

Applying Transformers or RNNs to Text and Sequence Data

AML Assignment 4



November 26, 2023

Jyothsna Peddireddigari

Contents

[Introduction 2](#_Toc151918622)

[Observations and Analysis 2](#_Toc151918623)

[Conclusion: 3](#_Toc151918624)

# Introduction

Text and sequence data present unique computational challenges and opportunities due to the information they contain in natural language or ordered sequences. This type of data is commonly encountered when analysing written content, predicting user behaviour, and identifying patterns in time series data. The key to analyzing text and sequence data is to uncover the underlying structures, patterns, and dependencies within the datasets, which enables the development of applications such as sentiment analysis, language modeling, and time series forecasting. For this purpose, the IMDB dataset is used. The Internet Movie Database (IMDB) dataset is a frequently utilized compilation of movie reviews designed for natural language processing (NLP) and machine learning tasks. This dataset comprises a range of movie reviews, including positive and negative sentiments, making it a favoured option for sentiment analysis projects.

# Observations and Analysis

This analysis includes exploring the effective application of Transformers to text and sequence data. We have modified the dataset several times to improve our model's performance. First, we truncated all reviews after the first 150 words and considered only the top 10,000 words. Second, we restricted the training dataset to 100 samples and validated the model on a large set of 10,000 samples. Especially, compared the effectiveness of using an embedding layer versus a pre-trained word embedding and how changing the number of training samples impacts the model's performance and to determine at what point the embedding layer gives the better performance.

**Table: Different models built with their Test accuracy values**

|  |  |
| --- | --- |
| **Models** | **Test Accuracy (**%**)** |
| Transformer encoder for Text Classification | 81.3 |
| Combining the Transformer encoder with positional embedding | 81.9 |
| A basic sequential model with training samples of 100. | 80.6 |
| A model that uses an Embedding layer trained from scratch using training samples of 100. | 81.5 |
| Using an Embedding layer with masking enabled. | 81.6 |
| A model with an embedding layer built from scratch with increased training samples of 200. | 80.1 |
| A model with an embedding layer built from scratch with increased training samples 1000. | 83.6 |
| A model with an embedding layer built from scratch with increased training samples 2000. | 83.8 |
| A model with an embedding layer built from scratch with increased training samples 5000. | 83.4 |
| Model using a pre-trained word embedding with 1000 training samples. | 83.2 |
| Model using a pre-trained word embedding with 5000 training samples. | 83.9 |

In this analysis of the above models, each model was evaluated based on the Accuracy obtained values on their test set. The Transformer architecture model core is a transformer encoder layer that uses self-attention mechanisms to capture contextual relationships in sequences. A global max pooling layer is applied to reduce variable-length sequences to a fixed-size representation. Dropout is used for regularization, and the model concludes with a dense layer that uses a sigmoid activation for binary classification.

A text classification model using a Transformer encoder has achieved an accuracy of 81.3%. This indicates that the attention and self-attention mechanisms in the Transformer architecture are successful in capturing contextual information in text, enabling the model to classify text with high accuracy.

When the Transformer encoder is combined with positional embedding, the model's accuracy is improved to 83.9% which is better than the text classification model using a Transformer encoder. This means that providing the positional information enhances the model's understanding of the input text's sequential nature. This emphasizes the significance of positional encoding for tasks that involve ordered sequences.

The basic sequential model has been trained on a small dataset of 100 samples and achieved an accuracy of 80.6%. This indicates that even with limited data, the model is capable of learning significant patterns and features for text classification.

The accuracy of a model with an Embedding layer trained from scratch using 100 training samples is 81.5 This accuracy is higher than the sequential model, which suggests that training a model with an embedding layer from scratch improves the accuracy, may also require a larger dataset to capture intricate language representations effectively.

When the masking feature is enabled in the Embedding layer, the model shows an accuracy of 81.6% which is almost closer to the model with an embedding layer from scratch. This indicates that the model benefits from considering sequence padding and excluding the padded values during training. By doing so, the model can avoid learning false patterns from padding tokens, thus contributing to the overall performance of the model.

After training a model with an embedding layer built from scratch and increasing the training sample size to 200, The accuracy obtained was 80.1%, which suggests that a slight increase in training samples does not contribute much to the model's improved performance.

Further increases in training samples have a better performance than above all models. After increasing the training samples to 1000 to the model built with an embedding layer from scratch, the model's accuracy improved to 83.6%, and when training samples were increased to 2000 and 5000 the accuracy obtained are as 83.8% and 83.4% respectively. which suggests that the significant increase in the training samples from 100 to 1000 or more brought a significant improvement in the performance of the model. It is clear that size of the data plays a very important role, and that the model's performance can be enhanced with more data.

The model that uses pre-trained word embedding with training samples of 1000 and 5000 have obtained an accuracy of 83.2% and 83.9%. Here it can be observed that the pretrained model with increased training samples performs the best with an accuracy of 83.9% which is higher than all other models compared. This highlights the efficacy of utilizing pre-existing language representations. It indicates that pre-trained embeddings capture much more information, providing a strong foundation for text classification tasks.

# Conclusion:

In conclusion, Transformer encoders with text classification itself on the IMDB dataset, or combined with positional embedding, achieved a better accuracy, which indicates that transformer architecture is good at capturing contextual information.

The model built with the embedding layer from scratch using 2000 training samples showed the best performance, with an accuracy of 83.8 %, compared to all other models with smaller training samples of 100, 200, and 1000 sample sizes. This indicates the importance of the size of the data; limited data decreases the model's performance.

However, leveraging pre-trained word embeddings with increased training samples yields the highest accuracy of 83.9%. It stands out among all the embedding layer models built from scratch, emphasizing the value of pre-existing language representations.

The following is a visual representation that shows the accuracy values for each model using embedding layers.