Good, but how good? Using RoBERTa to quantify consumer satisfaction based on IMDB movie reviews

Ole Jonas Liahagen and Martin Johannes Nilsen
NTNU, Computer Science, 2022
[olejlia, martijni]@stud.ntnu.no

Abstract

Every year consumers are bombarded by trailers and teasers for upcoming movies. Production studios promise the best movie ever made, only for the movie to fall short when it is finally released in cinemas. This comes at a cost of both time and money for the potential consumer. Due to this, consumers often opt to wait for reviews of the movie in order to judge whether or not it is worth watching. In this report, we want to explore an approach to further supplement and aid consumers in the process of selecting movies. We will fine tune the state of the art transformer model RoBERTa to attempt to improve on the results of movie rating prediction achieved by more traditional NLP approaches. Movie rating predictions can be of great use for streaming services and broadcasting companies utilising recommender systems to provide consumers with content fit for them. Providing a continuous scale of ratings rather than a binary good or bad could help improve the match between movies and consumer interest. Although the experiments in this report are conducted exclusively on movie reviews, the authors believe that the use of advanced language models in conjunction with recommender systems in other domains can be done in a similar manner with promising results.

1 Introduction

In recent years the field of natural language processing has been introduced to a new kind of model addressing many of the key problems of previous state of the art models such as LSTMs and recurrent neural nets. The new models are large language models based on the transformer architecture. The transformer architecture allows new language models such as BERT and RoBERTa to keep track of the relationships between all parts of a sequence - e.g. where the model should focus it's attention. This allows for a powerful new way to represent sentences and thus also extract further important smantic information. With these tools up their sleeve, large language models have quickly become the state of the art models for close to all NLP benchmark tasks. One such language model that has made waves recently is the RoBERTa model based on the well known BERT model. The RoBERTa model further builds upon BERT's qualities by constructing a larger training set and tweaking training steps to increase performance. In this report we wish to explore the capabilities of the RoBERTa model, both as an intermediary and as a standalone classification model. We will explore disadvantages to using large language models as well as judging the models ease of use. To do this we combine the already well researched IMDB movie review dataset (Maas et al., 2011) with a dataset also containing neutral reviews to combat the polarity of the original dataset.

2 Background Theory

2.1 The transformer architecture

Previous machine learning methods for natural language processing struggle to handle long sequences of text. This rings especially true when the models are applied to problems where context is important. The transformer architecture (Vaswani et al., 2017) is an architecture that combats this drawback by help of three key features.

- **Positional encoding** is a technique that allows transformers, which have no recurrency, to retain information about the order of words in a given text sequence. Each word in a sequence is given a word embedding as per usual NLP. Before feeding the word embeddings into a neural network, each word is given a positional vector consisting of sine and cos functions. The result of positional encoding is a vector that is the sum of the word embedding and the words positional encoding vector. This allows the model to learn the importance of the order in which words appear in sequences of text.

- **Attention and self-attention** in the context of transformer models can at a high level be seen as a way to return a specific value given a query, kind of similar to a database lookup. This is due to the nature of the attention mechanism where information about relations between each word in a sequence is stored in a matrix representation of each word. However, this analogy does not exactly mirror the functionality of the attention mechanism, as the value returned from this "lookup" is not one exact value, but rather the weighted similarity values between each word in the query and the "lookup table" convoluted with the values at each location in this "lookup table".

$$attention(q, k, v) = \sum_i similarity(q, k_i) \times v_i$$

  The output is a vector with dimensions corresponding to the dimensions of the input sequence. This scaling comes in the form of a softmax operation on the output values, squashing small values and emphasising values resulting from high similarities.

The attention mechanism in transformer models is what allows them to store and build understanding of semantic relationships between words. This is done in several steps. These steps are applied in encoder decoder fashion with the previously mentioned positionally encoded word embeddings being fed to the encoder part of the transformer. The output of the encoder is then fed to the decoder. The decoder consists of more attention layers, functionally very similar to the attention layer present in the encoder layer. The output from these is then finally fed to a linear classification layer to produce the output of the transformer model. The output corresponds to a probability distribution across words that could follow from the given decoder input sequence. As long as a special token is not predicted (e.g. end of sequence token), the output from the decoder is fed into the decoder itself again and appended to the decoder inputs.

2.2 BERT

The BERT model (Vaswani et al., 2017) is a state of the art transformer model introduced by researchers at Google Research. The model exploits transformer models' capability of parallellism and attention mechanisms to allow for training on datasets drastically larger than before. The parallellism comes from the masked multi-head attention step of the model, where each sequence is passed into the attention head with masks. These masks hide words later in the sequence from the model, forcing it to perform a prediction on the next word to come. For a sentence "I am a boy", the model would be fed the sentence as follows "I [mask] [mask] [mask]", where the next word has to be predicted. The next steps would remove masks one after one until the last prediction is made. The actual sentence is always known, so as to allow for actual training. This makes it possible to train all steps of the masking and prediction process at once!

The name BERT stands for Bidirectional Encoder Representations from Transformers. The bidirectional encoder in BERT refers to the capability of the model to consider both past and future words in the context of the present word by way of the attention mechanism. This enables the model to extract much deeper semantic meanings and relationships than previous models such as LSTMs where context often is lost if sequences become too long.

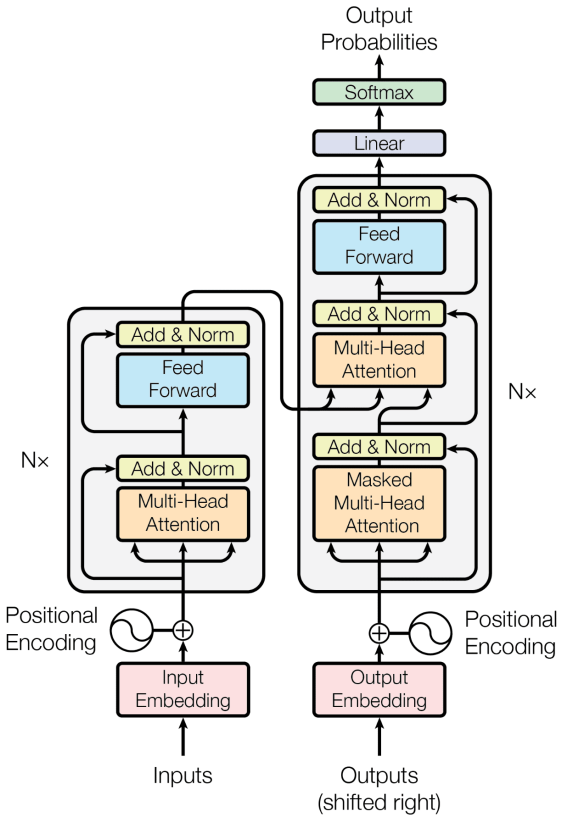


Figure 1: The transformer architecture

2.3 RoBERTa

After inspection of the BERT model, it seems that the proposed base model was severely undertrained (Liu et al.). The RoBERTa model is a continuation of the BERT architecture aimed at correcting some small flaws with the previous state of the art model. RoBERTa differs from BERT in some crucial aspects:

- RoBERTa is trained on a significant amount more data than the original BERT implementation. It is trained using larger batches, more data and longer sequences to further the model's performance on larger inputs.
- The paper removes the next sentence prediction objective, focusing solely on the sequence masking training approach.
- The masking step is changed from static to dynamic, allowing a greater permutation of sequence and mask sets to be seen during training, further generalising the model.

These changes together result in the RoBERTa model matching or surpassing BERT's performance on close to all downstream tasks (Liu et al.).

2.4 Transfer learning and fine-tuning

Transfer learning is a concept that has risen in popularity along with the emergence of large transformer models such as BERT and RoBERTa. Transfer learning is the act of applying a pre-trained model to a different domain or downstream task than what it was originally trained on. Fine-tuning is a form of transfer learning that is widely used in conjunction with large language models. In fact, BERT is

[0]Borrowed from the original representation in the Attention is All You Need paper by Vaswani et al. (2017)

designed to leverage the possibility of fine-tuning it for downstream tasks. The objective of fine-tuning is to present for example BERT with a new domain of data that it has not necessarily been trained on previously and train on this data. This contributes to significant improvements in performance, albeit for that specific domain.

3 Related Work

The act of utilising NLP to analyse user sentiment based on reviews is a relatively well studied area of research. Older studies conducted often consist of applying classical machine learning and statistical approaches to the problem domain. Pang and Lee were one of the first to attempt to give movie reviews a rating based on textual reviews. The paper compares one-vs-all, regression, and metric labeling on the problem.

Qu et al. (2010) introduced a bag of words based approach to combat the problem of sparse n-grams in short texts. They argue the problem arises due to common sentiment mining methods such as utilisation of uni- and bi-grams fail to capture the true sentiment of a document. They perform poorly when context is important. N-grams would be a potential solution, however, these occur far too seldom in short documents to be reliable features to use for sentiment mining. With their approach they managed to predict the rating given from amazon reviews (1-5) with their lowest MSE being 0.627 for music and highest MSE (0.928) for dvds.

A widely used dataset for NLP benchmarking containing movie reviews and their ratings was constructed by Maas et al. (2011). This dataset only contains reviews with ratings 1-4 and 7-10, avoiding, making the dataset highly polar. Later Scaria et al. (2011) implemented a multi-class SVM and a naive bayes model in conjunction with movie metadata to predict user ratings from 1 to 10. They chose to extend the dataset built by Maas et al. (2011) by adding ratings scoring 5 and 6 to balance the dataset. The best scoring model was a simple naive bayes classifier. Introducing metadata into the feature set only served to decrease the accuracy of predictions.

Jong (2011) chose to approach the problem using sentiment analysis as the basis for rating. The Yelp scientific dataset was utilized to train an SVM, multinomial naive bayes classifier and learning vectors. The author refrains from giving any specific rating, and instead classifies each Yelp review as either positive or negative.

In recent years, more focus has been put on applying neural networks to attempt to solve the problem of applying a rating to reviews. Tang et al. (2015) compared common sentiment analysis methodics such as SVM, BOW and VecAvg to feed to an SVM classifier. They also proposed a novel approach to review rating prediction taking the author of the review into account, arguing that one person might place different weight on a word than another individual. Combining vector representations of word weighting for a given user with the existing vector representation of the document gives a representation of the document that is modified based on it's respective author. The representation produced by this combination is then used as features to train a feed-forward neural network. Their model managed to beat current benchmark accuracies on rating reviews from 1-5.

4 Architecture

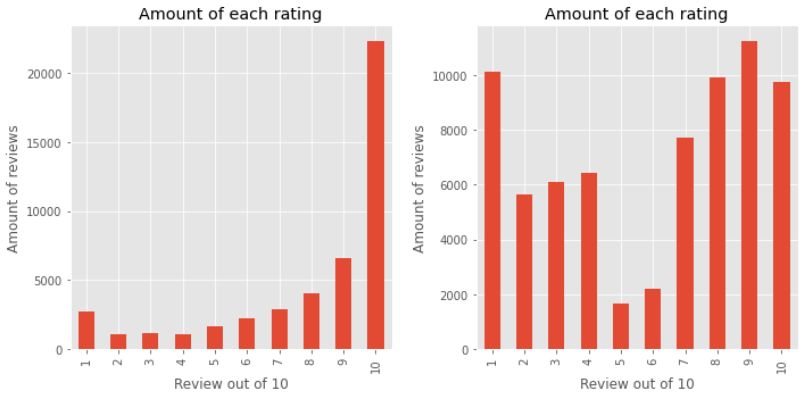
This section presents an overview of methodologies used in the experiments as well as the architectures used to perform the experiments presented.

4.1 Dataset

The original dataset used for this report was the IMDB dataset constructed by Maas et al. (2011) mentioned in section 3. This dataset was designed to be used for binary sentiment analysis and was therefore made higly polar on purpose, leaving out the ratings 5-6. To attempt to combat some of this polarity, we introduced a second dataset[1] fetched from Kaggle. This dataset contains close to 50000 user reviews from 10 different movies. Combining these two datasets yields a dataset that is still highly polar as can

[1] https://www.kaggle.com/datasets/sadmadlad/imdb-user-reviews

be seen from Figure 2. This combined dataset was then finally split into train, test and validation splits for use in the experiments.



(a) Kaggle dataset (b) Extended dataset

Figure 2: The dataset, concisting of approximately 70.000 movie user reviews

4.1.1 Preprocessing

To adhere to customs when working with large language models and transformer models such as BERT, little to no preprocessing was done on the corpus before feeding it to the BBPE tokenizer that was proposed in the original RoBERTa paper. (Liu et al.) This is done deliberately as to not risk losing any semantic meaning the model could pick up on seeing as even stop words or changes in punctuation can lead to changes in the nuance of sentences. Before tokenization, only user tags and spurious HTML tags were sanitized to prevent any contamination of the data. After data sanitization, each review was fed to the BBPE tokenizer with truncation set to RoBERTa's max length = 512.

4.2 Sentiment as features

For the first task the RoBERTa model was experimented with as an intermediary on the way to rating prediction. A RoBERTa model [2] pretrained on twitter data was leveraged to produce sentiment scores delivered as a vector containing scores for negativity, neutrality and positivity. Each score was bounded from -1 to 1. These sentiment scores were then fed forward as features to both an XGBoost regressor and classifier together with the corresponding rating of the review.

4.3 Fine-tuning

The second task experimented with RoBERTa's predictive power alone. Hence a bare-bones model; roberta-base was fetched from huggingface.co to serve as a base model for further fine-tuning. The model was also extended with a final pooling head, pooling all outputs of the model into one output node to use as a regressor. The model was then fed the output of the BBPE tokenizer mentioned in section 4.1.1 and the ratings accompanying each review as training data. This allows us to fine-tune the model to our problem domain - movie reviews and short user written texts.

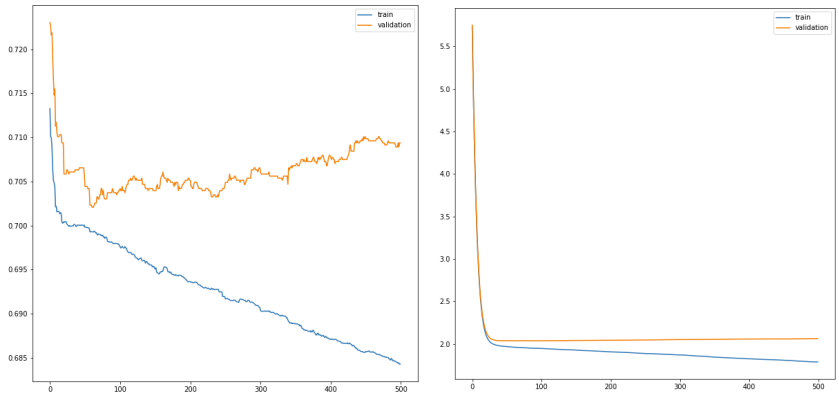
5 Experiments and Results

The experiments conducted in this paper aim to explore the capabilites of the RoBERTa model when applied as a predictive model to an already known dataset, both alone and in conjunction with other models. The experiments were conducted as three separate experiments;

5.1 RoBERTa sentiments with XGBoost classifier and regressor

The XGBoost classifier was trained on the training set presented earlier in the dataset section and validated by the validation set. It was implemented using the XGBoost XGBClassifier class with n estimators

[2] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment



(a) Mean error for XGB classifier (b) RMSE of XGB regressor

= 500, max depth = 2, learning rate = 0.1 and softmax as the objective function. The model was then fitted with the scikit-learn XGBoost function XGBoost.fit(). The model was then fed the sentiment data produced by the RoBERTa model.

The XGBoost regressor was trained on the training set presented earlier in the dataset section and validated by the validation set. It was implemented using the XGBoost XGBRegressor class with n estimators = 500, max depth = 7, learning rate = 0.1 and squared error as the objective function. The model was then fitted with the scikit-learn XGBoost function XGBoost.fit(). The model was then fed the sentiment data produced by the RoBERTa model.

5.2 RoBERTa regressor

The RoBERTa regressor was fine-tuned from the roberta-base model with parameters max length = 512, learning rate = 5e-5, batch size = 32 and epochs = 5. The batch size was chosen to not exceed memory limitations. Regression was performed by following the procedure presented in 4.3. The figure below shows the evaluation metrics gathered for each training epoch.

Epoch	Training Loss	Validation Loss	Mse	Mae	R2	Accuracy
1	2.401300	2.246503	2.246503	1.030272	0.780007	0.347621
2	1.634400	2.385510	2.385510	1.038969	0.766394	0.374941
3	1.203100	1.862939	1.862939	0.885512	0.817568	0.454310
4	0.880500	1.883312	1.883312	0.879030	0.815573	0.465379
5	0.662500	1.836324	1.836325	0.851984	0.820174	0.489637

Figure 4: Training of the RoBERTa regressor

5.3 Model accuracy

Each model's performance was also checked by a simple ratio test to make performance comparison easier.

$$accuracy = \frac{n_{correct}}{n_{total}}$$

Inspired by the work of (Scaria et al., 2011), this acuracy metric was also employed after allowing for a 1 rating error tolerance since very close ratings can hard to distinguish, even for humans. This provides some interesting results.

The accuracy of the regression models was calculated after rounding the output value to the nearest whole rating.

Model	0-tolerance	1-tolerance
XGBCLF	0.27	0.584
XGBREG	0.197	0.510
RoBERTa	0.462	0.822

6 Evaluation and Discussion

6.1 Evaluation of data

Looking at the results of the models, the XGBoost implementations leave a lot to be desired. They both performed worse than previous baselines implemented with Naive Bayes (52.61%) (Scaria et al., 2011). By further examining the dataset that was brought in to introduce ratings of 5 and 6, there seems to be some noisy reviews in the dataset that could bring down the accuracy of the predictions. There are some reviews containing random French that was not accounted for, as well as some reviews that contain a mix of French and gibberish. This is proof that further data sanitisation could and should have been performed to boost performance. Another striking fact about the data used was it's polarity. As mentioned earlier, the original IMDB dataset was on purpose made highly polar to ease binary classification tasks with sentiment analysis. Seeing as this is still one of the if not the most popular dataset for movie reviews, we still viewed it as a well studied and trusted dataset. When trying to make up for the lack of any neutral reviews in the original dataset, the new dataset that was required needed to meet two criteria; containing the ratings 5-6 and review ratings ranging from 1-10. Therefore when such a dataset was found, it was implemented, perhaps with too little hesitation, leading to the contamination of the data that was discussed. However, the addition of the dataset did contribute to balancing the total data to a small degree although reviews given a 5 or 6 star rating were still lacking.

This data skew can be seen affecting the rating predictions of the XGB regressor below.

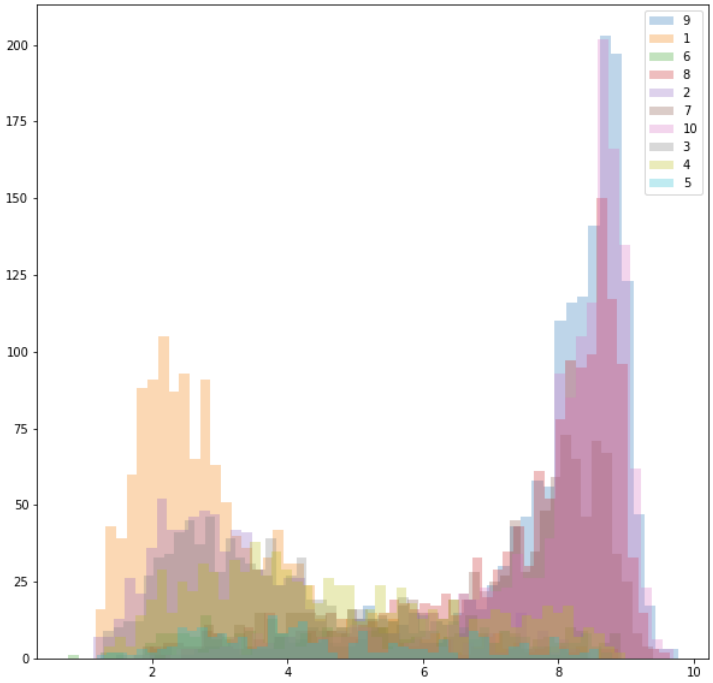


Figure 5: XGBRegressor prediction skew [3]

Note that a skew is only natural since the data has a skewed distribution to begin with and thus naturally more ratings at the extremes. However it should be noted how this affects the training of the model, which seems to result in predictions for other not as well represented ratings, being pulled towards the extremes.

This is not very surprising thinking about when and how ratings are often written - either in a moment of joy or a moment of frustration. This leads to a collection of reviews often being highly polar and biased towards the extremes of the rating scale. A quick trip to a movie review site or an app-store can confirm this.

Another factor when working on unsanitised user data from forums is the human unpredictability. There is nothing stopping a user from writing a review saying that a movie is the worst movie ever, only to proceed to give the movie a perfect 10, just for fun. Data points like this would also be unwanted noise to a machine learning model.

6.2 Feature choice

The features chosen to feed into the XGB based models were chosen due to them being easy to produce from the model of interest - RoBERTa. After closer consideration, looking at the results, especially the training and validation loss of the XGB classifier we can surmise that the feature set was perhaps a little small. The model seems to overfit to the training data, leading to a larger loss when evaluated on the validation set. However, when inspecting the pure sentiment ratings given from the RoBERTa model, both negativity and positivity seemed to have a near linear correlation with the positivity of the rating, while being close to inverse of eachother. Seeing this property made it seem like good features to choose. A possible explanation as to why this still did not work out as well as hoped, is again the fact that similar ratings are hard to distinguish, especially with the small increments resulting from a scale from 1-10. The three features, may not have been discriminative enough to capture the subtle differences of the review ratings.

Some of the motivation behind choosing the approach of utilizing RoBERTa as an intermediary stems from the established fact that large language models are very cmputationally expensive to use for inference due to their complex nature. Therefore it would be of interest to perform such inference operations preferrably once and then feed these values to a less complex, cheaper algorithm such as the proposed XGBoost. These values could also be stored for later use such as was done in this report. The sentiment values produced by RoBERTa were all saved to csv files for use in inference with the XGBoost models. It could be interesting to further investigate with other models and this data at a future date.

6.3 RoBERTa performance and tolerance

The RoBERTa model was expected to outperform the other implementations, which it ultimately did with quite a large margin. It did not manage to surpass the Naive-Bayes implementation of (Scaria et al., 2011), which is interesting when comparing it to the case where a rating deviation of 1 is tolerated. The RoBERTa model surpasses the Naive-Bayes implementation with a small margin (0.8221 vs 0.82). Although the difference in accuracy between our implementation and the classic Naive Bayes is small, one has to take into consideration the minimal amount of preprocessing needed to reach this degree of performance. Referring back to section 4.1.1 only removal of user tags and HTML tags as well as whitespace characters. The ease of use of these models make them incredibly beginner friendly, allowing for experimentation.

The accuracy for 0-tolerance leaves a little to be wanted, but as argued earlier, small increments of sentiment are hard to detect. In daily conversation one might rate something "like a 5 or 6" usually allowing for some leniency. Allowing some leniency for our model also greatly improves the performance, by almost doubling the accuracy. From the evaluation scores produced from the training step of the RoBERTa model one can see that the average error is 0.85. Taking the "tolerance accuracy" into account as well shows that the model manages to generalise and fit quite well to the domain. There seems to be a low variance despite the high polarity of the dataset and noise that both XGBoost implementations were hindered by. This again speaks to the versatility of the new large language models!

6.4 Limitations and disadvantages of large language models

Although the large language models come with a load of advantages, there are some downsides. The first apparent downside is the immense amount of computational power required to train such a model from scratch. The BERT model alone was trained on a corpus of around 3.3 billion words (Muller, 2022). The

RoBERTa model is trained on even more data seeing as it is an extension of the original BERT model. Because of the immense data consumed by these model, they grow to huge sizes to be able to store all the information collected. This leads to some issues, one of them being the environmental impact such large scale models can cause. This is worth mentioning, but is outside the scope of this report.

The second consequence of their large size is the memory needed to run and fine-tune sufficiently large models. This was a hurdle in the experimentation process. There was frequent exhaustion of memory when the project was initially started, leading to the choice of model falling on the roberta-base model instead of the roberta-large. Running the roberta-large model was simply too computationally expensive for both google colab and our computers (not to mention the power bill we could have racked up!). Further experimentation had to be conducted with batch sizes and max length to prevent out of memory errors until we finally arrived at the parameters described earlier.

Maybe the least obvious consequence of these large new language models is the fact that they are so good at almost everything they do. Even so, they are not a be all end all solution, as shown by our XGBoost implementation. One should still take care in selecting correct data to represent a domain as precise as possible while also not thinking that the model will do all the work for you. The authors of this article certainly were thinking a little along those lines in the beginning. Seeing the results of our proposed XGBoost approach quickly dismissed those thoughts though.

7 Future work

As evidenced by the discussion section of the report, there are improvements to be made to further the performance of the architectures used. We will list our reccommended future improvements below:

- A larger and more balanced dataset of movie reviews with accompanying ratings should be created. This will alleviate the problem the authors faced with data being highly skewed towards the two extremes at each end of the rating scale (1 and 10). This could also in theory further improve model performance across all architectures, not just RoBERTa.

- More thorough text processing should be performed on the data before being passed to the models. Even though it is recommended to keep preprocessing of data to a minimum when working with large language models, the loss of semantic meaning by removing extraneous symbols and gibberish would most likely be very minimal. It would highly likely lead to an increase in performance.

- It would be interesting to test the performance of the RoBERTa large model on the same corpus to compare the two models and see how much the difference in data affects them both. Other pre-trained implementations of the RoBERTa architecture should also be explored to see if there are models more relevant to the problem domain than what has been used in this report.

- The correlation between sentiment scores yielded from the RoBERTa model pre-trained on a Twitter corpus seemed to be highly correlated with the actual rating a review would give. Despite this, the results obtained from our experiments were less than promising, suggesting that either these features are not enough to capture the subtle differences from rating to rating or that the authors have made a mistake somewhere along the way. Further experimentation with these sentiment values, and possible combinations made with these and e.g. metadata of movies could be an interesting further step to investigate.

- The authours also wished to compare the RoBERTa model to previous models implemented on rating scales that contain only 5 or 4 stars. We believe that the model would perform much better here seeing as the margin between each step in the scale is twice as big as for the 10 star rating scale.

8 Conclusion

The results displayed by the RoBERTa model despite the low requirements needed to implement it surprised both authors. The amount of preprocessing needed to fine-tune an acceptable model is truly

remarkable. The possibility of adding a custom head on top of these fine-tuned language models further extend their reach in the NLP domain and possibly also other disciplines. Despite it's advantages, one can still make mistakes when implementing it, such as our implementation utilizing a combination between XGBoost and the RoBERTa arcihtecture. More is not always better!

The key takeaways from this project is that large language models are huge tools that time and time again sow their versatility in various areas of research. The baseline RoBERTa model outperformed previous classical approaches as well as our own implementation of XGBoost with little to no preprocessing steps required. One should however still take the steps used to train such a model into careful consideration, especially if one is trying to do so from scratch or on large amounts of data. The computation time for these models is very slow and heavy, making training environmentally taxing as well as expensive.

References

Jason Jong. Predicting rating with sentiment analysis. *Retrieved Maret*, 2:2017, 2011.

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. URL `http://arxiv.org/abs/1907.11692`.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. 2011.

Britney Muller. Bert 101 tate of the art nlp model explained, 2022. URL `https://huggingface.co/blog/bert-101`.

Bo Pang and Lillian Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, pages 115–124. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL `http://portal.acm.org/citation.cfm?doid=1219840.1219855`.

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. The bag-of-opinions method for review rating prediction from sparse text patterns. 2010.

Aju Thalappillil Scaria, Rose Marie Philip, and Sagar V Mehta. Predicting star ratings of movie review comments. page 5, 2011.

Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. User modeling with neural network for review rating prediction. 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.