Date of publication Dec 14, 2024, date of current version Dec 14, 2024.

Digital Object Identifier

Granular Genre Classification of Creative Literature Using Their Content

GIA PHAT HUYNH[1], THE MINH NGUYEN[2], and THAI SON TRUONG[3],
[1]Wilfrid Laurier University, Waterloo, ON N2L 3C5 Canada (e-mail: huyn8900@mylaurier.ca, student number: 235838900)
[2]Wilfrid Laurier University, Waterloo, ON N2L 3C5 Canada (e-mail: nguy6401@mylaurier.ca, student number: 169076401)
[3]Wilfrid Laurier University, Waterloo, ON N2L 3C5 Canada (e-mail: truo1520@mylaurier.ca, student number: 235841520)

ABSTRACT Text genre classification has always been a subject of interest within the natural language processing field, due to the complexity of human languages and the nature of machine learning models in taking texts at face value. As such, standalone sentences and literary devices may confuse these models, as there may be hidden context or meaning that are not typically recognized. Despite the need for text genre classification models to be trained on a larger context window, time and computing resource constraints mean that most projects have historically relied on traditional models (such as support vector machines) and are trained on a very small amount of tokens per entry. However, with the arrival of transformer models, and powerful and openly available training platforms, the task seems more approachable than ever. Our project aims to classify creative literary works into the appropriate genres, and is different from previous attempts at this task in two meaningful ways: we provided our models of choice with much more context per input, and used the trained models to predict one or more genres per work. Our experiments showed promising results: using a variety of transformer-based models, we achieved a high accuracy of 81% using the DistilBERT model and 83.2% using the Longformer model for multi-label classification tasks.

INDEX TERMS Creative works, genre classification, machine learning, natural language processing (NLP), text classification, transformer models.

I. INTRODUCTION

FROM the Epic of Gilgamesh to Harry Potter, fictional works have always been a part of human life. They provide an outlet for us to imagine the possibilities of an alternative universe, to explore abstract and complex ideas, and to express our inner thoughts and emotions. Picking up a fictional work leads us to an immersive world where the impossible becomes possible, where the unnatural becomes natural, and where the non-canonical becomes canonical. With the advent of the digital era, people now have more resources to find fictional works that they might be interested in than ever before. Different people may prefer different themes, whether it be steampunk 19th century London, or a distant world several centuries from today. While books are typically classified into certain major genres (such as romance or thriller), they might not be enough to completely satisfy what the reader is looking for. Certain sites like Goodreads [1] assign additional tags to further categorize each work, but these attempts are largely manual and time-consuming, as it requires the tagger to have a good knowledge of the contents of the work. Additionally, there may exist additional themes or tropes within the work that taggers may not pick up. An automated tagging system, using natural language processing (NLP) and examining the entire contents of the works, could be an effective and efficient way to tag each incoming work, without the need for manual labor.

Multi-label text classification is a task where each text instance can belong to multiple categories simultaneously. This is a common scenario in applications such as document tagging, recommendation systems, and genre classification for books or articles. Unlike single-label classification, multi-label classification requires models to predict all applicable labels, introducing additional challenges in encoding, modeling, and evaluation.

This report summarizes the implementation of a single-label and a multi-label classification model using transformer-based architectures, including DistilBERT, Longformer, and Reformer, to predict genres based on textual content. The findings provide insights into the effectiveness of the implemented approaches, as well as future directions for improving performance.

II. LITERATURE REVIEW

Existing research into genre classification has mainly focused on a work's cover, title, or summary [2] [3]. To our knowledge, there has only been one published attempt at classifying a work based on its entire content [4]. This is understandable, as the models that can handle such an input are typically heavily time or resource consuming; the large version of a model called "Longformer" [5] [6], capable of handling inputs of up to 16384 tokens in length (approximately the length of a chapter book [7]) was built and trained using 8 graphics processing units (GPUs) with 48 gigabytes of VRAM each - much more than what a consumer GPU typically has. However, this does mean that there is little precedent to our work that we can examine. The lone paper experimenting with full-text genre classification experimented with eight different models and achieved a highest accuracy rate of 67.4 percent, using a linear SVM model [4]. Nonetheless, we still feel that this is a task worth exploring, especially as powerful, openly available training platforms like Kaggle have made this task more accessible.

III. PROBLEM STATEMENT

Traditional single-label classification assigns one label to each instance, but many real-world tasks involve multi-label classification, where an instance can belong to multiple categories. For example, a piece of text can be simultaneously classified as "Action", "Adventure", and "Sci-Fi".

The goal of this project was to predict genres from the content of a text. Each text could belong to multiple genres, requiring a robust approach that could handle overlapping labels while providing accurate predictions. In addition, the project aimed to address challenges like handling long documents.

IV. METHODOLOGY

A. DATASET

Using Python's Scrapy library [8], we created our own dataset, which comprises 5803 works pulled from the fanfiction site Archive Of Our Own (also known as "AO3") [9]. Each work is 8000 or fewer words in length, has at least 100 kudos (also known as "likes", various depending on genre), and belongs to one or more of the following genres: **Action, Adventure, Comedy, Crime, Fantasy, Horror, Mystery, Romance, Sci-Fi, Superheroes**. 36% of the works have one of the aforementioned genres, 31% have two genres, and 32% have three genres. In total, there are 11,385 counts of genres across the dataset. Table 1 shows the distribution of each distinct genre, which indicates the large populations on Romance and Comedy genres while Superheroes genre is tagged with the minority samples.

During the scraping process, most of the special characters will be eliminated with exceptions for punctuation characters like comma, question mark or full stop... because those punctuations play key roles in context-sensitive content. The data is then organized into a comma-separated values (CSV) file with 6 columns: workid, worktitle, author, wordcount, content, genres. An example of a few entries is included in table 2 below. (Values for the "content" column are omitted to fit the format of this paper.)

Genre	Genre Count	Genre Distribution
Action	904	7.94%
Adventure	925	8.12%
Comedy	1,574	13.83%
Crime	949	8.34%
Fantasy	1,350	11.86%
Horror	947	8.32%
Mystery	799	7.02%
Romance	2,478	21.77%
Sci-Fi	886	7.78%
Superheroes	573	5.03%
	11,385	

TABLE 1. Genre distribution in the dataset

workid	worktitle	author	workcount	content	genres
102363	Marbles	mellish	1519	condition...	Horror
270750	Feathers	Daylight	3928	feather...	Action

TABLE 2. Sample entries in the dataset

B. DATA PREPROCESSING

1) Lemmatization

To process the data for our input, we applied lowercasing and lemmatization to the content of each work, the latter using NLTK's WordNetLemmatizer [10]. During the research process, an alternative way of processing the data was suggested, which is to stem the words. While that method would have been faster, we ultimately proceeded with applying lemmatization, as it ensures that we are left with real English words after they are processed, which stemming's tendency to focus on a word's root may fail to achieve. Additionally, we decided not to remove stop words, as the models we experimented with are capable of handling them natively.

2) Genres Encoding

Regarding single labels, we use LabelEncoder() from sklearn.preprocessing [11] to encode genres in single-label classification tasks. It assigns a unique integer to each genre, ensuring that one label corresponds to exactly one class. For instance, given the genres [Action, Comedy, Superheroes], the encoding would map Action to 0, Comedy to 1, and Superheroes to 2. This integer-based representation simplifies the data for models that require numerical input while maintaining the distinctiveness of each genre.

On the other hand, the MultiLabelBinarizer() from sklearn.preprocessing [12] is used to encode genres in multi-label classification tasks. Unlike single-label classification, multi-label predictions represent each genre as a multi-hot vector, where each position corresponds to a binary indicator of whether a label is present. For instance, given the genres [Action, Comedy, Superheroes], a sample with [Action, Superheroes] would be represented as [1, 0, 1], while [Comedy, Superheroes] would be [0, 1, 1]. Each class prediction treats as an independent binary classification task, and a probability threshold (e.g., >0.3) is applied to determine the final assigned labels. This thresholding approach helps balance precision and recall for each genre.

The primary difference between single-label and multi-label classification lies in the relationship between labels and predictions. In single-label classification, each sample belongs exactly to one class, represented as a single integer (e.g., 'Comedy' encoded as 1). Conversely, multi-label classification allows a sample to belong to multiple classes simultaneously, represented as a binary vector (e.g. [1, 0, 1] for 'Action' and 'Superheroes'). The latter approach introduces complexity, requiring independent binary predictions for each label and the use of thresholds to finalize the predictions, whereas single-label classification relies on simpler integer mappings and a single-class output.

3) Text tokenization

Tokenization is a fundamental process in Natural Language Processing (NLP) that involves segmenting text into smaller units, known as tokens, which can be words, subwords, or characters. This step is crucial as it transforms raw text into a structured format that models can interpret and process. The choice of tokenization method significantly impacts the performance of NLP models, especially in handling out-of-vocabulary words and managing computational resources.

In transformer-based models like BERT and its variants, subword tokenization techniques such as WordPiece are commonly employed. These methods decompose words into subword units, enabling the model to handle rare or unseen words by leveraging known subword components. For instance, the word "unbelievable" might be tokenized into ["un", "**believable"], allowing the model to process each segment effectively. This approach not only aids in managing vocabulary size but also enhances the model's ability to generalize across different linguistic constructs. A simple example of tokenization includes the input text "I love playing soccer," which, after tokenization, results in ["I", "love", "playing", "soccer"], producing four tokens. Techniques like lemmatization may further preprocess text into simplified forms, such as ["I", "love", "play", "soccer"].

Furthermore, the effectiveness of tokenization extends beyond merely compressing text into fewer tokens. Studies have shown that factors such as pre-tokenization choices and vocabulary construction play significant roles in downstream performance, indicating that optimal tokenization involves more than just minimizing token count [13]. Tokenization strategies are also intertwined with the model's overall architecture and its ability to effectively capture and utilize context within text. These findings underscore the importance of selecting tokenization methods tailored to specific model requirements and text characteristics.

In conclusion, while traditional models like BERT face constraints due to token limits, innovations such as Longformer and Reformer offer promising solutions for handling long sequences. Tokenization remains a cornerstone of NLP, bridging the gap between raw text and numerical representation, while advances in tokenization methods and attention mechanisms continue to expand the potential applications of transformer-based architectures.

C. MODELS

To train all of the following models, we employed binary cross-entropy loss to effectively handle the independent binary predictions required for each label in our multi-label classification task. To optimize the training process, we used the AdamW optimizer [14], which is known for its stability and efficiency in transformer-based models. Additionally, we incorporated a learning rate scheduler to dynamically adjust the learning rate during training, allowing the model to fine-tune its learning as validation performance evolved. To ensure robust training and prevent overfitting, we implemented an early stopping mechanism, which halted the training process if the validation loss did not improve over a specified number of epochs. This combination of techniques ensured that the model was trained efficiently and effectively.

1) DistilBERT

DistilBERT, introduced by Sanh et al. in 2019, is a distilled version of the Bidirectional Encoder Representations from Transformers (BERT) model, designed to retain much of BERT's language understanding capabilities while being more efficient in terms of size and speed [15] [16]. By applying knowledge distillation during the pre-training phase, DistilBERT reduces the size of the original BERT model by 40 percentage, achieving 97 percentage of BERT's performance and operating 60 percentage faster. The efficiency of DistilBERT has led to its adoption across various natural language processing (NLP) tasks. For instance, in sentiment analysis, DistilBERT has been employed to extract subjective information from textual data, categorizing it into classes such as positive, neutral, or negative [17]. Its ability to handle complex language patterns with reduced computational resources makes it suitable for real-time applications. Furthermore, DistilBERT has been applied in cybersecurity for web attack detection. By treating HTTP and HTTPS requests as text data, DistilBERT extracts deep semantic features, which, when fused with empirical features, enable comprehensive characterization and classification of requests [18]. This method has demonstrated high accuracy and precision in identifying anomalous requests, contributing to enhanced security measures. In the realm of sentiment classification, combining DistilBERT with multi-scale convolutional neural networks (CNNs) has led to the development of multi-task models capable of capturing both global and local features of text [19]. This integration has shown improved performance in understanding nuanced sentiments across different contexts. Additionally, the construction of domain-specific DistilBERT models through fine-tuning has addressed the challenge of domain dependency in BERT [20]. By initializing DistilBERT's parameters with those of a trained BERT model and tuning them using domain-specific corpora, these models achieve efficient and effective language representation tailored to specific fields. The versatility and efficiency of DistilBERT have made it a valuable tool in various NLP applications, offering a balance between performance and computational resource requirements.

In this project, we utilized the pre-trained 'distilbert-base-uncased' model as the foundation for our work. The model was fine-tuned on the entire dataset, consisting of 5,803 samples, to adapt it to the multi-label classification task. The training was conducted on the Google Colab platform, leveraging the computational power of an A100 GPU for efficient processing.

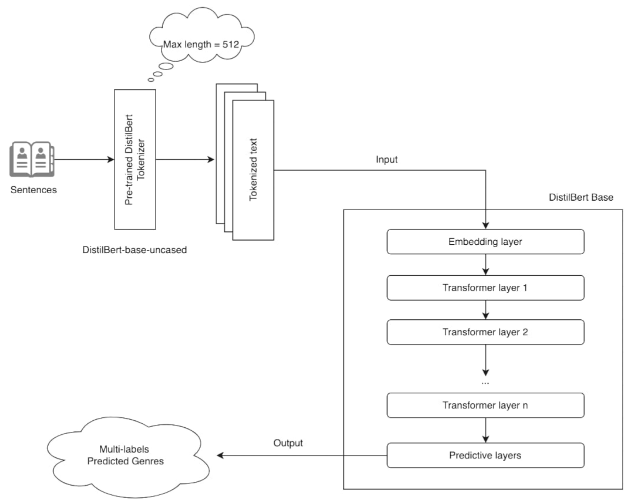


FIGURE 1. Flow of a DistilBERT model

2) Longformer

Due to DistilBERT's inherent token limit of 512 [15] (which would only realistically be suitable for a summary of a story), we opted to explore a few more models that can theoretically handle a much larger context size. Through our research, among the top contenders for this task was Longformer [5]. Deriving from another popular model, RoBERTa [21] [22], its architecture allows the model to scale linearly with the input length as opposed to quadratically like previous transformers, allowing training platforms to feasibly train the model on a much larger input. Two pretrained versions of Longformer have been released: a base version, longformer-base-4096, trained on inputs of up to 4096 tokens in length, and a larger version, led-base-16384, trained on inputs of up to 16384 tokens in length. We have opted to use the former version, due to time and resource constraints, and our assumption that such a token limit would still be sufficient for our input. The model was then trained using Kaggle's NVIDIA T4 x2 GPU, running in parallel for larger memory resources and faster training.

3) Reformer

Given the constraints of token limits of either 512 or 4096 with 2 aforementioned models, as well as the huge demand of super-efficient computing processors (with the 'led-base-16384' version of the Longformer model), we extended our research and arrived to an ample approach with the Reformer model [23] [24].

The Reformer model is a Transformer-based model that is designed to address long/ultra-long sequences tasks. The model is much memory-efficient but still retains the performance on par with Transformer model. Reformer is slightly less efficient than Longformer in term of accuracy but it's fine-tuned architect allows it to train faster for long texts. First solution is to "replace dot-product attention by one that uses locality-sensitive hashing", and second technique is to "use reversible residual layers instead of the standard residuals" [24]. Those changes foster the Reformer model to be trained on 64,000 token sequences making it sufficient for most of text classification tasks, taking in a slight trade-off in effectiveness and still achieving fast training performances. The experiment with the Reformer model was carried out using Kaggle's NVIDIA T4 x2 GPU to take advantage of the parallel architect of powerful processors.

D. EVALUATION METRICS

In our project, we employed a comprehensive set of evaluation metrics to assess the performance of our multi-label text classification model. These metrics included label-based accuracy, Hamming loss, and detailed analyses through classification reports. Each metric provided unique insights into the model's predictive capabilities, allowing for a nuanced understanding of its strengths and areas for improvement.

Label-based accuracy evaluates the proportion of correctly predicted labels out of the total labels for each label individually and computes the average across all labels. Unlike more stringent metrics like subset accuracy, label-based accuracy is less strict as it does not require an exact match of all labels in a prediction. Instead, it assesses each label independently, providing a granular view of the model's performance on a per-label basis. For example, consider the following scenario:

- True Labels: [1, 0, 1] (indicating that the first and third labels are relevant while the second is not).
- Predicted Labels: [1, 1, 1] (indicating that the model correctly predicts the first and third labels but incorrectly assigns the second label).

The label-based accuracy for this example is calculated as:

$$\text{Label 1: Correct} \rightarrow \text{Accuracy} = \frac{1}{1} = 100\%$$

$$\text{Label 2: Incorrect} \rightarrow \text{Accuracy} = \frac{0}{1} = 0\%$$

$$\text{Label 3: Correct} \rightarrow \text{Accuracy} = \frac{1}{1} = 100\%$$

The overall Label-Based Accuracy is the average across all labels:

$$\text{Label-Based Accuracy} = \frac{(100\% + 0\% + 100\%)}{3} = 66.67\%.$$

This example highlights how label-based accuracy provides meaningful insights even when predictions are partially correct. By focusing on individual labels, it allows for a more flexible and nuanced evaluation of multi-label classification models, especially when dealing with complex datasets where perfect predictions for all labels are rare.

Hamming loss quantifies the fraction of labels that are incorrectly predicted, providing a direct measure of prediction errors in multi-label classification. It is calculated as the proportion of misclassified labels to the total number of labels, offering a straightforward interpretation of the model's error rate. A lower Hamming loss indicates better performance, as it reflects fewer incorrect label assignments. This metric is particularly useful in scenarios where each label holds equal importance, as it treats all misclassifications uniformly. The significance of Hamming loss in multi-label classification has been explored in various research studies. For example, Wu and Zhu (2020) analyzed the learning guarantees of algorithms concerning Hamming loss, providing insights into its implications for model evaluation [25]. Their work highlights that optimizing for Hamming loss can improve general model performance, especially in scenarios with a moderate number of labels. Similarly, Dembczyński et al. (2010) conducted a regret analysis for performance metrics in multi-label classification, emphasizing the importance of Hamming loss as a reliable indicator of prediction quality [26]. Another study by Read et al. (2011) demonstrated the effectiveness of Hamming loss in evaluating algorithms designed for highly imbalanced datasets [27] These findings highlight its robustness and relevance in multi-label classification tasks.

The table 3 demonstrates the Hamming loss explanation.

	True Genres			Predict Genres		
Text	**Action**	**Comedy**	**Superheroes**	**Action**	**Comedy**	**Superheroes**
a lit cigar rests on a table, smoldering into a small dish on the countertop...	1	1	0	1	1	1
billy called from the front window, waving at the minivan pulling out of the driveway...	1	1	1	1	0	1
the nights were boring and long. when the sun disappeared over the horizon...	1	0	0	1	0	0
at the penthouse where the fantastic 4's base was, the remaining members were having argument...	0	1	1	0	1	0
Total number of predictions (TNP) = 12 Total number of incorrect predictions (TNIP) = 3 **Hamming Loss = TNIP / TNP = 3/12 = 0.25**						

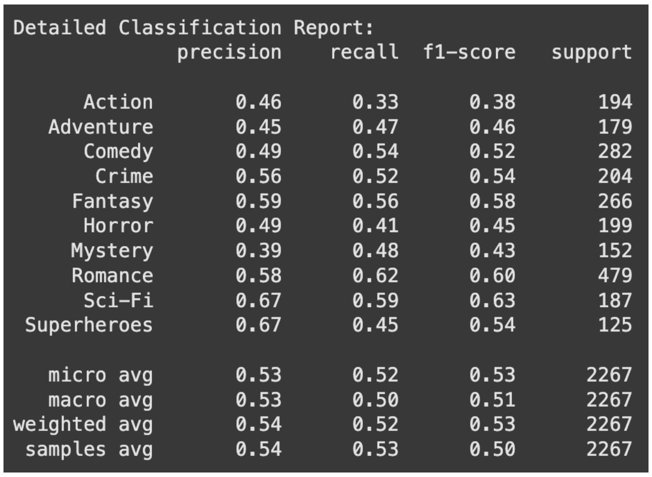
TABLE 3. Hamming Loss explanation

To gain a comprehensive understanding of the model's performance, we also generated classification reports. These reports detailed precision, recall, and F1-scores for each label, offering deeper insights into how well the model distinguishes between different classes. Precision indicates the proportion of correctly predicted positive observations to the total predicted positives, reflecting the model's accuracy in labelling. Recall measures the proportion of correctly predicted positive observations to all observations in the actual class, indicating the model's ability to capture all relevant instances. The F1-score, the harmonic mean of precision and recall, provides a balance between the two, offering a single metric that accounts for both false positives and false negatives. By employing this suite of evaluation metrics, we obtained a multifaceted view of our model's performance. This approach allowed us to identify specific strengths, such as high precision in certain labels, and areas needing improvement, such as recall in others. The combination of these metrics facilitated a balanced assessment, ensuring that the model's evaluation was thorough and aligned with the objectives of our multi-label classification task.

E. RESULTS

1) DistilBERT

In our project, we evaluated the performance of the multi-label classification model using DistilBERT, focusing on various metrics to capture its ability to predict multiple genres effectively. The classification report provided detailed insights into the model's precision, recall, and F1-scores for each genre, helping us assess its performance across both dominant and minority classes.



```
Detailed Classification Report:
             precision    recall  f1-score   support

      Action       0.46      0.33      0.38       194
   Adventure       0.45      0.47      0.46       179
      Comedy       0.49      0.54      0.52       282
       Crime       0.56      0.52      0.54       204
     Fantasy       0.59      0.56      0.58       266
      Horror       0.49      0.41      0.45       199
     Mystery       0.39      0.48      0.43       152
     Romance       0.58      0.62      0.60       479
      Sci-Fi       0.67      0.59      0.63       187
 Superheroes       0.67      0.45      0.54       125

   micro avg       0.53      0.52      0.53      2267
   macro avg       0.53      0.50      0.51      2267
weighted avg       0.54      0.52      0.53      2267
 samples avg       0.54      0.53      0.50      2267
```

FIGURE 2. Classification Report of the DistilBERT model

The report showed varying results across genres. For well-represented genres like Romance and Sci-Fi, the model performed relatively well, achieving F1-scores of 0.60 and 0.63, respectively, with high recall values. These results indicate the model's ability to identify these genres correctly in most cases, making it effective in scenarios where these labels are prominent. Similarly, Fantasy and Crime genres also demonstrated solid F1-scores of 0.58 and 0.54, reflecting the model's consistency in handling frequent labels.

However, the model struggled with some genres, particularly those with lower representation or overlapping features. For instance, Action had a low F1-score of 0.38, while Horror and Superheroes scored 0.45 and 0.54, respectively. This indicates challenges in correctly identifying these genres, possibly due to label imbalance or semantic similarities with other genres. Genres like Mystery also revealed moderate recall (0.48) but lower precision, resulting in an F1-score of 0.43, further highlighting the need for better handling of underrepresented classes.

The macro average F1-score of 0.51 reflects the model's overall performance across genres without considering class imbalance, while the weighted average F1-score of 0.53 accounts for label frequency, showing slightly better performance on dominant genres. The micro average F1-score of 0.53 provides an overall picture of the model's prediction capability across all genres, treating all labels equally.

Additional metrics, such as label-based accuracy and Hamming loss, further clarified the model's strengths and weaknesses. The best-performing model, achieved at epoch 3, recorded a label-based accuracy of 0.8061 and a Hamming loss of 0.1939, indicating that the model made errors on less than 20% of the total labels. While the overall validation accuracy of 0.1456 was lower, it reflects the strict nature of exact-match accuracy metrics in multi-label classification tasks.

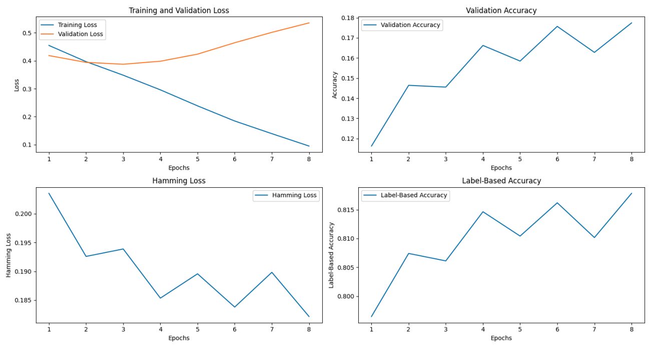


FIGURE 3. Performance of the DistilBERT model

In conclusion, the DistilBERT model demonstrated moderate success in multi-label classification, particularly for well-represented genres. However, challenges in identifying minority or semantically overlapping genres indicate opportunities for improvement through strategies like data augmentation, threshold tuning, or label imbalance handling. These findings provide a solid baseline for further optimizing the model and improving its versatility in multi-label tasks.

2) Longformer

Due to time and resource constraints, the Longformer model is trained on only 1500 samples randomly selected from the original 5803 works. However, the results still met our expectations: the model's early stopping mechanism triggered at epoch 9, with a best label-based accuracy rate of 83.2% at epoch 6, corresponding to a hamming loss of 0.168. Figure 4 shows the model's performance after each epoch.

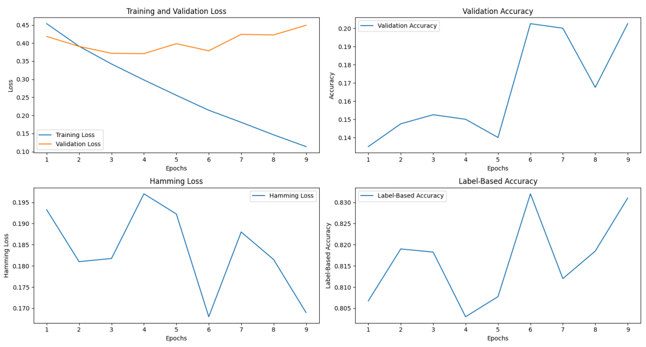
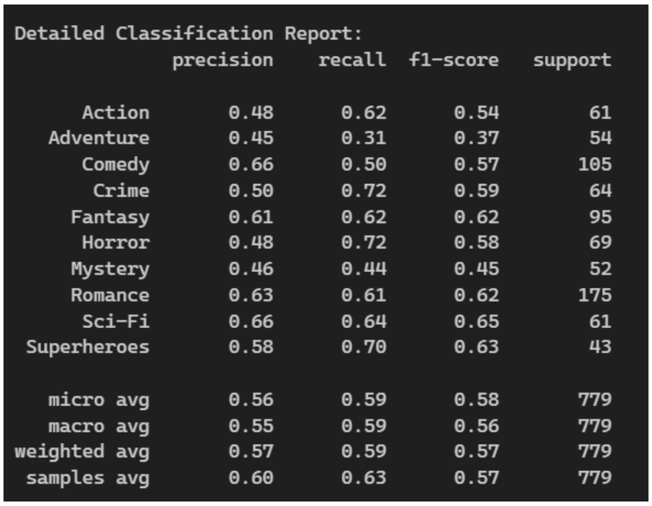


FIGURE 4. Performance of the Longformer model

The precision, recall, and F1-score of the model are available in Figure 5. Comparing the metrics to that of the DistilBERT model, there is a 4 percentage point (pp), 7pp, and 4pp increase, respectively, over the latter, suggesting that this model performed better overall. This is most likely due to the increased context that was provided to the model, allowing it to better understand the works and classify them accordingly.



```
Detailed Classification Report:
              precision    recall  f1-score   support

      Action       0.48      0.62      0.54        61
   Adventure       0.45      0.31      0.37        54
      Comedy       0.66      0.50      0.57       105
       Crime       0.50      0.72      0.59        64
     Fantasy       0.61      0.62      0.62        95
      Horror       0.48      0.72      0.58        69
     Mystery       0.46      0.44      0.45        52
     Romance       0.63      0.61      0.62       175
      Sci-Fi       0.66      0.64      0.65        61
 Superheroes       0.58      0.70      0.63        43

   micro avg       0.56      0.59      0.58       779
   macro avg       0.55      0.59      0.56       779
weighted avg       0.57      0.59      0.57       779
 samples avg       0.60      0.63      0.57       779
```

FIGURE 5. Classification Report of the Longformer model

3) Reformer

Reformer becomes a promising solution to our tasks when it is designed to cope with ultra-long sequences classification, up to 64,000 token. Similarly to DistilBERT and Longformer, we applied the Reformer model on the entire dataset of 5803 samples. The result for label-based accuracy achieved the top score at 78.5% at epoch 3 and 6, and not showing any significant improvement after that, so the early stopping mechanism was triggered at epoch 9. The respective hamming loss is 0.22. The figure 6 shows the performance and the figure 7 shows classification report of the Reformer model.

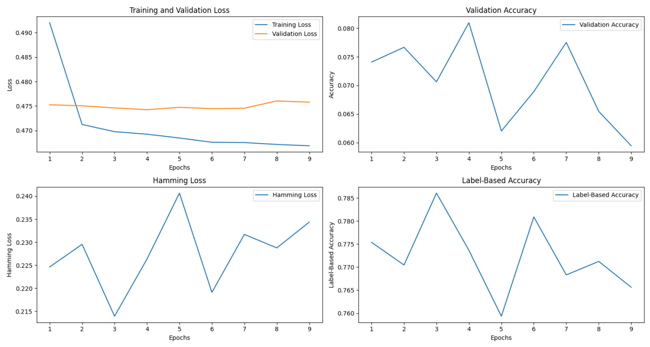
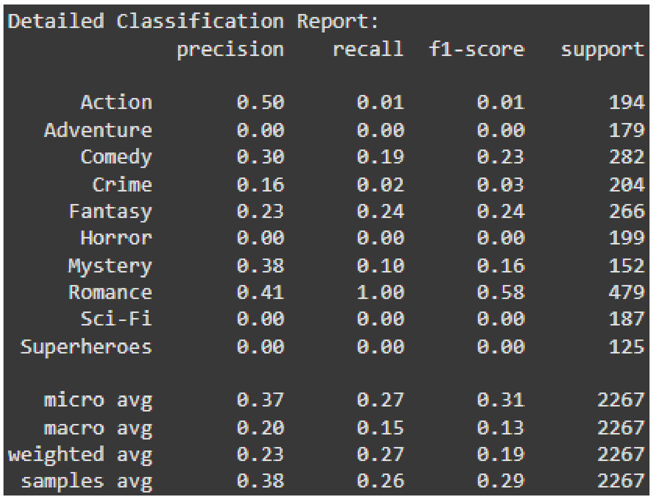


FIGURE 6. **Performance of the Reformer model**



```
Detailed Classification Report:
               precision    recall  f1-score   support

      Action       0.50      0.01      0.01       194
   Adventure       0.00      0.00      0.00       179
      Comedy       0.30      0.19      0.23       282
       Crime       0.16      0.02      0.03       204
     Fantasy       0.23      0.24      0.24       266
      Horror       0.00      0.00      0.00       199
     Mystery       0.38      0.10      0.16       152
     Romance       0.41      1.00      0.58       479
      Sci-Fi       0.00      0.00      0.00       187
 Superheroes       0.00      0.00      0.00       125

   micro avg       0.37      0.27      0.31      2267
   macro avg       0.20      0.15      0.13      2267
weighted avg       0.23      0.27      0.19      2267
 samples avg       0.38      0.26      0.29      2267
```

FIGURE 7. **Classification Report of the Reformer model**

The Reformer overall underperformed the DistilBERT and Longformer models, but this is known issue because Reformer trades-off the effectiveness and hardware resources to be able to address the ultra-long sequences without imposing a significant boost of computing power. Similar to DistilBert, the Reformer model performs better for genres that have more data and, in contrast, f1-score is relatively low for the minority subsets. There is also a note that the dataset's maximum length work is 8000, which is far under the capability of Reformer. With our key measurement is label-based accuracy, the Reformer model is still performing acceptable.

V. ANALYSIS

A. MODEL COMPARISON

A notable limitation of transformer models is their fixed maximum token limit, which constrains the length of input sequences they can process. For example, BERT and its derivatives typically have a maximum token limit of 512 tokens. This restriction poses challenges when dealing with longer texts, as exceeding this limit necessitates truncation, potentially leading to the loss of critical information. To address this issue, models like Longformer have been developed, extending the maximum token limit to 4096 tokens. Longformer employs a combination of local and global attention mechanisms, allowing it to efficiently process longer sequences without a proportional increase in computational complexity. Additionally, Reformer, another innovative transformer-based model, pushes the boundaries even further by handling sequences up to 64,000 tokens. Reformer achieves this by using techniques such as locality-sensitive hashing (LSH) for efficient attention computation, making it well-suited for ultra-long documents.

Table 4 summarizes the study on models' characteristics. The result and range of token limit supports from the experimented models enable the solution to expand to any bigger datasets.

Model	Token Limit	Long Content	Model Traits	Candidate for Tasks
DistilBERT	512	Truncate	Trained on 65M parameters. Lightweight and efficient.	Short sequences.
Longformer	4,096 16,384	Truncate	High resource demands, highly efficient.	Medium or long sequences.
Reformer	60,000	Suitable for most cases	Trained on 3M parameters. Medium resource demands, satisfactory efficiency to trade-off with hardware consumptions.	Long to ultra-long sequences.

TABLE 4. **Model Comparison**

In comparison with other published works on book genres classification, the proposed transformer-based methods show highly accurate results in table 5. This promising outcome suggests that aforementioned models have a wide range of potential applications.

Author	Dataset	Model	Accuracy
Rahul et al. [4]	Full text	Linear SVM	67.4%
Kundu et al. [28]	Cover page	Text-based models - Universal Sentence Encoder	52.6% - 73.7%
		Multi-modal models - Simple concatenation	56.1% - 77.7%
Proposed method	Full content	**DistilBERT**	**80.6%**
Proposed method	Full content	**Longformer**	**83.2%**
Proposed method	Full content	**Reformer**	**78.5%**

TABLE 5. **Proposed methods in comparison with other published works**

B. CHALLENGES

Throughout the development of our multi-label classification model, we encountered several significant challenges that impacted the overall performance and robustness of the system. Addressing these challenges is crucial for improving the accuracy and generalization of the model in future iterations.

One of the primary challenges was imbalanced data. As we self-created our data, the imbalance created difficulties for the model in learning to predict minority genres effectively, as it was more likely to optimize for the majority classes.

As a result, minority genres often had lower precision and recall, as seen in the classification report. Despite using oversampling techniques to rebalance the dataset, the model still struggled to generalize effectively for these underrepresented labels. This highlighted the need for more comprehensive data augmentation strategies or advanced loss functions, such as class-weighted loss, to better handle imbalanced datasets.

Another critical challenge was related to model limitations in text tokenization processes. Pre-trained tokenization models like those used in DistilBERT operate within a fixed maximum token limit (512 tokens), requiring truncation for longer texts. This truncation often resulted in the loss of contextual information, particularly for lengthy documents with rich narrative content. Additionally, tokenization models are highly sensitive to the structure of input data, and slight variations in text formatting or preprocessing could significantly impact performance. As result, ensuring compatibility between tokenization techniques and the dataset while maintaining computational efficiency proved to be a complex balancing act.

In short, these challenges emphasize the need for tailored solutions, such as collecting more diverse and balanced datasets, experimenting with advanced models capable of handling longer sequences or integrating hybrid tokenization strategies to preserve contextual information in longer texts. These efforts would improve model accuracy and ensure better predictions across all genres.

C. FUTURE WORKS

To improve the model's performance and address existing challenges, several enhancements are proposed. Firstly, rebalancing data for minority genres using advanced oversampling techniques or class-weighted loss functions is critical. These methods can help the model better learn underrepresented genres like "Action" and "Mystery," improving overall recall and precision for minority labels. Secondly, dynamic fine-tuning of classification thresholds for each genre based on performance metrics, such as precision-recall trade-offs, could significantly enhance prediction accuracy. This approach ensures more balanced and precise predictions across both dominant and minority genres. We also suggest implementing a hybrid framework combining multiple transformer models for optimized performance. For example, DistilBERT can efficiently handle shorter texts, Longformer can manage medium-length documents, and Reformer, capable of processing sequences up to 64,000 tokens, can tackle ultra-long content. This integrated approach would ensure versatility and scalability across diverse input lengths. Lastly, leveraging the vast text sources from fandoms, fan-wikis, Reddit, and Discord highlights the broad application potential of this project. Techniques like replacing characters' names with generic pronouns could further reduce overfitting and improve generalization across datasets. These strategies aim to address current limitations, enhance the model's robustness, and expand its applicability to various NLP tasks.

Recent research has explored various strategies to overcome the token limit constraints in transformer models. For instance, the Unlimiformer framework introduces a method that enables the processing of virtually unlimited input lengths by offloading cross-attention computations to a k-nearest-neighbor index, thereby circumventing the fixed token limit without modifying the model architecture [29] [30]. Another approach, ChunkBERT, proposes segmenting long texts into smaller chunks and applying convolutional layers to aggregate information, allowing BERT models to handle longer texts efficiently [31].

VI. CONCLUSION

Our experiments with several transformer models proved to be successful, as they all yielded results that are within our expectations and thus are sufficient for the task that we envisioned our models to perform. We hereby recommend these models (especially those with a larger token limit) to be used for text genre classification purposes. This will help websites hosting literature works partially automate the tagging process, ensuring that readers gain access to new works to their liking as quickly as possible. Further experimentation with more capable transformers, fine-tuning, and larger input may also allow tagging systems to detect certain themes that exist in the works, allowing readers to look for what they would like to read more precisely. Changing the input to text of a different domain may similarly help article-oriented websites automatically categorize their content.

ACKNOWLEDGMENT

We would like to thank Dr. ANK Zaman for his guidance and feedback throughout the course of this project, and Google Colab and Kaggle for providing an openly available training platform that we can use to train our models. The Hugging Face community also offered significant supports when we were working with models.

REFERENCES

[1] I. Goodreads, "Goodreads | meet your next favorite book," 2024. [Online]. Available: https://www.goodreads.com/

[2] E. Ozsarfati, E. Sahin, C. J. Saul, and A. Yilmaz, "Book genre classification based on titles with comparative machine learning algorithms," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 14–20.

[3] S. Srinivasan, S. G. Shivanirudh, S. Sathya, and T. T. Mirnalinee, "Exploring bayesian uncertainty modeling for book genre classification," in *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2022, pp. 179–183.

[4] Rahul, Ayush, D. Agarwal, and D. Vijay, "Genre classification using character networks," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 216–222.

[5] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[6] ——, "Longformer," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/longformer

[7] K. Appiah, "How many words in a novel? word counts by genre," Aug 2023. [Online]. Available: https://www.thenovelry.com/blog/how-many-words-in-a-novel

[8] Scrapy, "Scrapy 2.11.1," Feb 2024. [Online]. Available: https://docs.scrapy.org/en/latest/news.htmlscrapy-2-11-1-2024-02-14

[9] O. for Transformative Works, "Archive of Our Own," 2024. [Online]. Available: https://archiveofourown.org/

[10] NLTK, "nltk.stem.WordNetLemmatizer," Jan 2023. [Online]. Available: https://www.nltk.org/api/nltk.stem.WordNetLemmatizer.html

[11] scikit learn, "LabelEncoder," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[12] ——, "MultiLabelBinarizer," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MultiLabelBinarizer.html

[13] C. W. Schmidt, V. Reddy, H. Zhang, A. Alameddine, O. Uzan, Y. Pinter, and C. Tanner, "Tokenization is more than compression," 2024. [Online]. Available: https://arxiv.org/abs/2402.18376

[14] PyTorch, "AdamW," 2023. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[16] ——, "DistilBERT," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/distilbert

[17] S. Y. Ng, K. M. Lim, C. P. Lee, and J. Y. Lim, "Sentiment analysis using distilbert," in *2023 IEEE 11th Conference on Systems, Process Control (ICSPC)*, 2023, pp. 84–89.

[18] L. Nige, C. Lu, Z. Lei, T. Zhenning, W. Zhiqiang, S. Yiyang, and G. Xiaolin, "A web attack detection method based on distilbert and feature fusion for power micro-application server," in *2023 2nd International Conference on Advanced Electronics, Electrical and Green Energy (AEEGE)*, 2023, pp. 6–12.

[19] G. Xiong and K. Yan, "Multi-task sentiment classification model based on distilbert and multi-scale cnn," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2021, pp. 700–707.

[20] J. Bai, R. Cao, W. Ma, and H. Shinnou, "Construction of domain-specific distilbert model by using fine-tuning," in *2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2020, pp. 237–241.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[22] ——, "Roberta," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta

[23] K. Nikita, K. Łukasz, and L. Anselm, "Reformer: The efficient transformer," 2020. [Online]. Available: https://arxiv.org/pdf/2001.04451

[24] ——, "Reformer," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/reformer

[25] G. Wu and J. Zhu, "Multi-label classification: do hamming loss and subset accuracy really conflict with each other?" 2020. [Online]. Available: https://arxiv.org/abs/2011.07805

[26] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "Regret analysis for performance metrics in multi-label classification: The case of hamming and subset zero-one loss," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 280–295.

[27] T. Barszcz, M. Bielecka, A. Bielecki, and M. Wójcik, "Wind turbines states classification by a fuzzy-art neural network with a stereographic projection as a signal normalization," in *Adaptive and Natural Computing Algorithms*, A. Dobnikar, U. Lotrič, and B. Šter, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 225–234.

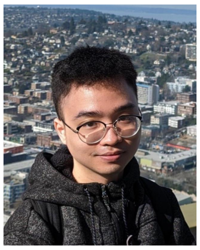
[28] C. Kundu and L. Zheng, "Deep multi-modal networks for book genre classification based on its cover," 2020. [Online]. Available: https://arxiv.org/abs/2011.07658

[29] A. Bertsch, U. Alon, G. Neubig, and M. R. Gormley, "Unlimiformer: Long-range transformers with unlimited length input," 2023. [Online]. Available: https://arxiv.org/abs/2305.01625

[30] ——, "Unlimiformer: Long-range transformers with unlimited length input (neurips 2023)," Jan 2024. [Online]. Available: https://github.com/abertsch72/unlimiformer

[31] A. Jaiswal and E. Milios, "Breaking the token barrier: Chunking and convolution for efficient long text classification with bert," 2023. [Online]. Available: https://arxiv.org/abs/2310.20558

GIA PHAT HUYNH received the B.S in Software Engineering from FPT University, Vietnam, in 2017, where he was one of three students selected for a fully sponsored internship at Kyushu Institute of Technology, Japan. With over five years of experience as an IT Business Analyst, he played versatile roles in software development for Kaplan International Education, bridging technical and business needs effectively. He is now pursuing a Master of Applied Computing at Wilfrid Laurier University, Waterloo, Ontario.



THE MINH NGUYEN received the B.S. in software engineering from Washington State University, Everett, WA, USA in 2022. He is currently pursuing the Master of Applied Computing at Wilfrid Laurier University, Waterloo, Ontario.

THAI SON TRUONG received the B.E. degree in computer science and engineering from Ho Chi Minh City University of Technology, Vietnam National University, in 2008. He spent 3 years as a software developer in Ho Chi Minh city, Vietnam, before heading south to Singapore. In Singapore, he spent 12 years playing various roles in software development for Singtel and Schroder Investment Management SG. The tenure of 8 years with an influencing financial services player, Schroders group, has ignited his interests for the computing and machine learning engineering for domain-specific businesses. That leads him to currently pursuing the Master of Applied Computing at Wil- frid Laurier University, Waterloo, Ontario.