Trip Advisor Reviews Sentiment Analysis using RNN and Transformers

Salvatore Mastrangelo
s.mastrangelo4@studenti.unipi.it

Abstract

This report presents a comprehensive analysis of sentiment classification on Trip Advisor reviews using Recurrent Neural Networks (RNNs) and Transformers. The study explores the effectiveness of these models in understanding and predicting sentiment from textual data. A detailed methodology, experimental setup, and results are provided, highlighting the strengths and weaknesses of each approach. In particular, given a rather small dataset of Trip Advisor reviews, the aim is to classify the sentiment of each review into both stars classes (1 to 5 stars) and ternary classes (positive, negative, neutral). RNN models are implemented using Long Short-Term Memory (LSTM), while Transformers are implemented using the BERT architecture, in particular the RoBERTa model. The results show that while Transformers require much less epochs to converge, RNNs are able to achieve better performance in terms of accuracy and F1 score on such small dataset. On the other hand, Transformers show much better capabilities in terms of ambiguous reviews, achieving a better F1 score and accuracy in the middle part of the scale (3-4 stars).

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.

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].

HLT final project report (2025).

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.

2.1 models

The first part is about the models used in this project. LSTM-RNNs have been used for a long time in Natural Language Processing for its capabilities in capturing long-term dependencies among data, overcoming the problem of vanishing gradients that affects RNNs [Hochreiter and Schmidhuber, 1997]. Such models 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.

Some simpler models, like the Gated Recurrent Unit (GRU) [Cho et al., 2014], have been proposed 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]

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, that uses self-attention mechanisms to capture relations among words, allowing parallelization too.

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.

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.

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, 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. [Zhang et al., 2019]

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:

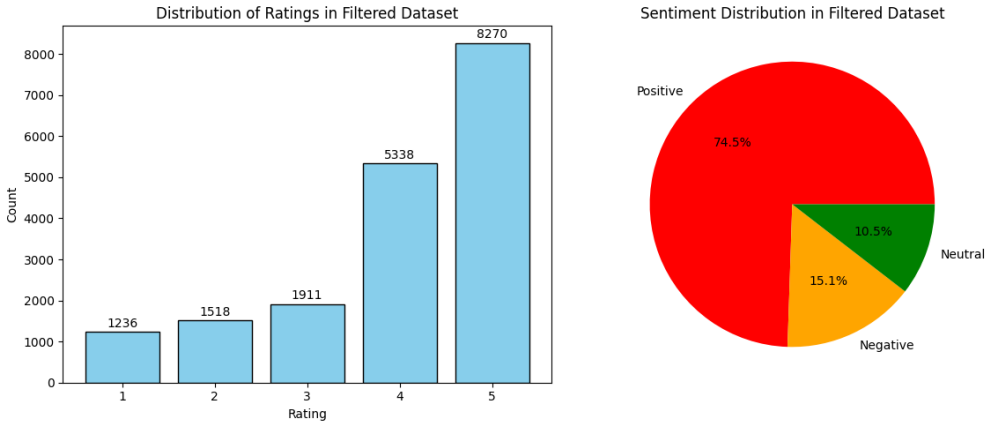

Figure 1: Label distribution

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;

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

- a fully connected layer, which maps the output of the LSTM to the final output classes.
- a softmax activation function

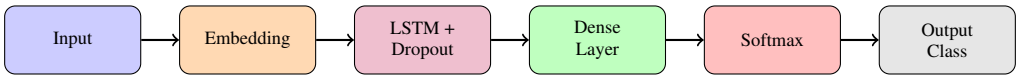

Figure 2: LSTM-based sentiment classifier.

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 accuracy on the validation set. Ultimately, the hyperparameters used for the model are:
(**bold** values are the chosen ones)

Hyperparameter	**Values**		
Dropout	0	0.1	**0.3**
Learning rate	0.001	5e-4	**3e-4**
Patience	0	**5**	10

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.

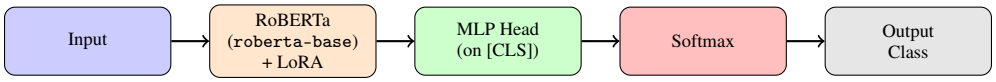

Figure 3: RoBERTa-based sentiment classifier.

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. The final hyperparameters used for the model are:
(**bold** values are the chosen ones)

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

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. 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.

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

Describe all the relevant aspects of the experimental setup used in your experiment (e.g., how you performed model selection for fine-tuning of hyper-parameters, all details regarding the learning / fine-tuning of your model, etc.)

4.3 Results

Provide results (figures and tables with mounerical results should go here). Provide insights and comments on the achieved results (also comparatively with literature). Possible ablation studies go here.

5 Discussion

Overall discussion on the results, relevant insights. Frame your considerations in the current landscape of relevant research.

6 Conclusions

Draw conclusions and possibly delineate future works / possible improvements.

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.

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.

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.

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.

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in appendix.