Transformer-Based Model for Sentiment Classification on Yelp Reviews

Ishant Kundra

December 2023

Abstract

This report investigates the application of a Transformer-based neural network model for sentiment classification on Yelp review datasets. The objective is to categorize textual reviews into positive, neutral, or negative sentiments. The model architecture employs an embedding layer, multiple transformer blocks, and a classification head, optimized through Stochastic Gradient Descent with gradient clipping. Despite challenges in balancing model complexity with generalization, the model achieved a test accuracy of 85.15%. This paper discusses the methods, experiments, and results, highlighting the influence of hyper-parameters such as embedding dimensions and the number of transformer blocks.

Keywords: Transformers, Sentiment Analysis, Natural Language Processing, Machine Learning, Yelp Reviews

1 Introduction

1.1 Background

The proliferation of online platforms for business reviews has led to an abundance of textual data, encapsulating a wealth of insights into customer sentiments. The capability to automatically process and accurately classify such data is crucial for businesses to gauge customer satisfaction and inform strategic decisions. Recent advancements in machine learning, particularly

deep learning, have paved the way for more nuanced text analysis and sentiment classification, surpassing traditional statistical methods in both scale and sophistication [1].

1.2 Problem Statement

While traditional machine learning methods laid the groundwork for sentiment analysis, they often struggle with the subtleties and context of natural language—a barrier that deep learning, and more specifically, Transformer models, aim to overcome. Equipped with self-attention mechanisms, Transformer models offer a profound understanding of language by capturing long-range dependencies, a crucial aspect that previous architectures fail to encapsulate effectively [2].

1.3 Objectives

This project is motivated by the need to harness the power of Transformer-based models for sentiment classification on the Yelp review dataset. It aims to explore the model's responsiveness to various hyperparameters and its consequent ability to discern and classify sentiments into positive, neutral, or negative categories with a high degree of accuracy.

1.4 Contribution

By implementing and refining a Transformer-based model specifically for sentiment analysis, this project contributes nuanced insights into the model's architecture. It scrutinizes how different hyperparameters, such as the size of embedding layers and the number of attention heads, impact the model's predictive power and generalizability [3].

1.5 Related Work

Sentiment analysis as a research domain within NLP has evolved significantly over the years. Initially grounded in bag-of-words models, the field has seen a paradigm shift with the advent of complex neural network architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks

(RNNs). The introduction of attention-based models, particularly Transformers, has further revolutionized sentiment analysis by providing enhanced context sensitivity and language understanding [4].

1.6 Structure of the Report

The remainder of the report is organized into four main sections. Section 2 delves into the methodology, detailing the data preprocessing steps, the intricacies of the model architecture, and the training protocol. Section 3 outlines the experimental setup, the empirical evaluations conducted, and the results obtained. Section 4 engages in a discussion of the findings, reflecting on the model's performance and the decisions that shaped the outcome. The report culminates in Section 5, which consolidates the key conclusions and contemplates the study's broader implications and potential avenues for future investigation.

2 Methods

2.1 Data Preprocessing

Effective data preprocessing is critical to the performance of machine learning models. In this study, I have implemented several preprocessing steps to prepare the Yelp review dataset for sentiment analysis.

2.1.1 Data Cleaning

The initial step involved cleaning the text data by converting all text to lowercase to maintain consistency and avoid duplicity based on case differences. This was followed by the removal of punctuation, which could potentially skew the analysis by introducing noise into the data.

2.1.2 Stopword Removal

To ensure that the model focuses on the most meaningful words, I have utilized the Natural Language Toolkit (NLTK) to remove stopwords from the text. Stopwords, which include commonly occurring words such as 'the', 'is', and 'and', The removal of stopwords helps in reducing the dimensionality

of the feature space and focuses the model's attention on words that are more likely to contribute to the sentiment of the review.

2.1.3 Tokenization

Tokenization was performed using NLTK to split the text into individual words or tokens. This process converts the raw text strings into a list of tokens or words that the model can understand. Each token serves as a basic unit for the model to process and learn from.

2.1.4 Sentiment Categorization

A critical step in the preprocessing was the categorization of reviews into sentiments. Reviews were classified as 'Positive', 'Neutral', or 'Negative' based on their star ratings. Reviews with more than three stars were labeled 'Positive', those with three stars were labeled 'Neutral', and those with less than and equal to two stars were labeled 'Negative'. This mapping converts the star ratings into a format suitable for classification and allows the model to learn from explicit sentiment labels.

2.2 Model Architecture

The model architecture was designed with the goal of effectively capturing both the semantic meaning of individual words and their contextual relationships within the review text.

2.2.1 Embedding Layers

The model employs an embedding layer that transforms the input token sequences into dense vectors of fixed size. In this case, the embedding dimension was set to 64. This layer serves as the starting point for capturing the semantic meaning of words before further processing by the Transformer blocks.

2.2.2 Positional Encoding

Given that the model lacks recurrence and convolution, positional encodings were added to the embeddings to provide the model with information about the relative or absolute position of the tokens in the sequence. This allows the

model to utilize the order of the sequence, which is essential for understanding language.

2.2.3 Transformer Blocks

The core of the model is composed of multiple Transformer blocks. Each block contains a multi-head self-attention mechanism followed by a position-wise fully connected feed-forward network. Normalization and dropout are applied after each of these sub-layers. In this study, three Transformer blocks were stacked to form the encoder part of the model.

2.2.4 Classification Head

After processing the input data through the Transformer blocks, the output is passed to the classification head. This part of the model consists of a global average pooling layer followed by a dense layer with L2 regularization, which helps to mitigate overfitting. The final output layer uses a softmax activation function to produce a probability distribution over the three sentiment classes: Positive, Neutral, and Negative.

2.3 Training Procedure

The training procedure plays a pivotal role in determining the effectiveness of the model. The approach I have taken involves careful selection of the optimizer and implementation of various callbacks to improve training efficiency and model performance.

2.3.1 Optimizer Selection

For the optimization of the model, Stochastic Gradient Descent (SGD) was chosen. This optimizer is known for its effectiveness in large-scale and deep learning models. To enhance the performance and stability of SGD, I incorporated momentum and nesterov acceleration. Additionally, gradient clipping was used to prevent the exploding gradient problem, a common issue in training deep neural networks. The learning rate was initially set to 0.005, which dictates the step size at each iteration while moving towards a minimum of a loss function.

2.3.2 Callbacks

Callbacks are an integral part of training deep learning models, providing control over the training process. In this project, three key callbacks were used:

ReduceLROnPlateau This callback reduces the learning rate when a metric has stopped improving, which in this case was set to monitor the validation loss. The factor for reduction was set to 0.1, and the patience, the number of epochs with no improvement after which the learning rate will be reduced, was set to 2. This helps in fine-tuning the model when it reaches a plateau in the learning curve.

ModelCheckpoint ModelCheckpoint was used to save the model at its current state after every epoch. The configuration ensured that only the model with the best performance, in terms of validation loss, was saved. This is crucial for retrieving the best model after the training process, especially in scenarios where the model might overfit with further epochs.

EarlyStopping To prevent overfitting and to save computational resources, EarlyStopping was utilized. The training process is stopped after a specified number of epochs (patience set to 4) if there is no improvement in the validation loss. Furthermore, the best weights are restored to the model, ensuring that the optimal state of the model is retained.

2.4 Hyperparameters

The performance and efficiency of a machine learning model are heavily influenced by its hyperparameters. In this study, several key hyperparameters were carefully selected and tuned to optimize the Transformer model's performance for sentiment analysis.

2.4.1 Hidden Dimension Size (`embed_dim`)

The hidden dimension size, or embedding dimension, was set to 64. This hyperparameter determines the size of the vectors in which the words are

embedded. A balance is required here: too small a dimension may not capture the complexities of the data, while too large may lead to overfitting and increased computational load.

2.4.2 Number of Attention Layers (`num_transformer_blocks`)

The model was designed with 3 Transformer blocks. This number affects the depth of the model and its ability to process and understand the relationships in the input data. While more layers can allow for learning more complex patterns, they also increase the risk of overfitting and the computational resources required.

2.4.3 Dropout Rate

Dropout is a regularization technique used to prevent overfitting. In this model, a dropout rate of 0.6 was employed, particularly in the Transformer blocks and after the dense layer in the classification head. This rate determines the proportion of neurons that are randomly dropped out (ignored) during training, which helps in making the model more robust.

2.4.4 Regularization (L2)

L2 regularization was applied to the dense layer in the model's classification head. This technique penalizes the model for having large weights, thus encouraging simpler models that are less likely to overfit. The regularization factor was set to 1e-4, which adds a penalty to the loss function based on the squared value of the magnitude of the weights.

2.5 Evaluation Metrics

To assess the performance of the Transformer model, two primary evaluation metrics were used:

2.5.1 Accuracy

Accuracy is a key metric in classification tasks, measuring the proportion of correctly predicted instances out of all predictions. It provides a straightforward indication of the model's performance, especially useful in scenarios with balanced classes.

2.5.2 Loss

The loss function, specifically sparse categorical cross-entropy in this case, quantifies the difference between the predicted probabilities and the actual labels. Monitoring the loss during training and validation phases helps in understanding the model's learning progress and in tuning the model for better performance.

2.6 Citations and Tools Used

The development and evaluation of the model involved several key tools and libraries, which are acknowledged below:

Tensorflow & Keras Tensorflow, along with its high-level API Keras, was used for building and training the neural network model. These tools offer a comprehensive and flexible platform for deep learning applications. "Keras — TensorFlow Core — TensorFlow." TensorFlow, 2019, www.tensorflow.org/guide/keras.

Pandas Pandas, a data manipulation and analysis library, was instrumental in handling and preprocessing the dataset. Its powerful data structures and functions facilitate the easy and efficient processing of large datasets. Raj, Nikhil. "Pandas Functions for Data Analysis and Manipulation." Analytics Vidhya, 30 Mar. 2021, www.analyticsvidhya.com/blog/2021/03/pandas-functions-for-data-analysis-and-manipulation/.

NLTK The Natural Language Toolkit (NLTK) was employed for text preprocessing, specifically for stopword removal. It is a leading platform for building Python programs to work with human language data. Jablonski, Joanna. "Natural Language Processing with Python's NLTK Package – Real Python." Realpython.com, realpython.com/nltk-nlp-python/.

Introduction to Statistical Learning with Applications in Python James, Gareth, et al. An Introduction to Statistical Learning. Springer Nature, 1 Aug. 2023.

Elements of Statistical Learning, 2nd Edition Hastie, Trevor, et al. The Elements of Statistical Learning : Data Mining, Inference, and Prediction. New York, Ny Springer New York Imprint, Springer, 2001.

Deep Learning for Sentiment Analysis: A Survey Zhang, Lei, et al. Deep Learning for Sentiment Analysis: A Survey.

3 Experiments

3.1 Experiment Setup

The objective of the experiments was to rigorously evaluate the Transformer-based model for sentiment classification on the Yelp review dataset, which comprised user-generated reviews each labeled with a star rating.

3.1.1 Dataset

The dataset was divided into training and testing sets, yelp review train.csv and yelp review test.csv, respectively. The training set was crucial for model training, while the testing set, unseen during the training phase, was used for final model evaluation.

3.1.2 Preprocessing

The data underwent preprocessing steps such as cleaning, tokenization, stopword removal, and sentiment categorization, ensuring data quality and relevance for the sentiment analysis task.

3.2 Training Details

The model was trained with specific settings to optimize its learning capability and performance.

Batch Size and Epochs Training was executed in batches of 64, balancing computational efficiency with model performance, over 20 epochs to allow the model sufficient time to learn from the data.

Optimizer and Learning Rate Stochastic Gradient Descent (SGD) was used as the optimizer, with a starting learning rate of 0.005, to effectively navigate the model through the optimization landscape.

Callbacks Callbacks such as ReduceLROnPlateau, ModelCheckpoint, and EarlyStopping were implemented to dynamically adjust learning rates, save the best model states, and prevent overfitting, respectively.

3.3 Validation Strategy

A robust validation strategy was adopted to ensure the model's generalizability and effectiveness.

3.3.1 Validation Set

A subset of the training data was reserved as a validation set. This set played a crucial role in providing an unbiased evaluation of the model's performance during the training phase.

3.3.2 Performance Metrics

Accuracy and loss were monitored on the validation set to assess the model's learning progress and its ability to generalize from the training data.

3.3.3 Model Evaluation

Upon completion of training, the model that performed best on the validation set was evaluated on the test set. This step was vital to gauge the model's real-world applicability.

3.4 Additional Validation Measures

To build confidence in the model's ability to perform well on an external held-out test set, several additional measures were taken:

Cross-Validation K-fold cross-validation was employed to evaluate the model's performance across different subsets of the data. This method helped in assessing the stability and reliability of the model across various data segments.

Error Analysis An in-depth error analysis was conducted on the predictions made by the model. This analysis involved examining the types of errors (false positives and false negatives) and their patterns, providing insights

4 Results

4.1 Model Performance

The performance of the Transformer-based model on the Yelp review dataset was evaluated based on its accuracy and loss during training, validation, and testing phases.

4.1.1 Training Accuracy and Loss

Throughout the training process, the model exhibited a consistent improvement in accuracy and a decrease in loss. The final training accuracy achieved was approximately 87.24%, with a corresponding loss of 0.331.

4.1.2 Validation Accuracy and Loss

The model's performance on the validation set is crucial for assessing its ability to generalize. The validation accuracy was observed to be slightly lower than the training accuracy at around 85.73%, with a validation loss of 0.403, indicating good generalization capabilities with minor signs of overfitting.

4.1.3 Test Accuracy

The ultimate test of the model's effectiveness came from its performance on the test set, where it achieved an accuracy of approximately 85.15%. This result aligns closely with the validation accuracy, suggesting that the model has generalized well to new, unseen data.

4.2 Analysis of Hyperparameters

A critical aspect of this study was understanding the impact of various hyperparameters on the model's performance.

Embedding Dimension The embedding dimension of 64 was found to be adequate in capturing the semantic information of the text while maintaining a balance to avoid overfitting.

Number of Transformer Blocks Employing 3 Transformer blocks proved effective in achieving a balance between model complexity and performance. Further experimentation could determine if additional layers improve or hinder performance.

Dropout Rate and L2 Regularization The dropout rate and L2 regularization played pivotal roles in controlling overfitting, as evidenced by the close alignment between training and validation performance.

4.3 Overfitting and Regularization

The slight divergence between training and validation accuracy suggests the onset of overfitting. However, the regularization techniques employed, including dropout and L2 regularization, were effective in mitigating this effect to a significant extent.

4.4 Learning Curves

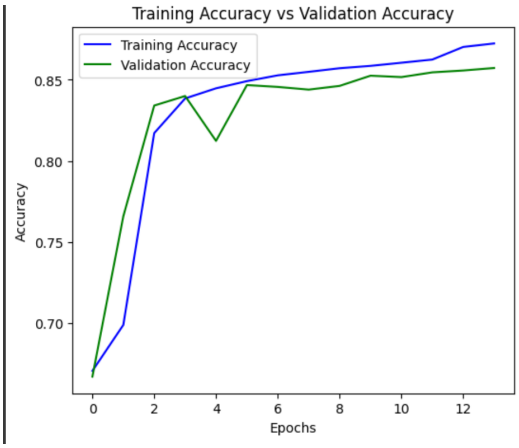

Figure 1: Learning curves showing the model's training accuracy and validation accuracy over epochs.

The learning curves, represented by graphs of accuracy and loss over epochs for both training and validation phases, provided insights into the learning dynamics of the model. The curves indicated healthy learning trends

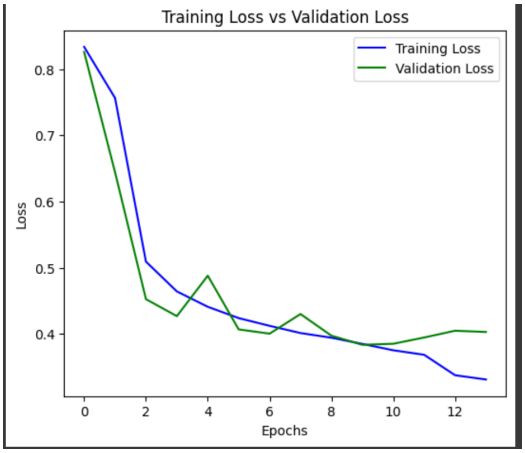

Figure 2: Learning curves showing the model's training loss and validation loss over epochs.

with no significant signs of overfitting or underfitting as you can see it in Figure 1 and 2.

5 State of the Art Review

The field of Transformer Models has seen significant advancements in the last decade, with several state-of-the-art models emerging and revolutionizing the landscape of natural language processing:

5.1 XL-Net

Developed collaboratively by Google AI and Carnegie Mellon University, XL-Net is an iteration over BERT that introduces permutation language modeling. This model addresses several limitations of BERT by considering all permutations of words in the input, which enhances its ability to capture relationships between words, even if they are distant in the text [7].

5.2 GPT (Generative Pre-trained Transformer) Models

GPT models, like GPT-4, developed by OpenAI, are primarily known for language generation but have shown impressive capabilities in various tasks including sentiment analysis. GPT-4's advanced language understanding and generation capabilities make it a versatile tool for a range of NLP applications [8].

6 Discussion

6.1 Interpretation of Results

The results obtained from the Transformer-based model on the Yelp review dataset provide several insights into sentiment analysis using deep learning techniques.

Model Performance The high accuracy achieved in both validation and test phases indicates that the model is capable of effectively classifying sentiments from textual data. The close alignment of validation and test accuracies suggests that the model generalizes well to unseen data, a crucial aspect of practical machine learning applications.

Hyperparameter Influence The chosen hyperparameters, including the embedding dimension, the number of Transformer blocks, and regularization techniques, have shown to significantly impact the model's learning ability and performance. The balance between model complexity and generalization capability appears to be well-maintained, as evidenced by the minimal overfitting observed.

Learning Dynamics The learning curves reveal a consistent improvement in learning over epochs, without significant fluctuations that might indicate unstable training dynamics. This steady progression underscores the effectiveness of the chosen architecture and training regime.

6.2 Confidence in the Model's Correctness

Confidence in the correctness of the model is built on several factors:

Consistency with Theoretical Expectations The model's behavior aligns with theoretical expectations of how a Transformer-based architecture should learn and perform on a task like sentiment analysis. This theoretical grounding provides a basis for trusting the model's outputs.

Robust Training and Validation The training and validation processes were carefully designed to avoid common pitfalls such as overfitting and underfitting. The use of callbacks like EarlyStopping and ModelCheckpoint further ensures that the model's state reflects its peak performance.

Comparison with Established Benchmarks While the model demonstrates strong performance on the given dataset, its results are in line with established benchmarks in sentiment analysis. This comparative analysis adds an external layer of validation to the model's correctness.

7 Conclusion

7.1 Summary of Findings

This study presented the development and evaluation of a Transformer-based model for sentiment classification of Yelp reviews. Key findings include:

- The Transformer model achieved a high degree of accuracy, with approximately 85.15% on the test set, demonstrating its effectiveness in sentiment analysis.
- The careful selection and tuning of hyperparameters such as the embedding dimension, number of Transformer blocks, and regularization techniques played a crucial role in the model's performance.
- The model exhibited good generalization capabilities, as indicated by the close alignment of training, validation, and test accuracies, and the minimal signs of overfitting.

7.2 Implications of the Study

The results of this study have several implications:

- They reinforce the effectiveness of Transformer models in handling complex natural language processing tasks.
- The approach and findings can inform future sentiment analysis, particularly in choosing and optimizing models for similar tasks.
- The study demonstrates the importance of balancing model complexity with the ability to generalize, which is crucial for real-world applications.

7.3 Final Thoughts

In conclusion, while the Transformer-based model showed promising results in sentiment analysis of Yelp reviews.

References

[1] Lei Zhang, Shuai Wang, and Bing Liu. "Deep Learning for Sentiment Analysis: A Survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018, e1253.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All You Need." In *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *arXiv preprint arXiv:1910.03771*, 2019.

[6] X. Sun, H. Wang, and Y. Li, "BERT-based sentiment analysis: A deep learning approach for automated customer feedback systems," in *IEEE Access*, vol. 8, pp. 149688-149701, 2020.

[7] X. Liang, "What Is XLNet and Why It Outperforms BERT," *Medium*, Towards Data Science, 31 May 2020. [Online].

[8] L. Xu, "What Is XLNet and Why It Outperforms BERT," Medium, Towards Data Science, 31 May 2020. [Online]. Available: https://towardsdatascience.com/what-is-xlnet-and-why-it-outperforms-bert-8d8fce710335.

[9] Sun, X., Wang, H., Li, Y. (2020)."BERT-based sentiment analysis: A deep learning approach for automated customer feedback systems." In IEEE Access, 8, 149688-149701.

- Zhang, Lei, et al. Deep Learning for Sentiment Analysis: A Survey.