Trip Advisor Reviews Sentiment Analysis using RNN and Transformers

Salvatore Mastrangelo
s.mastrangelo4@studenti.unipi.it

Abstract

This study investigates document-level sentiment classification of hotel reviews by contrasting a lightweight Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) with a pretrained Transformer (RoBERTa-base) fine-tuned through Low-Rank Adaptation (LoRA). The analysis is conducted on 18'273 English hotel reviews trimmed to 256 tokens and split 72.3 / 12.7 / 15 % for training, validation and testing, respectively, in a strongly imbalanced five-star rating distribution.

The LSTM comprises an embedding layer, a single LSTM layer with mean-pooled hidden states, and a soft-max classifier, achieving convergence below 240 s on a consumer GPU. A RoBERTa based model was fine-tuned with FP16 precision and a LoRA adapter, thet reduced trainable parameters to 1.9 % of the backbone, enabling single-GPU training.

On the native dataset, RoBERTa outperformed the LSTM with 67 % accuracy / weighted F1 = 0.67 in the five-class task and 88 % / 0.88 in the ternary task, compared with 58 % / 0.58 and 84 % / 0.83 for the LSTM, respectively. Transformer models shows great capability in classifying minority ratings (1–3 stars).

These findings underscore three insights: (i) pre-trained attention models confer a clear advantage on imbalanced, ambiguous opinion data; (ii) conventional RNNs remain competitive under balanced conditions and deliver rapid training; (iii) LoRA enables cost-effective fine-tuning of large encoders. The work thus provides a reproducible benchmark and practical guidelines for choosing architectures under resource and data-distribution constraints.

1 Introduction

1.1 Nature of sentiment analysis

Sentiment Analysis is a subfield of Natural Language Processing (NLP) that focuses on identifying and classifying subjective information in text data. In particular, it allows to cathegorize the sentiment expressed by pieces of text, such as reviews, comments, or social media posts, into predefined classes, that can be whether:

- favorable or unfavorable opinions towards a product, service, or entity,
- expressed emotions such as joy, anger, sadness, or fear,
- opinions about specific aspects or features of a product or service (aspect-based sentiment).

Sentiment analysis is widely used in various domains, including marketing, customer service, and social media, to gain insights into public opinion or provide control over allowed content, like in the case of hate speech detection.

HLT final project report (2025).

1.2 Approaches to Sentiment Analysis

The methods used for sentiment analysis can be broadly categorized into:

- **Rule-based systems**, which use lexicons and pattern-based approaches, typically hand-crafted, that require large efforts to develop and mantain [Gupta et al., 2024].
- **Feature engineering and Machine Learning**, that is based on extraction of features as bag-of-words, n-grams, or word embeddings, followed by machine learning classifiers [Gupta et al., 2024].

In particular, machine learning models have gained in recent years a lot of attention, as recurrent models like Long Short-Term Memory (LSTM) networks have shown effective capabilities in capturing relations among distant words in a sentence [Staudemeyer and Morris, 2019], and Transformers like BERT [Devlin et al., 2019] and its variants have shown state-of-the-art performance in many NLP tasks.

1.3 The project

In this project, the focus is posed on implementation and evaluation two models, one based on LSTM-RNN and one based on RoBERTa [Liu et al., 2019], a variant of BERT [Devlin et al., 2019], that are able to classify the sentiment of hotel reviews, and return a score from 1 to 5 stars, as well as a ternary classification of the sentiment expressed, that can be positive, negative or neutral.

2 Background

The background of this project can be divided into two main parts: the models used and the related works in the field of sentiment analysis.

2.1 Models

The first part is about the models used in this project. Long Short-Term Memory networks have been used for a long time in Natural Language Processing for its capabilities in capturing long-term dependencies among data, breakthrough that allowed the overcoming of the problem of vanishing gradients that affects RNNs [Hochreiter and Schmidhuber, 1997]. Such models, unlike standard RNN, implement a memory cell that can block or allow the flow of information from the past to the future using three gates: input, forget and output gates. Such gates allowed the development of more complex sequence processing, making LSTM effective in tasks like machine translation, classification and text generation

Some simpler models, like the Gated Recurrent Unit (GRU), proposed by Cho et al. [2014], have been deployed as an alternative to LSTM to reduce the number of parameters, but still keeping the same concept of memory cell. Both LSTM and GRU have been widely used and show strenghts in different tasks, with LSTM being more suitable for higher-complexity sequencies and GRU for simpler ones [Cahuantzi et al., 2023], preferred in environments where computational efficiency is critical.

Transformers, on the other hand, are a more recent architecture [Vaswani et al., 2023] that has revolutionized the field of sequence processing and thus Natural Language Processing. with the introduction of self-attention mechanisms they made possible to capture relations among words even at great distance. Such innovation allowed the development of models with more complex context understanding capabilites. Furthermore, since Attention is computed separately for each pair of tokens, it allowes massive parallelization too, advantage tackled by the use of GPUs.

Nowadays, Transformers have set the state of the art standard in many NLP tasks. In particular, BERT [Devlin et al., 2019] and its variants, like RoBERTa [Liu et al., 2019], are pre-trained models that are born from such Transformer architecture used as encoders for text data. Trained on large corpora of text, can be fine-tuned on specific tasks to achieve brilliant results with very few variations in the

architecture, effectively performing transfer learning very efficiently and with a small amount of data. [Torrey and Shavlik, 2010]

2.2 Related Works

The second part is about the related works in the field of sentiment analysis. Many works have been done in the past, using different approaches and models, but the most recent ones focus on the use of Transformers and pre-trained models like BERT and its variants due to the much improved performance with respect to RNN. Nevertheless, it should be noted that until the introduction of Transformers they were the State of The Art models to perform tasks like Machine translation [Cho et al., 2014] and next sentence prediction [Ganai and Khursheed, 2019]

In particular, Gupta et al. [2024] provide a comprehensive study on sentiment analysis, comparing different approaches and models, including rule-based systems, machine learning classifiers, and deep learning models like LSTM and Transformers. They highlight the strengths and weaknesses of each approach, showing that while rule-based systems can be effective for specific tasks, machine learning and deep learning models are more suitable for general-purpose sentiment analysis.

An interesting work is done by Wen et al. [2023], who provides a similar study on sentiment analysis using another variant of BERT, called ERNIE [Zhang et al., 2019], that uses knowledge graphs to improve the understanding of language structure and semantics. They show that ERNIE outperforms BERT in various knowledge-intensive tasks, while retaining comparable performance in other NLP tasks.

In summary, the field of sentiment analysis has seen a shift from traditional rule-based systems and machine learning classifiers to deep learning models. In particular, the use of pre-trained models has become a standard practice, on which this work is based too.

3 Methods

In this section the methodologies implemented are illustrated, starting from the dataset used, the preprocessing steps and the models implemented.

3.1 Dataset

The dataset used in this project is a collection of Trip Advisor reviews, taken from kaggle[1]. Such dataset contains 20491 english reviews of hotels, labelled with a score from 1 to 5 stars (labels 0 to 4). To grant a more balanced benckmark, all the reviews with a length greater than 256 tokens according to the RoBERTa tokenizer were dropped, resulting in a final dataset of 18273 reviews. Such choice was made to make a compromise between the maximum depth of the LSTM Networks in order to avoid vanishing gradients, and the maximum length of the *roberta_base* tokenizer and model, which is 512 tokens. Then, some preprocessing steps were done, such as removing HTML tags, converting the reviews to lowercase and removing stop words according to the NLTK default list. The final distribution of labels is the following:

[1] `https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews`

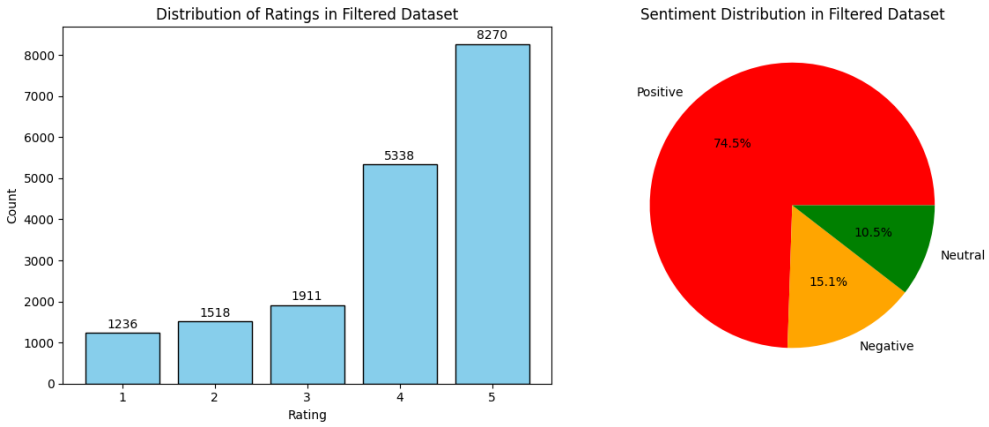

Figure 1: Label distribution

The dataset is quite unbalanced, with a vast majority of positive reviews (4-5 stars).

The reviews were then split into:

- training set: 72.3%
- validation set: 12.7%
- test set: 15%

Based on the used model, the reviews were converted to tokens either using The RoBERTa tokenizer, either using the NLTK tokenizer, which was used to create the vocabulary for the RNN model. Both tokenized datasets were then padded to a maximum length of 256 tokens.

3.2 Models

3.2.1 LSTM-RNN

The first model implemented is a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) implemented in Torch, made of:

- an embedding layer, which converts the input tokens to a dense vector representation;
- a LSTM layer, which processes the sequence of embeddings and captures the sequence temporal dependencies;
- a fully connected layer, which maps the output of the LSTM to the final output classes.
- a softmax activation function

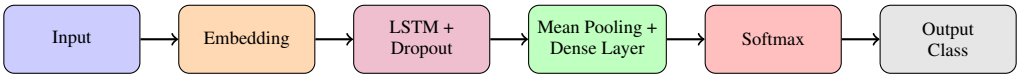

Figure 2: LSTM-based sentiment classifier.

In particular, to evaluate the performance of the model, only the higher probability class is selected. The model is trained using the ADAM optimizer [Kingma and Ba, 2017], based on the cross-entropy loss function. Some form of dropout is applied to the LSTM layer to avoid overfitting. the model is trained for a maximum of 100 epochs, with early stopping kicking in almost always before the epochs upper limit, based on the f1 score on the validation set.

The tokenizer used in this model, as said before, is the NLTK tokenizer, with a vocabulary size of 10002, corresponding to the 10000 most frequent words in the training set, plus the `<PAD>` and `<OOV>` tokens, corresponding to the padding token and the out-of-vocabulary token.

3.2.2 RoBERTa Transformer

The second model implemented is a `RoBERTaForSequenceClassification` transformer from the HuggingFace Transformers library, initialized at the *roberta-base* checkpoint. Furthermore, a classification MLP head is present, that takes information from the first output of the RoBERTa transformer, which corresponds to the `[CLS]` token, and maps it to the final output classes using softmax activation.

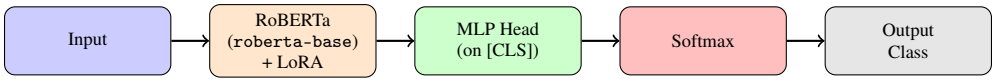

Figure 3: RoBERTa-based sentiment classifier.

The model, even though is pretrained, is fine-tuned on the datset using a rank-32 LoRA (Low-Rank Adaptation) PEFT with dropout. Like with the LSTM-RNN model the training has place using the cross-entropy loss function, with the optimization performed using the ADAMW optimizer [Loshchilov and Hutter, 2019] It should be noted that the model is trained using fp16 mixed precision, allowing to speed up the training and reducing the memory usage, allowing the training of the model on a single GPU with 6GB of memory.

Furthermore, a Learning Rate Warmup [Kalra and Barkeshli, 2024] of 10% of the training steps is applied. The tokenizer used in this model is the *roberta-base* tokenizer.

4 Experimental analysis

4.1 Task

As mentioned in the introduction, the task consists in classifying sentences based on the sentiment expressed by them. The dataset is composed of reviews of hotels from the TripAdvisor website, labelled with a score from 1 to 5 stars.

After the training of the models, the desired output is a class, in the range from 1 to 5 stars in the first place, and in the range [0, 2] in the second place, where 0 is negative, 1 is neutral and 2 is positive. The second classification task is expected to perform better, since it reduces the amount of possible classes and the ambiguity of intermediate scores.

Furthermore, such task is performed with the purpoose of not only gather metrics about the generalization capabilities of the architectures implemented, but also to compare the performance of the two.

4.2 Experimental settings

All the experiments were run on a machine with a NVIDIA RTX 3060 mobile GPU, 6 GB of VRAM. All the validations were performed using 12.7% of the total dataset.

4.2.1 LSTM-RNN

In order to determine the best hyperparameters for the LSTM-RNN model, a grid search was performed in the space of the hyperparameters, using the f1 score on validation set as metric to choose the best configuration. The hyperparameters involved were the learning rate, the dropout rate, the number of hidden units and the weight decay.
(**bold** values are the best ones found):

Hyperparameter	Values		
Dropout	0	0.1	**0.3**
Learning rate	0.001	5e-4	**1e-4**
hidden units	64	**128**	256
weight decay	0	**1e-5**	1e-4

It is interesting to note that the f1 score and validation loss of the model in the first experiments remained unchanged for the first $25 \approx 40$ epochs, then the model started to learn. Since such behavior was present without regularization too, and was probably due to the fact that the model is initialized with random weights and the gates of the LSTM were still learning when to open and close, thus leading to a stable high validation loss, immediate consequence of gradient vanishing.

As a solution, in the second iteration the model was modified to make a mean pooling of the hidden states of the LSTM, which are only then passed to the classification head. Such change allowed the model to learn from the beginning of the training, leading to better performances.

It should be noted that the patience of early stopping was set to 10 epochs and triggers on f1 score. In the first implementation a minimum of 40 epochs to be trained was set to avoid the early stopping to stop the training before starting to learn. In the last version such threshold was set to 0, since the vanishing problem wasn't present anymore with the same relevance.

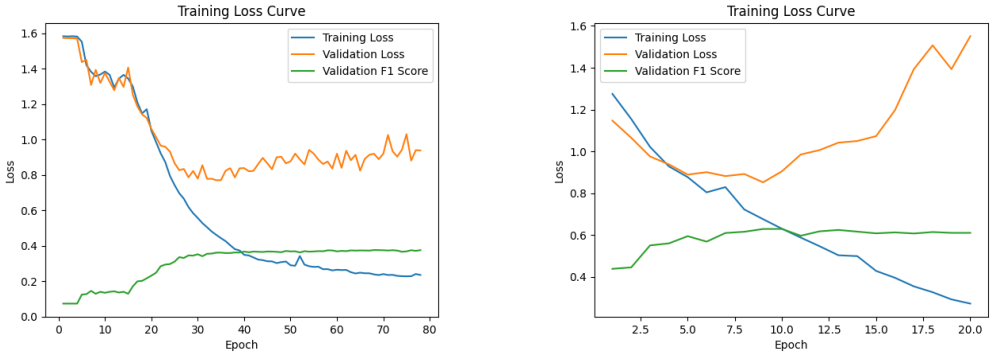

Figure 4: Training curves in first (left) and second (right) implementation of the LSTM-RNN model.

In both cases, the model was trained using the ADAM optimizer, with a batch size of 16, exploiting the functions implementend in the torch library to calculare loss and gradients. Overall, the training of the model took approximately 90 seconds.

4.2.2 RoBERTa Transformer

The second model implemented is a `RoBERTaForSequenceClassification` transformer, already pretrained, thus the fine-tuning is performed on the dataset adjusting only the classification head and a rank-32 LoRA applied to query and value matrices of the attention layers. Dropout was set to 0.1, effectively introducing a form of control over model complexity. Such design choices allowed to substantially reduce the number of parameters to train:

$$\text{Trainable parameters:} \quad 2,368,522/126,423,562 \approx 1.87\%$$

Allowing to train the model on a single GPU and perform model selection too. The grid search was performed taking into account the loRA dropout, the loRA alpha, the learning rate and the weight decay. Here too the validation f1 score was used as metric to choose the best hyperparameters. In

particular, the hyperparameters chosen for the model were:
(**bold** values are the best ones found):

Hyperparameter	**Values**		
LoRA Dropout	0	**0.1**	0.3
LoRA Alpha	16	**32**	64
Learning rate	1e-5	2e-5	**3e-5**
Weight decay	0	**1e-5**	1e-4

The model was trained using the ADAMW optimizer, with a batch size of 16. Raising such value even more might have led to better results, but the GPU VRAM was not enough to support it. Thanks to the HuggingFace `trainer` structure it was possible to enable mixed precision training with FP16, thus reducing substantially the memory required to train the model effectively.

Differently from the LSTM-RNN model, the RoBERTa model required much less iterations to learn, stopping training after only 5 epochs. Nevertheless, each epoch required a longer time to be perfomed, of about 6 minutes each, for a total of 28 minutes.

4.3 Results

In this section the results of the two models are provided, based on the three main metrics used to evaluate the performance of classification:

- **Precision**: the ratio of true specific class predictions over the total number of same specific class predictions made by the model.
- **Recall**: the ratio of true specific class predictions over the total number of same specific class instances in the dataset.
- **F1 score**: the harmonic mean of precision and recall, which is a good measure of the model's accuracy when dealing with imbalanced datasets, calculated as:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

4.3.1 LSTM-RNN

The confusion matrices of the LSTM-RNN model is shown in the following figure:

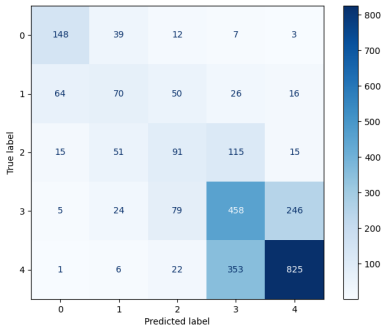

Figure 5: 5-class confusion matrix, LSTM.

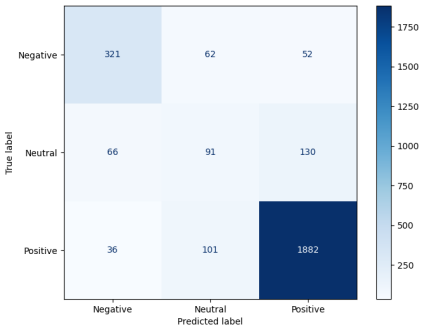

Figure 6: 3-class confusion matrix, LSTM.

from such matrices it is possible to derive the following metrics:

Class	Precision	Recall	F1
1 star	0.64	0.71	0.67
2 stars	0.37	0.31	0.34
3 stars	0.36	0.32	0.34
4 stars	0.48	0.56	0.52
5 stars	0.75	0.68	0.71
weighted avg.	0.59	0.58	0.58

Class	Precision	Recall	F1
Negative	0.76	0.74	0.75
Neutral	0.36	0.32	0.34
Positive	0.91	0.93	0.92
weighted avg.	0.83	0.84	0.83

It appears that the model is able to classify the reviews quite accurately with 3 classes, but it struggles with the 5-class classification as the actual discrimination between 4 and 5 stars is not very concise, with a lot of 4 stars reviews classified as 5 stars and vice versa. Furthermore, the model appears quite confused for all the minority classes, which is to be expected given the small proportion of such classes in the dataset.

The overall accuracy of the model is 58% for the 5-class classification, that still is a good result given the small dataset and the well known difficulty of RNN to learn over quite long sequences. The accuracy for the 3-class classification is 84%, and in contrast to the 33% of a random classifier, it holds quite well, given the minimal amount of training required.

4.3.2 RoBERTa Transformer

The confusion matrices of the RoBERTa model is shown in the following figure:

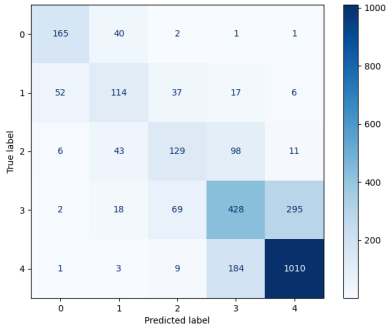

Figure 7: 5-class confusion matrix, RoBERTa.

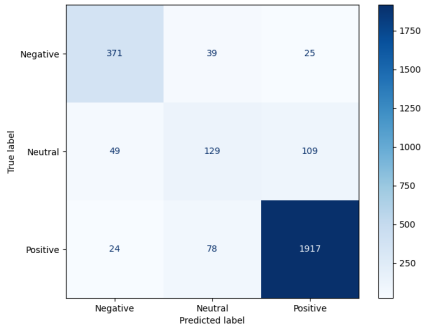

Figure 8: 3-class confusion matrix, RoBERTa.

that bring to the following metrics:

Class	Precision	Recall	F1
1 star	0.73	0.79	0.76
2 stars	0.52	0.50	0.51
3 stars	0.52	0.45	0.48
4 stars	0.59	0.53	0.56
5 stars	0.76	0.84	0.80
weighted avg.	0.66	0.67	0.67

Class	Precision	Recall	F1
Negative	0.84	0.85	0.84
Neutral	0.52	0.45	0.48
Positive	0.93	0.95	0.94
weighted avg.	0.88	0.88	0.88

The results of the RoBERTa model appear to be better than the LSTM-RNN model, and the level of generalization the model achieved is quite good, with an acceptable classification capability for minority classes too. The addition of attention mechanisms and the pretraining yield better long

distance information dependecies.

Such additional features bring accuracies of 67% and 88% for the 5-class and 3-class, with improvements of 15.5% and 4.8% respectively. The great improvement in 5-star classification is a demonstrarion of the superior context understanding of the transformer architecture, which is able to capture finer differences in the reviews.

4.4 Comparison with Literature

As term of comparison, the results of the presented model will be compared with the results of the paper *"Sentiment Analysis of Hotel Reviews With LSTM And ELECTRA"* by Husein et al. [2023], which presents a 3-class sentiment analysis on scraped hotel reviews. The models used in the paper are an LSTM and a discriminator encoder called ELECTRA.

To make a fair comparison, some further training instances will be done to the presentented models with an oversampled and an undersampled version of the dataset used in this project, since the Husein et al. [2023]'s one is undersampled too to equalize the amount of entries for each class. For this purpose, the oversampled dataset will have all the classes as numerous as the 5-star reviews, that amount to 8270, and the undersampled done will have all classes as popolated as the 1-star class, that has dimension 1236.

Model	**Accuracy**	**Precision**	**Recall**	**F1**
LSTM - US	0.73	0.74	0.73	0.73
LSTM	0.84	0.83	0.84	0.83
LSTM - OS	0.82	0.84	0.82	0.83
RoBERTa - US	0.80	0.79	0.80	0.79
RoBERTa	0.88	0.88	0.88	0.88
RoBERTa - OS	0.86	0.86	0.86	0.86
ELECTRA Husein et al. [2023]	0.47	0.33-0.56	0.38-0.62	0.35-0.59
LSTM Husein et al. [2023]	0.30	0.38-0.40	0.29-0.50	0.28-33

The results shows that on the native dataset transformers work better than RNNs, capable of capturing finer differences in the reviews, but within over-sampling the LSTM-RNN results to be comparable to the RoBERTa based model. Such behavior hints that LSTM based architectures are better suited for learning from balanced datasets. On the other side, the RoBERTa implementation suffers from ripetitive data, which leads to a worse performance, probably due to overfitting.

Furthermore, the undersampled versions show that, even if the trasformer based implementation is able to generalize better than ELECTRA [Husein et al., 2023], it definitely suffers from the lack of data, leading to a worse performance. Nevertheless, both models are able to outperform the LSTM based and ELECTRA model introduced by Husein et al. [2023].

5 Discussion

The results obtained from the experiments conducted on the Trip Advisor reviews dataset provide valuable insights into the performance of current state-of-the-art models for sentiment analysis. The comparison between the LSTM-RNN based models and the transformers based ones, particularly RoBERTa, highlights how different architectures can yield varying results depending on the nature of the dataset and the specific applications.

LSTM models, despite requiring more epochs to converge, demostrated quite impactul results, achieving levels of performance in understanding the sentiment expressed in the reviews that can be compared to Transformers architectures. On the other hand, Transformers, demonstrated that can be

quite elastic in learning knowledge from very unbalanced datasets, particularly important in case of big datasets where oversampling techniques are unfeasible.

In all the experiments, the hardest class to identify is the neutral one, corresponding to the 3-star reviews. This limitation is likely due to the inherent ambiguity in the language used in these reviews, which often includes mixed terminology and sentiments. Both models seem to struggle to "grasp indecision" in the language, leading to lower accuracy and F1 scores.

5.1 Limitations

The main limitation about the choice of which model to use resides in the computational resources available, particularly at training time. Being trained on a consumer-grade GPU, Transformers models require much more time to train, limited by the memory available on the GPU, besides the bigger amount of trainable parameters. Due to these limitations, it was not possible to try larger models, such as larger versions of RoBERTa, that could have potentially yielded better results

Another limitation is the size of the dataset, which is relatively small with respect to the complexity of the models used. It is easy to lead a model to overfitting, especially in the case of Transformers, which are known to require large amount of data to generalize well. Furthermore, the specificity of the dataset might lead to non generalizable results, thus limiting the expected performance in other topics.

6 Conclusions

In the experiments conducted, two models were successfully implemented and trained to classify hotel reviews into star ratings and ternary sentiment classes. The approach differed between the two models, and brought to two quite satisfactory results.

As expected, the task was quite easier for high sentiment reviews, since the majority of the reviews belonged to that category, but the results showed that both models performed far better than random guessing. It would be interesting to explore the possible further expressiveness of the models considering the confidence scores of each prediction, since it would indicate far better understanding of the review to fall into classes closer to the actual sentiment, particularly for the 5-star classes.

The results also showed that the RNN model was able to achieve good performance, particularly in terms of precision and F1 score, but also in amount of time needed to train the model, with merely 4 minutes in the worse case. The use of mean pooling also brought to a good increment in training stability and time, also learning better representations of the corpora. finally, the LSTM-RNN showed accuracy rates of 58% and 84% which resemble state of the art results, aligned with the results of other works in the literature.

The RoBERTa based model, on the other hand, were reliable in all the experiments, keeping almost all metrics above 80% in 3-class classification. It emerged once more how models based on Transformers suffer from lack of data, but are able to still grasp context also in minority classes, such as the low score reviews. On the other side, too much training on the same data easily leads to overfitting, reducing performance and increasing the training time. Nevertheless, the amount of computing power required to train and infer with such models demonstrated how in some simpler cases is better to opt for lighter models.

Future work could focus on use of larger datasets or implement forms of data augmentation to increment training quality. Also use some different form of readout layers might a be a topic of future investigation, maybe exploiting attention mechanisms on LSTM hidden states before readout to improve the context capabilities of the model.

References

Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A comparison of lstm and gru networks for learning symbolic sequences, 2023. ISSN 2367-3389. URL http://dx.doi.org/10.1007/978-3-031-37963-5_53.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL https://arxiv.org/abs/1409.1259.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Aejaz Farooq Ganai and Farida Khursheed. Predicting next word using rnn and lstm cells: Stastical language modeling. In *2019 fifth international conference on image information processing (ICIIP)*, pages 469–474. IEEE, 2019.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. Comprehensive study on sentiment analysis: From rule-based to modern llm based system, 2024. URL https://arxiv.org/abs/2409.09989.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory, 1997.

Amir Mahmud Husein, Nicholas Livando, Andika Andika, William Chandra, and Gary Phan. Sentiment analysis of hotel reviews on tripadvisor with lstm and electra. *Sinkron: jurnal dan penelitian teknik informatika*, 7(2):733–740, 2023.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements, 2024. URL https://arxiv.org/abs/2406.09405.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

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. URL https://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019. URL https://arxiv.org/abs/1909.09586.

Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Yu Wen, Yezhang Liang, and Xinhua Zhu. Sentiment analysis of hotel online reviews using the bert model and ernie model—data from china. *Plos one*, 18(3):e0275382, 2023.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities, 2019. URL https://arxiv.org/abs/1905.07129.