Movie Multi-Genre Predictor using DistilBERT

NLP model that predicts movie genres based on film overview

Balete, Immanuel Josiah A.
Ducay, James Daniel P.

June 10, 2024

1 Introduction

Movies have developed greatly since their initial conception over a century ago. As technology and visual effects became more complicated, so did movie plots and themes. The 2022 film Everything Everywhere All at Once is a good example of this, as it has been described to contain elements from science fiction, fantasy, and black comedy, among several other genres. Previous works have developed models to predict a single genre based on a textual summary, but this may be lacking when put up against multi-genre films. As such, this project aims to improve on the classification model by identifying when a movie would be categorized under multiple categories, while still being accurate for films that mainly feature only one genre.

Accurately classifying a movie plot or synopsis by its genre can be beneficial for selecting suitable movies for specific audiences, as certain themes may not be appropriate for them. Additionally, this multi-label classification model can be applied in generating precise movie ratings, such as general patronage, parental guidance, or strict parental guidance. Moreover, it can also be applied in identifying an audience members's perception of a movie's underlying themes. This would be done by inferring genres based on their version of the synopsis, perhaps from reviews or blog sites. Aside from genre predictions, the model could also be generalized for solving any classification problem involving multiple categories.

Implementing multi-label classification with numerous possible labels presents a significant challenge. Therefore, this project intends to fine-tune and train the DistilBERT NLP model using preprocessed data specifically designed for this purpose. Its effectiveness will be evaluated by measuring its accuracy, ensuring the model is suitable for the intended uses.

2 Related Work

There are numerous works that predict movie genres, each employing different datasets, models, analyses, and outputs. The references noted here used movie overviews as input, or used techniques to allow their model to assign multiple labels.

One method, as detailed by Gowtham, G. (2023), utilizes TF-IDF in tandem with the Naive Bayes classification algorithm, a probabilistic classifier which assumes independence among the present features. This technique operates on a comprehensive dataset sourced from IMDb on Kaggle. Notably, it predicts a single genre that best aligns with the movie overview. [4] Data-Driven Science (2023) also uses a plot synopsis to predict a single genre, but utilizes the Long Short-Term Memory (LSTM) model instead. This model, a variant of recurrent neural networks (RNNs), excels in capturing and retaining long-term dependencies in sequences, making it ideal for text classification tasks. [6] The work of Alwyn et al. (2024) also uses LSTM and the BERT model to predict single genres based on plot summaries, with both models achieving a 95.72% accuracy. [2] To improve on the use of the BERT model, this project aims to let the model predict more than one genre for each film.

Previous works have tackled the problem of assigning multiple genres, but with the use of a different model. Joshi (2022) uses an approach inspired by multi-label image classification. By employing the Binary Relevance method, his model generates multi-label classifications based on plot summaries. In essence, it creates individual classifiers for each genre label, allowing for multiple genre predictions per movie. [5] Similarly, the model developed by Akbar et al. (2022) uses the film synopsis to predict more than one genre. After preprocessing, they compared three different models for classifying the synopses: support vector machine, logistic regression, and the naive Bayes method. All three approaches used grid search to find the optimal parameters. [1]

There were also works that did not use textual inputs but had interesting approaches to solving the multi-label problem. One of these was Chu and Guo's work (2017), which used deep neural networks to classify movie posters. In order to assign more than one genre to a movie, they identified optimal thresholds for each genre's probability by using a grid search scheme together with the Matthews correlation coefficient. Thus, different genres could be given different thresholds, allowing for better accuracy. [3]

3 Methods

The project utilized the '9000+ Movies Dataset', a CSV file sourced from Kaggle, originally designed for constructing a movie recommendation system. This dataset encompasses nine columns, notably featuring movie titles, overviews or brief summaries, and genres, which are essential for the project. Despite its inclusion of movies from various origins, the dataset exclusively comprises English content. This includes 9515 unique movie titles and 9824 unique movie overviews, which may imply that there are movies of the same title.

Using this dataset, the project aimed to predict the genres of movies by utilizing only the overview and genre columns. Given that all overviews are in English, an efficient implementation of Natural Language Processing (NLP) techniques was chosen to train the model. Since the dataset also provided a finite set of possible genres, the project was treated as a multi-label classification problem.

To parse the genres column, which was formatted as a string in the original dataset, one-hot encoding was applied via MultiLabelBinarizer from the sklearn package.

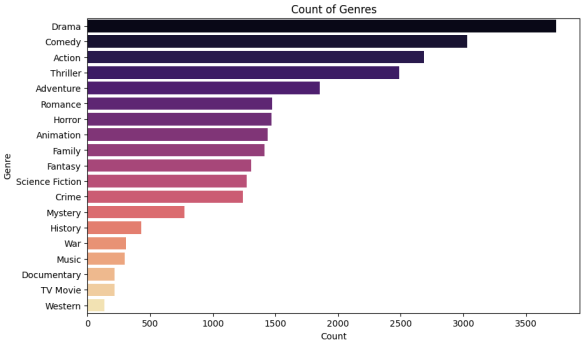

Figure 1: Initial Dataset Plot

As observed in the dataset plot in Figure 1, out of the 9,824 movie overviews, the most frequent genre was 'Drama' with 3,743 instances, while the least frequent was 'Western' with only 20 samples. This disparity may introduce bias in the classification of overviews, leading to a higher likelihood of predicting 'Drama' and a lower likelihood of predicting 'Western'. A similar analysis was conducted for other genres.

To address this issue, oversampling of minority genres was employed. First, all movies which were not classified under one of the top four genres—Drama, Comedy, Action, and Thriller—were extracted. Using this subset, back-translation via the BackTranslation library was performed. Each text was translated into the following five languages: Chinese, Filipino, Spanish, Japanese, and Korean. Then, all the foreign texts were translated back to English, thus augmenting the original texts.

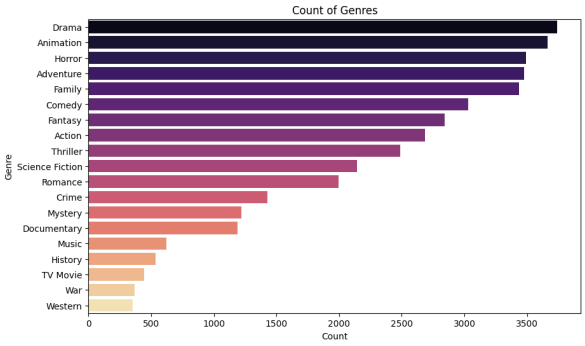

Figure 2: Oversampled Dataset Plot

Figure 2 illustrates the result of the oversampling done via back-translation, which shows a more balanced distribution of the genres, albeit still featuring a large range of values.

After preprocessing, the dataset was divided into three sets: 70% for the training set, 20% for development, and 10% for testing. The DistilBERT NLP model and a custom dataset class were initialized for the training, validation, and test sets. These sets now contained tokenized versions of the text data, as processed by the DistilBERT model's tokenizer, along with their corresponding target labels.

DistilBERT is a transformers model, smaller and faster than BERT, which had been pre-trained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. This means it was pre-trained on the raw texts only, with no humans labelling them in any way. When taking a sentence, the model randomly masks 15% of the words in the input then runs the entire masked sentence through the model and tries to predict the masked words.

The resulting data was formatted as PyTorch Tensors, tailored for deep learning tasks. Training parameters were initialized using the TrainingArguments function from the transformers library, with both training and evaluation batch sizes set to eight samples, and number of epochs set to five. Subsequently, the Trainer function from the transformers library was used to initialize a trainer, which enabled the model to be trained using the preprocessed training dataset.

After training, a sigmoid activation function was used to convert the model's output values into probabilities for each genre. These probabilities were then turned into binary predictions, based on threshold values. Initially, there was a fixed threshold of 0.5 for all genres, but these values were subsequently adjusted using the validation set.

The model iterated through the data in the validation set ten times. In each iteration, the thresholds were updated after every prediction. For example, if the model incorrectly predicted the presence of a genre (indicated by a value of 1), that meant

the threshold for that specific genre had to be made stricter by increasing its value. On the other hand, if the model missed a genre, the threshold had to be relaxed by decreasing its value. The first iteration would adjust thresholds by 0.001 each time, while subsequent iterations changed the threshold values by smaller and smaller steps, similar to the concept of simulated annealing.

4 Evaluation

To assess the model's accuracy, two custom metrics were devised. Since there are 19 genres, the model, along with the finetuned thresholds, produces an array of 19 zeroes and ones, predicting which genres a text falls under. This array can then be compared to the correct classification. For the first custom metric, which shall be called 'accuracy', the number of matching zeroes and ones for each text is tallied, then divided by the total number of predictions (i.e. 19 times the number of texts in the test set). Thus, this metric counts both true positives and true negatives.

However, this metric may give a false sense of correctness, since most arrays are made up of zeroes, and the model generally predicts a lot of zeroes as well. As such, a second metric was defined, called 'score', which gives more emphasis to correctly guessing the presence of a genre. To elaborate, the model earns three points for each correctly predicted value of 1 (indicating presence of a genre), one point for each correctly predicted value of 0, and no points otherwise. Then, the total number of points earned is divided by the maximum possible number of points to generate the score. The two metrics can be summarized in the following equations:

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{(\#\text{ of genres}) \cdot (\#\text{ of samples})}$$

$$\text{score} = \frac{3(\text{true positives}) + \text{true negatives}}{\text{maximum possible points}}.$$

By experimenting with different ways of training the thresholds, the following accuracies and scores were obtained:

Description	Acc.	Score
original model	0.9579	0.9296
control (always guess 0)	0.8779	0.7057
1 iter, steps of 0.001	0.9519	0.9186
5 iters, steps of 0.001	0.9523	0.9211
10 iters, steps of 0.001	0.9519	0.9222
10 iters, steps of $0.001/i$	0.9528	0.9205

Figure 3: Accuracy and Scores

Note that the "step" refers to step size of threshold adjustment. Also, the first row refers to an older version of the model which was not trained on an oversampled dataset.

The control row indicates the baseline performance where the model simply guesses that all movies do not belong to any genre. This was chosen as the control since movies are generally not part of most genres (around 16 to 18 out of 19).

5 Discussion and Conclusion

The final model that was trained with an oversampled dataset seemingly achieved a lower accuracy and score than the original model. However, it must be taken into account that while the training had an uneven distribution of genres, so did the test set. This suggests that the earlier model only did "better" because it predicted the majority genre more often.

This is illustrated in Figures 4 and 5, which shows the confusion matrix for the 'Drama' genre, both when using the original model and the model trained with an oversampled dataset.

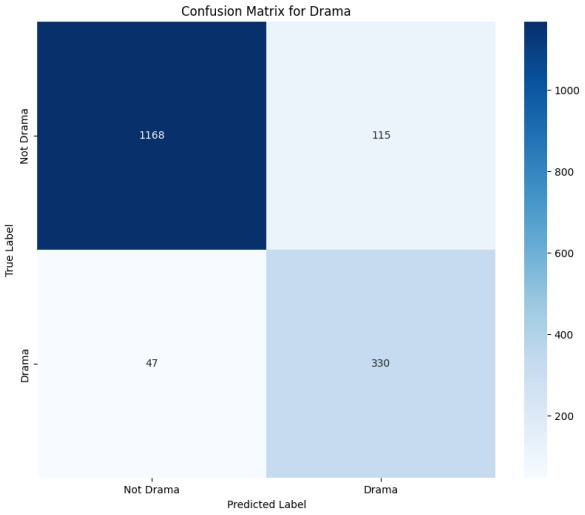

Figure 4: Original Drama Confusion Matrix

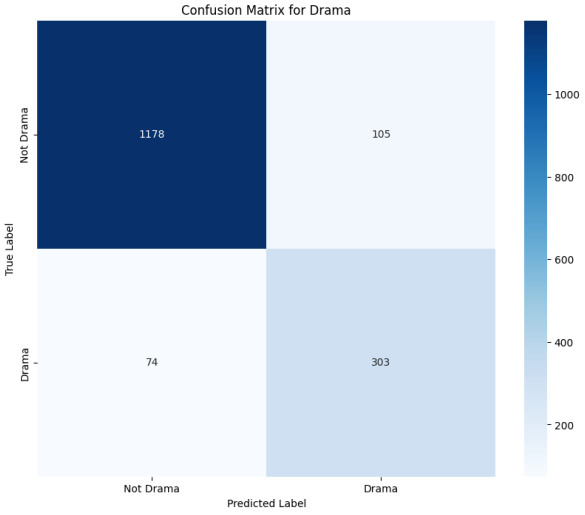

Figure 5: Oversampled Drama Confusion Matrix

While the original model achieved more true

positives, it also had more false positives, reflecting how the original model simply liked to predict majority genres more often. The test set's uneven distribution thus benefits the original model's preference for the top occurring genres, giving a false sense of better accuracy.

It can also be observed from Figure 3 that the accuracy seemed to decrease when performing 10 iterations with constant steps, while the score increased. This suggests that the thresholds had begun overfitting already to the validation set and could be making a lot of false positives. By using smaller steps for succeeding iterations, this issue was resolved.

The score generated from the modified scoring system is notably consistently lower than the accuracy level. The final model's accuracy is 95.79%, whereas the score is 92.96%. This follows from the fact that there are significantly more true negatives. It can be observed that the model needs improvement in avoiding false positives, at least for the drama genre. The large occurrence of false positives may be attributed to the fact that 'Drama' is one of the top occurring genres. Additional training or adjustments might be required to discourage the model from making false positives.

Nevertheless, the model was able to perform significantly better than the baseline of simply guessing zero, i.e. predicting that every movie belongs to no genres. A difference of 7.49% could be observed for the accuracy and 21.48% for the score. Thus, the model successfully identified the genres of the test dataset and can now be applied to other movie synopses outside the dataset, thereby addressing the initial motivation for developing the model.

There is still much room for improvement in this project. For one, the hyperparameters of the model, such as batch size and number of epochs, were not modified due to time constraints. Furthermore, the authors were more concerned with how the thresholds could be made to dynamically fit to each genre. Thus, future researchers may want to finetune the distilBERT model's parameters itself to gain better performance.

Second, the method of oversampling may be suboptimal due to introducing repeat copies of the same movies. Also, the final preprocessed dataset still featured an uneven distribution of genres. This may lead to the model overfitting on repeated movies or majority genres. Testing on a larger dataset, while ensuring a fairer distribution of genres, may lead to better results.

Lastly, the method of identifying the thresholds for each genre may be improved upon. While more iterations on the validation set may not necessarily lead to better performance (due to the thresholds themselves overfitting to the validation set), a different approach such as grid search, local beam search, or genetic algorithms may be considered. Furthermore, the thresholds were adjusted independent of each other, so another idea would be examining possible relationships between genres, e.g. due to the prevalence of the romantic-comedy genre (romcoms), the threshold for comedy may change based on the threshold for romance.

6 References

[1] Akbar, J., Utami, E., and Yaqin, A. (2022). 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)*, pages 250–255. https://ieeexplore.ieee.org/abstract/document/10057828/.

[2] Alwyn, A., Pranoto, E. J. P., Ichsan, I., Halim, K., Justin, W., and Girsang, A. S. (2024). Movie genre classification using bert and lstm. In *AIP Conference Proceedings*, volume 2927. AIP Publishing. https://pubs.aip.org/aip/acp/article-abstract/2927/1/040001/3279172/Movie-genre-classification-using-BERT-and-LSTM?redirectedFrom=fulltext.

[3] Chu, W.-T. and Guo, H.-J. (2017). Movie genre classification based on poster images with deep neural networks. In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes*, MUSA2 '17, page 39–45, New York, NY, USA. Association for Computing Machinery. https://dl.acm.org/doi/abs/10.1145/3132515.3132516.

[4] Gowtham, G. (2023). Movie genre classification. https://www.kaggle.com/code/imgowthamg/movie-genre-classification.

[5] Joshi, P. (2022). Predicting movie genres using nlp – an awesome introduction to multi-label classification. https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/.

[6] Science, D.-D. (2023). Master movie genre prediction with nlp: A comprehensive guide to imdb dataset analysis and lstm modeling. https://shorturl.at/kIMQT.