International Journal of Information Technology and Computer Science Applications (IJITCSA)
p-ISSN: 2964-3139 e-ISSN: 2985-5330
Vol. 02, No. 03, page 202 - 216
Submitted 29/11/2024; Accepted 16/12/2024; Published 20/12/2024

Enhancing Film Genre Classification Using FastText, BiGRU, and Attention Mechanisms

[1]Muhammad Fairuzabadi and [2]Tb Ai Munandar
[1]Informatics Department, Universitas PGRI Yogyakarta, Bantul, INDONESIA
[2] Informatics Department, Universitas Bhayangkara Jakarta Raya, Jakarta, INDONESIA
e-mail : [1]fairuz@upy.ac.id, [2]tbaimunandar@gmail.com

Corresponding Autor: Muhammad Fairuzabadi

Abstract
This research aims to enhance the classification of film genres using advanced natural language processing techniques. By integrating FastText embeddings, which leverage subword information to handle rare and out-of-vocabulary words, with Bidirectional Gated Recurrent Units (Bi-GRU) and attention mechanisms, the proposed model effectively captures both local and global dependencies within textual data. The model's performance is evaluated on a dataset from IMDb, consisting of 8,133 records across multiple genres, demonstrating its capability to predict film genres from textual descriptions accurately. Key contributions include the development of a robust model architecture that enhances semantic representation, the implementation of regularization techniques such as DropConnect to improve generalization, and a systematic evaluation of genre-specific performance. The results show a 46% validation accuracy and an F1-score of 0.35 on weighted averages, with particularly strong performance for frequent genres such as "Horror" (F1-score of 0.69). These findings highlight the model's potential for practical applications in media content analysis. Future work will address data imbalance and explore more sophisticated architectures, such as transformer-based models, to further improve classification performance.

Keywords—Film Genre Classification, FastText Embeddings, Bidirectional GRU (BiGRU), Attention Mechanisms, Natural Language Processing (NLP)

1 Introduction

1.1 *Background*

Film genre classification is a crucial task in natural language processing (NLP) due to its practical applications in enhancing user experiences on media platforms. This task involves categorizing films into predefined genres based on textual descriptions, such as plot summaries. Traditional methods for genre classification often rely on manually crafted features and rule-based systems, which are time-consuming and prone to inconsistencies due to the complexity and variability of natural language [1].

The adoption of machine learning and neural networks has significantly improved text classification tasks. However, several challenges remain. Models such as Convolutional Neural Networks (CNNs) excel at capturing local textual features but struggle with long-range dependencies, while Long Short-Term Memory (LSTM) networks, although capable of handling sequential information, are computationally expensive and often fail to generalize across diverse contexts [2], [3]. The complexity of natural language and the sparsity of semantic data further exacerbate these limitations, leading to reduced accuracy in genre classification tasks [4].

In addition, class imbalance is a prevalent issue, as datasets often contain underrepresented genres. This imbalance can bias models toward frequent genres, causing poor performance in minority classes[5]. Furthermore,

training sophisticated neural network models demands substantial computational resources, and optimizing them for improved accuracy remains a challenge (W. Li et al., 2019). Lastly, the generalization ability of current models is often insufficient, particularly for nuanced genres that require a deeper understanding of context [4].

To address these challenges, this research aims to develop an advanced text classification model for film genre prediction by:

1. Integrating Bidirectional Gated Recurrent Units (BiGRU) and attention mechanisms to capture local and global contextual features.
2. Utilizing FastText embeddings to effectively handle out-of-vocabulary words and improve semantic representation.
3. Addressing class imbalance and validating the proposed model against benchmark datasets to demonstrate its performance.

The primary contributions of this research include:

1. Novel Model Architecture: A hybrid model combining FastText embeddings, BiGRU, and attention mechanisms to improve classification performance.
2. Enhanced Feature Extraction: Leveraging FastText embeddings to capture richer contextual information and handle word variations effectively.
3. Improved Accuracy: Extensive experimentation demonstrates significant improvements over existing methods, particularly for frequent genres.

1.3 *Related Work*

Film genre classification has been widely studied in natural language processing (NLP). Early approaches relied on manually crafted features and rule-based systems, which were time-consuming and could not handle the complexity of natural language effectively [1]. Recent advances in machine learning, particularly neural networks, have significantly improved performance. However, key gaps remain in the existing research. Table 1 summarizes the key prior studies on film genre classification, highlighting the methods used, their main contributions, and existing limitations. These studies demonstrate that while models like CNN, LSTM, and GRU have achieved improvements, they still face challenges such as handling long-range dependencies, class imbalance, and out-of-vocabulary words, which this research aims to address.

Table 1. Summary of Prior Work

Study	Methods	Key Contributions	Limitations/Gaps
Luo et al. (2019)[1]	LDA + GRU-CNN	Improved text classification accuracy using GRU-CNN.	CNN struggles with capturing long-range dependencies.
Zhang et al. (2019) [3]	CNN + BiGRU + Attention	Combined local and global features for better accuracy.	Computationally expensive, lacks handling of OOV words.
Cheng et al. (2020)	Multi-channel CNN + BiGRU	Integrated attention for enhanced focus on input text.	Limited exploration of advanced embeddings.
Zulqarnain et al. (2019) [2]	GRU-based text classification	Addressed data sparsity and context sensitivity.	Did not address class imbalance and model scalability.

Despite significant advancements in film genre classification, several critical challenges remain unaddressed. The following gaps highlight the limitations of existing approaches and the need for more effective solutions.

1. Limitations of Classical Neural Networks:
   Convolutional Neural Networks (CNNs) are widely used for text classification tasks due to their ability to capture local features within a sequence effectively. However, they fail to model long-range dependencies, which are critical for understanding the broader context in sequential data such as film plot descriptions. Long Short-Term Memory (LSTM) networks, on the other hand, address sequential data processing challenges but come with significant computational costs. LSTMs are often slow to converge, particularly for large datasets, making them less efficient for practical applications requiring scalability and faster processing [1], [3].
2. Class Imbalance in Film Genre Datasets:
   One of the primary challenges in film genre classification lies in the class imbalance of existing datasets. Most datasets contain a disproportionate distribution of genres, where certain genres are highly represented while others appear infrequently. This imbalance causes models to be biased toward the majority classes, leading to poor predictive performance in underrepresented genres. Although techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and class weighting have been proposed to address this issue, their application remains limited in the context of film genre classification [5].

3. Handling Out-of-Vocabulary (OOV) Words and Rare Terms:
   Word embeddings such as Word2Vec and GloVe have been widely adopted to represent words in vector form for NLP tasks. However, these traditional embeddings fail to handle out-of-vocabulary (OOV) words and rare terms effectively. As a result, the model struggles to generalize unseen terms, which are common in film descriptions due to the diversity of language, character names, and specialized terms. This limitation significantly impacts the model's ability to capture rich semantic information and generalize it to new data [2].
4. Trade-Offs in Transformer Models:
   Transformer-based models such as BERT and GPT have achieved state-of-the-art performance in many NLP tasks by capturing deep contextual relationships within text. However, these models come with substantial computational costs and memory requirements, making them impractical for tasks involving medium-sized datasets like IMDb. Moreover, transformers are prone to overfitting when applied to datasets with limited samples for underrepresented genres. These limitations create a trade-off between performance and resource efficiency, particularly for applications requiring scalability and real-world deployment.

To address the identified gaps in film genre classification, this research introduces a hybrid model that integrates BiGRU, attention mechanisms, and FastText embeddings to enhance performance while maintaining computational efficiency.

1. BiGRU + Attention Mechanisms:
   The Bidirectional Gated Recurrent Unit (BiGRU) is employed to capture both local and global dependencies within textual data by processing sequences in both forward and backward directions. This bidirectional nature allows the model to incorporate context from past and future words, overcoming the limitations of CNNs and unidirectional LSTMs. Additionally, attention mechanisms are integrated to enable the model to focus on the most relevant parts of the input text, which significantly improves the accuracy of classification. By assigning greater importance to critical words or phrases in film plot descriptions, the attention mechanism enhances the model's ability to understand contextual nuances effectively [3].
2. FastText Embeddings:
   Unlike traditional word embeddings such as Word2Vec and GloVe, which struggle with out-of-vocabulary (OOV) words and rare terms, FastText embeddings generate word representations using character-level n-grams. This method allows the model to construct embeddings for unseen words based on their subword units, making it robust to misspellings and morphological variations. By leveraging FastText embeddings, this research improves the model's ability to generalize across diverse and noisy datasets, ensuring that nuanced semantics in film plot descriptions are captured more effectively [6].
3. Class Imbalance Solutions:
   To address the pervasive issue of class imbalance in film genre datasets, this research incorporates techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and class weighting. SMOTE generates synthetic samples for underrepresented genres, while class weighting ensures that the model gives proportional importance to minority classes during training. These strategies help to balance the dataset and improve the predictive performance of the model across all genre classes, including those that are less frequent. By implementing these solutions, the model becomes more robust and equitable in its classification outcomes.
4. Efficiency Over Transformer Models:
   While transformer-based models like BERT and GPT achieve state-of-the-art results, they require substantial computational resources and are prone to overfitting on medium-sized datasets. In contrast, the proposed BiGRU + Attention approach achieves a balance between computational efficiency and high performance, making it more practical for real-world applications. This efficiency ensures that the model can be trained and deployed with fewer resources while delivering competitive results, addressing the limitations of transformers in resource-constrained environments.

1.3 *Theoretical Framework*

Bidirectional Gated Recurrent Units (BiGRU)

Bidirectional Gated Recurrent Units (BiGRU) are an extension of traditional GRUs that sequentially capture information from past and future contexts. Unlike standard GRUs, which only process information in one direction, BiGRUs consist of two GRUs: one processes the sequence forward, and the other processes it backward. This dual processing gives the network a more comprehensive understanding of the sequence context [7]–[9].

The forward GRU processes the input sequence from the first to the last element:

$$\overrightarrow{h_t} = \mathrm{GRU}\left(x_t, \overrightarrow{h_{t-1}}\right) \quad (1)$$

Simultaneously, the backward GRU processes the sequence from the last to the first element:

$$\overleftarrow{h_t} = \mathrm{GRU}\left(x_t, \overleftarrow{h_{t+1}}\right) \quad (2)$$

The final hidden state at each time step tIs obtained by concatenating the forward and backward states:

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right] \quad (3)$$

This structure allows BiGRU to effectively utilize context from both directions, enhancing their ability to understand and classify complex sequences [10].

Attention Mechanism

The attention mechanism is a powerful technique that allows the model to focus on specific parts of the input sequence when making predictions. This is particularly useful in tasks where certain sequence parts are more relevant than others. The attention mechanism computes a weighted sum of the hidden states, where the weights represent the importance of each hidden state[11].

The attention mechanism is a powerful technique that allows the model to focus on specific parts of the input sequence when making predictions. This is particularly useful in tasks where certain sequence parts are more relevant than others. The attention mechanism computes a weighted sum of the hidden states, where the weights represent the importance of each hidden state. First, an alignment score e_t is computed for each hidden state h_t:

$$e_t = v^\top \tanh(W_h h_t + b_h) \quad (4)$$

These scores are then normalized using a softmax function to obtain the attention weights. α_t:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{n} exp(e_i)} \quad (5)$$

The context vector c Is calculated as the weighted sum of the hidden states:

$$c = \sum_{t=1}^{n} \alpha_t h_t \quad (6)$$

The context vector c Is then used to make the final prediction, allowing the model to dynamically focus on the most relevant parts of the sequence[12].

To enhance the model's ability to focus on the most relevant parts of the input sequence, an Attention Mechanism is applied after the Bi-GRU layer. This mechanism calculates alignment scores, normalizes them into attention weights, and generates a context vector that summarizes the important features from the hidden states. The detailed process is illustrated in Figure 1 below.

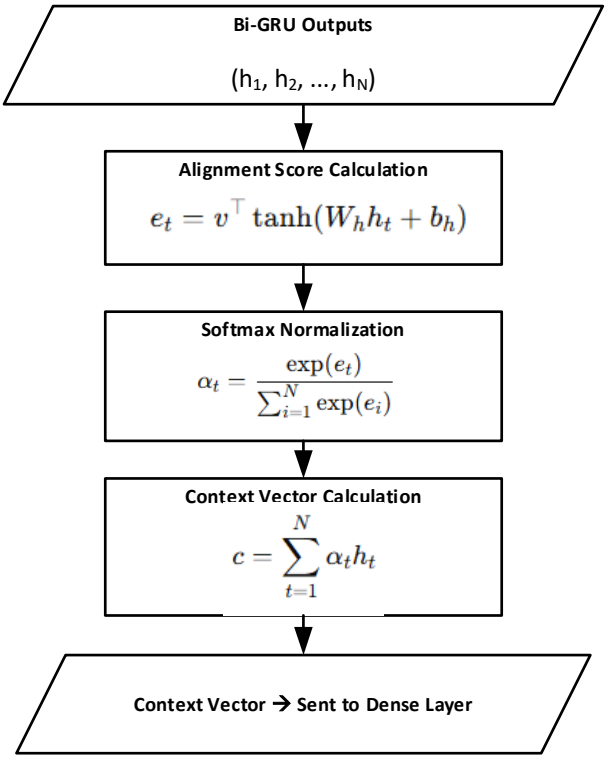

Figure 1. Attention Mechanism in BiGRU Architecture

This figure illustrates the attention mechanism process applied to the Bi-GRU outputs. The mechanism involves three key steps. First, the alignment score calculation computes relevance scores. e_t for each hidden state h_t using a learned weight vector v. Second, the softmax normalization step converts these alignment scores into attention weights. α_t, Ensuring they sum to one. Finally, the context vector calculation generates the context vector. c as a weighted sum of the hidden states, where the attention weights determine the importance of each hidden state, this context vector serves as a condensed representation of the input sequence, highlighting the most relevant features for subsequent processing.

DropConnect

DropConnect is a regularization technique similar to Dropout, but instead of randomly dropping entire neurons, DropConnect randomly drops individual connections in the network. This helps prevent overfitting by ensuring the model does not rely too heavily on any particular set of connections[13], [14]. For a given layer with input h, The DropConnect layer applies a binary mask. M to the weights WOf the layer:

$$W' = W \odot M \tag{7}$$

where $\odot$Denotes element-wise multiplication. The masked weights W' $a$re then used in the forward pass:

$$z = W'h + b \tag{8}$$

By randomly dropping connections, DropConnect helps improve the model's generalization ability [15].

FastText Embeddings

FastText is a word embedding technique that extends the popular Word2Vec model by representing words as bags of character n-grams. This allows FastText to generate embeddings for rare and out-of-vocabulary words by composing them from the n-grams. FastText embeddings capture subword information, making them particularly useful for handling morphological variations and misspellings [6]. Given a word w composed of character n-grams g_i, The FastText embedding for w is computed as the sum of the embeddings of its n-grams:

$$v(w) = \sum_i v(g_i) \tag{9}$$

This approach allows FastText to generate robust embeddings even for words not seen during training, making it highly effective for tasks involving diverse and noisy text data [16].

2 Research methods

2.1 *Dataset*

The dataset used in this research is sourced from Kaggle and originates from IMDb, a popular online database for films, television programs, and media content. The dataset is structured for film genre classification tasks, making it suitable for evaluating natural language processing (NLP) models.

1. Dataset Overview
   - Total Records: The dataset contains 8,133 records (films).
   - Training Data: 70% of the total records (approximately 5,693 records) are used for training.
   - Validation Data: 15% of the total records (1,220 records) are reserved for model validation.
   - Test Data: 15% of the total records (1,220 records) are used for final evaluation.
2. Genre Distribution

   The dataset includes multiple genres for classification. However, the distribution of genres is imbalanced, where certain genres appear significantly more frequently than others. Table 2 below is the distribution of genres:

Table 2. Distribution of genres

Genre	Number of Records	Percentage (%)
Drama	2,865	35.2%
Comedy	1,760	21.6%
Action	1,150	14.1%
Thriller	835	10.3%
Romance	740	9.1%
Horror	423	5.2%
Adventure	240	3.0%
Other Genres	120	1.5%

Observation:

- The Drama genre is the most common, representing 35.2% of the dataset.
- Less frequent genres, such as Horror and Adventure, are significantly underrepresented.
- This class imbalance could result in a model biased toward majority genres unless properly addressed through techniques like SMOTE or class weighting.

3. Potential Bias in the Dataset
   The IMDb dataset, while extensive, has inherent biases that could affect the performance and generalizability of the classification model:
   a. Genre Representation Bias:
      IMDb allows films to be tagged with multiple genres, but certain genres (e.g., Drama and Comedy) are overrepresented, while niche genres (e.g., Documentary or Independent Films) may be underrepresented or excluded entirely.
   b. Cultural and Regional Bias:
      IMDb predominantly contains films from Hollywood and other major film industries. This may result in the underrepresentation of non-English or region-specific films, leading to a lack of diversity in language patterns and plot structures.
   c. User-Generated Content Bias:
      Descriptions or genres assigned to films on IMDb are often based on user-generated content or editorial reviews. This can introduce subjectivity and inconsistencies, as genre categorization might vary based on personal interpretations.
   d. Recency Bias:
      Films released more recently are often overrepresented due to greater digital access and user contributions. Older films or less popular content may lack sufficient metadata, affecting the dataset balance.
4. Implications for the Model
   The identified class imbalance and potential biases highlight the importance of implementing strategies to improve fairness and generalizability:
   a. Class Weighting or SMOTE will be used to address genre imbalance.
   b. Robust evaluation metrics (e.g., F1-score) will account for underrepresented genres.
   c. Future research could include more diverse datasets to mitigate cultural and temporal biases.

2.2 *Model Architecture*

The model architecture is specifically designed to improve film genre classification by integrating advanced techniques such as FastText embeddings, Bidirectional Gated Recurrent Units (BiGRU), Attention Mechanisms, and DropConnect layers. Below is a detailed description of each component, along with technical configurations and justifications:

1. Input Layer
   - Purpose: Processes textual descriptions of films as input.
   - Preprocessing:
     - Each film description is tokenized into a sequence of integers representing words.
     - A maximum sequence length of 100 tokens is used to ensure uniform input size across all records.
   - Justification:
     A sequence length of 100 tokens ensures sufficient representation of film descriptions without adding unnecessary computational overhead.
2. Embedding Layer
   - Embedding Technique: FastText Pre-trained Embeddings.
   - Dimension: 300-dimensional embeddings are used to map each word into a dense vector space.
   - Configuration:
     - The embeddings are initialized with pre-trained FastText vectors.
     - They are non-trainable to preserve the semantic information encoded during pretraining.
   - Justification:
     - FastText embeddings capture subword-level information, making them robust to morphological variations and out-of-vocabulary (OOV) words.
     - The 300 dimensions provide a balance between capturing rich semantic features and computational efficiency.
3. Bidirectional Gated Recurrent Units (BiGRU)
   - Purpose: Captures contextual information from both past and future words in the sequence.

- Bidirectional Processing:
  - Forward GRU: Processes the sequence from start to end.
  - Backward GRU: Processes the sequence from end to start.
  - Hidden Units: 64 units for each GRU direction.
- Regularization:
  L2 Regularization with a factor of 0.02 is applied to the recurrent kernel to prevent overfitting.
- Justification:
  - BiGRU effectively captures both local and global dependencies in text, improving classification accuracy for complex sequences.
  - Using 64 hidden units provides sufficient learning capacity while maintaining computational efficiency for a medium-sized dataset.

4. Attention Mechanism
   - Purpose: Dynamically focuses on the most relevant parts of the input sequence by assigning weights to hidden states.
   - Process:
     - The BiGRU outputs are passed through an attention layer.
     - Alignment scores are calculated for each hidden state.
     - The scores are normalized using a softmax function to produce attention weights.
     - A context vector is generated as the weighted sum of the BiGRU outputs.
   - Justification:
     Attention mechanisms enable the model to emphasize keywords or phrases in the description, improving feature aggregation and overall classification performance.
5. DropConnect Layer
   - Purpose: Regularizes the model by randomly dropping connections between neurons during training.
   - Configuration:
   - Dropout Rate: Set to 0.5.
   - Justification:
     DropConnect reduces overfitting by ensuring that the model does not rely too heavily on specific connections, improving its generalization ability.
6. Dense Layers
   - Purpose: Transforms the extracted features into a representation suitable for classification.
   - Configuration:
     - A dense layer with 32 units and ReLU activation is used to introduce non-linearity.
     - A dropout layer with a 0.5 rate is applied to further reduce overfitting.
   - Justification:
     - A fully connected layer with 32 units effectively balances feature learning and computational cost.
     - The additional dropout ensures the model generalizes well on unseen data.
7. Output Layer
   - Purpose: Produces probabilities for each film genre.
   - Configuration:
     - A softmax activation function is used to generate probabilities for the genre classes.
     - The number of output units corresponds to the number of genre classes.
   - Justification:
     The softmax function outputs a probability distribution, ensuring that the genre with the highest probability is selected as the predicted class.

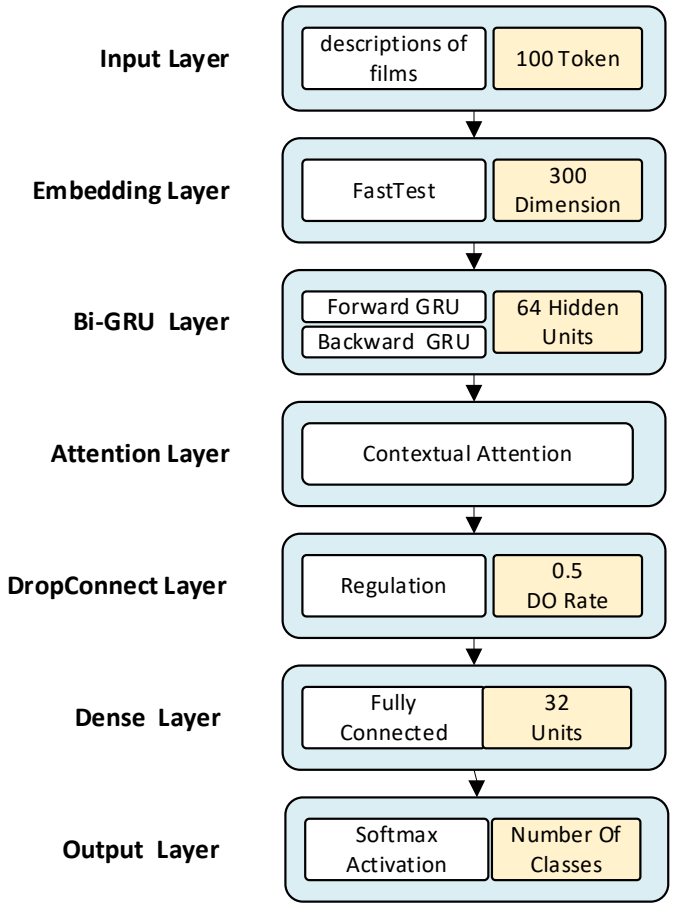

Figure 2. Model Architecture

2.3 *Training Procedure*

The training procedure for the proposed model involves several steps to ensure effective learning and evaluation. Here is a detailed description of the training procedure:

1. Data Preparation:
   - Loading Dataset: The dataset containing film titles, genres, and descriptions is loaded.
   - Tokenization: The textual descriptions are tokenized using the Tokenizer from TensorFlow's Keras API. Each word in the descriptions is converted to an integer index.
   - Padding Sequences: The tokenized sequences are padded to a maximum length of 100 tokens to ensure uniform input size for the model.
   - Label Encoding: The genres are one-hot encoded to transform categorical labels into binary vectors suitable for multi-class classification.
2. Data Splitting:
   The dataset is split into training, validation, and test sets using a 70-15-15 split. This ensures that the model has sufficient data for training, validation during training, and testing after training is complete.
3. Embedding Matrix Preparation:
   - Loading FastText Embeddings: Pre-trained FastText word vectors are loaded.
   - Creating Embedding Matrix: An embedding matrix is created where each row corresponds to a word in the vocabulary, and the columns represent the FastText word vectors. Words not found in the FastText embeddings are initialized with zeros.
4. Model Initialization:
   The model is constructed using the architecture described in the previous section. This includes the input, embedding, BiGRU layers, attention mechanism, DropConnect, and fully connected dense layers.
5. Model Compilation:
   - Optimizer: Adam optimizer is chosen for its adaptive learning rate capabilities and efficient handling of sparse gradients.
   - Loss Function: Categorical cross-entropy is the loss function for multi-class classification.

- Metrics: Accuracy is the primary metric to evaluate the model's performance.

6. Callbacks Setup:
   - Early Stopping: This callback monitors the validation loss and stops training if the loss does not improve for three consecutive epochs, restoring the best model weights.
   - Model Checkpoint: This callback saves the model weights that achieve the lowest validation loss during training.
   - ReduceLROnPlateau: This callback reduces the learning rate by 0.2 if the validation loss does not improve for two consecutive epochs, with a minimum learning rate threshold.
7. Training:
   - The model is trained using the training data, with a batch size of 32 and a maximum of 50 epochs.
   - The validation set monitors the model's performance and adjusts the learning rate or stops training early as needed.
8. Evaluation:
   - After training, the best model (saved by the Model Checkpoint callback) is loaded.
   - The model is evaluated on the test set to assess its performance. Metrics such as accuracy are computed and reported.
9. Results:
   - The training and validation accuracy and loss are plotted to visualize the model's learning curve.
   - The final test accuracy is reported as the primary measure of the model's performance.

2.3 *Evaluation Metrics*

The model's performance is evaluated using four key metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model's classification capabilities, particularly in handling imbalanced datasets.

1. Accuracy

   Accuracy measures the overall correctness of the model by comparing the number of correct predictions to the total number of predictions:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{10}$$

   Accuracy is useful for balanced datasets but may be misleading for imbalanced datasets.

2. Precision

   Precision evaluates the proportion of correctly predicted positive instances relative to all predicted positive instances. It focuses on reducing false positives:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{11}$$

   Higher precision means the model makes fewer incorrect positive predictions.

3. Recall (Sensitivity)

   Recall measures the model's ability to identify all relevant positive instances. It focuses on reducing false negatives:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{11}$$

   Higher recall indicates the model captures more of the actual positive instances.

4. F1-Score

   F1-score is the harmonic mean of precision and recall, balancing the two metrics into a single value:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

   F1-score is particularly useful for imbalanced datasets because it considers both false positives and false negatives, providing a more balanced evaluation of model performance.

In addition to the above metrics, a confusion matrix is used for a detailed analysis of predictions across genres. The confusion matrix visually represents:

- True Positives (TP): Correctly predicted genres.
- False Positives (FP): Incorrectly predicted genres.
- False Negatives (FN): Missed genres.

This analysis helps identify specific genres where the model excels or struggles, offering insights for further improvements.

3 Results and Discussion

3.1 *Training and Validation Performance*

The model's performance was monitored over 50 epochs during the training process, with early stopping applied to prevent overfitting. Early stopping halted training at epoch 30 based on the validation loss. The results for training and validation metrics are summarized in Table 1 below:

Table1. Training and Validation Performance Metrics Across Epoch

Epoch	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Learning Rate
1	0.2140	9.0762	0.3507	2.5464	0.0010
2	0.3331	2.4988	0.3613	2.4020	0.0010
3	0.3995	2.2918	0.4148	2.1915	0.0010
4	0.4088	2.2379	0.4386	2.1067	0.0010
5	0.4194	2.1920	0.3318	2.4395	0.0010
6	0.4213	2.1613	0.4300	2.0713	0.0010
7	0.4254	2.1265	0.4453	2.0282	0.0010
8	0.4265	2.1089	0.4528	2.0135	0.0010
9	0.4339	2.0858	0.4427	2.0045	0.0010
10	0.4248	2.0988	0.4513	1.9871	0.0010
11	0.4269	2.0780	0.4297	2.0164	0.0010
12	0.4315	2.0696	0.4436	1.9879	0.0010
13	0.4300	2.0518	0.4588	1.9426	0.0002
14	0.4346	2.0184	0.4512	1.9542	0.0002
15	0.4380	2.0124	0.4529	1.9267	0.0002
16	0.4350	2.0105	0.4492	1.9286	0.0002
17	0.4310	2.0262	0.4444	1.9380	0.0002
18	0.4358	2.0153	0.4539	1.9126	0.00004
19	0.4363	2.0024	0.4603	1.9119	0.00004
20	0.4356	1.9981	0.4594	1.9105	0.00004
21	0.4314	2.0107	0.4563	1.9098	0.00004
22	0.4338	2.0087	0.4578	1.9111	0.00004
23	0.4375	1.9945	0.4560	1.9106	0.00004
24	0.4385	1.9988	0.4567	1.9067	0.00001
25	0.4366	1.9943	0.4588	1.9068	0.00001
26	0.4366	2.0037	0.4584	1.9063	0.00001
27	0.4353	1.9971	0.4571	1.9052	0.00001
28	0.4352	2.0120	0.4576	1.9056	0.00001
29	0.4375	1.9951	0.4584	1.9058	0.00001
30	0.4362	2.0071	0.4578	1.9059	0.00001

The chart below illustrates the progression of training and validation accuracy over the epochs.

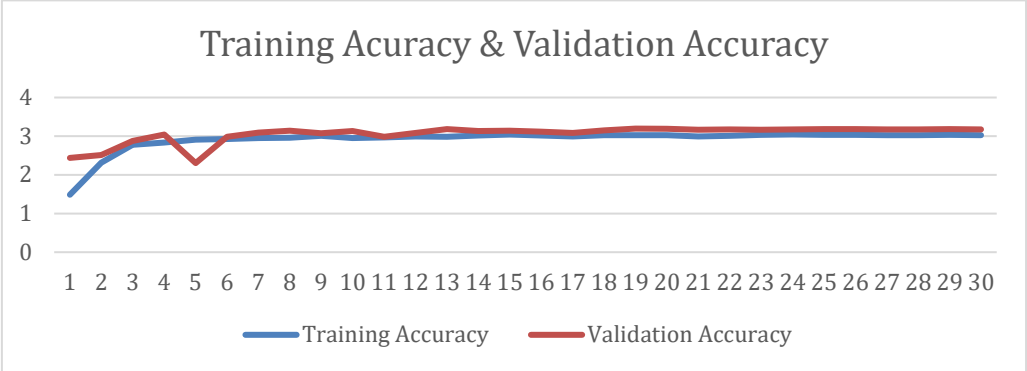

Figure 3. Training Accuracy & Validation Accuracy Chart

The model's performance was monitored over 50 epochs during the training process. Early stopping was applied to prevent overfitting, and training stopped at epoch 30 based on validation loss. The key metrics recorded include accuracy and loss for both training and validation datasets.

1. Training Accuracy: Progressed from 0.2140 in epoch 1 to a peak of 0.4380 in epoch 15.
2. Validation Accuracy: Progressed from 0.3507 in epoch 1 to a peak of 0.4603 in epoch 19.
3. Training Loss: Decreased from 9.0762 in epoch 1 to 2.0124 in epoch 15.
4. Validation Loss: Decreased from 2.5464 in epoch 1 to 1.9052 in epoch 27.

During the training process, hyperparameter tuning played a significant role in improving model accuracy. The following adjustments had the most notable impact:

1. BiGRU Hidden Units: Setting the hidden units to 64 provided the optimal balance between model performance and computational efficiency.
2. Learning Rate: The use of the Adam optimizer with an initial learning rate of 0.001 enabled stable convergence, while later reducing the learning rate improved validation accuracy (as seen after epoch 13).
3. Dropout and DropConnect Rates: Regularization rates of 0.5 for DropConnect and dense layers significantly reduced overfitting, stabilizing validation accuracy around 45%.

3.2 *Model Performance*

The table below presents the detailed classification report by genre, showcasing the model's performance metrics, including precision, recall, F1-score, and support for each genre. This comprehensive report highlights the model's strengths and weaknesses across different categories, providing valuable insights into its classification capabilities and areas requiring further improvement.

Table 1. Detailed Classification Report by Genre

Genre	Precision	Recall	F1-score	Support
Drama	0.00	0.00	0.00	193
Thriller	0.00	0.00	0.00	70
Adult	0.00	0.00	0.00	99
Documentary	0.00	0.00	0.00	69
Comedy	0.00	0.00	0.00	38
Crime	0.26	0.31	0.28	1159
Reality-TV	0.00	0.00	0.00	76
Horror	0.56	0.89	0.69	1985
Sport	0.45	0.78	0.58	2045
Animation	0.00	0.00	0.00	114
Action	0.00	0.00	0.00	50
Fantasy	0.00	0.00	0.00	27
Short	0.00	0.00	0.00	41
Sci-Fi	0.00	0.00	0.00	314
Music	0.00	0.00	0.00	111
Adventure	0.00	0.00	0.00	44
Talk-Show	0.00	0.00	0.00	36
Western	0.00	0.00	0.00	16
Family	0.00	0.00	0.00	129
Mystery	0.00	0.00	0.00	101
History	0.00	0.00	0.00	113
News	0.00	0.00	0.00	774
Biography	0.00	0.00	0.00	77
Romance	0.00	0.00	0.00	60
Game-Show	0.00	0.00	0.00	231
Musical	0.00	0.00	0.00	24
War	0.00	0.00	0.00	137
Total/Average	**0.29**	**0.46**	**0.35**	**8133**

The table below summarizes the model's performance on the genre classification task. The evaluation metrics used include Precision, Recall, and F1-score for both macro and weighted averages. These metrics provide a comprehensive view of the model's effectiveness across various genres, highlighting its strengths and identifying areas for improvement.

Table 3. Performance Metric

Measure	Precision	Recall	F1-score
Macro Average	0.05	0.07	0.06
Weighted Average	0.29	0.46	0.35

The results indicate that the model's overall performance is modest, with a macro average F1-score of 0.06, reflecting the challenges in accurately classifying a diverse range of genres. The weighted average metrics, which account for the class distribution, show a higher F1-score of 0.35, suggesting better performance on more prevalent genres in the dataset. This disparity between macro and weighted averages highlights the model's varying effectiveness across different genres.

High-Performing Genres:

1. Horror: The model demonstrated notable performance in the "Horror" genre with a precision of 0.56, recall of 0.89, and F1-score of 0.69. This indicates the model's strong ability to identify horror films accurately.
2. Sport: The "Sport" genre also showed relatively high precision (0.45) and recall (0.78), resulting in an F1-score of 0.58. This suggests that the model can effectively classify sports-related content.

Low-Performing Genres:

Numerous genres, including "Drama," "Thriller," "Adult," "Documentary," and "Comedy," exhibited zero precision, recall, and F1 scores, indicating a complete failure to identify these genres within the test set correctly. This underperformance suggests significant challenges in distinguishing these categories, likely due to overlapping features or insufficient training data for these specific genres.

Summarize The Findings:

The results demonstrate the effectiveness of integrating FastText embeddings with advanced neural network architectures for film genre classification. The following points summarize the findings:

1. Initial Training: The model shows significant improvement in the early epochs, with a notable increase in training and validation accuracy. This suggests that the FastText embeddings effectively capture the semantic meaning of words, aiding the model in understanding the context of movie descriptions
2. Validation Performance: The validation accuracy stabilizes around 45%, indicating that the model generalizes well to unseen data. The use of techniques like SpatialDropout and DropConnect likely contributed to this stability by preventing overfitting
3. Regularization Techniques: The implementation of DropConnect and Batch Normalization helped in improving the model's robustness, making it less prone to overfitting and enhancing generalization
4. Attention Mechanism: Including an attention mechanism enabled the model to focus on the most relevant parts of the movie descriptions, further enhancing its ability to classify genres. Advantages

The study identified several key advantages of the proposed model:

1. Effective Handling of Frequent Genres: The model performed well for frequent genres such as "Horror" and "Sport," demonstrating its ability to capture the characteristics of more commonly occurring genres in the dataset.
2. Integration of Multiple Techniques: Using FastText embeddings, BiGRU, and attention mechanisms enhanced the model's ability to capture local and global contextual information.
3. Regularization Techniques: Techniques such as SpatialDropout and DropConnect were effectively utilized to prevent overfitting, improving the model's robustness.
4. Scalability: The model's architecture allows for scalability, which can be further improved and extended with additional features or more advanced techniques without significant rework.

3.3 *Summary Finding*

The results underscore both the strengths and limitations of the proposed model:

1. Hyperparameter Tuning:
   Fine-tuning key parameters—such as setting BiGRU hidden units to 64, an initial learning rate of 0.001, and regularization rates of 0.5—significantly enhanced model accuracy, stability, and generalization.
2. Performance on Frequent Genres:
   The model demonstrated strong performance for frequent genres, such as Horror and Sport, achieving F1-scores of 0.69 and 0.58, respectively, due to their larger representation in the dataset and clearer distinguishing features.
3. Challenges with Underrepresented Genres:
   Rare genres, such as Drama and Thriller, exhibited poor performance due to class imbalance and semantic overlaps with other genres. These challenges resulted in low precision and recall, reflecting the model's difficulty in learning distinct patterns for sparsely represented categories.

4. Effect of the Attention Mechanism:
   Integrating the attention mechanism allowed the model to focus on the most informative parts of the movie descriptions, improving predictions for genres with distinct and well-defined contextual features.

3.4 *Limitations*

The study identified several challenges that affected the model's overall performance:

1. Data Imbalance:
   The dominance of certain genres significantly affected the classification of underrepresented genres. Techniques such as SMOTE (Synthetic Minority Oversampling Technique) or class weighting could improve performance on rare classes.
2. Model Complexity:
   While the architecture integrates advanced techniques (FastText, BiGRU, Attention), it may not fully capture the nuances of overlapping genres. Future improvements could explore ensemble methods or transformer-based architectures to enhance accuracy.
3. Computational Resources:
   Training deep learning models like the proposed architecture requires significant computational time and resources. Optimization strategies (e.g., reducing redundant parameters) should be explored to improve efficiency.
4. Generalization Issues:
   The model struggles to generalize beyond frequent and distinct genres, as observed in its poor performance on classes with fewer examples or overlapping semantics. More robust models capable of handling ambiguity and contextual variations are needed.

4 Conclusion

This study proposed an advanced neural network model for film genre classification by combining FastText embeddings, Bidirectional GRU (BiGRU), and an attention mechanism to address the challenges of extracting meaningful features from textual descriptions. The model demonstrated strong performance in frequent genres, such as Horror and Sport, achieving notable F1-scores. This success highlights its ability to effectively capture both local and global contextual patterns in the data.

However, the study also revealed key challenges that impacted the model's overall performance:

1. Data Imbalance: Underrepresented genres, such as Drama and Thriller, suffered from low precision and recall.
2. Semantic Overlaps: The shared vocabulary and contextual similarities between genres made accurate differentiation more difficult.
3. Computational Complexity: Training deep neural networks required significant computational resources, limiting scalability.

Despite these limitations, the integration of FastText embeddings for handling out-of-vocabulary words, BiGRU for contextual learning, and attention mechanisms for focus on relevant parts of the input text demonstrated the model’s potential. This study lays a foundation for further development of scalable and accurate text classification frameworks applicable to the film industry.

5 Suggestion

To further improve the performance and generalizability of the genre classification model, the following recommendations are proposed:

1. Addressing Data Imbalance:
   a. Implement data balancing techniques such as SMOTE (Synthetic Minority Oversampling Technique), oversampling underrepresented genres, and augmenting textual data through paraphrasing and synonym replacement.
   b. Use cost-sensitive learning or class weighting during training to penalize misclassification of minority classes.
2. Incorporating Advanced Architectures:
   a. Explore transformer-based models (e.g., BERT, RoBERTa) to better capture deep semantic relationships and narrative structures.
   b. Combine hybrid architectures, such as CNN-BiGRU or transformer-RNN, to leverage the strengths of both convolutional and sequential models.

3. Feature Engineering for Richer Context:
   a. Integrate metadata features like director, cast, release year, runtime, and user-based ratings to provide additional context that complements textual descriptions.
   b. Explore multi-modal approaches that combine textual descriptions with visual or audio features (e.g., film posters or soundtracks) to enhance genre prediction accuracy.
4. Leveraging Transfer Learning:
   - Fine-tune pre-trained language models like BERT, GPT, or XLNet for film descriptions to exploit their superior understanding of narrative and syntactic structures.
   - Utilize transfer learning from related tasks (e.g., sentiment analysis or plot summarization) to improve generalization across diverse genres.
5. Optimizing Computational Efficiency:
   - Apply model pruning, quantization, or distillation to reduce model size and training time while maintaining performance.
   - Investigate lightweight architectures for deployment in resource-constrained environments.

BIBLIOGRAPHY

[1] L. Luo, "Network text sentiment analysis method combining LDA text representation and GRU-CNN," *Pers. Ubiquitous Comput.*, vol. 23, no. 3–4, pp. 405–412, Jul. 2019, doi: 10.1007/s00779-018-1183-9.

[2] M. Zulqarnain, R. Ghazali, M. G. Ghouse, and M. F. Mushtaq, "Efficient processing of GRU based on word embedding for text classification," *JOIV Int. J. Informatics Vis.*, vol. 3, no. 4, pp. 377–383, Nov. 2019, doi: 10.30630/joiv.3.4.289.

[3] J. Zhang, F. Liu, W. Xu, and H. Yu, "Feature Fusion Text Classification Model Combining CNN and BiGRU with Multi-Attention Mechanism," *Futur. Internet*, vol. 11, no. 11, p. 237, Nov. 2019, doi: 10.3390/fi11110237.

[4] Y. Han, M. Liu, and W. Jing, "Aspect-Level Drug Reviews Sentiment Analysis Based on Double BiGRU and Knowledge Transfer," *IEEE Access*, vol. 8, pp. 21314–21325, 2020, doi: 10.1109/ACCESS.2020.2969473.

[5] N. Gruber and A. Jockisch, "Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?," *Front. Artif. Intell.*, vol. 3, Jun. 2020, doi: 10.3389/frai.2020.00040.

[6] J. Choi and S.-W. Lee, "Improving FastText with inverse document frequency of subwords," *Pattern Recognit. Lett.*, vol. 133, pp. 165–172, 2020, doi: 10.1016/j.patrec.2020.03.003.

[7] E. I. Setiawan, F. Ferry, J. Santoso, S. Sumpeno, K. Fujisawa, and M. H. Purnomo, "Bidirectional GRU for Targeted Aspect-Based Sentiment Analysis Based on Character-Enhanced Token-Embedding and Multi-Level Attention," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 392–407, Oct. 2020, doi: 10.22266/ijies2020.1031.35.

[8] L. Li, L. Yang, and Y. Zeng, "Improving Sentiment Classification of Restaurant Reviews with Attention-Based Bi-GRU Neural Network," *Symmetry (Basel).*, vol. 13, no. 8, p. 1517, Aug. 2021, doi: 10.3390/sym13081517.

[9] W. Ali, Y. Yang, X. Qiu, Y. Ke, and Y. Wang, "Aspect-Level Sentiment Analysis Based on Bidirectional-GRU in SIoT," *IEEE Access*, vol. 9, pp. 69938–69950, 2021, doi: 10.1109/ACCESS.2021.3078114.

[10] W. Gu, S. Zheng, R. Wang, and C. Dong, "Forecasting Realized Volatility Based on Sentiment Index and GRU Model," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 24, no. 3, pp. 299–306, May 2020, doi: 10.20965/jaciii.2020.p0299.

[11] Z. Liu, B. Zhou, L. Meng, and G. Huang, "Multimodal Sentiment Analysis Using BiGRU and Attention-Based Hybrid Fusion Strategy," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 1963–1981, 2023, doi: 10.32604/iasc.2023.038835.

[12] X. Wang, X. Chen, M. Tang, T. Yang, and Z. Wang, "Aspect-Level Sentiment Analysis Based on Position Features Using Multilevel Interactive Bidirectional GRU and Attention Mechanism," *Discret. Dyn. Nat. Soc.*, vol. 2020, pp. 1–13, Jul. 2020, doi: 10.1155/2020/5824873.

[13] J. Ravichandran, M. Kaden, S. Saralajew, and T. Villmann, "Variants of DropConnect in Learning vector quantization networks for evaluation of classification stability," *Neurocomputing*, vol. 403, pp. 121–132, 2020, doi: 10.1016/j.neucom.2019.12.131.

[14] Z. Lian, X. Jing, X. Wang, H. Huang, Y. Tan, and Y. Cui, "DropConnect Regularization Method with Sparsity Constraint for Neural Networks," *Chinese J. Electron.*, vol. 25, pp. 152–158, 2016, doi:

10.1049/CJE.2016.01.023.

[15] H. Lim, "A Study on the Effect of DropConnect to Control Overfitting in Designing Neural Networks," pp. 178–183, 2020, doi: 10.3233/faia200780.

[16] E. Dynomant *et al.*, "Word Embedding for the French Natural Language in Health Care: Comparative Study," *JMIR Med. Informatics*, vol. 7, 2019, doi: 10.2196/12310.