Bert-Powered Book Genre Classification

Stanford CS224N Custom Project
TA mentor: Swastika Dutta
No External Collaborators, Mentors, and Sharing Project

Jessica Chen and Karen Wang
Department of Computer Science
Stanford University
xjchen01@stanford.edu, karenw24@stanford.edu

Abstract

The vast number of e-books available presents a challenge in automatically labeling and organizing them. This paper addresses this issue by proposing a genre classification model that uses only book titles, a readily available and informative input, to predict one of 32 genres. The main contributions of this paper include extensive experiments on various approaches to finetune a Bidirectional Encoder Representations from Transformers (BERT) model for book title genre classification, including pretraining on a Masked Language Modeling objective, performing Supervised Contrastive Learning, and using the Sharpness-Aware Minimization optimizer. The final model achieved 71.2% accuracy, outperforming previous models and baseline architectures. However, a limitation of our model is that it assumes each book belongs to a single genre and does not account for the possibility of books fitting into multiple genres.

1 Introduction

The digital revolution has resulted in an abundance of digital books. However, the large number of e-books and unstructured information has made it challenging for maintainers to automatically label and organize them. Machine learning techniques can be used to tackle this problem and make subject labeling more efficient, especially when e-books are from anonymous sources and require significant human effort to label correctly. Using the entire book as input for genre classification can be computationally intensive and time-consuming due to the large amount of data to be processed.

In contrast, book titles are a more accessible option than using the entire book, as they are readily available and shorter and more concise, allowing for quicker and easier processing. Despite not always being indicative of a book's genre or topic, book titles are an essential component of a book's presentation, as they often contain clues about the book's content and purpose. Therefore, this project aims to create a genre classification model that analyzes book titles, which are typically the first point of attraction for potential readers, to simplify the classification process.

The limitations of previous models have led to the exploration of more advanced techniques for book genre classification. Bidirectional Encoder Representations from Transformers (BERT) is a pretrained deep learning model that has been shown to be highly effective in natural language processing tasks. BERT is capable of contextual understanding of text and can capture the nuances and complexities of language, making it an ideal candidate for book genre classification using only book titles. With the ability to handle both short and long sequences of text and detect relationships between words, BERT has the potential to outperform previous models and improve the accuracy of book genre classification with titles.

For this project, we implemented a BERT-based classifier that categorizes books by their titles into one of 32 genres (Table 5). We were able to achieve a 71.2% accuracy, 67.5% precision, 66.4% recall, and 66.9% F1-Score, better than the baseline architectures as well as human performance (Table 1). The rest of the paper is organized as follows. Section 2 describes the related works. Section 3 provides the approach of the project. Experimental results are presented in Section 4 and qualitatively analyzed in Section 5. Final conclusions are drawn in Section 6.

2 Related Work

The task of book genre classification has been the subject of numerous studies in literature. Previous literature have employed book summaries or complete texts as input data for classification (Worsham and Kalita, 2018), and others have utilized computer vision techniques to classify book genres based on their cover images (Gupta et al., 2019; Kundu and Zheng, 2020). Although previous research has explored the use of book titles for genre classification, the studies have been limited to simple neural network architectures such as RNN, CNN, and LSTM (Ozsarfati et al., 2019). More complex models often combine information from both book covers and titles and/or only categorize books into broad categories such as fiction versus non-fiction (Biradar et al., 2019). These approaches disregard the challenges and intricacies of genre categorization with only book titles.

Worsham et al. introduce the Gutenberg Dataset for Genre Identification and study how current deep learning models compare to traditional methods for this task (Worsham and Kalita, 2018). They explore various machine learning approaches and found that using an ensemble of chapters can significantly improve results in deep learning methods, and ultimately achieved 84% accuracy. Recognizing the computational expense of training on entire texts, Biradar et al. explored using the combination of both the cover and title of the book to predict the genre (Biradar et al., 2019). They used logistic regression as the classification model and obtained an accuracy of 87.2% when using both cover and title features. The authors also compared the performance of using only image features or title features separately and found that the title features were more expressive of the genre.

Ozsarfati et al. investigated different machine learning algorithms to predict the genre of a book solely based on its title (Ozsarfati et al., 2019). To prepare the data, they performed several preprocessing steps, including using word embeddings, and evaluated the performance of five different models. The authors found that the Long Short-Term Memory (LSTM) with a dropout achieved the highest accuracy of 65.58%. This paper is the first to present a book genre classifier based solely on the title, and we believe that building off more advanced models such as BERT can further enhance the performance of the classifier.

3 Approach

In this project, we are building a BERT-based model that takes the title of the book as input and classifies each title into one of 32 genres (Table 5). The WordPiece tokenizer is used to split input sentences into individual word pieces, which are then converted into token ids for use in the BERT model. The model then utilizes a trainable embedding layer that sums the token embeddings, the segmentation embeddings, and the position embeddings, each with a dimensionality of 768. The base BERT model uses 12 Encoder Transformer layers, each consisting of multi-head attention, followed by an additive and normalization layer with a residual connection, a feed-forward layer, and a final additive and normalization layer with a residual connection. Multi-head Self-Attention involves a scaled-dot product operation that is applied across multiple different heads and is then normalized using a softmax function, allowing the model to jointly attend to information from different representation subspaces at different positions.

3.1 minBERT

Our project leverages the Default Project Starter Code and implements the Multi-head Self-Attention and the Transformer Layer from Section 3 of the Default Project Handout. We also implemented the `step()` function of the Adam Optimizer based on Decoupled Weight Decay in Section 4. This functional minBERT uses Cross Entropy Loss and has been experimentally verified by training it for Sentiment Classification and achieving the expected evaluation metrics. With `classifier.py` as a reference, we built our own minBERT `genre_classifier.py` specific to our downstream task. This minBERT genre classifier, before any finetuning, will serve as our baseline model for future evaluation. Refer to Table 1 for more information regarding all baselines.

3.2 Masked Language Modeling

Masked Language Modeling (MLM) is a pretraining objective that is commonly used in the context of BERT-based models (Devlin et al., 2018). The MLM objective involves randomly masking a percentage of tokens in a given input sequence, with the goal of predicting the masked tokens using contextual information from the surrounding words. By doing so, the model can learn to better understand the semantics and relationships between different words and phrases in a given language.

For our project, we used MLM to further pretrain our BERT model with target-domain data. By doing so, we hope to enhance the model's understanding of the specific language used in the domain of book titles, which tend to be punchier, less grammatical, and more attention-grabbing than other texts. During the MLM pretraining process, we randomly masked 15% of all tokens in the title and replaced "masked" words with (1) `[MASK]` 80% of the time, (2) a random token 10% of the time, (3) the unchanged, original token 10% of the time. This should prevent the mismatch between pretraining and finetuning since `[MASK]` never appears in finetuning. While the concept is borrowed from the original BERT paper, the code was implemented from the ground-up without any outside references.

3.3 GloVe Embeddings

After pretraining our BERT model with MLM, we turned our attention to finetuning the model for the specific task of genre classification. We experimented with adding a layer of static GloVe embeddings to our BERT model to improve its performance. The goal was to leverage the benefits of both models – BERT's ability to capture the contextual meaning of words and GloVe's strength in representing the global co-occurrence statistics of words in a corpus (Pennington et al., 2014). Due to the difference in tokenization between BERT and GloVe, we converted between the two tokenization methods and concatenated a matrix of GloVe embedding lookups, which would produce fixed-length vector representations for each title, to our current sum of token, segmentation, and position embeddings.

3.4 Supervised Contrastive Learning

Meanwhile, we implemented our own version of Supervised Contrastive Learning (SCL) as it has shown promising results in improving the performance of models on downstream tasks by learning better feature representations (Khosla et al., 2020). In our case, we split the book titles into pairs of positive and negative examples, where the positive pair consists of two titles belonging to the same genre, and the negative pair consists of two titles belonging to different genres. The goal of the SCL algorithm is to maximize the similarity between positive pairs and minimize the similarity between negative pairs. This is achieved through the use of a Contrastive Loss instead of the original Cross Entropy Loss, which is defined as:

$$\mathrm{L}^{sup} = \sum_{i=1}^{2N} \mathcal{L}_i^{sup} = \sum_{i=1}^{2N} \left[\frac{-1}{2N_{\tilde{y}_i} - 1} \sum_{j=1}^{2N} \mathbf{1}_{i \neq j} \cdot \mathbf{1}_{\tilde{y}_i = \tilde{y}_j} \cdot \log \frac{\exp\left((z_i \cdot z_j / \tau)\right)}{\sum_{k=1}^{2N} \mathbf{1}_{i \neq k} \cdot \exp\left((z_i \cdot z_k / \tau)\right)} \right]$$

Here, z_i and z_j are the feature vectors and τ is a temperature parameter that controls the balance between smoothness and optimization difficulty. $N_{\tilde{y}_i}$ is the total number of titles in the minibatch that have the same genre, $\tilde{y}_i$, as the anchor, i. By contrasting examples from different genres, the model can learn more informative and distinct representations of different genres and avoid learning confounding factors that are shared among multiple genres.

3.5 Optimizers, BERT Models, and More

Optimizers We started with the AdamW optimizer we implemented. However, we wanted to explore other optimizer options as well, so we experimented with stochastic gradient descent (SGD) and the recently proposed sharpness-aware minimization (SAM) optimizer (Foret et al., 2020). SAM is a modification of SGD that addresses the problem of sharp local minima in the loss landscape by optimizing the loss function at different levels of sharpness, allowing the model to find a flatter minimum. Therefore, although our initial plan was to use smoothness-inducing adversarial regularization to combat overfitting, we ultimately opted for SAM due to its built-in regularization mechanism that promotes smoother decision boundaries, leading to a reduction in overfitting and an improvement in generalization performance, all without the need for supplementary regularization techniques. The code for SAM is borrowed from this repo.

BERT Models While we started finetuning the BERT-base-uncased model, which has 12 layers and 110 million parameters, we also considered whether the BERT-base-cased model would be better suited for our task. Additionally, we explored using the BERT-large-uncased model, which has 24 layers and 340 million parameters, to see if the increased model complexity would lead to further improvements in performance. Experimental results comparing the performance of finetuned BERT-base-uncased, BERT-base-cased, and BERT-large-uncased models are presented in Table 2 in Section 4.1.

Hyperparameter Finetuning Furthermore, we attempted a hyperparameter sweep using Weights and Biases, a cloud-based machine learning experimentation platform, to explore the effect of varying hyperparameters on model performance. However, due to time constraints, we were unable to conduct a comprehensive sweep and instead, opted for 10 random trials.

Data Imbalance We considered sampling methods such as undersampling and oversampling to account for the class imbalance between genres. However, as the ratio between the smallest class ("Gay & Lesbian", 1339 books) and the largest class ("Travel", 18338 books) is roughly 1:13, we have determined that the degree of imbalance is not significant enough and chose to prioritize other experiments.

4 Experiments

Data We trained and evaluated our minBERT model on a dataset publicly released by Akshay Bhatina for the Judging a Book by its Cover project at `https://github.com/akshaybhatia10/Book-Genre-Classification/tree/master/data` Iwana et al. (2016). This dataset contains 207,575 English book titles, each corresponding to one of 32 genres, labeled by genre ID. A distribution is provided in Table 5. The data will be split 70% train, 10% dev, and 20% test. We only used the train and validation sets to train and tune our model. At the end, we evaluated our final model on the test set. We converted all words to lower case (except when using BERT-base-cased) and then tokenized the titles using the existing BERT tokenizer. Additional pretraining with MLM is performed on all book titles in this dataset. All other tasks will use this dataset with book title as input and book genre ID as output.

Experimental Details For each model, Table 6 shows the Learning Rate (LR), Batch Size (BS), Number of Epochs, Dropout Rate, Option (Pretrain (P): the BERT parameters are frozen vs. Finetune (FT): BERT parameters are updated), Optimizer, and Temperature.

Evaluation Method Since accuracy, precision, recall and F-1 score are the standard classification metrics we have seen in other papers, we will also be using these existing scores to compare our model against our baselines.

Baselines We used the models in the IEEE paper *Book Genre Classification Based on Titles with Comparative Machine Learning Algorithms* Ozsarfati et al. (2019) as baselines for our initial minBERT.

Model	Accuracy	Precision	Recall	F1 Score
LSTM	65.58%	64.18%	**63.92**%	64.05%
GRU	58.28%	59.57%	59.30%	59.43%
RNN	55.91%	54.24%	53.97%	54.10%
CNN	63.10%	59.86%	59.72%	59.79%
Naive-Bayes	55.40%	55.40%	55.73%	55.56%
Bi-LSTM	64.28%	64.12%	63.73%	63.92%
Baseline minBERT	**69.01%**	**65.43**%	63.90%	**64.54**%

Table 1: Baseline minBERT compared to Ozsarfati et al. (2019)

Our baseline minBERT model (evaluated on the validation set) performed better than all of Ozsarfati et al.'s architectures in accuracy, precision, and F1 score; it is slightly worse than LSTM in recall by just 0.02%.

4.1 Results

BERT Models We trained our baseline minBERT from some different pretrained BERT models available on Hugging Face, and evaluated them on the validation set.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
BERT-base-uncased	69.0%	65.4%	**63.9**%	64.5%
BERT-base-cased	68.3%	64.5%	63.4%	63.6%
BERT-large-uncased	**69.4**%	**66.4**%	63.7%	**64.7**%

Table 2: Comparing Pretrained BERT Models

Here, BERT-large-uncased is the best performing model in terms of accuracy, precision, and F1 score. This is as expected because BERT-large-uncased has 12 more transformer layers, 230 more million parameters, and a larger embedding dimension, allowing the model to encapsulate much more information than BERT-base-uncased. Since BERT-base-cased underperforms BERT-base-uncased, we concluded that capitalization does not offer much insight into categorizing book titles as the first initial of most words are capitalized in titles.

Model Performances Table 3 illustrates the validation performance of our minBERT model after each modification described in Section 3. The hyperparameter details and differences among the models are specified in Table 6.

Model	Accuracy	Precision	Recall	F1 Score
Baseline minBERT	69.0%	65.4%	63.9%	64.5%
Base + MLM	69.4%	65.5%	64.3%	64.9%
Base + MLM + GloVe	68.5%	64.4%	63.6%	63.9%
Base + MLM + SCL	69.6%	66.1%	64.3%	65.3%
Base + MLM + SCL + SGD	68.9%	65.0%	63.6%	64.1%
Base + MLM + SCL + SAM	70.8%	67.0%	66.2%	66.6%
Finetuned Final Model	**71.2%**	**67.5%**	**66.4%**	**66.9%**

Table 3: Model Performances

4.2 Discussion

Pretraining on MLM improved the model accuracy from 69.0% to 69.4%. This is likely due to MLM's role in helping the model to better understands the semantics and contexts of book titles. However, the improvement is not as significant as anticipated, possibly because MLM is more beneficial for longer sequences, whereas book titles are usually shorter.

However, experimentation showed that adding GloVe embeddings did not improve our model's performance, but instead decreased each metric by $\sim 1\%$. We hypothesize that this is because BERT already encompasses the information that GloVe embeddings provide, and the additional layer of embeddings only adds noise and interferes with the learning of BERT. Thus, we decided to stick with the original BERT architecture without any additional GloVe embeddings.

Supervised Contrastive Learning (SCL) further improved the model accuracy from 69.4% to 69.6%. This result is as expected since SCL is effective at maximizing the dissimilarity between inputs in different classes. However, the temperature hyperparameter was set to a default value of 0.1 and was not tuned for our particular task, limiting the potential performance gains from SCL.

The Stochastic Gradient Descent (SGD) optimizer exhibits worse performance than the AdamW optimizer. We expected this result as the weight decay features of AdamW optimizer could have helped BERT to converge more quickly and effectively while reducing overfitting compared to SGD.

Based on these metrics, we can conclude that the model's performance is reasonable, but there is still room for improvement. The fact that the accuracy is higher than the precision and recall suggests that the model may be better at correctly identifying negative instances (i.e., instances that do not belong to the positive class) than positive instances. This could be due to class imbalance, where there are many more negative instances than positive ones. To further improve the model's performance, it may be worth exploring techniques such as oversampling, undersampling, or data augmentation to address class imbalance, or finetuning hyperparameters such as the regularization strength or learning rate.

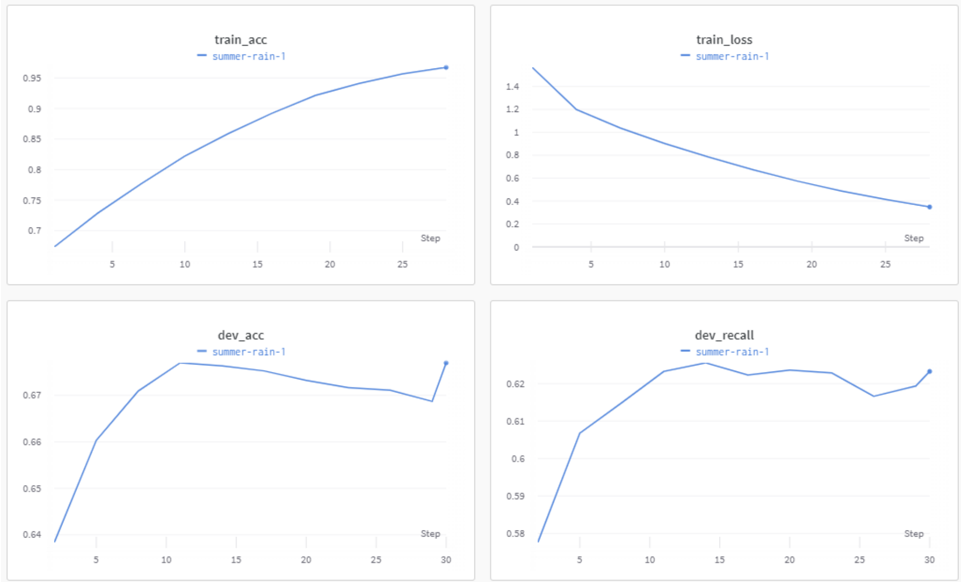

Figure 1: Performance Metric Graphs (Base + MLM + SCL)

From Figure 1, one observes that the train accuracy smoothly increases and is very high (> 97%). Meanwhile, the dev accuracy plateaus after some number of steps and even marginally decreased. This indicates that the model is overfitting to the train dataset and needs some regularization. The Sharpness-Aware Minimization (SAM) optimizer solved this issue as it avoids sharp minima and finds flat minima that generalizes better to new data.

From here, we conducted a random hyperparameter sweep with 10 trials and found a set of better parameters. Due to time constraints, we were unable to perform a more comprehensive search so we believe that there are more optimal parameter sets we have not explored yet. Nonetheless, we were able to develop a final genre classifier model that employs a pretraining approach with Masked Language Model objective, integrates Supervised Contrastive Learning to maximize dissimilarity between inputs in different classes, and utilizes the SAM optimizer for smoothness and regularization to achieve improved performance compared to the baseline model. Our Finetuned Final Model uses the current most optimal set of hyperparameters (Table 6).

Test Performance We evaluated our final model on the test set using 6 different random seeds.

	Accuracy	Precision	Recall	F1 Score
Average Validation Evaluation	71.2%	67.5%	66.4%	66.9%
Average Test Evaluation	71.0%	67.5%	66.3%	66.9%
Standard Deviation (on Test Set)	0.287%	0.562%	0.129%	0.238%

Table 4: Test Performance and Standard Deviation (Different Seeds)

The close match between the test performance and validation metrics suggests that our Finetuned Final Model is likely to generalize well to new, unseen data. As the validation set is used to tune the model's hyperparameters, and the test set serves as an independent evaluation of its performance on new data, a close alignment between the two metrics is indicative of the model's reliability and robustness. By performing well on both sets, the model has effectively captured the underlying patterns in the data without overfitting to the training set. To further assess the stability of the model's performance, we evaluated it on different random seeds and found that the standard deviation of the results was low. This indicates that the model is not highly sensitive to the choice of random seed and that its performance is consistent across multiple runs.

5 Analysis

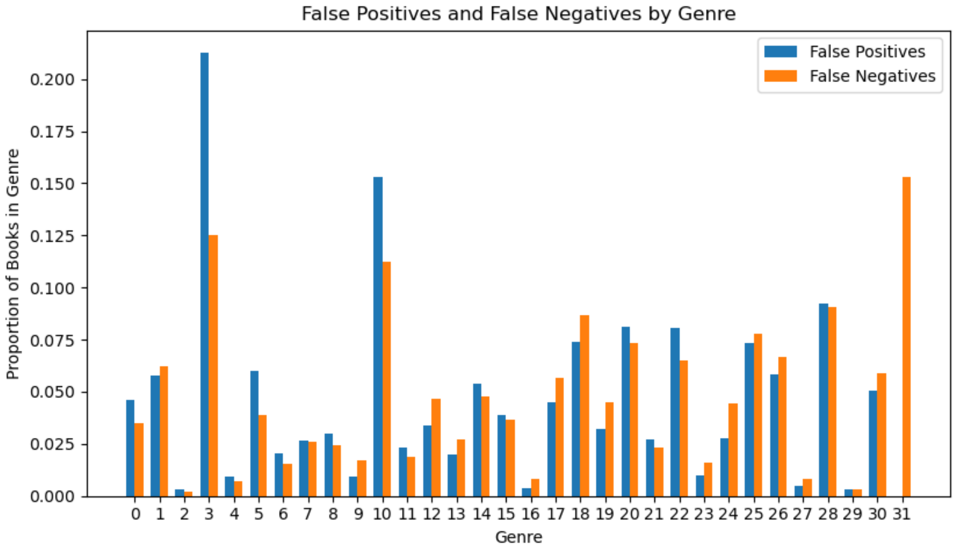

Figure 2: False Positive (FP) and False Negative (FN) Proportions by Genre in Test Evaluation

5.1 Error Analysis

Based on the FP and FN depicted in Figure 2, we observed that Genres 2, 4, 9, 16, 23, 27, and 29 (Table 5) have extremely low FP and FN (< 2%), indicating that the model has high accuracy in identifying them. This may be because they have distinct vocabulary and language patterns in their titles that are more easily distinguishable for the model. For example, titles of Medical Books (Genre 16) may contain more technical terms and jargon that are unique to the medical field. Furthermore, these genres may have more clear and distinct boundaries than other genres, and books within these genres may share more similarities in their titles than books in other genres.

On the other hand, Genres 3 and 10, Calendars, and Engineering & Transportation respectively, have exceptionally high FP compared to FN. This could be due to the fact that terms used in the book titles in these genres are also commonly used in other genres as well, leading to confusion for the model. For example, "year", a common indicator of a Calendar, is often also present in Young Adult novels or History books, making it difficult for the model to correctly categorize the titles. Moreover, Engineering & Transportation is a relatively broad category and the model may struggle to differentiate between book titles that fall under this category and those that belong to related fields such as Science & Math or Travel.

Furthermore, Genre 31 has no FP but relatively high FN. It is worth noting that Genre 31, Education & Teaching is a relatively small class (1,664 books, less than 1% of the dataset), which could result in less training data and less overall information for the model to learn from. The lack of false positives may also indicate that the model is avoiding classifying book titles as Education & Teaching in order to maximize its overall accuracy at the expense of lower recall, especially if there are other genres that the model is more confident in.

5.2 Human Performance

We conducted a manual labeling task for 100 book titles each to compare and contrast the performance of a model against human performance. We achieved an average accuracy of 44%, which is lower than the model's performance. This is not too surprising as it is quite difficult to distinguish the subtle difference among 32 genres without prior training. Despite this, both we and the model performed well for genres such as Business & Money and Travel. However, our accuracy on the Calendar category was noticeably higher than the model's, possibly because certain language used in the

Calendar category is more easily identifiable by humans. On the other hand, the model has a much higher accuracy than us in categories like Children's Books and Teen & Young Adult as we were often confused between the two genres. The model's higher accuracy could be due to the fact that children's books often have more distinct and recognizable patterns in their titles, such as the use of character names or specific themes. It is worth noting that both we and the model performed poorly on the Engineering & Transportation category, likely due to the challenge of distinguishing it from similar categories such as Science & Math.

During the manual labeling task, we discovered that it can be challenging to define each class (as some are open to interpretation), which can lead to inconsistencies and errors in the labeling process. For example, some books could be classified into multiple genres, while others did not fit neatly into any of the existing categories. However, the genre classification system used in our study is just one way of categorizing books, and there are many other possible sets of genres. Moreover, we identified a possibility of mislabeling books in our dataset, which is a prevalent problem in several other datasets and can have adverse effects on the model's performance. Also, due to the large difference between the two of us, we want to note that human performance in the labeling process can be affected by factors such as expertise, background knowledge, and personal bias. Therefore, it is crucial to carefully choose and train human annotators to ensure that the labeling process is consistent and accurate.

6 Conclusion

In this paper, we conducted extensive experiments to investigate the various approaches to finetuning BERT for classifying book genres based on their titles. Our experimental findings include: 1) while BERT-large-uncased overperforms BERT-base-uncased, BERT-based-cased decreases model accuracy compared to BERT-based-uncased; 2) additional pretraining on a Masked Language Modeling objective improves model performances; 3) additional GloVe embeddings adds noise and does not provide model improvements; 4) Supervised Contrastive Learning provides a small accuracy improvement; 5) Sharpness-Aware Minimization optimizer empirically prevents overfitting and increases our model performance. Our minBERT outperformed existing literature and original baselines in book genre classification task 1. Our Finetuned Final Model achieved a 71.2% accuracy, improving our minBERT by 2.2%.

The primary limitation of our model is that it assumes each book belongs to a single genre and does not account for the possibility that a book may fit into multiple genres simultaneously. In reality, books can often be associated with multiple genres that are not mutually exclusive, such as a historical romance novel or a science-fictional mystery. As our model only assigns a single genre to each book, it may not accurately capture the full range of genres that a book belongs to, which can impact the usefulness of the classification output. In order to overcome this limitation, a multi-label classification approach could be considered, where multiple genre labels can be assigned to each book. This would require modifications to the model architecture and training process, but would ultimately result in a more comprehensive and accurate genre classification system.

Based on our error analysis, we believe that a promising area of exploration is to incorporate the differing performance among genres in our model. For example, oversampling the worst performing genres can lead to more accurate classification results for all genres. Additionally, data augmentation can be utilized to create additional training data by applying transformations to the existing data. This can make up for the smaller classes and help the model to learn more robust features that are less sensitive to small changes in the input data.

References

Ganeshprasad R Biradar, JM Raagini, Aravind Varier, and Manisha Sudhir. 2019. Classification of book genres using book cover and title. In *2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT)*, pages 72–723. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Shikha Gupta, Mohit Agarwal, and Satbir Jain. 2019. Automated genre classification of books using machine learning and natural language processing. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 269–272. IEEE.

Brian Kenji Iwana, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, and Seiichi Uchida. 2016. Judging a book by its cover.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Chandra Kundu and Lukun Zheng. 2020. Deep multi-modal networks for book genre classification based on its cover. *arXiv preprint arXiv:2011.07658*.

Eran Ozsarfati, Egemen Sahin, Can Jozef Saul, and Alper Yilmaz. 2019. Book genre classification based on titles with comparative machine learning algorithms. In *2019 IEEE 4th International Conference On Computer And Communication Systems (ICCCS)*, pages 14–20. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Joseph Worsham and Jugal Kalita. 2018. Genre identification and the compositional effect of genre in literature. In *Proceedings of the 27th international conference on computational linguistics*, pages 1963–1973.

A Appendix

This table shows the genre distribution of our dataset.

Label	**Category**	**# Books**	**Label**	**Category**	**# Books**
0	Arts & Photography	6,460	16	Medical Books	12,089
1	Biographies & Memoirs	4,261	17	Mystery, Thriller & Suspense	1,998
2	Business & Money	9,965	18	Parenting & Relationships	2,523
3	Calendars	2,636	19	Politics & Social Sciences	3,402
4	Children's Books	13,605	20	Reference	3,268
5	Comics & Graphic Novels	3,026	21	Religion & Spirituality	7,559
6	Computers & Technology	7,979	22	Romance	4,291
7	Cookbooks, Food & Wine	8,802	23	Science & Math	9,276
8	Crafts, Hobbies & Home	9,934	24	Science Fiction & Fantasy	3,800
9	Christian Books & Bibles	9,139	25	Self-Help	2,703
10	Engineering & Transportation	2,672	26	Sports & Outdoors	5,968
11	Health, Fitness & Dieting	11,886	27	Teen & Young Adult	7,489
12	History	6,807	28	Test Preparation	2,906
13	Humor & Entertainment	6,896	29	Travel	18,338
14	Law	7,314	30	Gay & Lesbian	1,339
15	Literature & Fiction	7,580	31	Education & Teaching	1,664

Table 5: Book Genres and Sizes

This table indicates the hyperparameters associated with each experiment.

Model	LR	BS	Epochs	Dropout	Option	Optim	Temp
Pretrained minBERT	1e-3	64	10	0.3	P	AdamW	N/A
Pretrained casedBERT	1e-3	32	10	0.3	P	AdamW	N/A
Pretrained largeBERT	1e-3	32	4	0.3	P	AdamW	N/A
Baseline minBERT	1e-5	32	10	0.3	FT	AdamW	N/A
Finetuned casedBERT	1e-5	32	10	0.3	FT	AdamW	N/A
Finetuned largeBERT	1e-5	32	4	0.3	FT	AdamW	N/A
Base + MLM	1e-5	32	10	0.3	FT	AdamW	N/A
Base + MLM + GloVe	1e-5	32	10	0.3	FT	AdamW	N/A
Base + MLM + SCL	1e-5	32	10	0.3	FT	AdamW	0.1
Base + MLM + SCL + SGD	1e-5	32	10	0.3	FT	SGD	0.1
Base + MLM + SCL + SAM	1e-5	32	10	0.3	FT	SAM	0.1
Finetuned Final Model	1e-4	16	12	0.32	FT	SAM	0.1

Table 6: Hyperparameter Details.