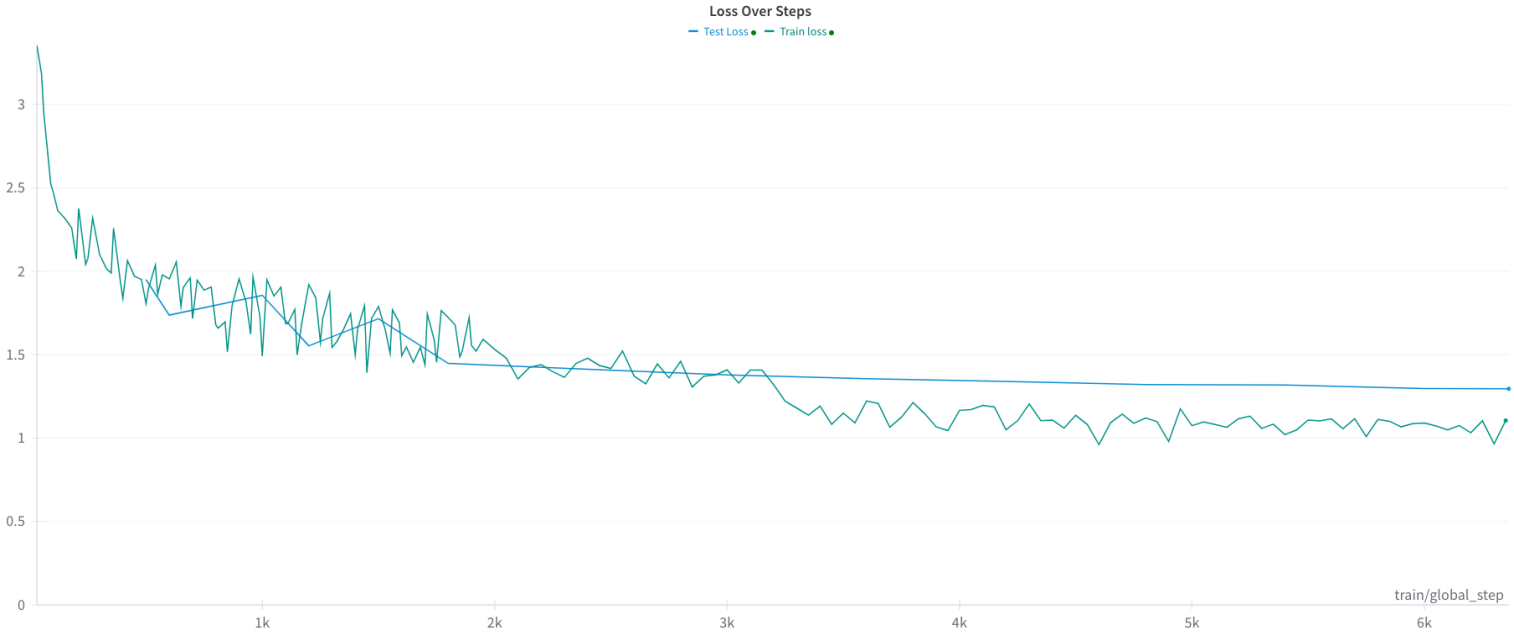
Appendix 9: Comparison of Test and Training Accuracy in Multinomial Naive Bayes



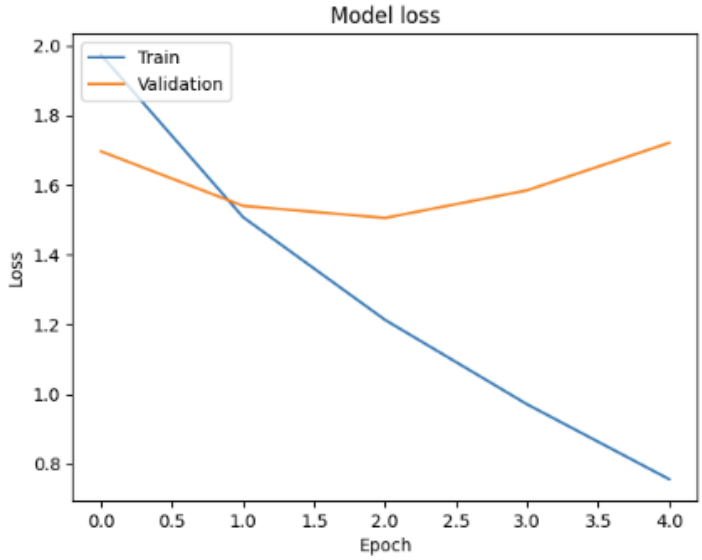
Appendix 10: Comparison of Test and Training Accuracy in RNN



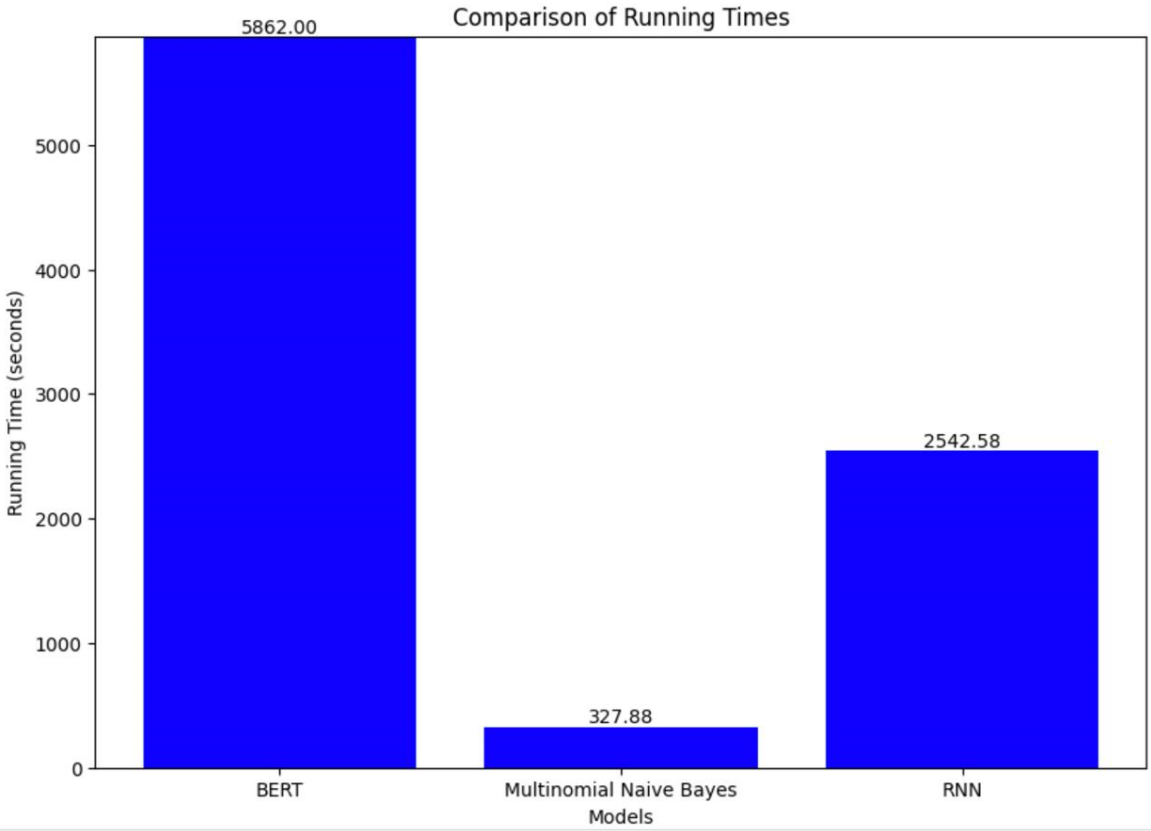
Appendix 11: Comparison of Test and Training Accuracy in BERT



Appendix 12: Loss and Accuracy of BERT



Appendix 13: Loss and Accuracy of RNN



Appendix 14: Comparison of Running Times