Sentiment Analysis

Mohina Ahmadi

December 12, 2023

1 Introduction

1.1 Background

Sentiment analysis, the automated process of determining the sentiment expressed in textual data, holds significant importance in understanding customer feedback, market trends, and public opinion. With the exponential growth of online reviews and user-generated content, the need for accurate and efficient sentiment analysis has become crucial for businesses and researchers.

1.2 Motivation

The motivation behind this project lies in the necessity for advanced natural language processing (NLP) models capable of capturing complex sentiments in customer reviews. Traditional sentiment analysis, also known as rule-based or lexicon-based sentiment analysis, refers to the use of predefined rules, linguistic patterns, or lexicons to determine the sentiment expressed in a piece of text. Unlike modern machine learning approaches that learn patterns from data, traditional sentiment analysis relies on handcrafted rules and resources. They often struggle with long-range dependencies and contextual understanding. The transformer model, with its self-attention mechanism, offers a promising solution to these challenges.

1.3 Objective

The primary objective of this project is the implementation and evaluation of transformer models in sentiment analysis tasks using the Yelp review dataset. The dataset comprises around 174,000 reviews with stars, and a subset of this dataset will be utilized for the experiments. The goal is to implement a transformer model for sentiment analysis based on the textual content of the reviews and their associated star ratings.

2 Related Work and State of the Art

The research paper "Attention is all you need" by Ashish Vaswani et al[1] introduces the Transformer model, challenging traditional recurrence and convolution in sequence-to-sequence tasks. It emphasizes the limitations of traditional models for handling long-range dependencies, advocating for the self-attention mechanism. The model, comprising encoder and decoder layers, demonstrates superior performance in machine translation compared to RNN and CNN models, showcasing its ability to capture linguistic structures through attention weight visualizations. The paper concludes that attention mechanisms alone are sufficient for sequence-to-sequence tasks, eliminating the need for recurrence and convolution, and highlights the Transformer model's parallelizability and broad applicability.

The paper by Ilgun et al[2] explores sentiment analysis on social media using transformers, particularly BERT, for enhanced performance on lengthy texts. It addresses challenges associated with unstructured social media data and compares transformer-based models, particularly BERT, against traditional models like Logistic Regression and Support Vector Machine. Despite the need for substantial computational resources, the paper underscores the transformative impact of transformers in sentiment analysis, emphasizing their effectiveness over traditional machine learning models.

The paper by Fangwu Wu et al[3] paper focuses on sentiment analysis of online product reviews, introducing SenBERT-CNN, a model combining BERT and CNN for sentiment information extraction. The study highlights the model's efficacy in influencing consumer decisions, validated on JD.com's mobile phone reviews dataset. Despite longer training times, SenBERT-CNN outperforms various models in accuracy, precision, recall, and f-beta score, emphasizing its significance in sentiment analysis for online product reviews.

The paper by Eman Saeed Alamoudi et al[4] research explores sentiment analysis and aspect extraction from Yelp restaurant reviews using diverse models, proposing an unsupervised approach based on semantic similarity. The ALBERT model achieves a maximum accuracy of 98.30%, and the proposed aspect extraction method reaches 83.04%. The paper categorizes sentiment classification and extraction approaches and identifies a research gap in sentiment analysis on Yelp data, providing comparisons between different models and proposing a novel aspect extraction approach, with results highlighting the effectiveness of ALBERT, BERT, and CNN models in sentiment classification on Yelp datasets.

Sentiment analysis today is shaped by advanced machine learning, especially transformer-based models like BERT, RoBERTa, ALBERT, DistilBERT, ELECTRA, ERNIE, XLNet, ULMFiT, and TinyBERT. These models have introduced bidirectional context, robust contextual embeddings, efficiency optimizations, knowledge integration, and versatility in pre-training. Transfer learning is a key trend, enabling generalization across domains with limited labeled data. The landscape also includes multimodal sentiment analysis with images and videos, aspect-based sentiment analysis, and a growing focus on explainability and interpretability. Domain adaptation, continuous learning, and refined evaluation metrics contribute to ongoing efforts for improved accuracy and applicability. The choice of a model depends on task requirements and available resources, reflecting the field's commitment to advancing contextual understanding, model efficiency, and adaptability in sentiment analysis.

3 Methodology

3.1 Data Preprocessing

To prepare the data for sentiment analysis, several preprocessing steps were followed. This included removing punctuation, converting text to lowercase to ensure consistency, and eliminating common English stopwords to focus on meaningful context. Additionally, numerical ratings were converted to sentiment labels. The input sequences are padded or truncated to a fixed length using the `pad_sequences` function from the Keras preprocessing module. This ensures uniformity in sequence lengths, which is necessary for input to the neural network. Tokenization is performed using a tokenizer, converting words in the text into integer indices. This step is crucial for feeding textual data into a neural network, as it transforms variable-length sequences into fixed-length numerical representations.

3.2 Transformer Model Configuration

3.2.1 Concepts of Transformers:

Built upon Vaswani et al.'s seminal paper 'Attention is All You Need' [1], this transformer model implementation derives inspiration and adopts crucial concepts to effectively handle sentiment analysis tasks.

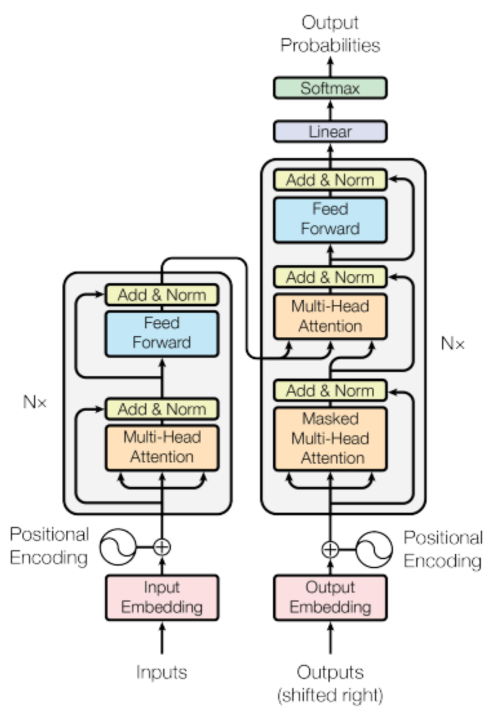


Figure 1: Transformer Model Architecture

i. **Self-Attention Mechanism:**
The self-attention mechanism allows the model to weigh the importance of different

words in a sequence differently when making predictions for a particular word. This attention mechanism enables the model to consider the context of each word in relation to all other words in the sequence, capturing long-range dependencies.

ii. **Multi-Head Attention:**
Multi-head attention is an extension of the self-attention mechanism. It involves using multiple sets of attention weights (heads) in parallel. Each head focuses on different aspects of the input, and their results are concatenated and linearly transformed. This allows the model to learn different relationships within the data simultaneously.

iii. **Positional Encoding:**
Transformers do not inherently understand the order of the input sequence since they operate on the data in parallel. To address this, positional encoding is added to the input embeddings to provide information about the position of each token in the sequence. This enables the model to consider the sequential nature of the data.

iv. **Feedforward Neural Network:**
After the attention mechanism, a feedforward neural network is applied independently to each position in the sequence. This layer introduces non-linearities and further captures complex patterns in the data.

v. **Layer Normalization and Residual Connections:**
Each sub-layer (e.g., self-attention, feedforward) in the transformer block is followed by layer normalization and a residual connection. This helps in stabilizing training and enables the efficient training of deep networks.

vi. **Encoder-Decoder Architecture:**
Transformers are often used in sequence-to-sequence tasks where an input sequence is transformed into an output sequence. The model consists of an encoder to process the input sequence and a decoder to generate the output sequence. The attention mechanism facilitates capturing relevant information from the input during the decoding process.

3.2.2 Module Implementation

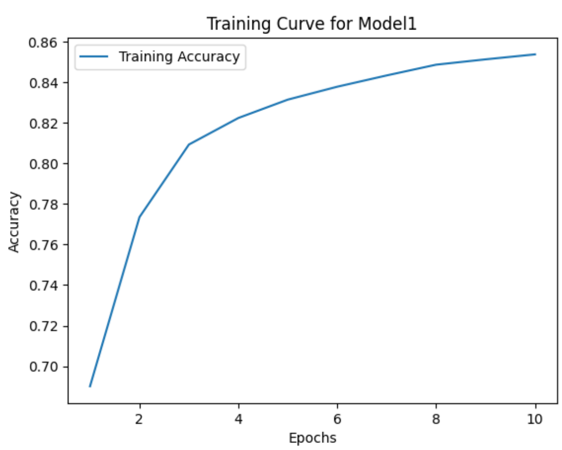
i. **Multi-Head Attention:** This module implements the multi-head attention mechanism, a key component of the Transformer architecture. It takes input queries, keys, and values and computes attention scores using linear transformations. The attention scores are used to calculate attention weights, which are then applied to the input values.The module uses a specified number of heads (`n_heads`) and the dimension of the model (`d_model`) to perform the attention computation.

ii. **Positional Encoding:** This module provides positional encoding for the input sequences to give the model information about the order of tokens. It generates positional encodings based on the position of each token in the input sequence. The positional encoding is added to the input embeddings of the model.

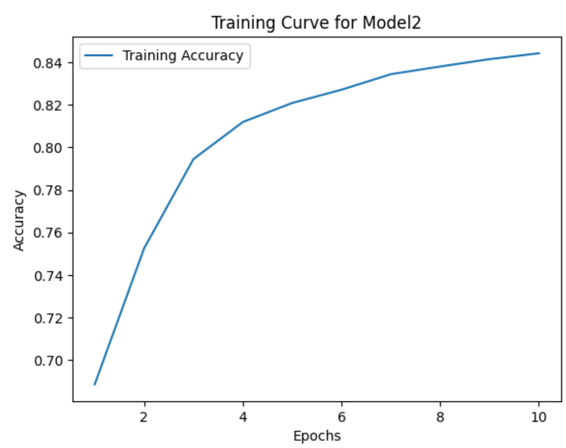
iii. **Transformer Model:** This module combines the multi-head attention mechanism and positional encoding to create a transformer model. It uses an embedding layer to convert input tokens into dense vectors, and adds positional encoding to the embeddings. The encoder is employed to process the input sequence using self-attention and feedforward layers. The final output is obtained by averaging over the sequence length and passing it through a linear layer.

3.3 Training Procedure Overview

The training procedure consists of several key steps. First, the required libraries, including PyTorch for neural network construction, DataLoader for batch handling, and TensorDataset for dataset creation, are imported. Following data preprocessing and splitting into training and validation sets, the data is converted into PyTorch tensors. These tensors, containing input sequences and corresponding labels, are then wrapped into datasets using TensorDataset and transformed into batch iterators using DataLoader. The transformer model is instantiated, and the optimization process begins. Stochastic Gradient Descent is used as the optimizer, and CrossEntropyLoss serves as the loss function. The training loop iterates for 10 epochs. Within each epoch, the training data is processed in batches. The optimizer is zeroed, and predictions are obtained through a forward pass, calculating the loss. Backward propagation updates the model's parameters. Fine-tuning was performed by modifying key hyperparameters to enhance the training process. Specifically, adjustments were made to parameters including the number of dimensions (`d_model`), the number of attention heads (`n_heads`), and the number of transformer layers (`num_layers`). Two distinct models were executed with varying hyperparameter configurations: `model1` (`d_model=256, n_heads=8, num_layers=4`) and `model2` (`d_model=128, n_heads=4, num_layers=3`). These modifications aimed to optimize the model's performance.



(a) Training Curve for Model 1



(b) Training Curve for Model 2

Figure 2: Training Curves for Model 1 and Model 2

3.4 Validation and Testing

The model's performance was evaluated on a validation set, demonstrating its ability to generalize to unseen data. The best model with the highest validation accuracy is saved. Subsequently, the best model was tested on a dedicated test set, yielding accuracy and other relevant metrics for sentiment classification. Between model1 and model2, model1 demonstrated a slightly superior performance as shown in Table 1.

Model	Training Accuracy	Validation Accuracy	Test Accuracy
Model 1	85.36%	84.92%	85.17%
Model 2	84.43%	84.54%	84.71%

Table 1: Accuracy Scores for Model 1 and Model 2

3.5 Impacts of Hyperparameters on the Transformer

3.5.1 Hidden Dimension (`d_model`):

Increasing `d_model` can enhance the model's capacity to capture complex patterns in the data but may also increase computational requirements. It's a crucial hyperparameter that affects the model's representation power.

3.5.2 Number of Transformer Layers (`num_layers`):

Increasing `num_layers` allows the model to learn hierarchical representations and capture more intricate dependencies in the data. However, it also increases the model's complexity and training time. Finding the right balance between hyperparameters and model complexity is important.

3.5.3 Number of Attention Heads (`n_heads`):

Increasing `n_heads` allows the model to focus on different aspects of the input simultaneously. It can improve the model's ability to capture diverse relationships in the data. However, too many heads may lead to increased computational demands.

The experiments conducted above affirm that increasing the number of dimensions, attention heads, and transformer layers contributes to an improvement in accuracy.

4 Results and Conclusion

4.1 Sentiment Analysis Evaluation

The accuracies presented in Table 1, showcase the performance of the trained transformer models across different datasets. For model1, the achieved training accuracy of 85.36%

indicates that the model successfully learned patterns and features from the training data. Moreover, the closely matching validation (84.92%) and testing accuracies (85.17%) highlight the model's ability to generalize well to previously unseen data. This consistency across training, validation, and testing sets is indicative of the model's reliability and effectiveness in sentiment analysis tasks.

Actual	**Predicted**		
	Class 0	Class 1	Class 2
Class 0	9051	243	125
Class 1	424	2596	125
Class 2	777	388	251

(a) Model1 Confusion Matrix

	Precision	**Recall**	**F1-Score**	**Support**
Class 0	0.88	0.96	0.92	9419
Class 1	0.80	0.83	0.81	3145
Class 2	0.50	0.18	0.26	1416
Accuracy: 0.85, Macro Avg: 0.73, Weighted Avg: 0.83				

(b) Model1 Classification Report

Actual	**Predicted**		
	Class 0	Class 1	Class 2
Class 0	9067	274	78
Class 1	442	2624	79
Class 2	839	425	152

(c) Model2 Confusion Matrix

	Precision	**Recall**	**F1-Score**	**Support**
Class 0	0.88	0.96	0.92	9419
Class 1	0.79	0.83	0.81	3145
Class 2	0.49	0.11	0.18	1416
Accuracy: 0.85, Macro Avg: 0.72, Weighted Avg: 0.82				

(d) Model2 Classification Report

The confusion matrix and classification report provide a detailed evaluation of the sentiment analysis model's performance across three classes: positive, negative, and neutral sentiments. Notably, the both the models exhibit high precision and recall for positive sentiments and negative sentiments. However, challenges arise in identifying neutral sentiments, where the models face lower precision and recall. These insights highlight the model's strengths and areas for improvement, particularly in enhancing its ability to identify neutral sentiments.

5 Future Work

The current work provides a foundation for future improvements and investigations. One avenue for further exploration involves studying further about the impact of hyperparameter tuning on the model's performance. Fine-tuning aspects like model architecture, learning rates, and optimization algorithms may yield enhanced results. Integrating more extensive or domain-specific datasets could improve the model's generalization across diverse contexts and handling of neutral sentiments. Additionally, delving into the interpretability of the model's predictions could offer insights into the features and patterns influencing sentiment analysis.

6 Conclusion

In conclusion, this project successfully implemented and evaluated a transformer model for sentiment analysis, achieving promising results on the provided datasets. The model demonstrated strong generalization capabilities, with consistent accuracies across training, validation, and testing sets. The analysis of results indicates the effectiveness of the transformer architecture in capturing complex relationships within textual data. While the current work provides valuable insights, there are opportunities for future improvements and refinements. Overall, this project contributes to the growing body of research on transformer models in natural language processing and sentiment analysis tasks.

7 References

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin `arXiv:1706.03762 [cs.CL]`,`https://doi.org/10.48550/arXiv.1706.03762`

2. H. E. İlgün and E. Kılıç, "Sentiment Analysis using Transformers and Machine Learning Models," 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 2021, pp. 42-45, doi: 10.1109/UBMK52708.2021.9558931.

3. Ashok Kumar Durairaj and Anandan Chinnalagu, "Transformer based Contextual Model for Sentiment Analysis of Customer Reviews: A Fine-tuned BERT" International Journal of Advanced Computer Science and Applications(IJACSA), 12(11), 2021. `http://dx.doi.org/10.14569/IJACSA.2021.0121153`.

4. Alamoudi, Eman & Alghamdi, Norah. (2021). Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. Journal of Decision Systems. 30. 1-23. doi: 10.1080/12460125.2020.1864106.