Text Classification Using Pre-Trained BERT

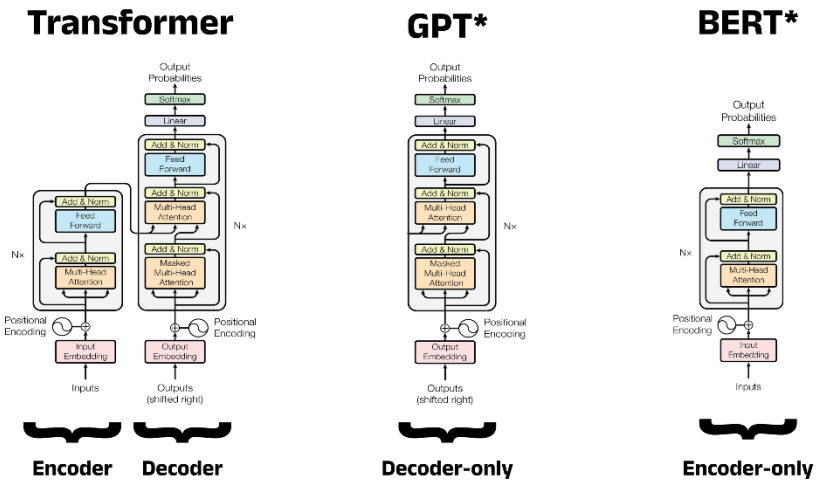
1. Introduction to Text Classification

Text classification is a fundamental task in **Natural Language Processing (NLP)** that involves assigning predefined categories or labels to textual data. Common applications include sentiment analysis, spam detection, topic classification, intent recognition, and document categorization. With the exponential growth of digital text data from social media, emails, reviews, and documents, automated text classification has become crucial.

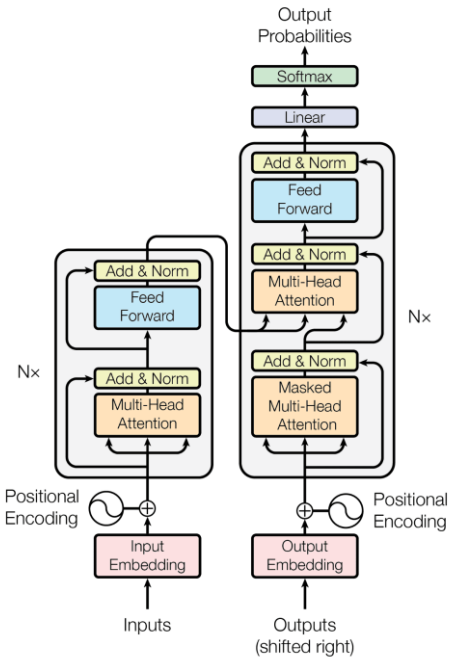
Traditional machine learning methods such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression rely heavily on manual feature engineering like Bag of Words (BoW) and TF-IDF. While effective to some extent, these methods struggle to capture **context, semantics, and long-range dependencies** in language.

The emergence of **deep learning** and **transformer-based models** has significantly improved text classification performance. Among them, **BERT (Bidirectional Encoder Representations from Transformers)** has become one of the most powerful and widely used pre-trained language models.

Overview of BERT



*Illustrative example, exact model architecture may vary slightly



BERT is a **pre-trained transformer-based model** designed to understand the contextual meaning of words in a sentence. Unlike traditional models that process text sequentially, BERT processes text **bidirectionally**, meaning it considers both left and right context simultaneously.

Key Characteristics of BERT

- Uses **Transformer Encoder architecture**
- Employs **self-attention mechanisms**
- Pre-trained on large-scale datasets such as:
    - Wikipedia
    - BooksCorpus

3. Pre-training Objectives of BERT

BERT is trained using two unsupervised learning objectives:

3.1 Masked Language Modeling (MLM)

In MLM, some words in a sentence are randomly masked, and the model learns to predict the masked words using surrounding context.

Example:

```
Input: I love [MASK] learning
Output: machine
```

This enables BERT to understand bidirectional context effectively.

3.2 Next Sentence Prediction (NSP)

In NSP, BERT learns the relationship between two sentences by predicting whether the second sentence logically follows the first.

This is especially useful for tasks involving sentence pairs, such as question answering and natural language inference.

4. Text Classification Using Pre-Trained BERT

Text classification using BERT typically involves **fine-tuning** a pre-trained BERT model on a labeled dataset.

4.1 Input Representation

Each input text is converted into tokens using **WordPiece Tokenization**. BERT introduces special tokens:

- **[CLS]** – Represents the entire sentence
- **[SEP]** – Separates sentences
- **[PAD]** – Padding for equal length sequences

The final input consists of:

- Token embeddings
- Segment embeddings
- Position embeddings

4.2 Classification Architecture

For text classification:

1. The input sentence is passed to BERT
2. The output corresponding to the **[CLS] token** is extracted
3. A fully connected (dense) layer is added
4. Softmax (or Sigmoid) is applied to predict class probabilities

This architecture allows BERT to leverage its deep contextual understanding for classification tasks.

5. Fine-Tuning Process

Fine-tuning is the process of training the pre-trained BERT model on a task-specific dataset.

Steps Involved

1. Load pre-trained BERT model and tokenizer
2. Prepare labeled dataset
3. Tokenize and encode input text
4. Add classification head
5. Train for few epochs (usually 2–5)
6. Evaluate using validation metrics

Advantages of Fine-Tuning

- Requires less training data
- Faster convergence
- Higher accuracy compared to training from scratch

6. Applications of BERT Text Classification

BERT-based text classification is widely used in real-world applications:

- **Sentiment Analysis**
  Classifying reviews as positive, negative, or neutral
- **Spam Detection**
  Identifying spam emails or messages
- **Topic Classification**
  Categorizing news articles or blogs
- **Intent Detection**
  Used in chatbots and virtual assistants
- **Document Classification**
  Legal, medical, and enterprise document tagging

7. Advantages of Using Pre-Trained BERT

- Captures deep contextual meaning
- Handles polysemy (same word, different meanings)
- Works well with limited labeled data
- Reduces need for manual feature engineering
- State-of-the-art performance on many NLP benchmarks

8. Limitations of BERT

Despite its strengths, BERT has some limitations:

- Computationally expensive
- High memory usage
- Slower inference time
- Not ideal for real-time systems without optimization

Variants like **DistilBERT**, **ALBERT**, and **TinyBERT** address these issues by reducing model size.

9. Evaluation Metrics for Text Classification

Common evaluation metrics include:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- These metrics help assess the effectiveness of the BERT-based classifier.