## horizontal line



Sentiment Analysis with BERT

28.10.2024

Hemanth A S, Deepak S l

Ai&Ds - A

# 

# Overview

This project implements sentiment analysis using BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art pre-trained language model developed by Google. The goal is to classify text into sentiment categories such as positive, negative, or neutral.

# Dataset and Preprocessing

We'll load the Google Play app reviews dataset, The dataset comprises the data that were sourced from various online reviews of movies and opinions from social media comment sections. These text data are used to train the 3 variants of BERT model: BERT-Base, Distil-BERT, XLNet.

* Add special tokens to separate sentences and do classification
* Pass sequences of constant length (introduce padding)
* Create array of 0s (pad token) and 1s (real token) called *attention mask*

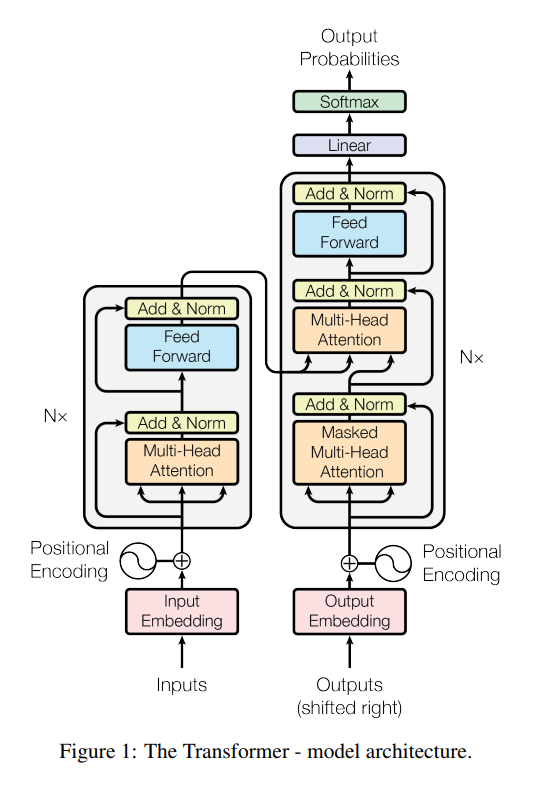
# Model Architecture & Description

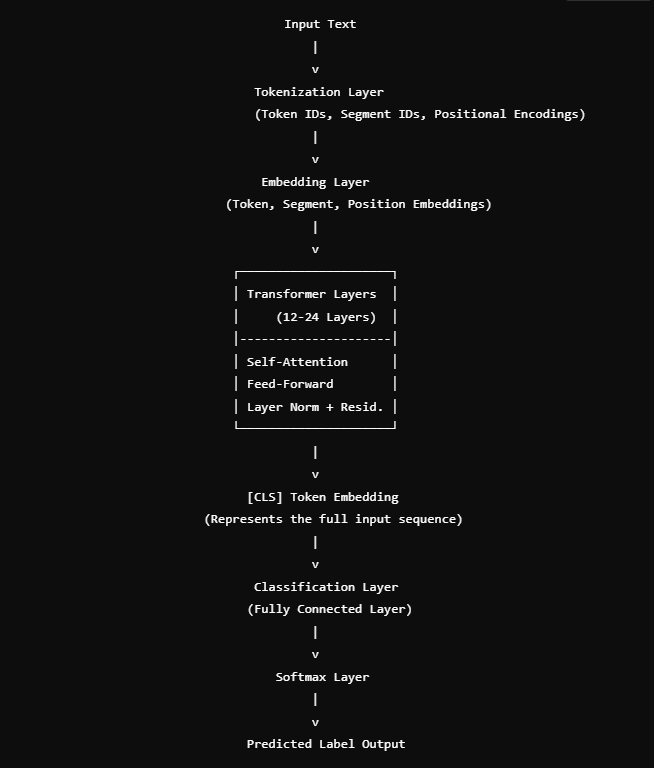
**BERT: Bidirectional Encoder Representations from Transformers**

BERT’s model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation of the **Transformer** model. The Transformer is based on an encoder-decoder architecture, but BERT, as a modification, only uses the encoder component.

BERT consists of multiple layers of transformers. The BERT BASE model has 12 layers (or transformer blocks). BERTBASE (L=12, H=768, A=12, Total Parameters=110M)

L= the number of layers, H=the hidden size, A=the number of self-attention heads





The key components of the Transformer model are:

**Self-Attention Mechanism:** The self-attention mechanism is designed to capture important sematic dependencies between words in a sentence, regardless of their distance from each other. This is highly crucial for sentiment analysis since this context can span multiple words or even sentences. For instance, in the sentence "The movie was slow, but the acting was brilliant," the word "but" modifies the sentiment of the phrase that follows. Self-attention helps in learning these kinds of relationships by computing attention scores between all pairs of words.

**Multi-Head Attention:** In practice, different parts of a sentence may require attention at different levels of granularity. For instance, one part of the network may need to focus on syntactic relationships, while another may need to focus on semantic meaning. To address this, Transformers use multi-head attention, where multiple self-attention mechanisms (heads) operate in parallel. The outputs of these heads are concatenated and transformed through a learned linear projection

**Positional Encoding**: Since the Transformer lacks the sequential structure of RNNs, it uses positional encodings to inject information about the order of words in the input sequence. Positional encodings are added to the input embeddings, providing the model with information about the position of words in the sentence.

**Feed-Forward Networks:** After the self-attention mechanism, the output is passed through a fully connected feed-forward network with a ReLU activation function. This step applies transformations to the attended embeddings, allowing the model to capture non-linear relationships between features.

**Layer Normalization and Residual Connections:** To facilitate training, each sublayer (e.g., multi-head attention, feed-forward network) is followed by layer normalization and residual connections. Residual connections help mitigate the vanishing gradient problem, allowing gradients to flow more easily through the network.

# Pre-training & Fine-tuning BERT

**Pre-training**

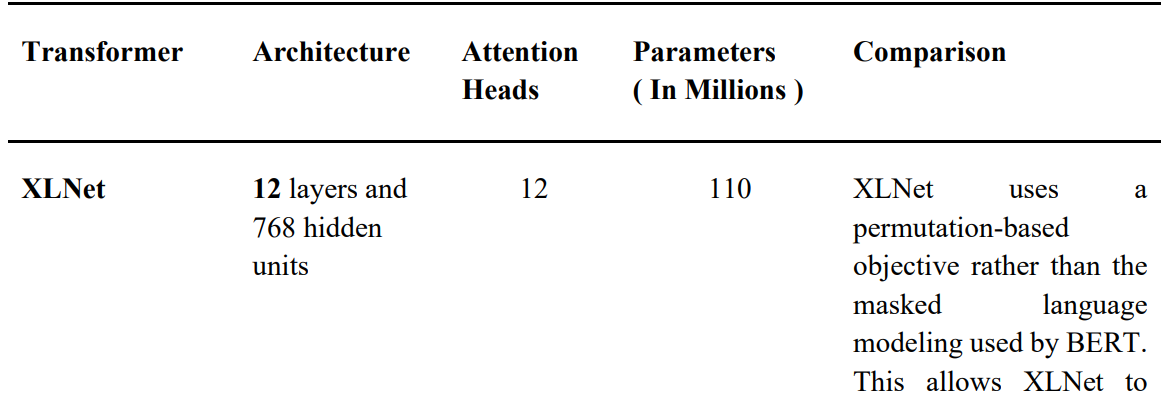
1. Masked Language Model (MLM): To learn bidirectional context, BERT randomly masks a percentage (typically 15%) of the input tokens and attempts to predict these masked tokens based on the surrounding words. This task forces the model to learn the relationships between words and their context, rather than just predicting the next word in a sequence.

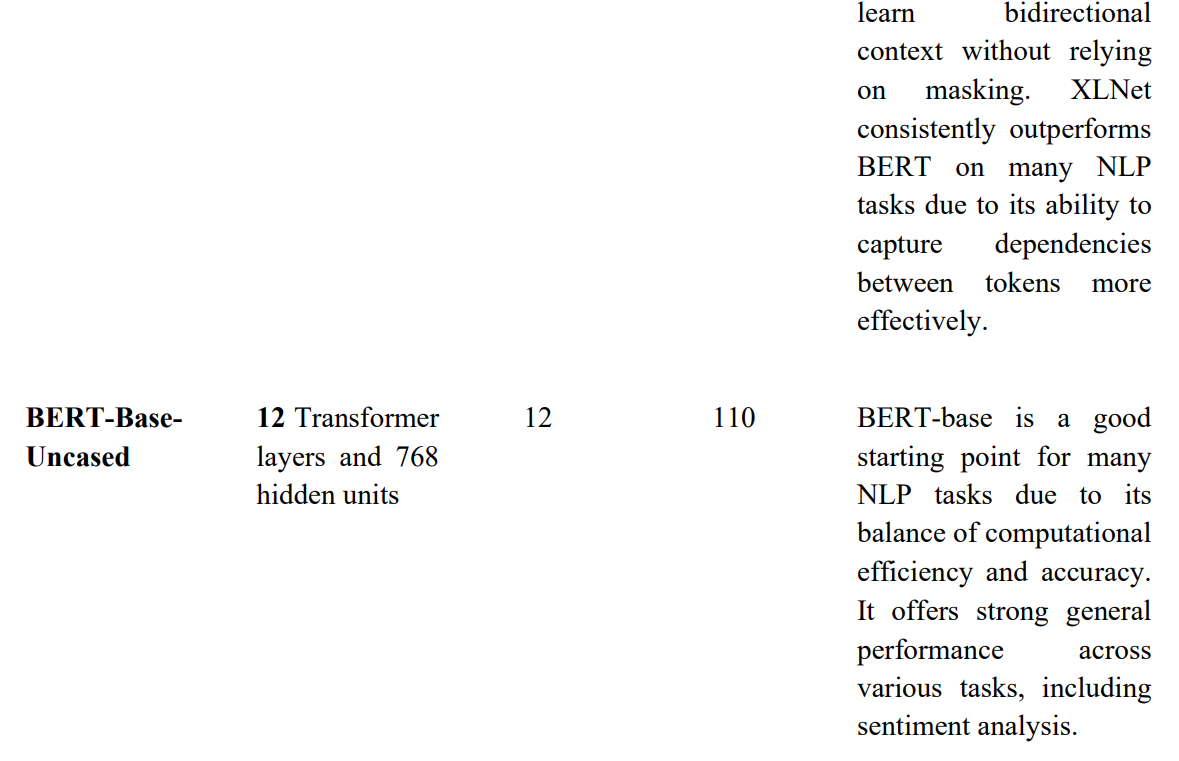
2. Next Sentence Prediction (NSP): In addition to the MLM task, BERT is also trained to predict whether two sentences appear consecutively in a text. (Ian Tenney, 2019) This task helps BERT capture inter-sentence relationships, which is important for tasks like sentiment analysis where the sentiment of a sentence can depend on previous or subsequent sentences. In this task, BERT is given two sentences, S1S\_1S1 and S2S\_2S2, and must predict whether S2S\_2S2 follows S1S\_1S1 in the original text. A classification layer is added on top of the pooled output of the [CLS] token, and the model is trained using binary cross-entropy loss.

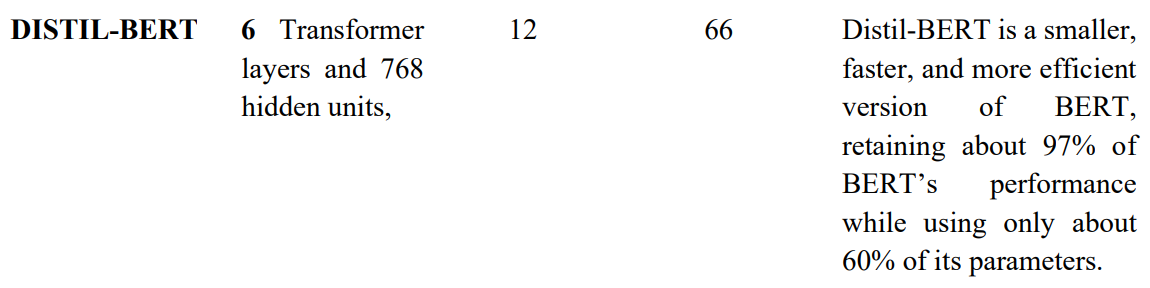
**Fine-tuning**

BERT's architecture is designed to be general-purpose to handle the prediction off the large sized data’s, allowing it to be fine-tuned for specific tasks like sentiment analysis. Fine-tuning involves adding a classification head on top of the pre-trained BERT model and training it on a labeled dataset for a few epochs. This process is relatively efficient, as the pre-trained BERT model already captures a deep understanding of language, and only the task-specific layers need to be learned from scratch.

# Model comparison (BERT-Base, XLNet, Distil-BERT)







# Evaluation Metrics

**Accuracy**: Overall correctness of predictions. Important for assessing general model performance.

**Precision**: Ratio of true positives to total predicted positives. Essential to minimize false positives in tumor detection.

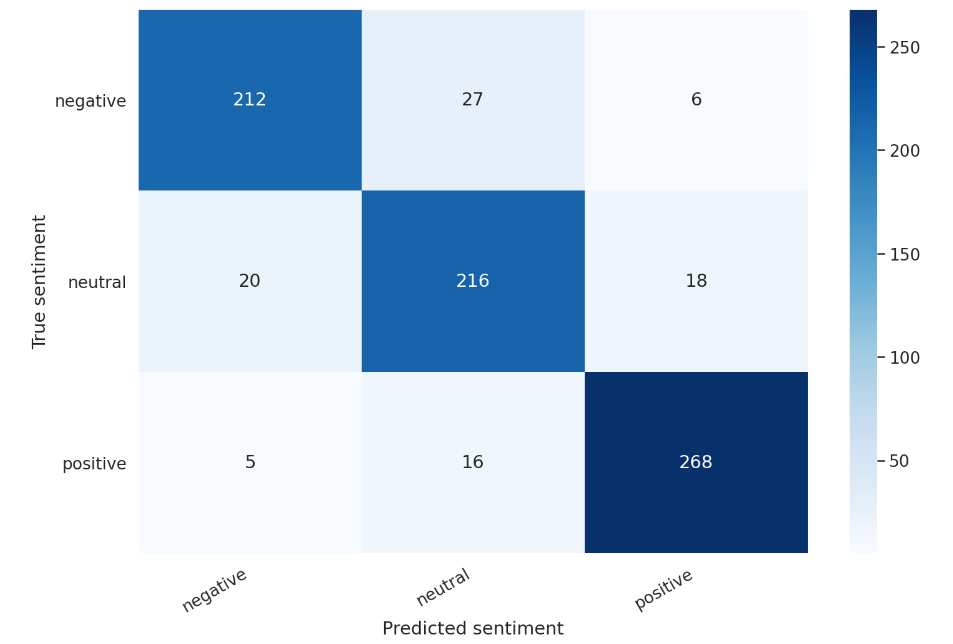
**Recall (Sensitivity)**: Ability to identify actual tumors. Critical to avoid missing any tumor cases.

**F1 Score**: Balances precision and recall, useful for handling class imbalances among tumor types.

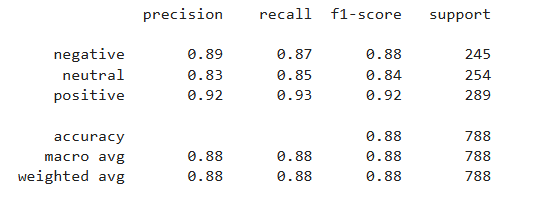
**Confusion Matrix**: Displays true positives, true negatives, false positives, and false negatives. Helps identify which tumor types are confused with others.

# OUTPUT

## 1. Confusion Matrix

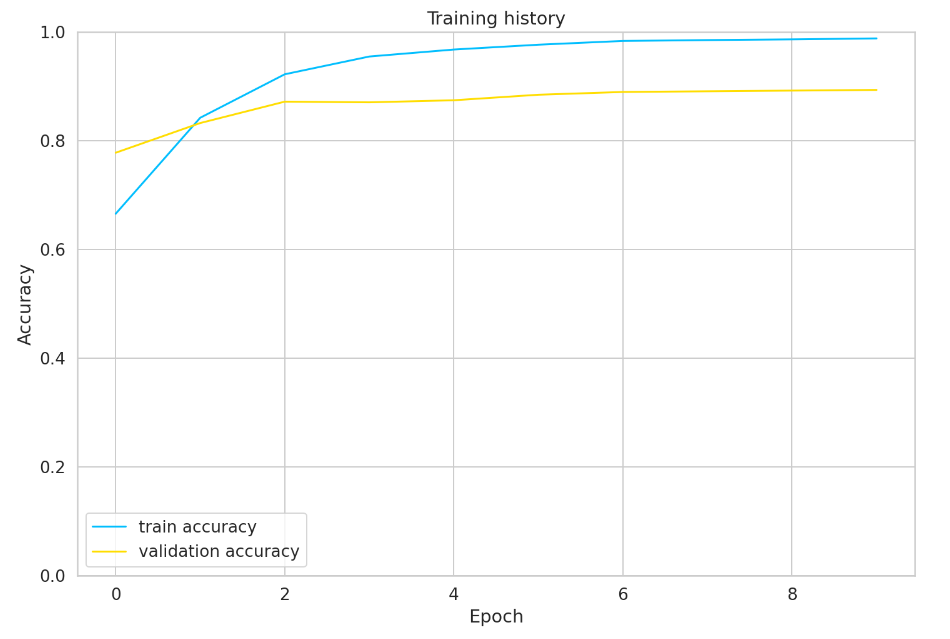


## 2. Precision, Recall, F1-score, support



## 

## 3. ROC curve



# Result Analysis

**Training and Validation Accuracy:**

The training accuracy improved significantly over the epochs, reaching about 98.8% by the final epoch, indicating that the model learned well from the training data.

The validation accuracy also improved, stabilizing around 89% after the third epoch. This high validation accuracy suggests good generalization to unseen data, with minimal overfitting.

**Loss Analysis:**

Training loss decreased steadily, reaching a very low value by the last epoch. This indicates the model’s confidence in its predictions.The validation loss, however, shows some fluctuations, initially decreasing and then stabilizing around 0.75. The higher validation loss compared to training loss may suggest minor overfitting, though the validation accuracy remains strong.

**Test Accuracy:**

The test accuracy of 88.3% is close to the validation accuracy, showing that the model performs consistently across validation and test datasets, which further confirms its generalizability.

**Overall Performance:**

The model achieved high accuracy and low loss, demonstrating effective sentiment classification. It shows strong predictive capabilities across the positive, neutral, and negative sentiment classes, as evidenced by the relatively balanced probability distributions in its predictions.

**Possible Improvements:**

further fine-tune performance, consider experimenting with different model architectures (e.g., using BERT variants like bert-large-cased or domain-specific models) or optimizing the dropout rate and learning rate schedule to mitigate minor overfitting.

# Summary

* **Benefits**:
  + BERT’s bidirectional context understanding improves the classification performance compared to traditional models.
  + The use of transfer learning reduces the data and computational resources required for high-quality results.
* **Challenges**:
  + BERT is computationally intensive and can be slow, especially during training.
  + Fine-tuning requires careful hyperparameter tuning to avoid overfitting on small datasets.

In summary, this project successfully applies BERT for text classification, showcasing its ability to understand nuanced language patterns and effectively categorize text data. The model can be adapted for various classification tasks, making it a versatile solution for NLP applications.

## References

1. **SENTIMENT ANALYSIS ON FEEDBACK USING VARIOUS BERT TRANSFORMERS**

Soorya Narayanan V

[Soorya+Narayanan+V.pdf](https://drive.google.com/file/d/1dHK2_v5bVvvQel2ZYJjRl9CaNeiS4kvk/view?usp=sharing)

2. **Attention is all you need**

paper by Vaswani et al. in 2017

<https://arxiv.org/pdf/1706.03762>