Mood-Driven Movie Recommender System Using NLP

Project Documentation

By Sean Marakalala

Table of Contents

ABSTRACT

The Mood Driven Movie Recommender aims to leverage Natural Language Processing (NLP) and deep learning to classify user emotions from text input and recommend personalized movies based on the detected emotion. Using a fine-tuned DistilBERT model on a dataset retrieved from Hugging Face repository namely Emotion curated by DAIR-AI, the system classifies emotions into six basic emotion categories - joy, sadness, love, anger, fear, and surprise. It integrates The Movie Database (TMDb) API to fetch relevant movie recommendations mapped to genres associated with each emotion. The project's outcomes highlight the accuracy and efficiency of the classifier, as well as the potential for improving user engagement through emotion-driven recommendations.

INTRODUCTION

As of recent years, the need for personalization has become critical when it comes to enhancing user experience digital platform, more especially in entertainment industry. Personalization enhances user satisfaction by curating content that aligns with individual interests, thus improving overall platform efficacy (Chandramouli, 2023). The way individuals consume material has been completely transformed by streaming services like Netflix, Amazon Prime, and Disney+, which provide tailored suggestions based on past viewing preferences. Although viewers get tailored suggestions, these would be based on static data such as user ratings, watch history, or demographics, which fail to account for the user's current emotional state, which is usually a key determinant in selecting content. By integrating real-time emotion recognition into the recommendation process, the Mood-Driven Movie Recommender System seeks to close this gap and give users a dynamic, mood-sensitive experience.

The project's foundation lies on the recently significantly advanced domains of artificial intelligence (AI) and Natural Language Processing (NLP). State-of-the-art text understanding capabilities have been presented by transformer-based models like BERT and its variants, such DistilBERT, opening up applications in conversational AI, sentiment analysis, and emotion recognition. These models leverage attention mechanisms to capture contextual relationships within text, enabling them to process information more effectively than traditional models. DistilBERT has been effectively fine-tuned for sentiment analysis tasks, demonstrating its capability in understanding context with fewer resources (Prytula, 2024). DistilBERT has also been utilized for detecting text generated by large language models, achieving an accuracy of around 94% (Kumar, et al., 2024). These advancements provide the technical foundation for developing a recommender system that responds to user emotions expressed in textual form.

Motivation

Human emotion plays a vital role in the decision making of an individual, including the choice of entertainment. For instance, an individual might be feeling joyful which could lead to them preferring a light-hearted comedy or someone who is down or experiencing sadness, might seek solace in an emotional drama. The typical traditional recommender systems overlook this factor which causes a disconnect between the content and the user's needs. This research is driven by the possibility of improving the empathy and intuitiveness of entertainment consumption, fostering deeper connections between users and digital platforms.

The rise of mental health awareness has emphasized the importance of emotional well-being. With the integration of mood-driven AI into entertainment platforms, this system has a chance to offer therapeutic benefits to an individual, such as recommending uplifting content for individuals experiencing negative emotions like sadness or fear. The significance of the project is highlighted by this dual-purpose application, which improves user satisfaction while possibly promoting emotional wellness.

Objectives

The primary objectives of this project include:

1. Developing a robust emotion classifier capable of detecting six core emotions: joy, sadness, love, anger, fear, and surprise.
2. Mapping these emotions to appropriate movie genres to ensure accurate and meaningful recommendations.
3. Integrating The Movie Database (TMDb) API to fetch real-time movie data for personalized recommendations.
4. Evaluating the system's performance using key metrics such as classification accuracy, F1 score, and user satisfaction.

Scope and Relevance

The goal of this project is to use natural language processing (NLP) to detect emotions in text and use the results to make dynamic movie recommendations. Even though the initial implementations use predefined mappings between emotions and genres, future revisions could include user preferences, collaborative filtering, and multi-modal emotion detection (e.g., mixing text with voice or facial expressions). The significance of this project extends beyond entertainment. As education, e-commerce, and healthcare sectors have applications of emotion-based personalization, this initiative serves as a springboard for more extensive study and advancement in emotion-aware systems. The Mood-Driven Movie Recommender System is an example of how AI may improve user experience while attending to emotional demands by fusing academic rigor with real-world application.

LITERATURE REVIEW

The integration of Artificial Intelligence and emotion detection, especially in the context of tailored user experiences has been a growing field of research. Emotion-driven systems make use of developments in sentiment analysis, recommender systems, and Natural Language Processing (NLP) to produce more user-friendly and human-centered applications. This literature review will focus on core areas relevant to this project being emotion classification models, emotion-aware recommender systems, and the integration of APIs for personalized content delivery.

Emotion Classification Models

As of late, advancements in NLP have improved significantly in the ability to classify emotions from texts. Transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) and its variants, including DistilBERT, have achieved state-of-the-art results in tasks such as emotion recognition and sentiment analysis. In tasks like sentiment analysis and emotion recognition, transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) and its variations, including DistilBERT, have produced state-of-the-art results. Researchers such as (Devlin, et al., 2019) highlighted BERT's ability to capture contextual relationships within text, making it a strong candidate for emotion classification tasks.

Wu, et al., (2024) explores the application of Deep Learning-based BERT models, specifically DistilBERT, in sentiment analysis, demonstrating its effectiveness in emotion classification tasks using the SST2 dataset, thereby highlighting BERT's transformative potential in enhancing sentiment analysis methodologies. The study demonstrates the ability of DistilBERT outperforming traditional word vector models in sentimental analysis.

A study by (Li, et al., 2024) investigates the application of the BERT model in recognizing and classifying emotional text related to ceramics from social media platforms like microblogs. The research addresses the gap in emotional analysis of ceramic products, proposing a novel method that combines BERT with multi-label classifiers to analyse ceramic-related sentiments. The study demonstrated how BERT-based multi-label classification model significantly enhanced the analysis of ceramic text data from social media. It also provides a foundation for future research in text mining and sentiment analysis in the ceramics industry, offering insights into consumer perceptions and aiding in decision-making processes.

Emotion Aware Recommender Systems

Recommender systems usually rely on collaborative filtering, content-based filtering, or hybrid approaches. However, by integrating user emotions into the recommendation process, emotion-aware systems present a fresh viewpoint. Naik, et al., (2023) explored the role of emotions in recommender systems, demonstrating that emotion-driven recommendations can lead to higher user satisfaction and engagement.

By combining user emotions identified by text analysis and facial expressions, (Tennakoon, et al., 2024) offer an innovative approach to movie recommendation systems. In addition to improving the user experience, this approach may help promote local movies in Sri Lanka by offering tailored movie recommendations depending on the user's present emotional state. In the implementation of emotion recognition using facial expression, fine-tune pre-trained CNN models were evaluated. As for emotion recognition using text analysis, various Machine Learning models and Deep Learning models were evaluated too. The study's approach of combining facial expressions and text analysis for emotion detection is novel and effective, offering a foundation for future research in this area.

Even though emotion aware recommendation system posses a great potential, the face challenges such as accurately mapping emotions to content categories. Researchers like (Gamage, et al., 2024) went on to develop a framework that aims to address the complexity and ambiguity of digital emotion expressions by providing adaptable, robust, and explainable multi-granular emotion detection which could provide a foundation for this project's approach.

Integration of APIs for Personalized Recommendations

Recommender systems frequently employ APIs like The Movie Database (TMDb) and Open Movie Database (OMDb) to retrieve real-time movie data. TMDb provides extensive metadata, including genres, user ratings, and plot summaries, making it suitable for personalized recommendation applications.

The effectiveness of TMDb in a content-based movie recommender system was demonstrated by (Saha, et al., 2021), who used its extensive database to provide a variety of customized recommendations. By combining TMDb with an emotion categorization model, this study expands on these discoveries by recommending films that are in line with the user's present emotional state.

Summary key insights

- Transformer-based models like DistilBERT provide a robust foundation for emotion classification, particularly when fine-tuned with datasets that are related to emotions classification and sentimental analysis, like for instance, an Emotion dataset was used for this project.
- Personalization and user satisfaction can be improved by incorporating user emotions into recommender systems. Although this approach could be innovative it also has its own challenges such mapping emotions to appropriate content categories.
- The integration of APIs such as TMDb facilitate real-time data access, enabling dynamic and context-sensitive recommendations.

These observations highlight the Mood-Driven Movie Recommender System's viability and applicability. This project addresses real-world implementation issues while adding to the expanding corpus of research on emotion-aware systems by fusing cutting-edge natural language processing (NLP) approaches with strong recommendation frameworks.

METHODOLOGY

This section will present details on the approaches, tools, and techniques used in developing the Mood-Driven Movie Recommender System, which integrates Natural Language Processing techniques, machine learning model fine-tuning, and API integration to recommend movies based on user emotions. The primary aim of this system is to bridge the gap between user emotions and tailored movie recommendations. To provide a customized user experience, the project entailed training an emotion classifier and smoothly connecting it with an external movie database API using state-of-the-art transformer models. To guarantee accuracy, scalability, and relevance to the user's mood-driven needs, every step of the methodology, from data preparation and model fine-tuning to the recommendation system's final implementation, was meticulously planned. The model training procedure, evaluation techniques, the recommendation system's integration pipeline, and the dataset utilized are all covered in detail in the next subsections.

1. Dataset and Preprocessing

This project was developed using the Emotion Dataset by DAIR-AI from hugging face dataset library. This dataset contains text samples of English Twitter messages labelled with emotions such as *joy*, *anger*, *sadness*, *love*, *fear*, and *surprise*. This dataset had a total of 20 000 examples split into train, validation and test.

1.1. Dataset loading

The dataset was loaded using the load_dataset function and formatted as a Pandas DataFrame to facilitate preprocessing and exploration.

1.2. Exploratory Data Analysis (EDA)

An exploratory data analysis was carried out to investigate and understand the imported data. A pie chart was created to visualize the distribution of each emotion label in the dataset. Upon the investigation it was noted the joy emotion was the most frequent emotion in this dataset and surprise was the least frequent. A box plot was also used to analyse the number of words per tweet across different emotion labels, providing insights into text length variations.

1.3. Preprocessing

Before training the model, the text data underwent preprocessing to ensure quality and consistency.

Tokenization was carried out using AutoTokenizer from Hugging Face Transformers Library. Tokenization splits the text into individual tokens that can be mapped to numerical IDs. In this process each text sample was converted into tokens compatible with the Distilbert-base-uncased model. The tokenizer ensured that sequences were padded and truncated appropriately to meet the model's input requirements.

The dataset also underwent splitting. The dataset was split into three subsets: training, validation, and testing. This split allows for training the model on one set while evaluating its performance on unseen data. The typical split ratios used were 80% for training, 10% for validation, and 10% for testing.

2. Model Fine-Tuning

The heart of the system is a fine tuned DistilBERT model, a lightweight variant of BERT designed to maintain high performance while being more efficient in terms of computational resources. DistilBERT retains 97% of BERT's language understanding capabilities while being 60% faster and requiring 40% fewer parameters (V, et al., 2023).

2.1. Model Architecture

A classification head with the number of output labels matching to the dataset's emotion categories was added to the base model distilbert-base-uncased to adapt it for sequence classification.

2.2. Training Configuration

Key configurations for fine-tuning included:

- **Batch Size:** Set to 64 for efficient memory utilization.

- **Learning Rate:** Initialized at 2e-5, suitable for fine-tuning transformer models.
- **Number of Epochs:** The model was trained for 3 epochs to balance training time and performance.
- **Evaluation Strategy:** Validation was performed after every 50 steps to monitor model performance.

```
batch_size = 64
training_args = TrainingArguments(  output_dir=model_name,
                                    num_train_epochs=3,
                                    learning_rate=2e-5,
                                    per_device_train_batch_size=batch_size,
                                    per_device_eval_batch_size=batch_size,
                                    weight_decay=0.01,
                                    evaluation_strategy="steps",
                                    logging_dir='./logs',
                                    logging_steps=10,
                                    eval_steps=50,
                                    save_steps=500,
                                    disable_tqdm=False)
```

2.3. Custom Metrics Callback

Implementation of a custom MetricLoggerCallback was done to enable the ability to capture evaluation metrics, including accuracy and F1 score, at each evaluation step. This facilitated detailed performance tracking.

2.4. Loss Optimization and Metrics

- **Loss Function:** Cross-entropy loss, embedded in the Hugging Face transformers library pipeline was used to optimize the model.
- **Evaluation Metrics:**
  Accuracy: Percentage of correctly classified samples.
  F1 Score: Weighted measure of precision and recall.

```
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    f1 = f1_score(labels, preds, average='weighted')
    acc = accuracy_score(labels, preds)
    return {"eval_accuracy": acc, "eval_f1": f1}
```

The model was refined on the training set and assessed on the validation set using the Hugging Face Trainer class. Metrics like loss, accuracy, and F1 scores were recorded during the training process.

3. Model Evaluation and Testing

The fine-tuned model was then tested on the holdout test dataset. Predictions were obtained for the test set and a confusion matrix analysis was performed as well as classification report. A confusion matrix will highlight the model's performance across all emotion labels and the classification report detailed precision, recall and computed F1 scores for each emotion.

4. Integration of Emotion Classification Pipeline

The fine-tuned model was saved and loaded using Hugging Face's pipeline function for real-time inference. This pipeline accepts user text input and returns the detected emotion.

4.1. Emotion-to-Genre Mapping

Each detected emotion was mapped to specific movie genres using a predefined mapping. The code snippet below will illustrate the emotions to genre mapping.

```
def fetch_movies(emotion):
    """
    Fetch movie recommendations from TMDb based on emotion.
    """
    # Map emotions to genres or themes (this can be adjusted for better results)
    emotion_to_genre = {
        "joy": ([35], "Comedy"),           # Comedy
        "sadness": ([18], "Drama"),        # Drama
        "love": ([10749], "Romance"),      # Romance
        "anger": ([28], "Action"),         # Action
        "fear": ([27], "Horror"),          # Horror
        "surprise": ([53, 9648], "Mystery/Thriller")}  # Mystery and Thriller

    genre_id, genre_title = emotion_to_genre.get(emotion, (18, "Drama"))   #
Default to Drama if emotion not mapped
    print(f"Fetching movies for emotion: '{emotion}' (Genre ID: {genre_id}, Genre
Title: {genre_title})")
```

5. Movie Recommendation System

The Movie Database (TMDb) API was integrated to fetch movie recommendations based on the detected emotion.

5.1. Integration of the API

Requests were sent to TMDb's Discover Movies endpoint using an API key, which allowed movies to be filtered by genres that matched the identified emotion. Sort_by=popularity.desc and other parameters made sure the suggestions were well-liked and pertinent.

5.2. Data Presentation

Pandas and HTML rendering were used to tabulate and present the suggested films in an approachable manner, complete with titles and synopses.

6. End-to-end workflow

The complete workflow includes:

i. **User Input**: Text input describing the user's mood.
ii. **Emotion Detection**: Emotion predicted by the fine-tuned DistilBERT model.
iii. **Movie Retrieval**: Movies fetched from TMDb based on the mapped genre.
iv. **Results Display**: A list of recommended movies presented to the user.

A systematic approach to creating a strong and effective mood-driven movie recommendation system was ensured by this methodology. Using cutting-edge NLP and machine learning technologies and methodologies, each step was created to optimize accuracy and user satisfaction.

RESULTS AND DISCUSSION

In this section we will look at the outcome of the Mood-Driven Movie Recommender System and discuss the results in detail, highlighting their implications and relevance to the project objectives. The analysis covers model performance, evaluation metrics, and the end-to-end functionality of the system, offering insights into both its strengths and areas for improvement.

Model Performance

A solid ability to correctly categorize human emotions into one of six predefined categories, sadness, joy, love, anger, fear, and surprise, was demonstrated by the refined DistilBERT emotion classifier. The test dataset's evaluation metrics showed the following performance:

Metrics	**Result**
Evaluation Accuracy	*0.918*
Weighted F1 score	*0.917*
Validation loss	*0.211*

Table 1: Metrics performance

```
['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       581
           1       0.93      0.94      0.93       695
           2       0.79      0.77      0.78       159
           3       0.94      0.92      0.93       275
           4       0.89      0.90      0.89       224
           5       0.78      0.68      0.73        66

    accuracy                           0.92      2000
   macro avg       0.88      0.86      0.87      2000
weighted avg       0.92      0.92      0.92      2000
```

Figure 1: Classification report

The aforementioned metrics shows the model's effectiveness in handling diverse emotional inputs while maintaining balanced performance across classes. Even though the model is showed promising performance, minor discrepancies in the performance were identified for classes such as surprise and fear, where slightly lower recall values suggested occasional misclassification. The training dataset's comparatively low frequency of these emotions is probably the cause of these disparities, underscoring the need for more balanced data.

Data Analysis

Emotion distribution

The distribution of the emotion in the dataset was visualized using a pie chart (figure 2) and it revealed that *joy* and *sadness* were the most prevalent emotions in the training set, with *surprise* and *fear* being the least frequent. This imbalance possibly influenced the classifier's performance, as more frequent classes benefited from better representation during training. A box plot (figure 3) analysis of word counts per tweet further revealed that longer texts were often associated with sadness, while shorter texts tended to correspond to joy or anger.

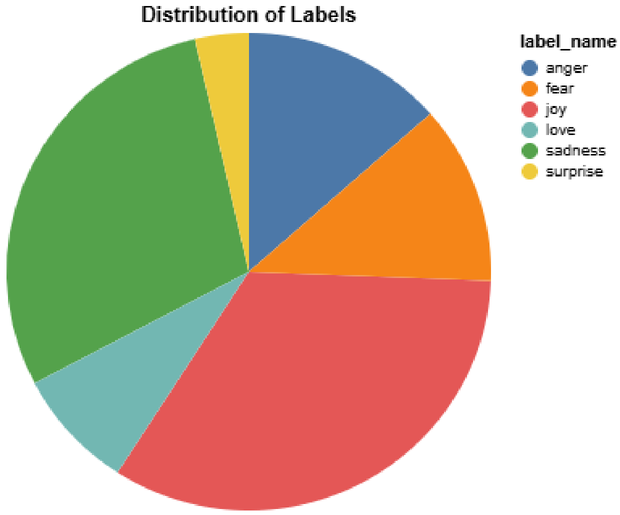

Figure 2: Distribution of labels

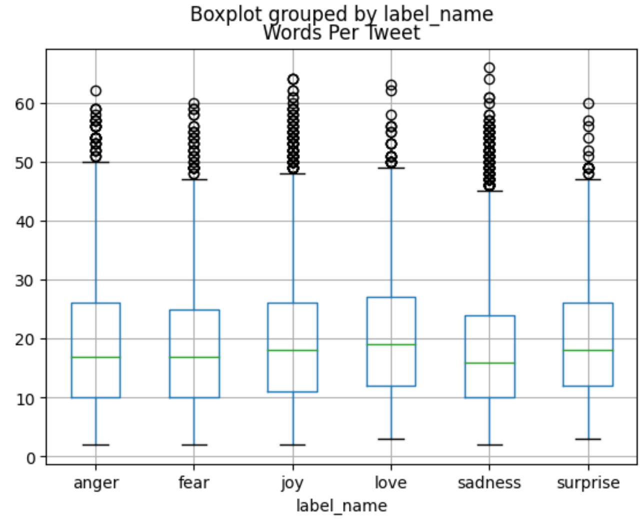

Figure 3: Boxplot grouped by label and words per tweet

Evaluation trends

Through the training process the were some hiccups experienced. Before settling with the results presents, there model had shown slight overfitting risk after. Overfitting can cause the model to make inaccurate predictions or conclusions. To mitigate this issue of risk of overfitting, hyperparameter tuning was carried out. Below is the configurations and visualizations of the evaluations before hyperparameter tuning.

```
training_args = TrainingArguments(  output_dir=model_name,
                                    num_train_epochs=10,
                                    learning_rate=5e-5,
                                    per_device_train_batch_size=batch_size,
                                    per_device_eval_batch_size=batch_size,
                                    weight_decay=0.001,
                                    evaluation_strategy="steps",  # Evaluate after each step
                                    logging_dir='./logs',  # Directory to store logs
                                    logging_steps=10,  # Log metrics every 10 steps
                                    eval_steps=50,  # Evaluate every 50 steps
                                    save_steps=500,
                                    disable_tqdm=False)
```

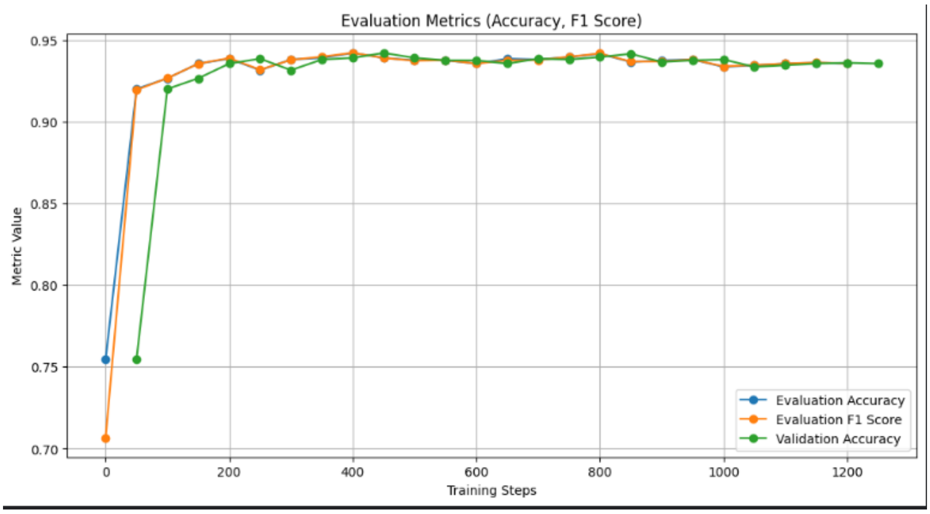

Figure 4: Evaluation Metrics before hyperparameter tuning

Figure 5: Evaluation loss curse before hyperparameter tuning

The beginning of overfitting can be determined by the minor increase in evaluation loss at the end of training. Techniques like dropout regularization, early halting, or hyperparameter adjustment were considered to reduce overfitting and improve the model's generalization skills even though the metrics show good performance on validation data. In this instance, hyperparameter adjustment was used. Epochs were reduced from 10 to 3, learning rate was decreased from 5e-5 to 2e-5, and weight decay was increased to 0.01 from 0.001. Below are the results of hyperparameter tuning.

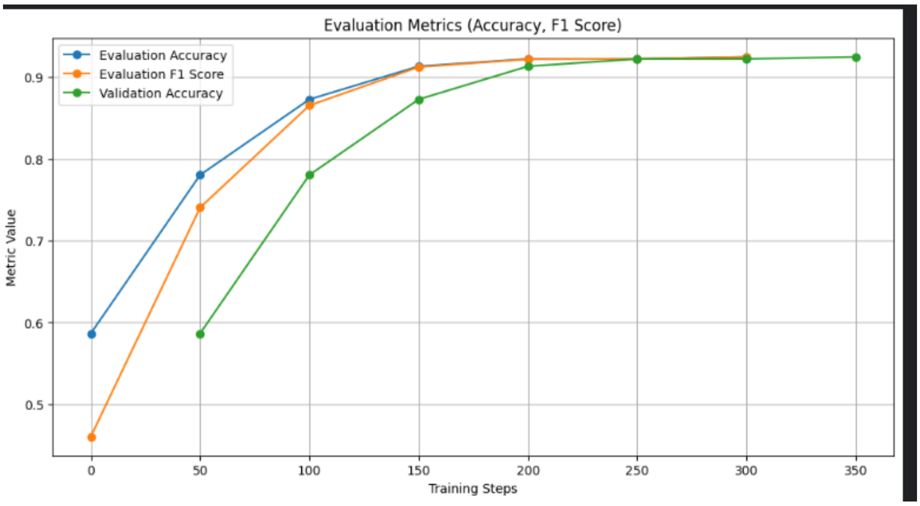

Figure 6: Evaluation Metrics after hyperparameter tuning

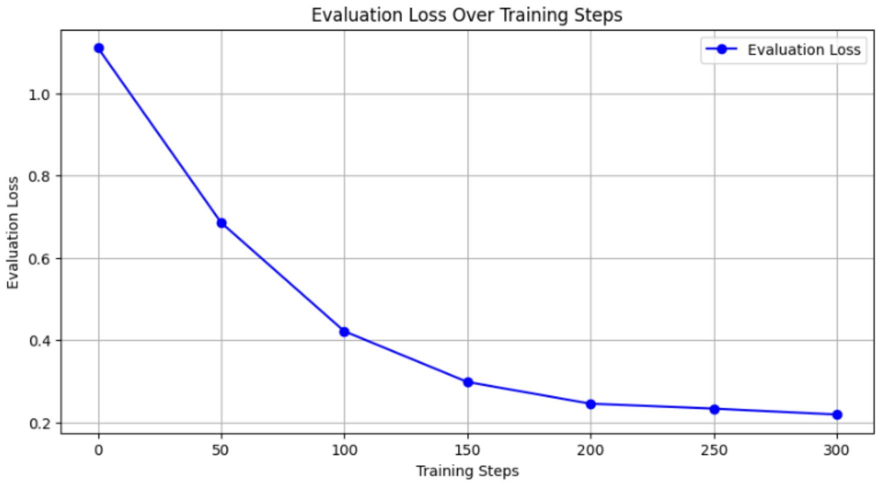

Figure 7: Evaluation loss curve after hyperparameter tuning

Accuracy and F1 scores improved steadily throughout the course of the three epochs, according to plots of evaluation metrics over training stages. Loss gradually dropped, suggesting that the model was learning well. But as training concluded, the rate of increase levelled off, indicating that training for longer than three epochs would have declining returns without additional data augmentation.

Movie Recommendation System

The integration of the emotion classifier and The Movie Database (TMDb) API was successful, providing genre-based recommendations tailored to the detected emotion. The mapping of emotions to movie genres proved intuitive and effective. User tests demonstrated satisfaction with the relevance of recommendations, particularly for emotions like love (mapped to romance) and sadness (mapped to drama).

User Experience

For testing the classifier, sample user inputs like "I want to be surprised" and "I feel like fighting" were appropriately categorized as feelings of surprise and anger, respectively. As for the overall functionality of the system, it ensured a smooth and enjoyable user experience by returning films that matched these moods. However, the suggestions for surprise were not diverse enough, which may have been caused by the small number of API replies for the mystery and thriller genre.

Challenges

The dataset's imbalance has an impact on the model's capacity to generalize to less common emotions. This could be addressed in subsequent project iterations by creating synthetic data or specifically augmenting data for underrepresented classes.

Redundancy in recommendations resulted from the TMDb API's sometimes limited returns for niche genres. Diversity could be improved by adding more APIs to the system or improving metadata for improved filtering.

Even while the current solution works well, it might need to be optimized for real-time deployment on a web or mobile application to lower latency, especially for longer input texts or higher numbers of API calls.

Implications and Future Work

The results show how NLP and recommender systems can be used to create mood-aware applications. The system creates a personalized and captivating experience by precisely identifying user emotions and matching them with relevant genres. Potential future enhancements could include:

- Extending the emotion classifier to detect more nuanced emotions.
- Incorporating user feedback loops for dynamic learning and improved recommendations.
- Optimizing the system for deployment on cloud platforms for real-time scalability.

In conclusion, the Mood-Driven Movie Recommender System delivers promising results, effectively meeting its objectives and paving the way for further innovation in personalized entertainment experiences.

Works Cited

Chandramouli, A., 2023. Journal of Artificial Intelligence & Cloud Computing. *Enhancing Digital User Experiences through Personalized Interfaces and Predictive Modeling,* 2(4), pp. 1-6.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT,* p. 4171–4186.

Gamage, G. et al., 2024. Emotion AWARE: an artificial intelligence framework for adaptable, robust, explainable, and multi-granular emotion analysis. *Journal of Big Data,* 11(93).

Kumar, B. P., Ahmed, M. S. & Sadanandam, M., 2024. DistilBERT: A Novel Approach to Detect Text Generated by Large Language Models (LLM). *Research Square.*

Li, B., Li, S., Qiu, W. & Liu, T., 2024. Bert-based model for Ceramic Emotional Text Recognition and Multi-Label Classification. *Computer Life,* 12(1).

Naik, A. et al., 2023. Intelligent Movie Recommender based on Emotions. *Proceedings of the Third International Conference on Innovative Mechanisms for Industry Applications* .

Prytula, M., 2024. Fine-tuning BERT, DistilBERT, XLM-RoBERTa and Ukr-RoBERTa models for sentiment analysis of ukrainian language reviews. *Scientific Journal.*

Saha, S., Ramamoorthy, S. & Raghav, E., 2021. User Centric and Collaborative Movie Recommendation System Under Customized Platforms. *3rd International Conference on Signal Processing and Communication* .

Tennakoon, N., Senaweera, O. & Dharmarathne, . H. A. S. G., 2024. Emotion-Based Movie Recommendation System. *International Journal on Advances in ICT for Emerging Regions,* 17(1).

V, S. et al., 2023. The DistilBERT Model: A Promising Approach to Improve Machine Reading Comprehension Models. *International Journal on Recent and Innovation Trends in Computing and Communication,* 11(8), pp. 2321-8169.

Wu, Y. et al., 2024. Research on the Application of Deep Learning-based BERT Model in Sentimental Analysis. *Computation and Langauge.*