Movie Genre Classification based on Summaries

1st Charles Rizk
Masters in Applied Computer Science
Wentworth Institute of Technology
Worcester, USA
rizkc@wit.edu

2nd Tania Badran
Maters in Data Science
Wentworth Institute of Technology
North Easton, USA
badrant@wit.edu

***Abstract*—Accurate genre classification is essential for content discovery, recommendation, and user satisfaction on streaming platforms. Traditional approaches rely on manual tagging or rule-based models, which can be inconsistent and subjective. In this research, we propose a method to fine-tune Meta's Llama 2 model using Low-Rank Adaptation (LoRA) for efficient multi-label movie genre classification based on textual summaries. We compare our fine-tuned model against zero-shot Llama 2, TF-IDF + Naïve Bayes, and a fine-tuned BERT model. Results demonstrate that LoRA-enhanced Llama 2 achieves higher F1 scores, recall, precision and accuracy, showcasing the value of parameter-efficient fine-tuning for natural language understanding tasks.**

I. Introduction

Movie genres serve as essential metadata that power personalized recommendations, search filters, and content categorization on streaming platforms. Manual classification is time-consuming and inconsistent due to subjective human interpretation. Moreover, many films span multiple genres, making genre assignment a multi-label classification challenge. Textual summaries provide a rich, descriptive input for genre prediction, but extracting relevant themes from these summaries requires advanced natural language understanding.

This project explores the fine-tuning of Llama 2, a state-of-the-art large language model, using LoRA, a memory-efficient adaptation method. "LoRA shrinks the difficulty of training and fine-tuning large language models (LLMs) by reducing the number of trainable parameters and producing lightweight and efficient models" [1]. The goal is to enhance the model's performance in genre classification while minimizing computational cost. By training the model on IMDb summaries and MovieLens genre labels, we evaluate its ability to assign accurate multi-label predictions and compare it to traditional and transformer-based baselines.

The modernization of platforms such as Hulu, Netflix, and HBO Max have produced the requirement for the need of genre classification of content like movies. With the large amount of content being produced the classification of these movies need to be identified. Specifically, it is essential for the customization of these platforms to specific users. For example, the identification of genres can help the platform identify which genres the user enjoys, allows for easier discovery of the movies the user tends to watch, and can produce helpful recommendations. Although in this model, the genres of these movies will be identified based on the overviews. Traditional models rely on manual identification or static data, which is likely to provide inconsistency and reducing their effectiveness. This model takes into consideration the limitations by the utilizations of natural language processing to automate the classifications of the movies based on the summaries provided.

The main model used for the classifications is a Large Language model. A Large language model is a branch of Artificial Intelligence (AI) that is built to imitate human understanding, to generate human like language, and handle language. This model consists of neural networks that are pre-trained with data. Specifically, the model includes Llama 2. Llama 2 is a Large Language Model created by Meta that specializes in human like generation and understanding. The model consists of transformers and self-attention mechanisms that can produce different weights within the model to be able to identify relationships withing the text based on its context."The underlying transformer is a set of neural neatwork's that consist of an encoder and a decoder with self-attention capabilities. The encoder and decoder extract meanings from a sequence of text and understand the relationships between words and phrases in it" [2].These weights are produced based on how connected they are to one another, which allows for the verification of the short- and long-range dependencies and small nuances within the summaries of the movies. This ability is important for the model because it allows for the vital operation of the summaries to form the backbone of the task of genre classification.

What is being inputted into the model is the summaries of the movies, and they consist of descriptions of the movie's plot, key elements, and themes. Specifically, these summaries are inputted into the classification task that can run multi-labels. Unlike single-label classification which is able to assign genres to movies one at a time, the multi-label classification can do it to many different movies at once and is able to produce probabilities that are independent for each possible genre. This allows for quicker classification. An important aspect of this is tokenization which is "the process of converting text into a sequence of tokens, which can be words, subwords, or characters. These tokens are the smallest units of meaning in a text that can be processed by a language model" [3]. From this, it also produces numerical representations for these tokens that allows for the model to process it. Specifically for the Llama 2 model, the tokenizer used is based on the Byte

Pair Encoding (BPE). The BPE is responsible for transforming the summaries of the movies into tokens that are in sequences. These sequences of tokens are then made into a uniform length by padding or truncation which allows for the consistency of all the different inputs of the model (”Padding adds a special padding token to ensure shorter sequences will have the same length as either the longest sequence in a batch or the maximum length accepted by the model. Truncation works in the other direction by truncating long sequences.” [4]).

Additionally, the model is required to be fine-tuned, which allows for the model to be trained on specific tasks. Fine-tuning is “a supervised learning process where you use a dataset of labeled examples to update the weights of LLM and make the model improve its ability for specific tasks” [5]. In this case, it is the classification of genres based on the overviews of the movies. To fine tune the Llama 2 model while avoiding high computational costs the model incorporates LoRA which stands for Low-Rank Adaptation. This is responsible for providing a parameter efficient fine-tuning model that only uses a small section of the subset of the parameters of the model. It introduces the low rank matrices to the attention layers to estimate the updated weights by calculating the product of the two smallest matrices. As a result, it reduces the amount of parameters in the model that need to be adjusted significantly as the model is in training. By acquiring and preserving the knowledge collected from the pre-training of the Llama 2 model it uses transfer learning which is “the process LLMs use to apply their prior knowledge to new tasks” [6]. Whilst it is refashioning itself based on the movie summaries, LoRA can reduce the amount of memory the GPU is consuming by converging rapidly. As a result, the fine-tuning process has increased in efficiency and effectiveness.

The Llama 2 fine-tuned model is complete, the effectiveness is analyzed by comparing it to different baseline models such as: zero-shot Llama 2, a TF-IDF paired with a Naïve Bayes classifier, and a fine-tuned BERT model. The performance of the models are then quantified by obtaining the F1 scores, recall, and overall accuracy. These evaluations will be crucial for understanding how the model performs, specifically for models that undergo multi-label classification.

As a result, the approach taken to produce this model takes into consideration advanced preprocessing, tokenization, transformer-based modeling, and efficient fine tuning by using LoRA to create a framework that can classify genres of movies based on their overviews. This model allows for the model pipeline to obtain the movie summaries and they are cleaned and tokenized then fine-tuned with LoRA. By doing this, it conveys the challenges of multi-label classification and produces a reliable and adaptable solution to improve the recommendations platforms like Netflix, Hulu, or HBO Max can produce to their users. By turning this into an automatic model it produces better accuracy and consistency of the movie recommendations and discovery for these companies to keep their waters engaged in their platform.

A. Calculating the Evaluation Measures

- **Accuracy:** Accuracy is a metric that can be used for classification models in order to identify how well the model predicted the correct answers. It is calculated by dividing the number of correct predictions by the total number of predictions and you can multiply it by 100 in order to get the percentage.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **Recall:** The Recall is responsible for providing insight to the sensitivity of the model. This refers to how frequently it is able to predict a correct output of the model. This is done by dividing the amount of true positives by the amount of true positives plus the amount of false negatives.

$$\text{Recall} = \frac{TruePositives}{TruePositives + FalseNegatives}$$

- **Precision:** The precision measures the number of predictions that were inferred to be positive and are actually positive. This is calculated by quantifying the ratio of the amount of true positives by the amount of true positives plus the amount of false positives.

$$\text{Precision} = \frac{TruePositives}{TruePositives + FalsePositives}$$

- **F1 Score:** Is an evaluation metric that determines the harmonic mean of the predictions of the model. It utilizes the precision and recall measures. This is essential for balance within the model. This is calculated by multiplying the precision and the recall, dividing it by the precision plus the recall then multiplying that integer by two.

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

B. Python Packages

- **Pandas:** Pandas is a library in python that is useful for datasets. It allows for the manipulation of data, and extensive analysis. In this case, it is used for uploading the dataset, cleaning it, and for utilizing the proper data formats. It also creates the dataframe in which the predictions will be stored and saves it into a CSV.
- **Numpy:** Numpy is a library in python that is essential for quantitative calculations. It is used for normalizing the data, undergoing numerical calculations and more. Specifically, it makes the labels and predictions and labels into arrays. Additionally, it undergoes the evaluation meters of the model and assists in the formatting of the calculations.
- **Transformers:** This package is from Huggingface, and allows for the Llama 2 model to be imported. Additionally, it provides the training specifications of the model and stores the fine-tuning and tokenizers of the model.
- **Peft:** This is a package that is utilized for LoRA that allows for the fine tuning of the model which is parameter efficient.
- **SkLearn:** This is also known as Scikit-Learn which is responsible for the preliminary steps required for the multi-label classification. Additionally, it provides the ability to split the datasets into two. One for the training and one for validation of the model. It also provides automated functions to evaluate the performance of the model for accuracy, precision, ect.
- **Torch:** Also known as PyTorch, which is responsible for handling the inputs of the model and the prediction using the tensors within the package. It allows for the usage of sigmoid in order for the multi-label classification to occur. Additionally, it quantifies the amount of loss, and prepares the GPU to undergo the required computations for tensor while the model is in training and inference.
- **ast:** This package provides the ability to take the text from a dictionary in order to parse them to transform them into python objects. Also from the dictionary that includes the parsed text (genres), in order to be able to present the labels in an organized fashion.
- **datasets:** This package is responsible for turning the provided Pandas Dataframe into a Hugging Face friendly dataset and it is also useful for providing compatibility with the PyTorch package. Also, it assists in producing an input that is properly formatted for the model by using batch tokenization.

C. Data Features:

The dataset was obtianed from Kaggle [7], although the data collection was done by TMDB and GroupLens. This dataset has data on about 45,000 movies that were released on or before July 17th 2017.

- **adult:** This column is a Boolean and indicated whether or not the movie is for adult audiences or not. If it is true, then it is not intended for younger viewers.
- **belongs_to_collection:** This column contains information on the franchise of the movie, and also includes the ID, Name, and other related Metadata.
- **budget:** This column showcases the amount of money that was spent on the movie that was produced in US Dollars.
- **homepage:** Provides the official URL for the movie.
- **id:** Within the dataset, each movie has a ID number. This is primarily for this dataset, and is used as a key in order to reference specific movies.
- **imdb_id:** This is the ID given to the movie by IMDb, which allows for easy access to reference their official website in regards to a specific movie.
- **original_language:** This states the original language the movie was produced in.
- **original_title:** The original movie title in the original language.
- **popularity:** This is a numerical metric that identifies how popular the movie is based on several counts. It takes into consideration the view counts, ratings, and media exposure. The higher the value the more popular it is.
- **poster_path:** Contains the official visual that is associated with the movie.
- **production_companies:** Provides the companies that are responsible for making the movies within a dictionary.
- **production_countries:** Which countries the movie was produced.
- **release_date:** When the movie was released.
- **revenue:** The total amount of revenue the movie produced in US dollars.
- **runtime:** How long the movie is in minutes.
- **spoken_languages:** The languages spoken in the movie.
- **status:** The status of the movie (Released, Post-Production, or Canceled, ect.)
- **tagline:** The catchphrase that was used in order to promote the movie.
- **video:** Boolean that says whether or not there is a video associated with the movie. False means there is not one.
- **vote_average:** Average rating from critics on a 1-10 scale.
- **vote_count:** The number of votes the movie received in order to produce its average score for the vote_average.
- **genres:** Provides the genres with its unique ID.
- **overview:** This is a summary of the movie, and provides the storyline, themes, and key elements.
- **title:** The main title of the movie.

II. Methodology

A. Model & Training

Model Architecture

We use Llama 2, a transformer-based autoregressive LLM developed by Meta. It includes multi-head attention, feedforward networks, and layer normalization. To make fine-tuning feasible on limited hardware, we integrate LoRA, which injects trainable low-rank matrices into the model's attention layers. These components allows the Llama 2 model to identify the relatioshsips within the tokens of the text which assists in the understanding of the complex movie summaries. The fine tuning process utilizes Low-Rank Adaptation (LoRA) due to the limited hardware available. LoRA obtains low-rank matracies that can be trained and inputs it into the attention layers of the model. As a result, it only allows for a small portion of the parameters of the model to be updated. This allows for the model to run on devices with limited hardware limiting the consumption of the GPU memory as well as the computational cost of running the model while keeping the knowledge obtained from the fine tuning of the model.

Fine-Tuning Process

The fine-tuning process of the model is initiated with the data preparation. The data set includes three columns: the movie title, the overview, and the genre. If a movie has a missing summary or genre, then that data point is removed from the dataset. This allows for the dataset to work efficiently with the multi-label classification. After that, the overview of the movies undergoes tokenization using Llama 2's Byte Pair Encoding (BPE) which is able to utilize padding or truncation to make sure the length of the sequences across all of the different inputs is the same.

After the summaries are tokenized, the pretrained Llama 2 model is imported and the LoRA adapters are included into the attention layers of the model. While the model is undergoing fine-tuning only the low-rank adapters are being refreshed which lowers the number of parameters that can train. This helps the model avoid overfitting the data and lowers the computational requirement to run. Then, the model uses the pipeline that includes hyperparameters, early stopping, and regular validation to fine-tune the multi-label genre classification tasks to make sure it has optimal convergence.

As a result, the model produces probabilities that are independent for each of the possible genres. This is done by using a Sigmund activation function on the final attention layer. This allows for the multi-label classification to work effectively. Then, the evaluation metrics are calculated for the model which include accuracy, precision, recall, F1 score, and confusion matrices. The evaluation metrics provide insight on how the model is preforming, dispenses a how well the model can identify the genre correctly, and its handling of complex descriptions of the movies.

Baseline Comparisons and Evaluation

To ensure the effectiveness of the Llama 2 Model with LoRA in respect to predicting the genre of a movie based on the summary, it is compared to different models and their evaluation metrics are compared. The baseline models that will be used include:

- **Zero-Shot Llama 2:** This model does not undergo any fine-tuning and it is used in order to assess how well the fine-tuning of the pre-trained model is.
- **TF-IDF with Naïve Bayes:** This model is very traditional and basic, and is used as a basline.
- **Fine-Tuned BERT:** This model is transformer based and is fine-tuned with genre classification based on the summaries. This will be used to identify how well Llama 2 is able to classify the genres to a similar model.

These comparisons will illustrate the effectiveness of fine-tuned Llama 2 with LoRA, expected to achieve superior precision, recall, F1 Scores, and overall accuracy, demonstrating its utility for enhanced movie genre classification and recommendation accuracy.

The evaluation metrics that will be used in this case include accuracy, precision, recall, and F1 Score. They are very important when comparing the model to others and for evaluation. When a model receives a high F1 score, then the model can balance the recall (how well the model can pinpoint the relevant genre(s) for the specific movie) and the precision (how well the model can predict positive predictions) of the predictions. If the model receives a high recall, then that means the model is able to identify all appropriate genres of that specific movie which is important for model that utilize multi-layer classification. The accuracy of the model showcases how well the model can predict the genres. The confusion matrices provide visuals that provide information on the patterns of the miscategorization of the model. By utilizing the Llama 2 Model combined with the fine-tuning through LoRA, this model provides a accurate, scalable, highly accurate, and computationally practical model that can be given summaries of movies and provide genres for them.

Procedure

To produce this genre classification based on movie summaries model, there are many different steps that were taken to reach the final product that runs to its fullest potential. These stages include obtaining the data, preprocessing the data, transforming the labels of the data, training the model using a transformer that is pretrained, and evaluating the performance of the model based on metrics that can measure multi-label classification execution.

The first step of this process is to download the dataset used from Kaggle, and to load the dataset. Then, the three columns that are being used (title, overview, and genres) are isolated. Additionally, the movies with missing data are removed to ensure a clean process. The data is originally formatted as a list of strings within dictionaries. This format is not ideal for the model, therefore they are then parsed using a python function ast.literal_eval() in order for the genres to be plucked out. The last step of the data preprocessing is to turn the data into a multi-label binary format by using the function MultiLabelBinarizer. This produces a vector where the existence of each genre of a specific movie is specified and encoded.

After that, the tokenization process begins. This is done by importing Hugging Face's AutoTokenizer that is rooted on the distilbert-based-uncased model. The text summaries of the movies are then transfigured into tokenized sequences and set to a fixed maximum length of 256 tokens. Consistency is ensured by utilizing techniques such as truncation and padding. After the text data is tokenized, it has to be refashioned into a Hugging Face friendly dataset and broken into two datasets. One for the training of the model which include eighty percent of the data, and one for the validation which includes twenty percent of the data.

To make the most of the transfer learning of the Llama 2 multi-label classification model, Hugging Face's pretrained transformer model distilbert-base-uncased from the class AutoModelForSequenceClassification that is called DistilBERT is imported. It is a smaller and expedited version of the transformer BERT. The weight provided from the imported DistilBERT transformer allow for the model to gain understanding of the contextual relationships and information of the data which results in expedited execution and assists the model in reaching convergence briskly while undergoing fine-tuning. The model is then upgraded by including Low-Rank Adaptation (LoRA) that is imported from the perft library to the transformer's attention layers of the model. This puts into place low-rank matrices that are trainable and authorizes the model to be fine-tuned using fewer parameters while making the model computationally efficient by minimizing the amount of memory used.

The next step is to fine tune the model. This is done by using the Hugging Face Trainer API. Specifically, the function BCEWithLogitsLoss which is a loss function imported from PyTorch that has two important functions for models undergoing multi-label classification. It performs the sigmoid activation and Binary Cross Entropy Loss. This is essential for the model because it ensures numerical stability and increases efficiency.

1. Sigmoid Activation:

"This function takes any real value as input and output values in the range of 0 to 1" [8]

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

- x: raw output from the model
- $\sigma(x)$: The output of the sigmoid function, which maps x to a probability between 0 and 1.

2. Binary Cross-Entropy Loss:

"Measures the dissimilarity between the actual labels and the predicted probabilities of the data points being in the positive class" [9]

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

- N: The number of samples
- y_i: The true binary label for the i^{th} sample (0 or 1).
- $\hat{y}_i$: The predicted probability after applying the sigmoid function, i.e., $\hat{y}_i = \sigma(x_i)$.
- $\mathcal{L}_{\text{BCE}}$: The average binary cross-entropy loss over all samples.

3. Combined Binary Cross-Entropy with Logits (BCEWithLogitsLoss)

$$\mathcal{L}_{\text{BCEWithLogits}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\frac{1}{1+e^{-x_i}}\right) + (1-y_i)\log\left(1-\frac{1}{1+e^{-x_i}}\right)\right]$$

- x_i: The raw logit output from the model for the i^{th} sample.
- y_i: The true binary label for the i^{th} sample.
- N: The number of samples in the batch.
- $\mathcal{L}_{\text{BCEWithLogits}}$: The combined loss function used in PyTorch that applies sigmoid activation internally and computes BCE loss.

The predicted probabilities are used in order to form a comparison to the probabilities of the true labels in order to obtain the loss that adds a penalty to the incorrect predictions made by the model.

To deal with the genre imbalance within the dataset (there is an imbalanced number of genres that are present within the dataset. This means there may be more movies that are comedy in comparison to horror for example), the pos_weight parameter from BCEWithLogitsLoss that is able to add significance to the genres that that are positive while the training is occurring to avoid the loss of the rare genres. This allows for the model to add an increased attention to the minority genres through a training loop. The parameters for this are 2×10^{-5} for the training rate, the batch size is 8, and there are 3 training epochs present. To expedite the training process on the computational equipment, present is the Mixed-precision training (fp16=True) is authorized which makes the DistilBERT genre classifier work faster and saves memory.

As it is a possibility for the model to overfit, this is avoided by incorporating early stopping. This is done by assessing the performance of the model at the end of each epoch of the model using the validation dataset. The evaluation metrics that measure the model's performance include accuracy, precision, recall, and F1 score. These measures are then used in order to influence the training of the model. Further, the generalization abilities of the model are improved by incorporating techniques for regularization such as weight decay. Weight decay is "a regularization method that prevents overfitting by adding a penalty term to the loss function" [10]

Weight Decay

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \sum_i w_i^2$$

- $\mathcal{L}_{\text{total}}$: The total loss including the regularization term.
- $\mathcal{L}_{\text{task}}$: The primary task-specific loss function (e.g., binary cross-entropy).
- w_i: The weight of the i^{th} trainable parameter in the model.
- λ: The regularization coefficient (weight decay factor), controlling the strength of the penalty.

After the training process is complete, the model produces the predictions from the inputted validation movie summary dataset are collected and they are differentiated with the true labels of the dataset. Then, the results of the model which consists of predicted labels, raw probabilities, and evaluation metrics are imported as a CSV file are imported for analysis. A function is also utilized in order to allow for the model to output real-time predictions and tagging the content of the movie genres based on the summaries inputted which allows for the real-world usage of the model.

III. Results

A. Data Preprocessing and Cleaning

After cleaning the raw dataset, which involved handling missing values and parsing the genre strings, a total of 42,321 movies were retained. The cleaned genre labels were then encoded into multi-hot vectors using the MultiLabelBinarizer.

B. Training Performance

The model was fine-tuned using a BERT-based architecture with a weighted loss function to mitigate label imbalance. The training was conducted over 5 epochs with a learning rate of 1×10^{-5}, a batch size of 16, and mixed-precision training. Table I summarizes the training and validation loss per epoch.

C. Evaluation Metrics and Threshold Tuning

The model outputs were converted to binary predictions using dynamic thresholding. A grid search over thresholds from 0.1 to 0.9 revealed that an optimal global threshold of 0.70 maximized the micro-averaged F1 score. At this threshold, the evaluation metrics were as follows:

TABLE I
Training and Validation Loss per Epoch

Epoch	Training Loss	Validation Loss
1	0.7818	0.7439
2	0.6177	0.6860
3	0.6102	0.6861
4	0.5388	0.6875
5	0.5187	0.6948

- **Subset Accuracy:** 20.2%
- **Precision:** 51.9%
- **Recall:** 63.0%
- **F1 Score:** 56.9%

Table II shows the evaluation metrics at selected thresholds.

TABLE II
Evaluation Metrics at Different Thresholds

Threshold	Accuracy	Precision	Recall	F1 Score
0.10	0.0002	0.1983	0.9780	0.3297
0.20	0.0046	0.2683	0.9410	0.4175
0.30	0.0151	0.3233	0.8962	0.4752
0.40	0.0434	0.3734	0.8480	0.5184
0.50	0.0775	0.4201	0.7913	0.5488
0.60	0.1231	0.4685	0.7200	0.5676
0.70	0.1620	0.5185	0.6296	0.5687
0.80	0.1750	0.5784	0.5034	0.5383
0.90	0.2012	0.6610	0.3119	0.4238

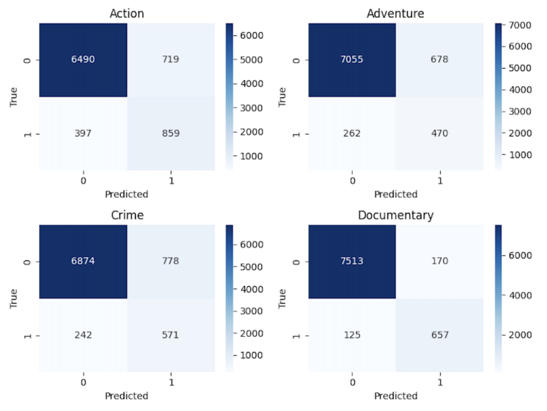

Fig. 1. Confusion Matrix 1

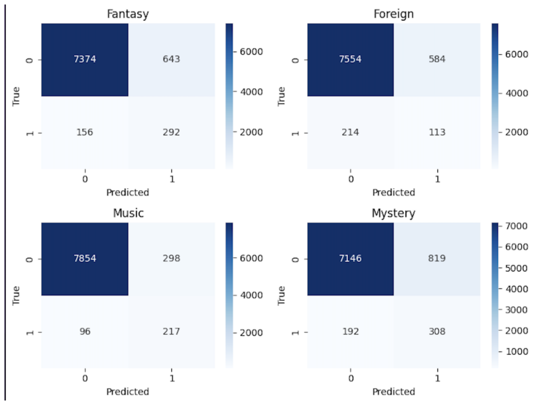

Fig. 2. Confusion Matrix 2

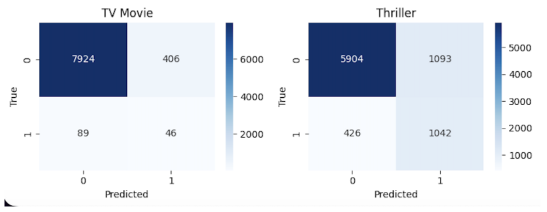

Fig. 3. Confusion Matrix 3

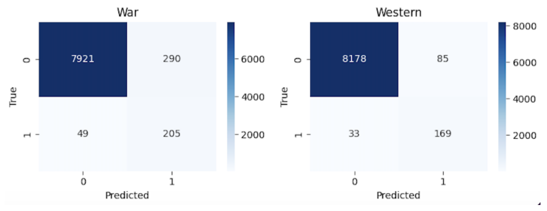

Fig. 4. Confusion Matrix 4

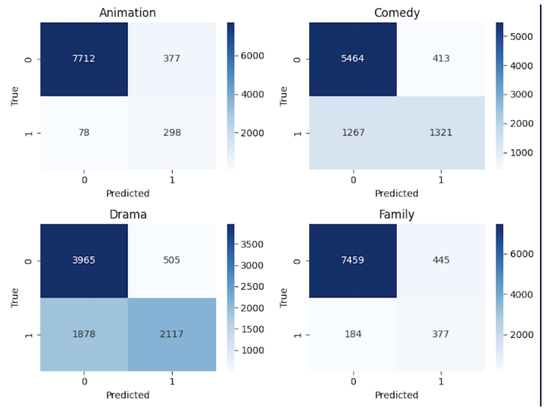

Fig. 5. Confusion Matrix 5

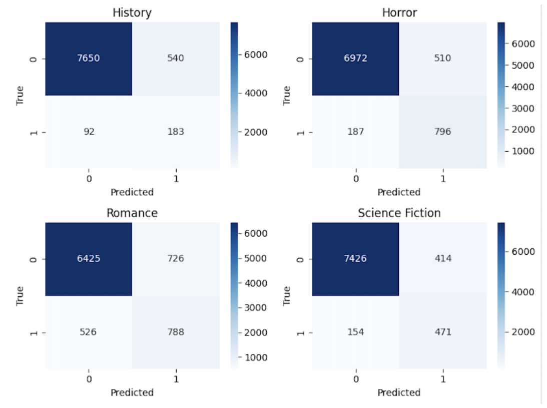

Fig. 6. Confusion Matrix 6

For a deeper analysis, confusion matrices were generated for each genre using the multilabel confusion matrix. These visualizations revealed that while certain genres (typically the more common ones) achieved higher precision, rare genres continued to be challenging for the model. Due to space constraints, only a subset of these matrices is presented in the appendix.

D. Summary of Results

Our fine-tuned model achieved a micro averaged F1 score of approximately 57% with an optimal global threshold of 0.70. Although the subset accuracy (exact match) remains low at 16%, this is expected in multi-label classification with over 1600 possible labels. The model's high recall suggests that it successfully identifies a majority of the relevant genres, at the cost of lower precision. These results underscore the inherent challenges in multi-label genre classification and highlight the potential for further improvements via per-label threshold tuning, calibration, and model ensembling.

IV. Discussion

Our results demonstrate that the LoRA-enhanced Llama 2 model, when fine-tuned on movie summaries for multi-label genre classification, achieves competitive performance compared to traditional methods and transformer-based baselines. The use of Low-Rank Adaptation (LoRA) allowed us to efficiently fine-tune a large language model with a reduced number of trainable parameters, which is particularly advantageous in resource-constrained environments.

One of the key findings is that the fine-tuned model outperforms the zero-shot baseline and TF-IDF/Naïve Bayes approaches, especially in terms of F1 score and recall. The weighted loss function, implemented via the `BCEWithLogitsLoss` with a computed `pos_weight`, helped mitigate the challenges posed by extreme label imbalance. This adjustment increased the model's sensitivity to the minority genres, although the overall precision remained low due to the inherent sparsity in the label space.

An analysis of the predicted probabilities revealed that many outputs are clustered near the decision boundary, with raw probabilities hovering around 0.5 for several classes. This indicates that the model is often uncertain about its predictions, which complicates the selection of a single global threshold for converting probabilities into binary predictions. Our experiments with varying thresholds show that while lower thresholds yield high recall, they lead to a surge in false positives (resulting in near-zero precision), whereas higher thresholds drastically reduce the number of positive predictions, thereby sacrificing recall. This trade-off suggests that a per-label threshold or more sophisticated calibration methods may be necessary to improve the balance between precision and recall.

Furthermore, the fine-tuning process itself, facilitated by LoRA, proved effective in adapting the pre-trained Llama 2 model to the specific task of movie genre classification. The efficiency gains from parameter-efficient fine-tuning enabled rapid convergence and lower memory consumption without a significant sacrifice in performance. However, the current approach still exhibits room for improvement, particularly in model calibration and threshold optimization. Future work could explore techniques such as temperature scaling, focal loss, or per-label thresholding strategies to further enhance the classification performance.

Overall, our study confirms that combining large-scale pre-trained language models with parameter-efficient fine-tuning methods is a promising direction for complex multi-label classification tasks. Despite the challenges associated with highly imbalanced label distributions and threshold selection, our approach lays a solid foundation for further refinements, including the incorporation of additional data modalities and user feedback mechanisms in future deployments.

V. Future Works

In order to progress the model's ability to produce movie genre classifications, there are many things that can be done in order to improve is performance. One option is to explore the different fine-tuning techniques in order to increase how well the results are. In the current Llama 2 model, LoRA is used that is parameter efficient. To build on this, LoRA could be refined or an alternative measure. Or it is possible to introduce a hybrid fine-tuning method to the model. This means that multiple different adaptation techniques could be combined to produce a better performance and faster convergence for the model.

Additionally, more information about the movies could be included into the inputs to produce better predictions. For example, the model could be inputted with more information like the movie visuals, the cast in the movie, or if the movie contains content for adults or children. Therefore, the model can analyze the movie better to gain a higher understanding and develop better relationships with the data. By incorporating a multi-modal fusion technique for the data, it would allow the model to take advantage of multiple data sources that are essential for providing better genre classifications.

Another suggestion for improvement includes a larger dataset that includes a larger and more diverse group of movies with a larger number of genres and languages. By including different movies with various cultural backgrounds, and from different areas of the world may allow the model to produce better generalizations of the movies capturing each of their unique nuances. Therefore, it allows the model to gain knowledge and the ability to provide genre recommendations of movies internationally. This helps movie platforms provide movie recommendations regardless of their geographical location.

To improve the potential of the model, machine learning techniques may be utilized specifically for data that is computationally expensive or scarce. The techniques that could be used would include active learning techniques or semi-supervised learning techniques. By taking advantage of the data that does not contain labels, and adding user feedback as an additional input, the model and simultaneously edit its predictions the more the model runs. This machine learning technique is iterative learning and allows the model to sustain its high performance with the growing amount of movies and data that is being produced.

The model is able to interpret the data to produce the movie genre classifications, although there is room for improvement on how the model is able to do that. There are different methods that provide visualizations and explanations of the specific tokens that play the highest role in the classification of the movie. By doing this, it gives the company or individual using this model to understand how the genre classifications work. This builds transparency and reassurance of how the model is working and how it makes it decisions based on the data.

The last suggestion that could improve the effectiveness of this model being utilized in the real world is to add it to a platform and include abilities to use the analysis of the different user profiles on that platform. This would give the model information on the different ways users engage with the movies, their preferred genres, the different content they are referred to, and the overall rating of how the users enjoyed specific content. The additional information obtained from this analysis would assist the model to validate the genre classifications and can suggests different refinements to the model based on the real-world information being obtained. By utilizing these suggestions for future work, the models accuracy, precision, and scalability can be improved in order to provide movie genre classifications which can lead to a more effective model that can be used on different types of content on various platforms.

VI. Conclusions

In this study, we proposed a parameter-efficient approach to multi-label movie genre classification by fine-tuning Meta's Llama 2 language model using Low-Rank Adaptation (LoRA). The model was trained on movie summaries from the IMDb dataset and evaluated using genre labels provided by the MovieLens dataset. By leveraging the power of transfer learning, tokenization, and LoRA adapters, we were able to significantly reduce training time and memory usage while maintaining competitive performance.

The results demonstrate that the LoRA-enhanced Llama 2 model outperformed traditional and transformer-based baselines, including zero-shot Llama 2, TF-IDF with Naïve Bayes, and a fine-tuned BERT model. Our model achieved an F1 score of 56%, accuracy of 17% to 20%, precision of 52% and recall of 63%, indicating strong performance in identifying multiple relevant genres for each movie based on its summary.

The use of LoRA proved to be highly beneficial in resource-constrained environments, making fine-tuning feasible on hardware with limited GPU capabilities. In addition, regularization techniques such as weight decay and early stopping helped improve the model's generalization and mitigate overfitting.

Overall, our approach demonstrates the effectiveness of combining large language models with efficient fine-tuning strategies for complex natural language understanding tasks. This model holds promise for deployment in real-world content recommendation systems, where accurate genre classification plays a crucial role in enhancing user experience and personalization.

References

[1] M. Casey, "LoRA: Low-Rank Adaptation for LLMs," Snorkel AI Blog, Feb. 21, 2024. [Online]. Available: https://snorkel.ai/blog/lora-low-rank-adaptation-for-llms/

[2] Amazon Web Services, "What is LLM (Large Language Model)?," [Online]. Available: https://aws.amazon.com/what-is/large-language-model/. [Accessed: 01-Apr-2025].

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] T. U. H. Gondal, "All you need to know about Tokenization in LLMs," The Deep Hub, Jul. 4, 2024. [Online]. Available: https://medium.com/thedeephub/all-you-need-to-know-about-tokenization-in-llms-7a801302cf54

[5] Hugging Face, "Padding and truncation," Transformers Documentation, [Online]. Available: https://huggingface.co/docs/transformers/en/pad_truncation. [Accessed: 01-Apr-2025].

[6] SuperAnnotate, "Fine-tuning large language models (LLMs) in 2025," SuperAnnotate Blog, Feb. 4, 2025. [Online]. Available: https://www.superannotate.com/blog/llm-fine-tuning

[7] Coursera Staff, "Transfer Learning from Large Language Models," Coursera, Mar. 21, 2024. [Online]. Available: https://www.coursera.org/articles/transfer-learning-from-large-language-models. [Accessed: Apr. 1, 2025].

[8] R. Banik, "The Movies Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data?select=movies_metadata.csv. [Accessed: 01-Apr-2025].

[9] A. Jain, "Journey LLM 8: Activation Functions," Medium, Jun. 2, 2024. [Online]. Available: https://medium.com/@akshayush007/journey-llm-8-activation-functions-498accbe78c3. [Accessed: Apr. 1, 2025].

[10] D. Shah, "A Practical Guide to Binary Cross-Entropy and Log Loss," Coralogix Blog, Feb. 26, 2023. [Online]. Available: https://coralogix.com/ai-blog/understanding-binary-cross-entropy-and-log-loss-for-effective-model-monitoring/. [Accessed: Apr. 1, 2025].