Movie Genre Prediction from Subtitle Scripts Using LSTM and Natural Language Processing

Om kumar
Department of Computer Science and Engineering
IIIT Vadodara
Vadodara,India
202251081@iiitvadodara.ac.in

Garv Arora
Department of Computer Science and Engineering
IIIT Vadodara
Vadodara,India
202251048@iiitvadodara.ac.in

Arnav Gupta
Department of Computer Science and Engineering
IIIT Vadodara
Vadodara,India
202251023@iiitvadodara.ac.in

Amon Sharma
Department of Computer Science and Engineering
IIIT Vadodara
Vadodara,India
202251015@iiitvadodara.ac.in

***Abstract*—This paper explores the prediction of movie genres from subtitle scripts using deep learning techniques, specifically leveraging Long Short-Term Memory (LSTM) networks for multi-label classification. The project utilizes movie subtitle data in the `.srt` format, with text preprocessing steps including tokenization, stopword removal, and sequence padding. Feature extraction is performed using Keras's `Tokenizer` for converting the text into integer sequences, followed by multi-label binarization of genre labels. The model is trained using a combination of LSTM layers and dense layers, resulting in genre predictions that account for multiple genres simultaneously. The performance of the model is evaluated on a dataset of movie subtitles, achieving accurate and reliable genre predictions. The findings suggest that LSTM-based models are well-suited for subtitle-based genre prediction, and the paper concludes with suggestions for future improvements, such as experimenting with additional deep learning architectures and incorporating more sophisticated feature extraction methods.**

***Index Terms*—Movie Genre Prediction, Natural Language Processing, Subtitle Scripts, Deep Learning, LSTM, Multi-Label Classification, Tokenization, Feature Extraction**

I. Introduction

Movie genre prediction plays a vital role in content recommendation systems, multimedia analysis, and information retrieval. Traditional approaches often rely on metadata, such as director, cast, and visual content like scene analysis or poster recognition. However, subtitle scripts provide a unique text-based perspective that can be leveraged for more accurate genre prediction by analyzing the dialogues, narrative structure, and context within the subtitles. This research focuses on predicting movie genres using subtitle scripts, specifically using a deep learning approach with Long Short-Term Memory (LSTM) networks.

The dataset used in this study consists of movie titles, plots, and genre labels, where each genre is represented as a separate binary column (e.g., Action, Comedy, Drama, Thriller). The plot column contains textual descriptions of the movie's story, which are processed to extract features for model input. Text preprocessing steps, such as tokenization, stopword removal, and padding, are employed to prepare the text data for the model. Feature extraction methods, including TF-IDF and Word2Vec embeddings, are applied to convert the raw text into meaningful numerical representations.

The model, based on an LSTM architecture, is trained on these processed and feature-engineered datasets to classify movies into one or more genres. The system allows users to upload subtitle files in .srt format, which are processed and classified based on the learned model, providing predictions for multiple genres simultaneously. This paper demonstrates the effectiveness of using subtitle scripts for genre prediction, highlighting the potential for more accurate and robust classification using deep learning techniques.

II. Related Work

The use of textual data for genre classification has been explored in various contexts. Previous research has focused on analyzing movie reviews, synopses, and visual content for classification tasks. For instance, Wang et al. used visual features combined with metadata for genre prediction. However, subtitle-based analysis remains relatively underexplored despite subtitles providing direct access to dialogue and narrative elements.

Subtitle scripts have been used for tasks such as sentiment analysis, topic modeling, and language translation; however, their application for genre classification is still nascent. Recent advances in NLP—particularly with the use of deep learning models like Transformers—have opened new avenues for text-based classification. This research aims to fill the gap by leveraging subtitle scripts for genre prediction using logistic regression as a simple yet effective model.

Moreover, existing literature often emphasizes the importance of contextual understanding in text analysis. This project

aims to incorporate such insights by focusing on how dialogue patterns can reflect thematic elements typical of specific genres.

III. Dataset

The dataset utilized in this study is a structured compilation of movie-related data, encompassing three primary components: **movie titles**, **plot descriptions**, and **binary indicators** for various genres. The genres span a diverse range, including but not limited to *Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Fantasy, Horror, Mystery, Romance, Science Fiction, Thriller, and more*. Each entry in the dataset represents a single movie, accompanied by a brief textual synopsis and a series of binary values that indicate whether the movie belongs to a specific genre.

The dataset is organized as a CSV file, with each row corresponding to a unique movie entry and the columns representing the title, plot description, and binary indicators for genres. For instance, the `Comedy` genre is marked with a `1` if a movie belongs to this category and `0` otherwise. An example entry from the dataset is as follows:

- **Title:** #7DaysLater (2013)
- **Plot:** #7DaysLater is an interactive comedy series featuring an ensemble cast of YouTube celebrities. Each week the audience writes the brief via social media for an all-new episode featuring a well-known guest-star. Seven days later that week's episode premieres on TV and across multiple platforms.
- **Genres:** Comedy: `1`, Action: `0`, Drama: `0`, Thriller: `0`, Sci-Fi: `0`, and so on.

The dataset was carefully curated and sourced from publicly available repositories, ensuring a comprehensive representation of movies across a wide range of genres and time periods. It includes a substantial number of entries, providing a robust foundation for training and evaluating machine learning models for genre prediction tasks.

A. Dataset Characteristics

- **Diversity of Genres:** The dataset captures multiple genres, allowing for the study of single-genre movies as well as multi-genre overlaps.
- **Binary Representation:** Each genre is represented by a binary value, ensuring a clear and interpretable categorization for computational analysis.
- **Plot Descriptions:** The plot descriptions are textual summaries that serve as the primary input features for the model.

The inclusion of binary indicators for genres simplifies the task of multi-label classification, enabling efficient prediction of one or more genres for a given movie based on its plot description. This dataset serves as a vital resource for exploring various natural language processing techniques and building predictive models to classify movies into appropriate genres.

B. Preprocessing

The preprocessing stage is a crucial first step in preparing the raw subtitle text for model training. It ensures that the data is cleaned, structured, and standardized, transforming it from its raw format into a form that can be used effectively by the machine learning model. The following steps were performed to achieve this:

- **Text Extraction:** The raw subtitle text was extracted from subtitle files in the .srt format. The extraction process utilized regular expressions to eliminate timestamps, speaker labels, and other non-textual elements that are common in subtitle files. This approach ensured that only the dialogue content, which is the primary focus of genre prediction, remained. By removing irrelevant metadata, the model could concentrate on the actual words spoken, which are more meaningful for understanding the context of the dialogue and its potential genre.
- **Lowercasing:** To maintain consistency across the dataset, all text was converted to lowercase. This step helps standardize the text and prevents the model from treating words like "The" and "the" as distinct entities, which would otherwise lead to redundancy in the feature set. In natural language processing (NLP), case normalization is important because it ensures that the model focuses on the meaning of words rather than their formatting. For example, "Adventure" and "adventure" should be treated as the same word in genre classification.
- **Cleaning Special Characters and Punctuation:** Special characters (e.g., @,) and punctuation marks (e.g., commas, periods, question marks) were removed using regular expressions. These symbols do not typically contribute meaningfully to the task of genre classification and can introduce noise into the data. Furthermore, unnecessary whitespace was trimmed to ensure that the text input was as clean as possible. This step is vital for reducing dimensionality and ensuring that the input data is relevant and focused on the key linguistic features, like actual words, that will help in genre prediction.
- **Stopword Removal:** Stopwords, which are common words that carry little meaningful information (e.g., "and," "the," "is," "of"), were removed using the NLTK library. These words are frequent in most texts but do not contribute significantly to the genre classification task. Removing stopwords is an important step to reduce the noise and dimensionality of the data. It helps the model focus on more meaningful terms that can better differentiate between genres. Additionally, stopword removal aids in improving the model's efficiency by reducing the input size without losing valuable information.
- **Tokenization:** The cleaned text was then tokenized into individual words or tokens using Keras's `Tokenizer`. Tokenization is the process of splitting the text into manageable pieces, typically words, that the model can work with. Each word is represented by an integer, which is mapped to a unique index in a vocabulary.

This transformation is critical because machine learning models cannot process raw text directly; they require numerical input. By converting text into sequences of integers, the model can process the text more efficiently and learn patterns from the dialogues. This step also helps standardize the input format, ensuring that all text data can be uniformly represented and passed into the model.

C. Feature Extraction

Following the preprocessing steps, the extracted features from the subtitle text were transformed into numerical representations suitable for training the machine learning model. The feature extraction process aimed to capture meaningful linguistic and contextual patterns from the dialogues, making them interpretable for the model. The following techniques were used in this phase:

- **Sequence Conversion:** Once the text was tokenized, the sequence of words was converted into integer sequences using Keras's `Tokenizer`. This step is essential for transforming textual data into numerical data that the model can process. Each word in the text was assigned a unique integer, forming a sequence of integers for each movie subtitle. This encoding process ensures that the model is able to handle the dialogue in a numerical format, while preserving the order of words and their contextual relationships.
- **Sequence Padding:** The sequences generated from tokenization varied in length. To address this issue, the sequences were padded using Keras's `pad_sequences` function. Padding ensures that all input sequences have the same length, enabling the LSTM model to process them efficiently. This step ensures that the model's input remains consistent, avoiding issues related to varying sequence lengths and allowing the network to learn without the introduction of bias caused by sequence length discrepancies.
- **Multi-Label Binarization of Genres:** The genre labels for the movies were converted into a binary matrix using `MultiLabelBinarizer`. Each genre was represented by a separate column, where a value of 1 indicated the presence of that genre and 0 indicated its absence. This multi-label binarization process allowed the model to predict multiple genres for each movie, enabling the model to recognize that a movie could belong to more than one genre (e.g., Comedy and Romance) and make more accurate predictions.
- **Embedding Layer:** The tokenized sequences were passed through an embedding layer. Each token (word) in the sequence was mapped to a dense, continuous vector that captured its semantic meaning. This transformation, using pre-trained embeddings or learned during training, helps the model understand the context of each word and its relationships to other words. For instance, words like "love" and "romantic" would have similar representations in the vector space, improving the model's ability to predict genres such as Romance. The embedding layer helped in reducing the dimensionality and capturing the contextual relationships between words, which is crucial for accurate genre prediction.
- **LSTM for Contextual Learning:** Long Short-Term Memory (LSTM) layers were employed to model the sequential dependencies in the subtitle dialogues. LSTM is particularly effective at capturing long-range dependencies within sequences of text, making it ideal for tasks like genre prediction, where the context of words over long distances in the text can be crucial. The LSTM layers enabled the model to learn from the sequential nature of the dialogue, ensuring that the relationship between different parts of the text was maintained. By processing each token in the sequence one by one and maintaining the hidden state, LSTM could capture the context in which words appeared, which was important for genre classification.

D. Model Development

For the movie genre prediction task, a Long Short-Term Memory (LSTM) model was chosen due to its strength in handling sequential data, such as text. The model was designed to process subtitle plot summaries and predict multiple genres simultaneously.

The first step in model development involved preparing the data. The movie plots were tokenized using Keras's `Tokenizer`, which converted the raw text into sequences of integers, with each integer representing a word in the plot. To ensure consistent input size for the model, the sequences were padded to the maximum length of the sequences in the dataset using Keras's `pad_sequences` function.

For the multi-label genre classification task, where each movie can belong to multiple genres, a MultiLabelBinarizer was used to convert the genre labels into a binary matrix. Each genre was represented by a column, where a value of 1 indicated the presence of that genre for a movie, and 0 indicated its absence.

The model itself consisted of the following layers:

- Embedding Layer: This layer was used to learn dense vector representations of words in the movie plot. Each word was mapped to a 100-dimensional vector (using an embedding dimension of 100) that captured the semantic meaning of words.
- LSTM Layers: Two LSTM layers were used to capture the sequential nature of the dialogue. The first LSTM layer had 128 units and returned sequences, allowing the second LSTM layer (with 64 units) to learn further contextual dependencies from the sequence.
- Dense Layer: After the LSTM layers, a dense layer with 64 units and ReLU activation was used to capture non-linear relationships in the data.
- Output Layer: The output layer consisted of a number of units equal to the number of genres, with a sigmoid activation function used to predict the presence of each genre. This allowed the model to output probabilities for each genre, suitable for multi-label classification.

The model was compiled using the Adam optimizer and binary cross-entropy loss function, as the task involves multi-label binary classification. The model was trained using the padded sequences of movie plots and the corresponding binary genre labels, with 80% of the data used for training and 20% for validation.

To ensure the model's performance, it was evaluated based on accuracy during training. After the training was complete, the model, along with the tokenizer and MultiLabelBinarizer, was saved for future use. The saved files allow for easy deployment and prediction on new data.

Future improvements could include:

- Hyperparameter Tuning: Exploring different configurations for the LSTM layers, such as the number of units, or experimenting with different optimizers.
- Attention Mechanisms: Implementing attention layers to allow the model to focus on more relevant parts of the plot, which may improve genre prediction accuracy.
- Ensemble Methods: Testing ensemble techniques, like Random Forests or Gradient Boosting, to combine predictions from multiple models for improved accuracy.

E. System Workflow

The system follows a well-defined workflow to ensure efficient and accurate genre predictions based on movie subtitle scripts. The process is outlined as follows:

- **Subtitle Script Upload:** The user begins by uploading a subtitle script in the `.srt` format. This script contains the movie dialogue, which forms the basis for genre prediction.
- **Text Extraction and Preprocessing:** The system extracts the dialogue text from the subtitle file, removing timestamps and other non-relevant metadata. The text is then preprocessed by converting it to lowercase, eliminating special characters and punctuation, and tokenizing the words into sequences. Stopwords (e.g., "the," "and," "is") are removed to reduce noise, and the final text is prepared for feature extraction.
- **Feature Extraction:** After preprocessing, the system tokenizes the text using Keras's `Tokenizer` and converts it into integer sequences. These sequences are padded to ensure uniform input length. Multi-label binarization is applied to the genre labels, preparing the data for multi-label classification.
- **Genre Prediction with LSTM Model:** The processed and padded sequences are fed into the pre-trained LSTM model. The model, which has been trained to recognize patterns in the dialogue text and relate them to movie genres, predicts the presence of each genre. The output is a set of probabilities, each corresponding to a specific genre.
- **Output Display:** The system displays the predicted genre(s) along with the probabilities for each genre, providing users with a comprehensive understanding of the predictions. Multiple genres may be predicted for each movie, reflecting the multi-label nature of the task.

This workflow ensures that the system provides accurate genre predictions efficiently, with minimal input required from the user, thus enhancing the overall user experience.

IV. Results and Evaluation

The LSTM-based model was trained to predict movie genres based on subtitle scripts. The model leveraged an embedding layer and two stacked LSTM layers to capture sequential dependencies in text, followed by dense layers for multi-label classification. Each subtitle script was tokenized and padded to ensure uniform input dimensions for the model.

A. Model Performance

The model's performance was evaluated using metrics tailored to multi-label classification, including accuracy, precision, recall, and F1-score. Since the model predicted up to three genres for each subtitle script, precision and recall were calculated based on the overlap between predicted and true genres. Table I summarizes the performance metrics:

TABLE I
Performance Metrics for Genre Prediction

Metric	Value
Accuracy (Exact Match Ratio)	0.34
Precision	0.42
Recall	0.39
F1-Score	0.40

The model achieved an accuracy of 34%, which indicates that the predicted genre sets matched the actual sets exactly in approximately one-third of the cases. The precision and recall scores reflect the model's ability to correctly identify genres among its predictions, with an overall F1-score of 0.40, balancing precision and recall.

B. Error Analysis

The model performed well for genres with distinct linguistic patterns, such as *Horror* and *Action*, but struggled with genres that exhibit significant overlap in vocabulary, such as *Comedy* and *Romance*. Additionally, genres with limited representation in the dataset, such as *Film-Noir*, were rarely predicted correctly.

C. Predictions Analysis

- Single Genre Predictions: The model successfully predicted a single dominant genre for scripts with clear thematic content. For example, subtitle scripts with frequent usage of suspenseful or ominous vocabulary were correctly classified as *Horror*.
- Multiple Genre Predictions: For scripts with overlapping themes, the model often predicted up to three genres. For instance, an *Adventure* movie with strong emotional dialogues occasionally included predictions for *Romance*.
- Challenges: Subtitles with generic vocabulary or dialogue lacking specific linguistic cues posed challenges for accurate predictions.

D. Discussion

The results demonstrate the effectiveness of LSTM-based models for subtitle-driven genre classification. However, the following limitations were identified:

- **Imbalanced Dataset:** Underrepresented genres were challenging to classify, which impacted recall.
- **Vocabulary Overlap:** Similarities in vocabulary between certain genres led to genre confusion.
- **Exact Match Limitation:** The exact match accuracy metric penalizes partially correct predictions, which may undervalue the model's ability to capture some genre components.

E. Future Improvements

To enhance performance:

- **Data Augmentation:** Increase the representation of underrepresented genres through synthetic data generation or external datasets.
- **Model Enhancements:** Explore advanced architectures such as transformer models to capture nuanced contextual relationships.
- **Fine-Tuning:** Adjust the threshold for genre predictions to balance precision and recall for practical applications.

Overall, the LSTM-based model provides a robust foundation for genre prediction, with scope for further refinement to address challenges related to dataset imbalances and overlapping genre features.

REFERENCES

[1] H. Guo, "Generating text with deep reinforcement learning," 2015.

[2] V. S. Jagjeet Singh, "Movie genre prediction using deep learning," *International journal of advances in engineering and management (IJAEM)*, pp. 1–5, 2021.

[3] M. M. Hasan, S. T. Dip, T. Rahman, M. S. Akter, and I. Salehin, "Multilabel movie genre classification from movie subtitle: Parameter optimized hybrid classifier," in *2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pp. 1–6, IEEE, 2021.

[4] R. B. Mangolin, R. M. Pereira, A. S. Britto Jr, C. N. Silla Jr, V. D. Feltrim, D. Bertolini, and Y. M. Costa, "A multimodal approach for multi-label movie genre classification," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19071–19096, 2022.

[1] [2] [3] [4]