Finding the Genre of Movie Plots

GROUP 66 - JOSHUA REDELBACH (1112470), HENRIETTA SUNDBERG (1112651), and LOANE RABILLARD (1112780)

1 Introduction

The classification of movies into genres is crucial in the film industry, shaping audience expectations and influencing marketing strategies. With the rise of digital media, there is a growing need for automated systems that can accurately predict movie genres based on available textual data.

This paper presents a combination of a transformer model and a Support Vector Classifier designed to estimate a film's genre using its plot, title, country of origin, and director. First, a detailed description of the final version of the developed model is given. Second, the experimental setup and the obtained results described. Afterwards, the results are interpreted and discussed, before an outlook on future work is given in conclusion.

2 Model

In our project, we developed a hybrid model that leverages the strengths of the DistilBERT architecture in combination with a Support Vector Classifier (SVC). This approach capitalizes on DistilBERT's ability to understand contextual relationships in textual data while utilizing the SVC's effectiveness in classification tasks.

DistilBERT is a lightweight version of the Bidirectional Encoder Representations from Transformers (BERT) model, which has gained prominence for its state-of-the-art performance in various natural language processing tasks. DistilBERT retains 97% of BERT's language understanding capabilities while being 60% faster and requiring 40% fewer parameters [1]. This makes it an ideal choice for this project as the goal is to achieve high accuracy while not having many computational resources available for developing and training the model.

To further enhance our classification capability, we employed a Support Vector Classifier (SVC) as well. SVC is a robust machine learning algorithm that excels in high-dimensional spaces and is particularly effective in scenarios where the number of features exceeds the number of samples. By finding the optimal hyperplane that maximizes the margin between different classes, the SVC can efficiently separate genres based on features, extracted e.g. by applying the Term Frequency-Inverse Document Frequency (TF-IDF) method [4]. In detail, the processing pipeline is divided into two parallel stages being described in the following.

Before applying the classifiers, the plot is first preprocessed by converting all text to lowercase. This normalization step ensures that words are consistently treated in their lowercase form simplifying the tokenization process. No preprocessing steps are applied to the remaining inputs (title, director, country of origin).

The first pipeline employs the DistilBERT model. The preprocessed plot, along with the title, director, and country of origin, are tokenized using the DistilBERT tokenizer. The resulting tokens are passed to the DistilBERT classifier, which outputs probabilities for the different movie genres.

The second pipeline uses only the preprocessed plot as input. The text is vectorized using TF-IDF. TF-IDF is effective at capturing the importance of specific terms in the plot, which can be highly indicative of genre (e.g., "alien" for sci-fi, "love" for romance) [5]. These vectors are then fed into a SVC, which outputs genre probabilities.

Once both pipelines have independently predicted genre probabilities, the average of the two probability distributions is computed. The genre with the highest average probability is selected as the final prediction.

During training, additional techniques are applied for improving the performance. First, the labels are encoded to numerical values (0 to 8). Furthermore, the AdamW optimizer is used during the DistilBERT training. It includes weight decay which is a form of regularization that helps to prevent overfitting by penalizing large weights. The AdamW optimizer is particularly well-suited for training transformer models such as DistilBERT [2]. Additionally, a scheduler is applied when training DistilBERT. It adjusts the learning rate during training by increasing it linearly from zero to the initial learning rate during the warmup phase, and then decreases it linearly after the warmup phase until training ends. This helps the model converge better, as the learning rate becomes smaller as the model gets closer to optimal parameters [3].

3 Experimental Setup and Results

For developing the model, a dataset is used that contains 8041 movie samples each composing of a title, director, country of origin, plot and a corresponding genre label. In total nine different genres occur in the dataset: drama, comedy, horror, action, romance, western, crime, animation and sci-fi. The distribution of the different genres over this set can be seen in fig. 1. This dataset is split into a training (80%) and a test (20%) set. For evaluating the resulting performance of the model, the precision, recall and the F1-score is determined for each category which are listed in tab. 1. Furthermore, a detailed confusion matrix is created, which can be seen in fig. 2. Lastly, the overall accuracy is determined which results in a value of 71.47%.

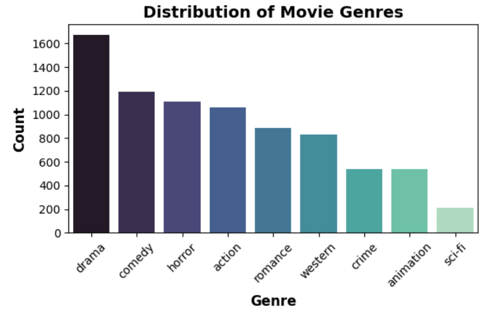

Fig. 1. Distribution of the movie genres in the dataset that is split and used for training and testing.

Genre	Precision	Recall	F1-score
Action	0.67	0.62	0.69
Animation	0.83	0.76	0.79
Comedy	0.68	0.52	0.59
Crime	0.76	0.63	0.69
Drama	0.59	0.73	0.65
Horror	0.79	0.86	0.82
Romance	0.66	0.58	0.62
Sci-fi	0.61	0.49	0.54
Western	0.93	0.94	0.94

Table 1. Results for each genre when applying the model on the test dataset. The scores are rounded to two digits.

4 Discussion

The performance of the model in classifying movie genres, as detailed in tab. 1, demonstrates strong results across most categories, though several key challenges can be identified. While genres like animation, western or horror perform well, others such as drama or sci-fi exhibit lower metrics which explains the overall accuracy of 71.47%.

Animation, western, and horror show strong performance with high precision, recall, and F1-scores. Animation and horror have F1-scores of 0.79 and 0.82 respectively, indicating accurate predictions. Western stands out with an even higher precision of 0.93 and recall of 0.94, leading to a F1-score of 0.94. These results may be because of the distinctiveness of these genres, which likely contain highly characteristic terms in their plots. For instance, westerns often have unique plot elements like historical settings and thematic elements that distinguish them from other genres. Horror movies often deal with fear, suspense and supernatural themes that can easily be set apart from other genres as well.

The crime genre performs decently with an F1-score of 0.69, benefiting from its distinctive narrative elements, like recurring twists. However, it's surprising that the model captures those elements well, given the limited number of crime samples in the dataset.

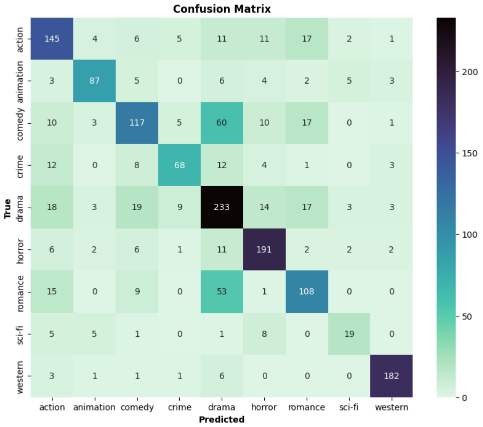

Fig. 2. Confusion matrix for the results of the test data.

The action genre, despite being more represented in the dataset, also has an F1-score of 0.69. This is likely due to action films often blending elements from other genres like drama, comedy, crime, and romance, which are reflected in their plot descriptions. The confusion matrix confirms this, as the model frequently mislabels samples from these genres as action.

Even though plot descriptions of sci-fi movies should contain indicative vocabulary, such as "space", "alien" or "robot", which should be easily captured by the TF-IDF vectorization and the deeper semantic understanding of the transformer, the model is not predicting them correctly resulting in a F1-score of 0.54. This might be caused by the lack of sci-fi samples in the dataset leading the model to misclassify them as similar genres like animation or more common genres like horror.

Drama, the most represented genre in the dataset, achieves an F1-score of 0.65, with significant confusion between drama, comedy, and romance as the model predicts the label drama for 60 comedy and for 53 comedy instances. One reason for this could be that the genre drama is simply overrepresented in this dataset and such the model gets biased. One reason for this lower performance could be that drama plots often share similar linguistics and overlapping narrative elements with other genres, as e.g. romances, leading to miss-classifications. Those films often include general terms like "love", "relationship" or "family" which are not always exclusive to one genre, making it more challenging for the model to differentiate and correctly assign these emotional themes. This can be observed as well when analyzing the results of comedy and romances leading to F1-scores of 0.59 and 0.62 respectively. Furthermore, comedy films often contain humor-related language, which is in general more difficult to capture as well.

In summary, while the hybrid DistilBERT-SVC model shows promising results, there are clear challenges in distinguishing between overlapping or less well-represented genres. Data imbalance and the overlap in film narratives across genres are likely the primary contributors to these performance issues.

5 Future Work

In order to further improve the model, several aspects can be considered in the future. First, it can be tried to balance the dataset more by applying data augmentation techniques, such as back-translation, for the genres which have less samples in the dataset. Furthermore, it was noticed that some plots of the data samples contain characters from non-latin scripts. To improve the performance, those parts could be first translated into english before applying the model. Lastly, it can be analyzed if the performance can be improved by applying more methods during preprocessing, e.g. stop word removal.

References

[1] Hugging Face. [n. d.]. *DistilBert.* Retrieved October 16, 2024 from https://huggingface.co/docs/transformers/model_doc/distilbert
[2] PyTorch. [n. d.]. *AdamW.* Retrieved October 16, 2024 from https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html
[3] PyTorch. [n. d.]. *Optimization.* Retrieved October 16, 2024 from https://huggingface.co/transformers/v3.0.2/main_classes/optimizer_schedules.html
[4] Scikit-Learn. [n. d.]. *SVC.* Retrieved October 16, 2024 from https://scikit-learn.org/dev/modules/generated/sklearn.svm.SVC.html
[5] Scikit-Learn. [n. d.]. *TF-IDF.* Retrieved October 16, 2024 from https://scikit-learn.org/1.5/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html