Genre Classifications using Book and Film Descriptions

Stanford CS224N Default Project

Shaunak Bhandarkar
Department of Mathematics
Stanford University
shaunakb@stanford.edu

Mattheus Wolff
Department of Computer Science
Stanford University
mbwolff@stanford.edu

Ari Webb
Department of Computer Science
Stanford University
arijwebb@stanford.edu

Abstract

In the domain of transfer learning, it is common to pretrain a language model on a large corpus of out-of-domain (but relevant) training data and then fine-tune the model on a limited dataset of in-domain examples. In recent years, transformer-based pretrained language models such as BERT and GPT-3 have proven to be effective for fine-tuning on many downstream tasks. However, fine-tuning to a different domain can often be tricky, given that the in-domain examples typically stem from a different distribution than the data that the model was originally trained on. In our project, we specifically aimed to investigate how well BERT can be fine-tuned to the downstream task of genre classification of movies given IMDb movie descriptions as well as to the downstream task of genre classification of books given Goodreads book descriptions. We also combined these tasks with three other unrelated tasks in a multitask classifier using the SMART regularization and optimization framework to see if that would improve results for all tasks. We found that fine-tuning for a small number of epochs (3-4) led to the performance across all models; moreover, multitask training with two of the component tasks being IMDb and Goodreads genre classification led to comparable performance on these classification tasks to if BERT is individually fine-tuned on these tasks, but improved accuracies for the three other tasks.

1 Key Information

- External collaborators (if you have any): N/A
- Mentor (custom project only): N/A
- Sharing project: N/A

2 Introduction

In the last decade, models like BERT and GPT-3 have performed impressively on various language classification and generation tasks. Such large language models are often pretrained on massive language datasets to build flexible, general-purpose internal representations; subsequently, these models are fine-tuned on specific downstream tasks of interest. The primary goal of fine-tuning is to adapt a model to a particular task while maintaining relevant information from the source distribution that the model was pretrained on.

Given that downstream tasks typically involve (relatively) smaller datasets with different data distributions, effectively fine-tuning large language models is difficult. Data availability is one such challenge. Additionally, having too little training data can lead to overfitting, which hurts test accuracy and generalization. Furthermore, it can be unclear how many epochs of fine-tuning tend to produce the best performance, or whether training on multiple tasks while using shared internal parameters can add stability to the fine-tuning process.

In this paper, we investigate the fine-tuning problem in a specific context: book and movie genre classification. In particular, we explore how well BERT can be fine-tuned to two key tasks:

1. Genre classification of IMDb movies given movie titles and descriptions.
2. Genre classification of books listed on goodreads.com given their descriptions.

Additionally, one of the benefits of exploring book and movie classification is that these two tasks are highly similar, meaning that they may be especially well suited to a multitask training regime along with other tasks. We thus investigated how well genre classification could be learned from a multitask learning framework as opposed to individually fine-tuning a model on a single genre classification task, and whether the the performance on other unrelated tasks improved with the addition of these genre classification tasks.

Outside of exploring genre classification, we also investigate regularization as a potential technique for improving BERT's fine-tuning capability. Recently, Jiang et al (2020) proposed a dual regularization technique, Smoothness-inducing Adversarial Regularization and Bregman Proximal Point Optimization (SMART), for robust fine-tuning of pretrained langugage models, such as BERT [1]. This regularizer firstly incorporates a notion of smoothness, meaning that the model's output distribution over genre classes should not shift wildly for slight perturbations in the input to the model. Second, they propose using the Bregman proximal-point method, which adds a regularization term for preventing aggressive weight updates to the model.

For our problem of genre classification, we ask whether SMART regularization can help improve genre classification after fine-tuning BERT. In particular, if SMART does not lead to an increase in classification accuracy, this would motivate the need to further refine a suitable regularization for genre classification.

3 Related Work

Fine-tuning: There is great interest in fine-tuning as it relates to the problem of transfer learning. Work on transfer learning spans over two decades, and encompasses classification, regression, and clustering problems [2]. Various fine-tuning frameworks, with a particular application towards transformers, have been proposed in recent years to retain knowledge over the course of fine-tuning and prevent catastrophic interference [3, 4]. Work has also been done to make fine-tuning less computationally intensive [5].

Genre Classification: Deep learning approaches - often involving transformers or attention - have been used for various classification tasks such as music genre classification [6, 7]. More general approaches in machine learning have investigated genre prediction for books and movies from titles and user reviews [8, 9].

Regularization: Various regularization techniques have been proposed to streamline the fine-tuning process for large language models. Such methods involve experimenting with dropout [10], contrastive self-training [11], injecting noise into hidden layers [12], or by imposing word embedding-related constraints [13].

4 Approach

4.1 Baselines

Our baselines are very standard: our basline model is BERT [14] and we use AdamW [15, 16] as our optimizer. For our baseline tests, we performed pretraining and finetuning (separately) of BERT for sentiment analysis on the Stanford Sentiment Treebank (SST) and CFIMDB datasets (provided to us).

To fine-tune a separate copy of BERT on each of these datasets, a linear layer was added to the output of each BERT copy (for each of these datasets).

Additionally, we performed multitask training of BERT on the SST, Question Pairs Quora Dataset, and the ACL Anthology Dataset (STS). In order to do this, we added separate linear layers for each of these tasks to the end of a single BERT model; this way, all three tasks share the same underlying BERT embeddings.

4.2 Genre Classification

To perform book and movie genre classification using BERT, we utilized a similar approach to that mentioned above. In particular, for each of IMDb and Goodreads genre classification tasks, we appended a linear layer to a separate copy of BERT for each of these tasks. We used the Adam [15] optimizer for training.

For IMDb movie genre classification, BERT (plus the final linear layer) was trained to perform single-genre classification, as each IMDb movie in our dataset (detailed in Section 5 below) is only assigned a single genre label. Therefore, we measured the accuracy of IMDb movie genre classification by tallying how many examples (out of the total number of examples) BERT classified correctly. On the other hand, books on goodreads.com are listed with multiple genre labels, meaning that BERT's prediction of the genre labels could be "partially" correct. Given this additional complexity, our approach was to use 3 different metrics to gauge the accuracy of fine-tuned BERT's performance on book genre classification:

1. **All-or-None:** For each example, we note whether fine-tuned BERT exactly predicts the multi-hot vector corresponding to the true genre labels of the given book; if so, we call this a correct prediction. The accuracy of the model is then the number of correct predictions divided by the number of examples.
2. **Equal-Book-Weight:** Equal book weight accuracy computes the percentage of correct genres on a per book basis, then averages to get the accuracy. The following equation computes equal book weight. n is the number of books in the set, c_b is the number of correct genres for book $b \in \{1, 2, \ldots, n\}$, and $F(b, g) = 1$ if the model correctly predicted that genre g belongs to book b and 0 otherwise. For notational purposes, let $g_{b,1}, g_{b,2}, \ldots, g_{b,c_b}$ denote all c_b correct genres for book b. Then, the equal-book-weight accuracy over all n books is

$$EBW = \frac{\sum_{b=0}^{n} \frac{\sum_{i=1}^{c_b} F(b, g_{b,i})}{c_b}}{n}.$$

3. **Equal-Genre-Weight:** Equal genre weight computes overall how many genres were predicted correctly in the entire batch. Using the same notation as above, the following equation computes equal-genre-weight accuracy:

$$EGW = \frac{\sum_{b=0}^{n} \sum_{i=0}^{c_b} F(b, g_{b,i})}{\sum_{b=0}^{n} c_b}.$$

Additionally, we performed unified multitask training on the following five datasets: SST, Question Pairs Quora dataset, ACL Anthology (STS), IMDb genre dataset, and Goodreads genre dataset. Again, in order to do this, we added separate linear layers for each of these tasks to the end of a single BERT model; this way, all five tasks shared the same underlying BERT embeddings.

4.3 SMART Regularization

Recall that SMART consists of two regularizer terms: a smoothness-inducing adversarial regularizer (for adding stability in predictions despite slight perturbations in the input) and a Bregman term (for preventing aggressive weight updates).

Smoothness-inducing Adversarial Regularizer: Let w denote the first embedding layer of BERT, and let f be all the subsequent layers of the BERT model (plus the additional linear layer) including an additional softmax to obtain a discrete probability distribution over the genre classes. Moreover, letting θ be the parameters of the model, the smoothness-inducing term is obtained through an approach similar to the one used in [1]. For any current training input x, let $w := w(x) \in \mathbb{R}^k$ denote the (continuous) word embedding obtained after applying the first layer of BERT. We first sample a small noise term $\epsilon \sim \mathcal{N}(0, \sigma^2 I_{k\times k})$ (we set $\sigma =$ 1e-5). We then define $\tilde{w} := w + \epsilon$. We first compute $l_s(f(w;\theta), f(\tilde{w};\theta))$, where

$$L := l_s(P, Q) = \text{KL}(P||Q) + \text{KL}(Q||P).$$

To encourage robustness against noise, we aim to minimize L (i.e. encourage the distributions P and Q to be close). Our approach (similar to [1]) is to use a single projected gradient ascent to obtain the largest possible value of L in a small neighborhood of w (and subsequently try to minimize this value through the main Adam optimization loop). We perform the update

$$\epsilon \leftarrow \frac{\epsilon + \eta\nabla_\epsilon L}{||\epsilon + \eta\nabla_\epsilon L||_\infty}$$

(where we set the step size $\eta :=$1e-3). Finally, we define our regularization term to be

$$R_s(\theta) := l_s(f(w;\theta), f(\tilde{w};\theta)),$$

where again $\tilde{w} = w + \epsilon$. Additionally, when the input x to the model is a batch of B examples, then we simply average over the regularization terms for each example:

$$R_s(\theta) := \frac{1}{B}\sum_{i=1}^{B} l_s(f(w_i;\theta), f(\tilde{w}_i;\theta)).$$

Bregman Point-Proximal Optimization: To obtain the full SMART regularization, we let θ_t denote the parameters of the model at each timestep t; in particular, θ_0 denotes the parameters of pretrained BERT (plus the final linear layer). At each timestep, we perform the optimization

$$\theta_{t+1} := \arg\min_\theta \left(\mathcal{L}(\theta) + \lambda_s R_s(\theta) + \mu\mathcal{D}_{\text{Breg}}(\theta, \theta_t)\right),$$

where $\mathcal{D}_{\text{Breg}}(\theta, \phi) := \frac{1}{B}\sum_{i=1}^{B} l_s(f(w_i;\theta), f(w_i;\phi)$; we perform this computation through a single gradient step using the aforementioned Adam optimizer. Moreover, during our experiments (detailed in Section 5), we set $\mu := 1$ and λ_s to 1 or 3.

5 Experiments

5.1 Data

We used a total of six datasets. For our base multitask implementation, we used four datasets: two for sentiment analysis (SST and CFIMDB), one for paraphrasing detection (Question Pairs Quora dataset), and one for similarity rating (ACL Anthology / STS). For our genre classification tasks, we used two more datasets. We used one dataset for movie genre classification: the IMDb movie dataset (found here). For book genre classification, we produced our own custom scraped goodreads.com book genre dataset (found here, we drew inspiration for the scraper from here). For each dataset, we randomly split the samples in the following way: 70% training, 20% dev, and 10% test.

For the sentiment analysis task, we used the Stanford Sentiment Treebank dataset [17] and the CFIMDB dataset (both provided to us). The SST dataset consists of 11,855 sentences from movie reviews, which were labelled as one of the following 5 classes: negative, somewhat negative, neutral, somewhat positive, positive. The CFIMDB dataset contains 2,434 movie reviews which are each more than one sentence. For the paraphrase detection task, we used a subset of the Quora dataset (used for paraphrase detection), found here. Our subset contains 200,000 examples of question pairs, which are marked with a binary classification indicating whether they are paraphrases of each other. For the similarity task, we used the SemEval STS Benchmark dataset, which consists of 8,628 sentence pairs alongside a number on a scale from 0 (unrelated) and 5 (same meaning).

We sampled a total of $\sim$ 50,000 examples from the IMDb dataset; each example consisted of a movie ID, a title, a true genre label (consisting of a single genre), and a movie description. Additionally, we sampled a total of 5,820 book examples from goodreads.com, in which each example consisted of a book description, a book title, and a list of true genre labels (often with multiple labels).

5.2 Evaluation method

We describe two evaluation metrics: accuracy and correlation. We used the all-or-nothing accuracy metric to evaluate the performance of the sentiment analysis, paraphrase, movie genre prediction tasks and book genre prediction tasks. This metric is simply the number of correct predictions divided by the total number of predictions. Movie genres were encoded as one hot vectors, so thus were easy to judge as correct or incorrect. Each book had multiple genre tags, so were represented as a multi-hot vector. For the all-or-nothing accuracy metric, the model needed to predict all of the multi-hot vector correctly in order to be correct.

Because of the unforgiving nature of all-or-nothing multi hot classification, and the fact that Goodreads books are listed with multiple genre labels, we decided to compute two other accuracy metrics for the book genre prediction task: equal book weight and equal genre weight (detailed in Section 4.2). Equal book weight accuracy computes the percentage of correct genres on a per book basis, then averages to get the accuracy. This approach is useful because it gives each book in a batch equal weighting, regardless of how many genres it was tagged with. Equal genre weight computes overall how many genres were predicted correctly in the batch. Equal genre weight is useful because it shows how many genres overall the model is able to predict, yielding intuition regarding the models general performance at deducing genres from textual prompts.

We used the correlation metric to evaluate the performance of the similarity task (i.e. ACL Anthology / STS), since measuring similarity is a graded phenomenon. Correlation values range from -1, which means a negative correlation, to 1 which is perfect positive correlation between ground truth and prediction.

5.3 Experimental details

As a baseline, we pretrained and fine-tuned BERT for just sentiment analysis on the SST and CFIMDB datasets, using the AdamW optimizer. Training was done for 10 epochs each time, and a hidden dropout rate of 0.3 was used. During pretraining, BERT's internal parameters are frozen and gradients are only applied to the linear heads attached to BERT (learning rate = 1e-3); during fine-tuning, gradients are passed through the full model, including the BERT module (learning rate = 1e-5). We use a small learning rate for fine-tuning to ensure that the pretrained parameters of BERT will not be drastically changed in a way that catastrophically interferes with its existing body of knowledge.

Then, we developed the multitask BERT for just the three default tasks (SST, Question Pairs Quora dataset, and STS). We used cross-entropy loss for the sentiment task (SST), binary cross-entropy for the Quora paraphrase task, and mean-squared error loss for the similarity task (STS). For multitask BERT training on these three tasks, we utilized a batch-level interleaving process, whereby one epoch consisted of randomly interleaving batches from each task, until all batches from the three tasks have been processed. Training was done for 10 epochs using a batch size of 64 and a hidden dropout rate of 0.3.

Building on our baselines, we built models (consisting of BERT plus a linear head) for movie genre prediction and book genre prediction, using the Adam optimizer [15] and fine-tuned using cross-entropy loss. For the IMDb training data, we explored how many epochs of fine-tuning would lead to optimal performance on the dev set; accordingly, we fine-tuned the IMDb classifier for 10 epochs, so as to examine at which epoch dev and test performance would peak. We fine-tuned the Goodreads classifier for 4 epochs. Once these genre classification models were working, we combined them with the previous three tasks into a full five-task classifier. Further, we developed batch-level interleaving for training, meaning that each epoch consists of a random interleaving of batches from each task until all batches from all five tasks have been run through the multitask model.

We ran the above three models (IMDb genre classification, Goodreads genre classification, unified multitask model) two times each, once with SMART regularization and once without. Each model (excluding the original IMDb classifier, with and without SMART) was fine-tuned for 4 epochs with a learning rate of 1e-5 and a batch size of 8. For runs involving SMART, the smoothness-inducing adversarial regularization rate was set to 1 or 3 and the Bregman learning rate was set to 1.

As a final step, we performed zero-shot cross-predictions for genre classification. That is, we loaded in our best-performing IMDb classifier and computed the zero-shot accuracy for genre classification on

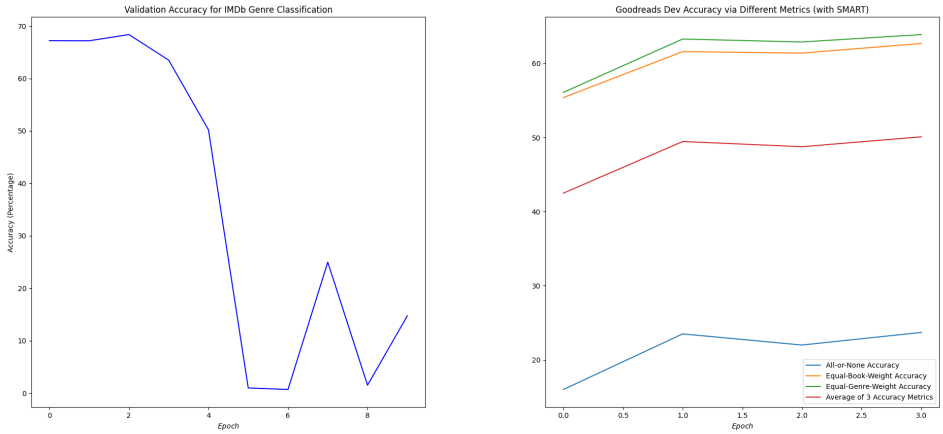

Figure 1: Accuracy of IMDb genre classifier on the IMDb dev set (left) and of Goodreads genre classifier on the Goodreads dev set (right). Both of these models used SMART regularization.

the Goodreads test dataset; we also loaded in our best-performing Goodreads classifier and computed the zero-shot accuracy for genre classification on the IMDb test dataset.

5.4 Results

We first pretrained and fine-tuned the single task BERT for sentiment analysis on both the SST and CFIMDB datasets, and the resulting accuracies are shown in Table 1. For our baseline tests of pretraining and fine-tuning BERT on the three default tasks (SST, Quora paraphrase, and STS), our results are shown in Table 2.

We then developed both the movie and book genre classifiers. We first ran both classifiers on their respective datasets (IMDb for movie and goodreads for book) both with and without SMART regularization. We then configured each classifier to evaluate "zero-shot" on the opposite dataset to see whether a BERT model fine-tuned for movie genre classification could do well on book genres and vice versa. The zero-shot accuracies are not directly comparable to the original accuracies, as the genre sets for movies were different than the ones for books, with only some overlap. The final results of these tests are shown in Table 3.

We ran IMDb fine-tuning for 10 epochs, shown on the left in Figure 1. After about epoch 4, validation accuracy plummets, bolstering the idea that fine-tuning of BERT should be done for $\sim$ 3-4 epochs [14]. On the right of Figure 1 is fine-tuning on the goodreads dataset, which we only ran for 4 epochs. The different lines represent the three different book genre accuracy metrics (discussed in Section 4.2), and there is one additional line that represents the average of the three metrics. All three metrics increased over the course of the 4-epoch fine-tuning period.

Once we had all 5 models working, we unified them into a multitask classifier. We then ran the 5 task classifier with and without SMART regularization. The results of the full 5-task classifier are shown in Table 3. The results for the other three tasks are shown in Table 4. The training and dev accuracies of all five tasks for the unified 5-task classifier without SMART are shown in Figure 2, while the same metrics for the 5 task classifier with SMART are shown in Figure 3.

	SST	CFIMDB
Pretraining	39.1%	77.6%
Fine-tuning	52.5%	96.3%

Table 1: Baseline Accuracies for Pretraining and Fine-tuning on Sentiment Classification

	Pretrain Final Accuracy	Fine-tuning Final Accuracy
Sentiment (SST)	19.8%	51.4%
Paraphrase (Quora)	68.9%	75.7%
Similarity (STS)	.290	.237

Table 2: BERT Three-Task Multitask Classifier Results

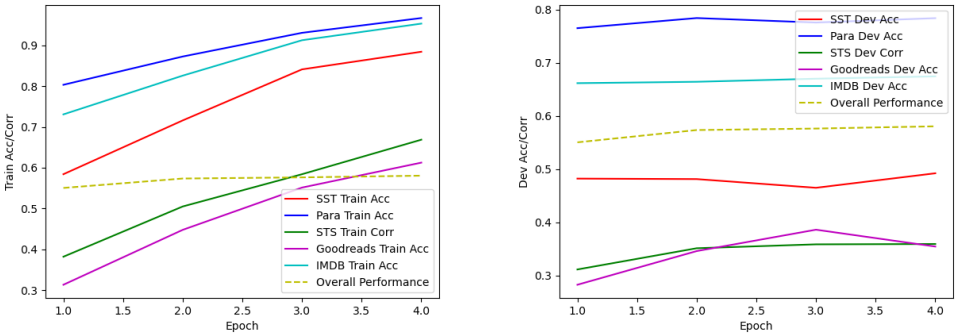

Figure 2: Train and dev accuracy plots for unified multitask training (all 5 tasks) without SMART.

6 Analysis

IMDb validation accuracy plummeted after the 4th epoch when running finetune for 10 epochs. This indicates that finetuning for too long could possibly lead to internal BERT parameters being overwritten, causing worse test accuracy. After this initial result, we ran the rest of our finetuning training sessions for only 4 epochs.

The training accuracy for the IMDB, SST and paraphrase tasks in the unified 5-task classifier without SMART became very high over just 4 epochs, especially relative to the dev accuracies (see Figure 2). This indicates a certain degree of overfitting. The same model, when run with SMART, does not have nearly as high training accuracy, seems to prevent overfitting to the training set (see Figure 3), and IMDb/Goodreads fine-tuning accuracy peaks after 2 epochs.

Overall, the genre classification results obtained from unifying all tasks into a five-task model were comparable to those of fine-tuning a single IMDb classifier on the IMDb dataset and to fine-tuning a single Goodreads classifier on the Goodreads dataset (see Table 3). This suggests that unified multitask training may be a suitable regime under which to perform genre classification (especially

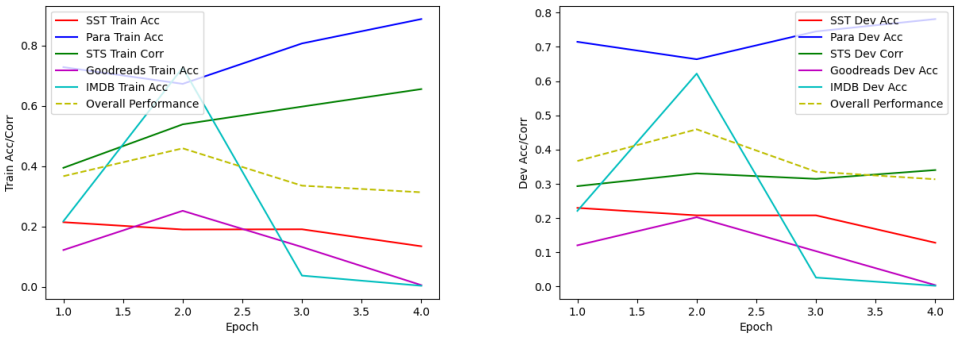

Figure 3: Train and dev accuracy plots for unified multitask training (all 5 tasks) with SMART regularization.

Dataset	Unified 5-Task	Unified 5-Task (SMART)	IMDb	IMDb (SMART)	Goodreads	Goodreads (SMART)
IMDb	**67.2%**	62.6%	**68.3%**	**68.7%**	Zero-shot: 23.9%	N/A
Goodreads[1]	**36.3**, **69.4%**, **71.5%**	22.4%, 60.2%, 61.1%	Zero-shot: 48.8%	N/A	31.1%, **70.1%**, **68.4%**	25.7%, 62.3%, 64.5%

Table 3: Movie and Book Genre Classification Results

	Unified Multitask	Unified Multitask (SMART)
Sentiment	49.2%	20.8%
Paraphrase	78.4%	66.4%
Similarity	.359	.331

Table 4: BERT Five-task (Multitask) Classifier Results For Original Three Tasks

given the underlying similarity of book and movie genre classification). Ultimately, unified multitask training encompasses a tradeoff between two competing effects. First, having more tasks could lead to "forgetting" pretrained parameters, which would cause accuracy to decrease. On the other hand, having multiple (related) tasks could lead to more robust shared embeddings, leading to better performance across the board.

For the classifier that only did Goodreads predictions, SMART regularization led to lower test accuracy. For IMDb, SMART led to slightly higher (although statistically insignificant) test accuracy. This result was unexpected. It could potentially be explained by underfitting caused by too much regularization, or in the Goodreads case, dataset size limitations.

For the zero-shot evaluations, the IMDb model performed better on the Goodreads dataset than vice versa (see Table 3). One possible reason for this is that the Goodreads model was trained to perform genre classification involving multiple labels per book, potentially causing the model to distribute its attention over various genres; in contrast, the IMDb classifier is trained to produce a single genre label, meaning that the label it does ultimately produce for a book has a nontrivial chance of being at least one of the listed genres for that book.

7 Conclusion

Our achievements in this project are as follows. We obtained solid baseline results for the original three tasks in a multitask training regime. We then fine-tuned BERT for two additional, similar downstream tasks: book and movie genre classification. We then used these fine-tuned models to make zero-shot predictions on each others datasets. We learned that BERT fine-tuning is best capped at fewer than 10 epochs; we believe the best number is 3-4. We implemented SMART regularization for all of these tasks, which had mixed results and is an area for future work. Finally, we improved upon our preliminary accuracies for the three-task classifier by adding two new genre classification tasks (IMDb and Goodreads) to form a five-task multitask training regime.

Future work would involve designing better datasets for genre classification as well as further development under the SMART framework. First, given that many IMDb movies have multiple genres, we would create a dataset that reflects this property. Additionally, improving the design of our zero-shot experiments would involve mapping book and movie genres into word vector embeddings and subsequently rating BERT's predictions (for both model evaluation and loss functions) using the notion of distance in word vector space. Future work would also involve a more principled search across SMART regularization hyperparameters. We might also consider incorporating momentum in calculating SMART regularization.

[1]Goodreads accuracies presented as a list of three values: all-or-nothing, equal-book-weight, equal-genre-weight.

References

[1] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[2] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[3] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

[4] Raymond Li, Wen Xiao, Lanjun Wang, Hyeju Jang, and Giuseppe Carenini. T3-vis: visual analytic for training and fine-tuning transformers in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 220–230, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[5] Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulić. Modular and parameter-efficient fine-tuning for NLP models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 23–29, Abu Dubai, UAE, December 2022. Association for Computational Linguistics.

[6] Yingying Zhuang, Yuezhang Chen, and Jie Zheng. Music genre classification with transformer classifier. In *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, ICDSP 2020, page 155–159, New York, NY, USA, 2020. Association for Computing Machinery.

[7] Yang Yu, Sen Luo, Shenglan Liu, Hong Qiao, Yang Liu, and Lin Feng. Deep attention based music genre classification. *Neurocomputing*, 372:84–91, 2020.

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

[9] Chandra Kundu and Lukun Zheng. Deep multi-modal networks for book genre classification based on its cover, 2020.

[10] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models, 2020.

[11] Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach, 2021.

[12] Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. Fine-tuning pre-trained language models with noise stability regularization, 2022.

[13] Kosuke Nishida, Kyosuke Nishida, and Sen Yoshida. Task-adaptive pre-training of language models with word embedding regularization, 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017.

[17] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.