Movie Scale Rating Classification using Multi-Task Learning for LSTM and Fine-Tuning Transformers

James Frech

May 20, 2025

Abstract

This analysis seeks to determine the sentiment of movie reviews by predicting scale ratings of the reviews (between 0 and 1). Multiple models are compared including a traditional machine learning approach, the random forest classifier based on a rule based sentiment analysis, and neural networks, namely the long short-term memory network (LSTM), and a ROBERTA transformer. Further, multi-task learning is used on the LSTM to determine if well aligned tasks (binned rating classifications and authorship of review) can improve the accuracy of scale rating classification for movie reviews. Results show that when trained on all five tasks considered, the LSTM shows improved performance. This improved LSTM performs on par with the random forest classifier, and both are outperformed by ROBERTA.

1 Introduction

Sentiment analysis is one of the most common tasks in natural language processing (NLP) and has many applications. Sentiment analysis is used to determine if a customer will buy a product, which candidate a person would vote for in an election, the general public's perception of a movie, and more. In the case of movie ratings, sentiment is not only used to determine if someone likes a movie (polarity), but also how strongly they feel about it (intensity). Sentiment of movies can be inferred by analyzing its reviews, often times which come accompanied with a scale rating, which allows readers to determine if the reviewer enjoyed a movie or not, as well as how strongly they felt about the movie. Many researchers have attempted to learn the explicit scale ratings of a review using many natural language processing techniques, including various machine learning models. Further, some have used reviews to not only predict the overall scale rating, but its different aspects including polarity and intensity. As scale ratings have different aspects that contribute to the overall rating, the task of learning scale ratings is an ideal case for the use of multi-task learning, a method in which a single deep learning model is trained to perform multiple tasks at once. Generally, if the tasks are well aligned, meaning each task is able to help a model learn features useful for another task, training the tasks together increases the accuracy of specific tasks compared to when only trained by themselves. Multi-task learning has shown promise for various use cases in NLP including emotion recognition [TD18, ACG+19], abstractive text summarization [KWRG22], and sentiment score classification combined with polarity and intensity of the sentiment [TLM18]. A more thorough overview of multi task learning in NLP can be found in [CZY21]. This analysis seeks to perform scale rating classification on 11 class (0.0, 0.1, ..., 1.0) using machine learning models, namely the random forest classifier (RFC) and neural networks including the long short-term memory (LSTM) recurrent neural network and ROBERTA transformer. Analysis determining if scale rating classification for movie reviews can be improved by training with additional tasks is done using the LSTM.

2 Data

This analysis makes use of the Cornell Movie Review Scale Rating Dataset [PL05]. The dataset contains 5006 paragraph movie reviews and their associated scale ratings (0.0, 0.1, ..., 1.0) provided from four authors. The data is accompanied by binned class ratings for three class classification (0: rating ≤ 0.4, 1: $0.4 <$ rating < 0.7, 2: rating ≥ 0.7) and four class classification (0: rating ≤ 0.3,

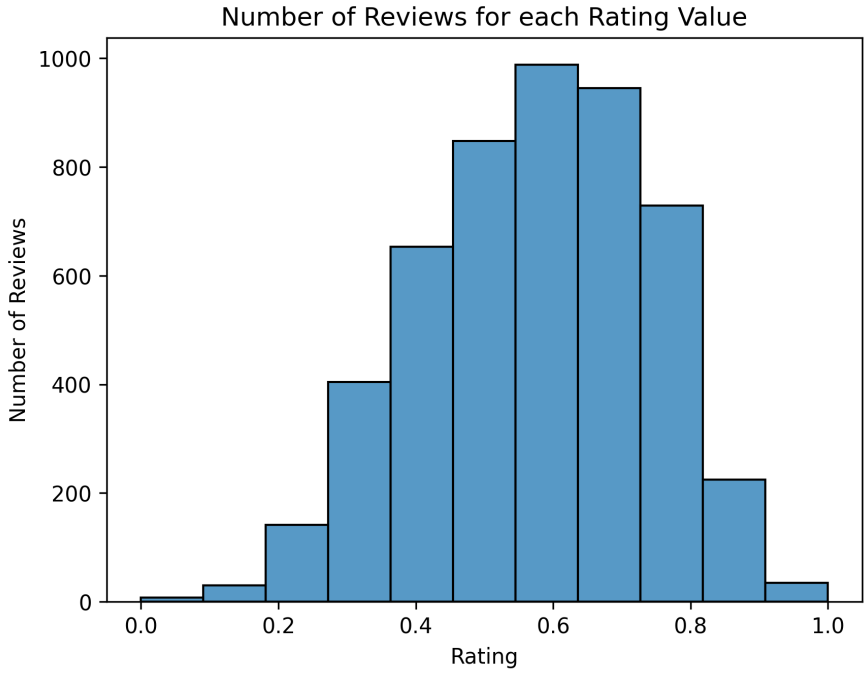

Figure 1: Distribution of scale rating reviews.

1: $0.4 \leq$ rating ≤ 0.5, 2: $0.6 \leq$ rating ≤ 0.7, 3: $0.8 \leq$ rating). While the main task in this analysis is to predict the full scale ratings (11 classes), three class and four class classification given above, as well as authorship and polarity (Negative: rating ≤ 0.5, Positive: rating ≥ 0.6) are performed as well, both individually, and as auxiliary tasks for multi-task learning. The data is split into train (70%), validation (20%), and test (10%) splits for robust tuning of hyperparameters and computation of error metrics. It should be noted that this dataset has issues with class imbalances (Fig. 1), and as such certain classes, (0.0, 0.1, 1.0) will have much higher errors. Ideally all classes should be more equally represented.

3 Methodology

3.1 Random Forest Classification

The random forest, a common, yet powerful traditional tree-based machine learning model, has shown robust performance across many disciplines for both classification and regression tasks [Bre01]. While only movie reviews are provided as features in the Cornell movie review dataset, the RFC requires features to be extracted from the raw text to make predictions. The first feature extracted from the raw text is a rule-based sentiment score computed on each individual movie rating. The rule-based sentiment analysis is performed using spacy's "SpacyTextBlob" which, among other analyses, returns a polarity score normalized between -1 (negative) and 1 (positive). As such, this feature gathers the intensity of the sentiment in the movie review, which helps to predict how someone might rate the movie on a scale. Using the spacy tokenizer, each movie review is further split into appropriate tokens, and the top 50 bigrams and unigrams from the training data are computed as input into the model. To avoid uninformative unigrams and bigrams, stopwords are removed from the analysis and only unigrams and bigrams containing adjectives are considered. This helps capture bigrams such as "excellent_show", "low_budget", "poor_show" and unigrams such as "average", "poor", "emotional", and "interesting" which all are key words which contribute to determining the sentiment behind a movie review. Once the features were prepared for the model, the RFC is tuned using five fold cross validation combining the train and validation sets. Final model accuracy is assessed on the held-out test set using F1 score. Optimal parameters chosen through five fold cross validation include a total of 600 trees in the forest, 75 maximum features allowed to make each split, and three minimum samples

per leaf node.

3.2 Long Short-Term Memory

The second model considered for the analysis is the LSTM recurrent neural network [HS97]. Unlike the random forest, the LSTM is specifically employed for sequential data, allowing it to learn features from the raw text itself once split into a sequence of tokens. For this model, tokenization is done by first lowercasing the data, splitting words and punctuations by whitespace, and adding <start> tokens to the beginning of sentences (here they are placed after each ".", "!", or "?") as well as <unk> tokens as a placeholder for unknown words. The top 5000 tokens in the training dataset are only considered, and each review is 0 padded (or cut) to a length of 500 tokens to avoid overfitting and unnecessary padding. Each token not in the top 5000 tokens is replaced with the <unk> token. Once tokenized, the tokens are passed to the LSTM in an embedding layer of 300 dimensions that are tuned for each token as part of the model training. After embedding the tokens, the data are passed to an LSTM cell with a hidden size of 64. From there, the last hidden state in the LSTM cell is input into a classification head consisting of a dropout layer followed by a fully connected layer of 10 nodes with ReLU activation, followed by another dropout layer and a final fully connected layer with output nodes equal to the number of classes. Both dropout layers have probability of dropout set to 0.3. The Adam optimizer with a learning rate of 0.001 is used along with cross-entropy loss. The batch size is set to 128.

In order to analyze the potential of multi-task learning in movie review classification, the LSTM is trained to learn each of five tasks described above (scale rating, three and four class classification, polarity, authorship). All five tasks are learned individually, scale rating is learned in conjunction with each of the other four tasks one by one for a total of four models trained on two tasks, and a final model is trained to perform all five tasks at once. Accuracy between the models for each task is compared to assess whether the tasks are well aligned and contribute to higher performance when paired together by sharing model features. All tasks only share features in the embedding layer and LSTM cell, the output of which is then sent to individual classification heads, each of which follow the same description as above. The single task models are run for 250 epochs using Google Colab's T4 GPU taking approximately two minutes each. The two and five task models are trained for 200 epochs taking approximately three and eight minutes respectively. Both three and four class classification tasks show high class imbalance, and therefore have weighted cross entropy loss functions to allow for more predictions in the less frequent classes. While the scale rating task showed even higher class imbalance, no weights were used in the loss function as weighing the rare classes more highly resulted in much lower overall model accuracy.

3.3 ROBERTA Transformer

The field of NLP has seen great strides in recent years by the introduction of new state-of-the-art models, transformers [VSP+23]. For sequence classification tasks such as the scale rating classification performed in this analysis, encoder only architectures show great promise. In order to compare the prior two models described above to a state-of-the-art model, a pretrained Hugging Face model, ROBERTA [LOG+19], is fine tuned for each task given above. For each task, the final two transformer layers and the classification head are tuned at a learning rate of 0.001 using the Adam optimizer and cross entropy loss. The fine-tuning achieves maximal results by five epochs for all tasks with the exception of scale rating classification, which achieved better performance at 15 epochs. Runtime for the transformer increased compared to the LSTMs with five epochs taking ∼3.5 mintues and 15 epochs taking ∼11.5 minutes, showing the importance of pretraining when using transformers. Fine-tuning the transformer for more than one task at a time was attempted, however, didn't show any real marked improvement. As such only single task transformer models are elaborated upon in the results.

4 Results

4.1 Multi-Task Learning

The LSTM is trained on many combinations of tasks. First, a baseline of training only on one task at a time for each task is implemented. Then, the model is trained on a combination of full scale rating

classification, and one of each of the other four tasks at a time to see whether or not a given task helps improve the scale rating classification. Finally, the LSTM is trained on all tasks at once. For consideration of the scale rating task, both the two task model with authorship as a second task and the five task model show improvement over the baseline single task model with an improvement in F1 score of 0.017 and 0.09 (out of 1) respectively, while a negligible difference is found training scale rating with three class classification and decrease in accuracy is found training with polarity and with four class classification (Fig 2). The increase in accuracy in training authorship is likely due to the differences in how each author rates movies based on their personal opinions. One author may tend to rate an average movie as a 0.5, whereas another sees an average movie as a 0.7. When trained to identify an author of a review, the model is more likely to pick up on these differences. In addition, this could possibly be part of the reason why polarity does not improve the model. Polarity for this analysis is defined as positive with a 0.6 or higher rating and negative as 0.5 and below. As different authors might view the boundary for a scale rating of a positive versus negative review should get, the actual polarity of the reviews may differ from the target in this analysis. If the authors were to provide their actual polarity on the reviews along with their scale rating scores, those provided values may actually be more informative than using the hard boundary between 0.5 and 0.6. Interestingly, the model trained on all five tasks, while showing great improvement, includes polarity and the other tasks that did not improve scale rating classification by themselves. This shows that while each task individually may not improve the accuracy of scale ratings, that learning all the different sentiment based tasks may provide enough information of the overall sentiment to improve the performance. In addition, with five tasks, the amount of data points the model is trained on increases five fold from $\sim$5000 to $\sim$25,000 which in itself is likely to provide higher accuracy.

Analyses for the other tasks give somewhat varying results depending on if a model is trained for a single task, two tasks (with scale rating), or all five tasks (Fig. 3). For the four class binned classification, results interestingly show improved performance both when trained alongside scale rating, and further when trained on all five tasks with an increase in F1 score of 0.09 and 0.13 respectively. Interestingly the similar three task classification shows decreased performance for both the two and five task models. While polarity here was determined to decrease the performance of scale rating, the opposite is not quite true as polarity shows relatively negligible change in performance for two and five task models. Finally, authorship shows differing results, where it has an increase in F1 of 0.08 when trained with scale ratings, but a decrease in F1 of 0.05 when trained on all tasks. This is likely due to similar reasons above, where being able to tell the differences in scale ratings for different authors will show improvement, but when implementing all five tasks together, the four sentiment based tasks likely dominate the weights of the model and reduce accuracy for authorship.

4.2 Model Comparison

While training an LSTM on all five tasks shows marked improvement over the base LSTM, the five task model does not actually outperform the RFC or ROBERTA (Fig. 4). The baseline RFC has comparable accuracy to the five task LSTM for polarity and scale rating, and outperforms all LSTMs for three and four class rating classification. The only time the LSTM outperforms the RFC is the authorship task, where it also almost achieves the performance of ROBERTA when trained on two tasks (scale rating, authorship) (Fig. 5). This shows that the LSTM extracts features that are able to distinguish the writing styles of the different authors quite well, but is not better for sentiment analysis than the rule based sentiment analysis, unigrams, and bigrams input into the RFC. As the features for the RFC were chosen specifically for sentiment analysis, it may show improved performance for authorship classification using input features extracted specifically to distinguish the authors' writing styles. Regardless of task, the ROBERTA transformer shows highest accuracy across all tasks.

While ROBERTA shows highest accuracy across all tasks, the predictive ability of the model (and others) varies greatly between tasks. Authorship has the highest overall accuracy by far, followed by polarity, three class classification, four class classification, and scale rating in that order. This shows that the task of authorship is easier than sentiment analysis for each of these models. In addition, while not surprising, the analysis shows that model performance on sentiment analysis deteriorates with a more fine grained scale. Similarly, we could expect the accuracy of authorship to decrease with an increase in the number of authors in the dataset. However, even though the performance decreases with the number of classes, with scale rating having a rather poor maximum F1 score of 0.31 with ROBERTA, the models still generalize rather well. Inspecting the confusion matrices for the RFC (Fig.

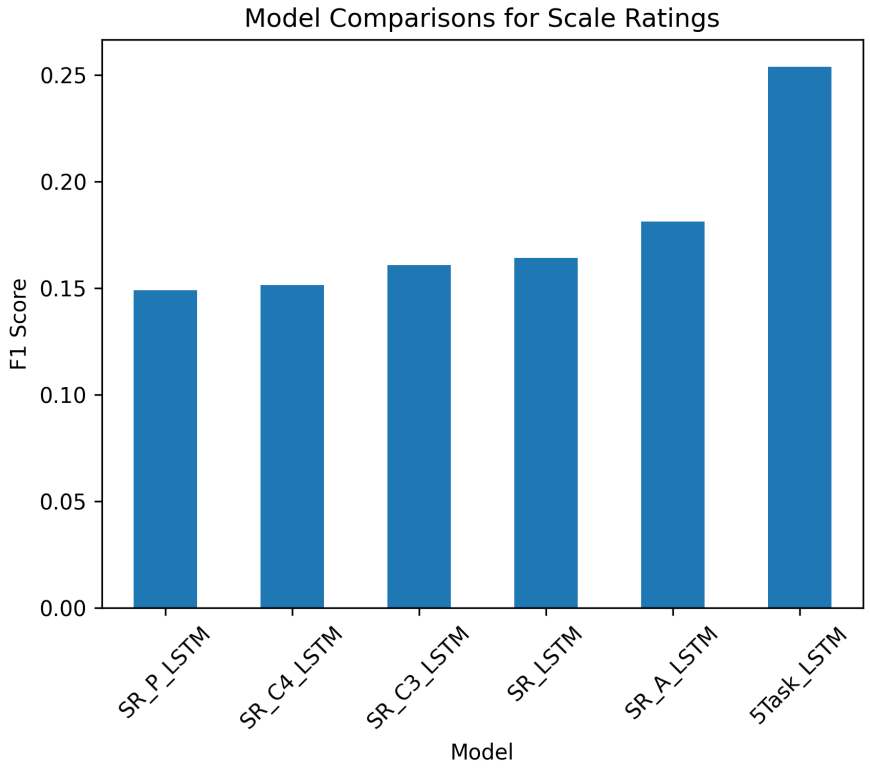

Figure 2: F1 scores for Multi-Task LSTMs trained for scale rating (SR), polarity (P), authorship (A), three class classification (C3), and four class classification (C4).

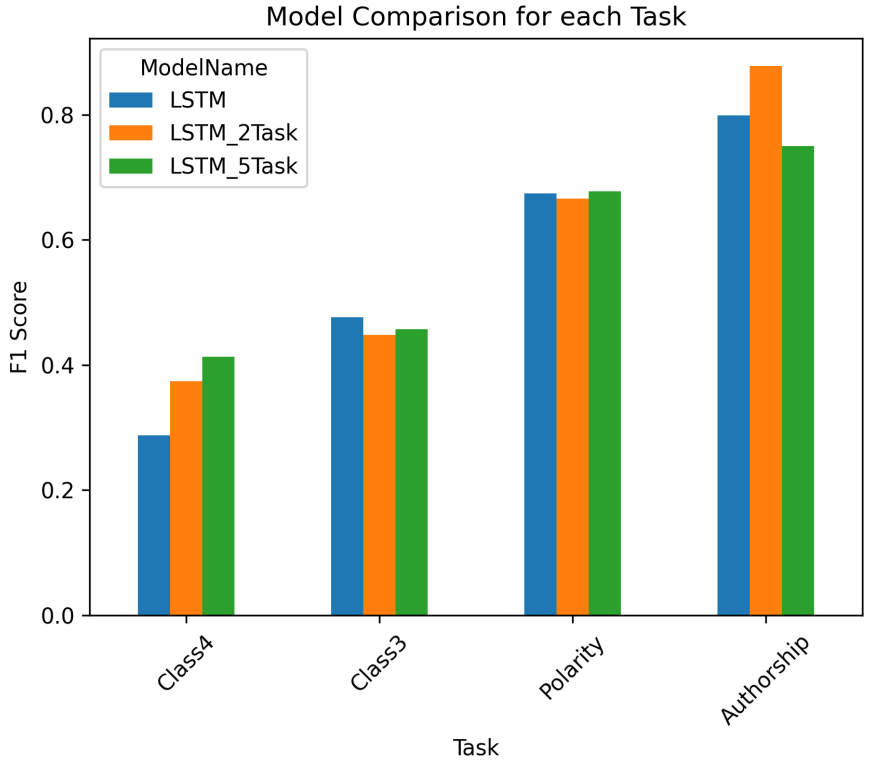

Figure 3: F1 score comparison for single-task, two-task, and five-task LSTM models on auxiliary tasks.

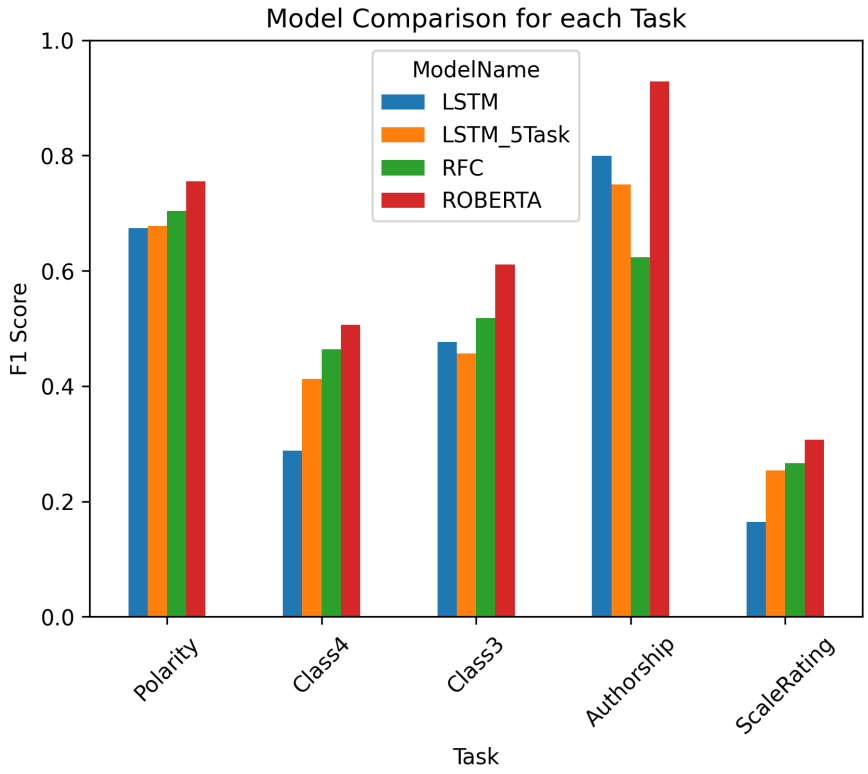

Figure 4: F1 scores for single task LSTMs, five task LSTM, random forest classification, and ROBERTA transformer.

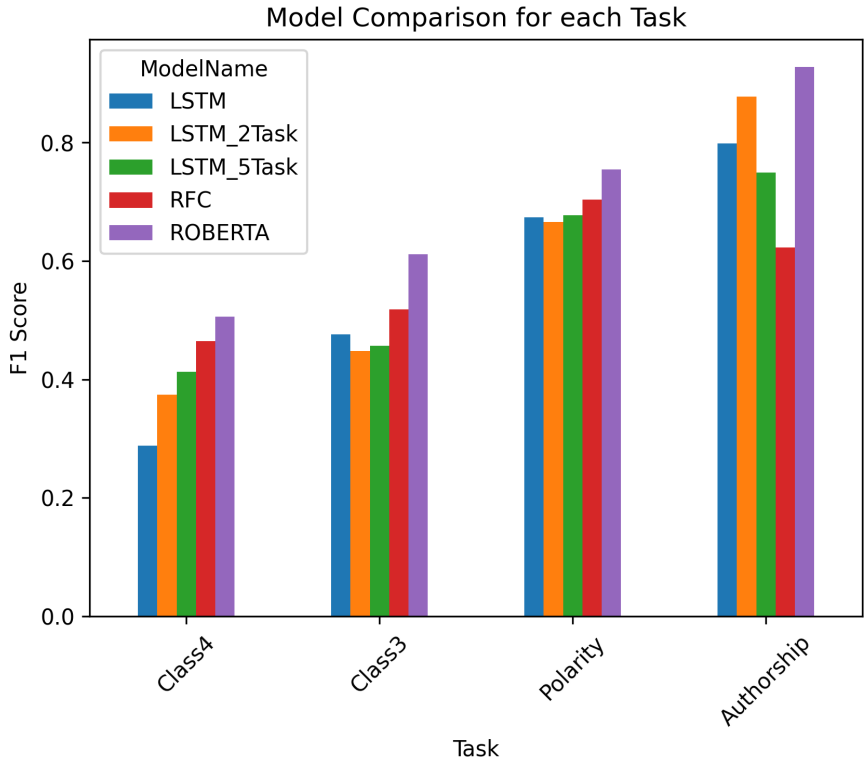

Figure 5: F1 scores for single task LSTMs, two task LSTMs, five task LSTM, random forest classification, and ROBERTA transformer.

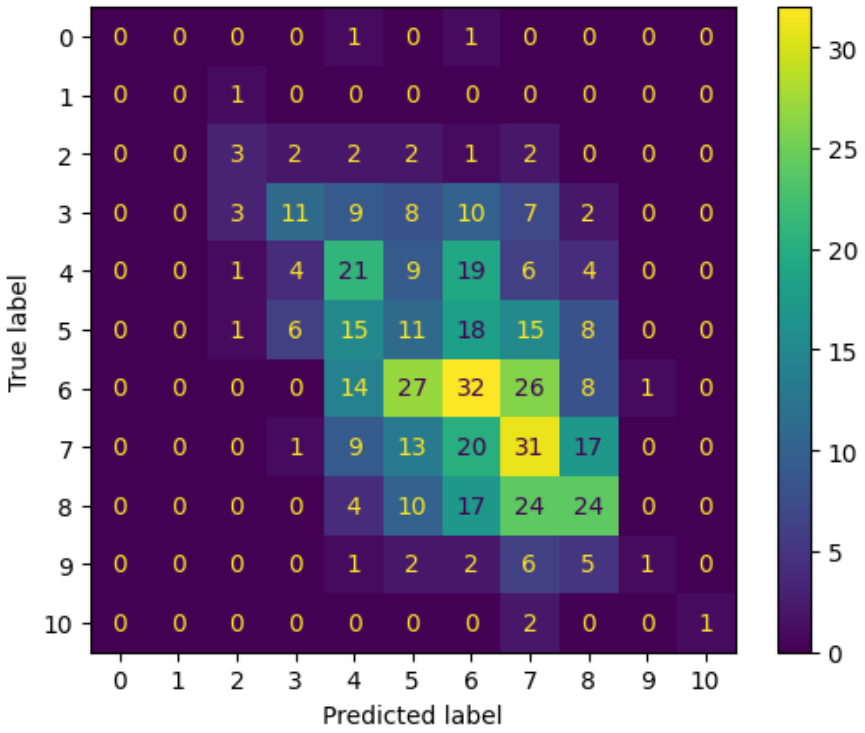

Figure 6: Confusion Matrix for the scale rating task using the RFC.

6), LSTM (Fig. 7), and ROBERTA (Fig. 8), the predictions are mostly near the diagonal, showing that while the models struggle more with getting exact results, the predictions tend to be within one or two ratings of the true value. Further analysis using top two or top three accuracy metrics may show similar values for scale ratings compared to the other binned sentiment analysis tasks.

5 Limitations

While rather in depth, this study has some limitations. The first is that there is a very large class imbalance (Fig. 1) which drives down the F1 score greatly for the scale rating task. As shown in the confusion matrices, the extremely high and extremely low ratings are almost never predicted correctly. A more balanced dataset would solve this issue. When it comes to the random forest, there are a few improvements that could potentially be made. For the sake of simplicity and due to time constraints, the top 50 unigrams and bigrams were selected, however when looking closely into them, some likely do not contain information for sentiment analysis like "Jurassic Park". In addition, stopwords were filtered out, which includes the word "not". However, bigrams such as "not_good" or "not_bad" would provide more information. Perhaps even some trigrams being included such as "not_good_movie" would improve performance. Similarly, improvements could be made for the LSTM model. The model was trained from scratch, including the embeddings, however, it is common to initialize the weights of the embeddings with Word2Vec or other embedding methods. Word2Vec is not necessarily great for sentiment analysis in particular, but perhaps tuning the Word2Vec embeddings to the sentiment analysis tasks could lead to higher accuracy. In addition, making the model bidirectional could further capture features in the reviews not captured by the one directional model. For example, if the negation of a word is after the word itself in a sequence, the bidirectional model could potentially figure this out, whereas the current model only contains information from previous tokens at each step. Lastly, if this model were to be extrapolated to a larger dataset with more authors, it would not make sense to use authorship to train the model. With the model trained to separate the reviews of authors in the dataset, it may overfit to their writing styles and fail to extrapolate to other authors not seen at train time.

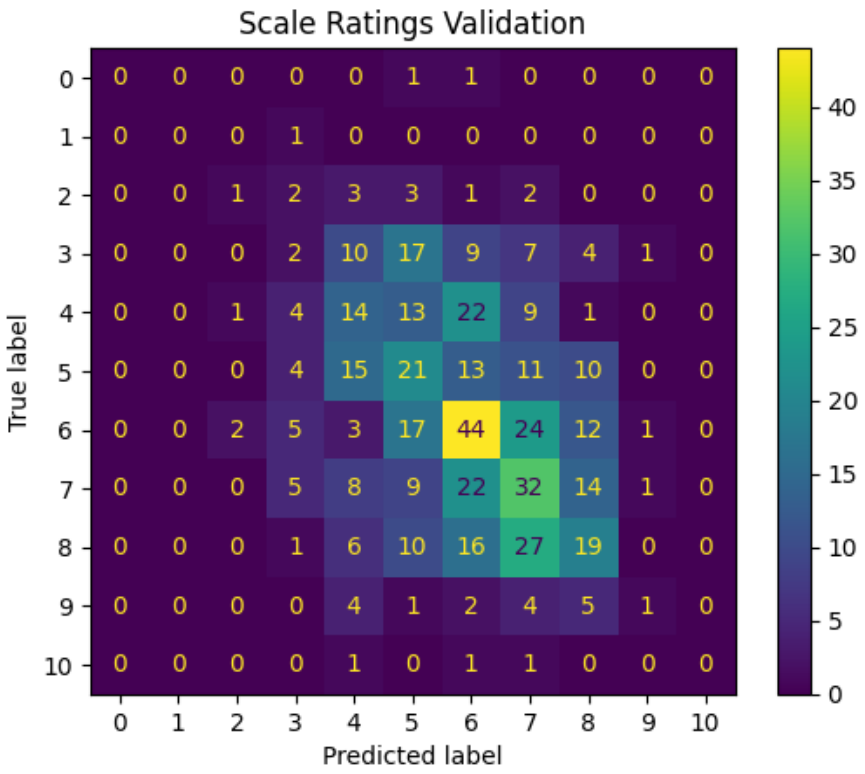

Figure 7: Confusion Matrix for the scale rating task using LSTM with five tasks.

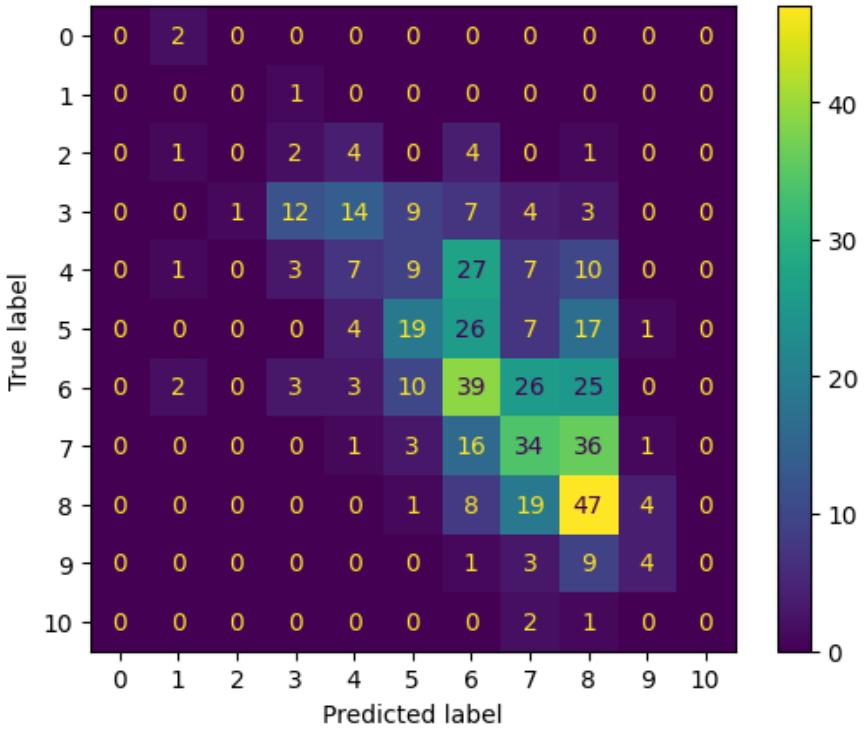

Figure 8: Confusion Matrix for the scale rating task using ROBERTA.

6 Summary

Multi-task learning shows some promise in the ability to increase the accuracy of movie review scale ratings prediction, however caution should be shown when choosing auxiliary tasks as performance may degrade with a task that is not well aligned. Even with multi-task learning, the LSTM, did not show better results than the RFC for any sentiment analysis task, but did show better results for authorship classification. In addition, while the accuracy decreased with the number of classes for sentiment analysis, the models were still able to generalize well and predict within a few scale ratings of the true value for most observations. Overall, the ROBERTA transformer outperformed the other models in every case with little tuning, showing that transformers are state-of-the-art and maintain high performance for various NLP tasks with a bit of tuning.

Code Availability

Code for this project is provided at https://github.com/JamesFrech/NLP-Movie-Scale-Rating-Classification-using-LSTM-and-Transformers.

References

[ACG+19] Md. Shad Akhtar, Dushyant Singh Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. Multi-task learning for multi-modal emotion recognition and sentiment analysis. *CoRR*, abs/1905.05812, 2019.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[CZY21] Shijie Chen, Yu Zhang, and Qiang Yang. Multi-task learning in natural language processing: An overview. *CoRR*, abs/2109.09138, 2021.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[KWRG22] Frederic Thomas Kirstein, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Analyzing multi-task learning for abstractive text summarization. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, page 54–77. Association for Computational Linguistics, 2022.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[PL05] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

[TD18] Shabnam Tafreshi and Mona Diab. Emotion detection and classification in a multigenre corpus with joint multi-task deep learning. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2905–2913, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[TLM18] Leimin Tian, Catherine Lai, and Johanna D. Moore. Polarity and intensity: the two aspects of sentiment analysis. *CoRR*, abs/1807.01466, 2018.

[VSP+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.