8-2023

Optimized Generative Pre-training EEG to Sentiment Classification (OGPTSC)

Amira Hassan
amh7246@rit.edu

EEEE 790 – Thesis

Optimized Generative Pre-training EEG to Sentiment Classification (OGPTSC)

Date	:	24 August 2023
Presenter	:	Amira Hassan
Advisor	:	Dr. Ali Sayyad Assistant Professor of Mathematics RIT Dubai
Committee members	:	Dr. Abdulla Ismail Professor of Electrical Engineering and Computer Science RIT Dubai Dr. Jinane Mounsef Assistant Professor of Electrical and Computer Science RIT Dubai

Optimized Generative Pre-training EEG to Sentiment Classification (OGPTSC)

by

Amira Hassan

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Electrical Engineering Department of Electrical Engineering and Computing Sciences.

Department of Electrical Engineering and Computing Sciences

Rochester Institute of Technology - Dubai

August 2023

Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification and that it is the result of my independent work.

Amira Hassan

Date: 24 August 2023

Abstract

This research aims to revolutionize electroencephalogram (EEG) analysis by proposing and developing an innovative method for sentiment classification, introducing the groundbreaking "Optimized Chat Generative Pre-training EEG to Sentiment Classification (OGPTSC)." This novel approach harnesses the power of Generative Pre-training (GPT2) as the vocabulary language model while utilizing an intelligent hyper-parameter selection method to fine-tune and optimize the model's performance. Extensive testing is conducted on a widely recognized dataset to evaluate the OGPTSC's prowess, demonstrating its exceptional capabilities in sentiment classification. Notably, this research extends its scrutiny beyond the OGPTSC by applying the same hyper-parameter selection technique to well-established models such as BERT, BART, and Multi-Layer Perceptions, thus enhancing these models' overall reliability and generalizability.

A key innovation of this research lies in improving model structure through a comprehensive tuning process. This process dynamically adapts the models' configurations by strategically leveraging classification loss and error during training. As a result, the refined models exhibit unprecedented levels of accuracy and robustness. The outcomes of the experiments vividly portray the superiority of the OGPTSC model over existing approaches that utilized the same dataset and techniques (BERT Model, BART Model, and Multi-Layer Perceptions). The OGPTSC's exceptional performance in sentiment classification firmly establishes it as a groundbreaking solution, surpassing previous benchmarks and setting a new standard in EEG analysis and sentiment classification.

Furthermore, the optimized versions of the renowned BERT Model, BART Model, and Multi-Layer Perceptions also showcase remarkable improvements over their predecessors. These enhancements solidify their positions as formidable contenders in the sentiment classification domain, providing researchers and practitioners with an invaluable toolkit to tackle complex EEG-based sentiment analysis tasks. In summary, this research presents an impressive and influential contribution to EEG analysis and sentiment classification. The development and evaluation of the OGPTSC model and the enhancement of existing state-of-the-art models inspire a new wave of research and application possibilities in the field. These optimized models demonstrated superiority and versatility and laid the foundation for future advancements in natural language processing, sentiment analysis, and beyond.

Keywords: EEG, BERT, BART, Multi-Layer Perception, GPT2, Hyper-parameters, Sentiment Classifications.

Table of Contents

List of Figures

List of Tables

Abbreviation

EEG	ElectroencEphaloGraphy
BCI	Brain-Computer Interface
DL	Deep Learning
ZuCo	Zurich Cognitive
BMI	Brain-Machine Interface
BLEU	BiLingual Evaluation Understudy
MLP	Multilayer PerCeptron
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
BART	Bidirectional and Auto-Regressive Transformer
SVM	Support Vector Machines
PCA	Principal Component Analysis
LSTM	Long Short-Term Memory
OGPTSC	Optimized Generative Pre-training Sentiment Classification
Adam	Adaptive Moment Estimation
SGD	Stochastic Gradient Descent
GPT	Generative Pre-training

1. Introduction

Electroencephalography (EEG) is a non-invasive way to measure the electrical activity in the brain using small sensors placed on the scalp. The EEG examines the electrical impulses that the brain emits during communication and stores them as wave patterns. These wave patterns can reveal a person's mental state, such as being awake or sleeping, peaceful or worried, or having a seizure. They can also reveal varying degrees of brain activity. EEG is frequently employed in medical settings to identify and keep track of ailments, including epilepsy, sleep issues, and brain traumas. It may also be utilized in studies to examine the brain's functioning during various mental or emotional states. EEG is an effective technique for learning about the brain's inner workings without intrusive treatments in general.

1.1. Research Objectives

Electroencephalography (EEG) analysis is used to examine brain activity and understand how it links to various cognitive and behavioral processes. The non-invasive EEG method analyzes the brain's electrical activity through electrodes on the scalp. [1].

Various aspects of brain functioning, including attention, memory, language, emotion, and sleep, are studied by researchers using EEG analysis. Additionally, aberrant brain activity linked to neurological conditions, including epilepsy, Alzheimer's disease, and Parkinson's disease, can be detected using EEG readings. Finding patterns and correlations in the EEG signals that might shed light on brain function and dysfunction is often one of the particular research aims of EEG analysis. The precise research objectives of EEG analysis can vary depending on the study issue. EEG analysis can also be utilized in combination with other neuroimaging methods, such as functional magnetic resonance imaging (FMRI), to provide a more thorough knowledge of brain activity.

1.2. Research outlines

EEG is commonly used in various fields, including neurology, psychology, and cognitive neuroscience. Here are some outlines of EEG analysis in these fields:

a) Neurology: To detect and monitor seizures, epilepsy, and other neurological disorders, EEG is utilized in neurology. EEG analysis in neurology involves visually inspecting the EEG recordings to identify abnormal patterns such as spikes, sharp waves, and sluggish waves. Quantitative EEG (QEEG) analysis can also identify brain electrical activity abnormalities by statistically analyzing EEG signals.

b) Psychology: EEG is utilized in psychology to investigate cognitive functions like emotion, memory, and attention. Psychology uses EEG analysis to examine event-related potentials (ERPs) that are time-locked to particular stimuli or occurrences. ERPs quantify the brain's reaction to certain stimuli by averaging EEG signals from several trials [2][3].

Cognitive Neuroscience: EEG is used in cognitive neuroscience to investigate the neurological underpinnings of cognitive functions such as attention, perception, and decision-making. EEG analysis in cognitive neuroscience entails identifying the brain areas and oscillatory activity involved in these processes by applying cutting-edge signal processing techniques, including time-frequency analysis and source localization.

Problem Statement

The problem statement can be briefly summarized in this question: How to predict sentiment using EEG signals? EEG signals might be available as electrical signals or by extracting features from electrical signals. The main things are how to use EEG as a feature for sentiment predictions and how to predict sentiments using raw text without EEG features.

1.3. Research Aims

In this research, the primary goals that have been achieved are:

- Understanding EEG signals: its specifications, applications, how to read them.
- Understanding how to use EEG signals in sentiment predictions using deep learning models (DL).
- Improving the DL model by using optimization methods.

- Comparing the proposed approach with existing related methods.

To achieve this aim, the dataset of EEG needs to be handled by a deep learning model and then mapped or matched with a linguistic model. The final results are used for sentiment predictions.

1.4. Dataset Description

Two datasets are used to validate our approaches:

1.4.1. ZuCo dataset

ZuCo is derived from 'A simultaneous EEG and eye-tracking resource for analyzing the human reading process.' The Zurich Cognitive Language Processing Corpus (ZuCo) is a state-of-the-art dataset that combines EEG and eye-tracking recordings collected from twelve subjects as they read natural English text. These recordings encompass active and passive reading tasks and apply to various fields, including neuroscience, psycholinguistics, and natural language processing. This resource offers valuable insights into the human reading and language understanding processes. It can be utilized to train machine learning models for various tasks, notably information extraction tasks such as entity and relation extraction and sentiment analysis. The dataset is particularly well-suited for such applications and can significantly enhance the performance of machine-learning models. The ZuCo 2.0 dataset includes recordings for more sentences and subjects and is also available in the same format, complete with text, EEG features, and sentiment labels [4]. The number of participants is 12 healthy adult native English speakers, each reading natural English text for 4-6 hours.

The dataset contains the following files: each task folder contains one Matlab file for each subject:

- Task 1 - Normal reading (NR such as Wikipedia)
- Task 2 – Sentiment reading (SR such as relations, sentiments, novels that contain feelings or jokes)
- Task 3 - Task-specific reading (TSR such as history, economics, etc.)

These tasks include raw text sentences with sentiment and relation labels for each sentence. Additionally, they provide Matlab files containing eye-tracking and EEG data. This data is recorded at both the word level (Word-level data) and the sentence level (Sentence-level data)

Structure of the .mat files: Each line within the structure contains information and features of one sentence. Forty-eight electrode pairs were used. The differences are always based on left–right values of homologous electrode-pairs. For each sentence (one line within the Matlab struct), the substructure "word" contains all features on the word level. The following eye-tracking features were extracted:

- nFixations: number of fixations
- Mean pupil size (pupil size is the pupil area measured in arbitrary units).
- FFD: first fixation duration.
- TRT: total reading time.
- GD: gaze duration.
- SFD: single first fixation (this field only contains a value if the fixation was fixated only once, which means it will be empty often).
- GPT: go-past time

The eye-tracking features are measured in samples with a rate of 0.5 (1 sample = 2ms). It also contains information about all fixations between the onset and offset of the sentence (including fixations outside of wordbounds). Word-level EEG features: For example, gamma activity of the EEG data during the first fixation duration of that specific word is measured, and other possible measurements (More details are available at https://osf.io/q3zws/wiki/home/). Sentiment levels in this dataset are negative, positive, and neutral.

Table 1-1 contains examples of three sentiment types. Some words are bolded to show the reasons for their labels.

Table 1-1: EEG ZuCo Sentiment dataset examples

Sentence	Sentiment label
-1 = negative, 0 = neutral, 1 = positive	
Presents a good case while failing to provide a reason for us to care beyond the very basic dictums of human decency.	0

Beautifully crafted engaging filmmaking that should **attract** upscale audiences hungry for quality and a nostalgic twisty yarn that will keep them guessing	1
Bread, My Sweet has so many **flaws** it would be easy for critics to shred it	-1
Slow, silly, and unintentionally hilarious	0
Ultimately feels **empty and unsatisfying**, like swallowing a Communion wafer without the wine.	-1
Exudes the fizz of a Busby Berkeley musical and the **visceral excitement** of a sports extravaganza.	1
If **Deuces** Wild had been tweaked up a notch it would have become a camp adventure, one of those movies that's so bad it starts to become good.	0
The film often achieves a mesmerizing poetry.	1
A work of astonishing delicacy and force.	1
Flaccid drama and exasperatingly **slow** journey	-1
I **like** the new footage and still **love** the old stuff	1
Could The Country Bears really be as bad as its trailers?	-1
It depends on how well flatulence gags fit into your holiday concept.	0

The number of positive sentiments in the dataset is 140, negative sentiments are 24, and neutral ones are 137. Figure 1-1 below shows a histogram of labels

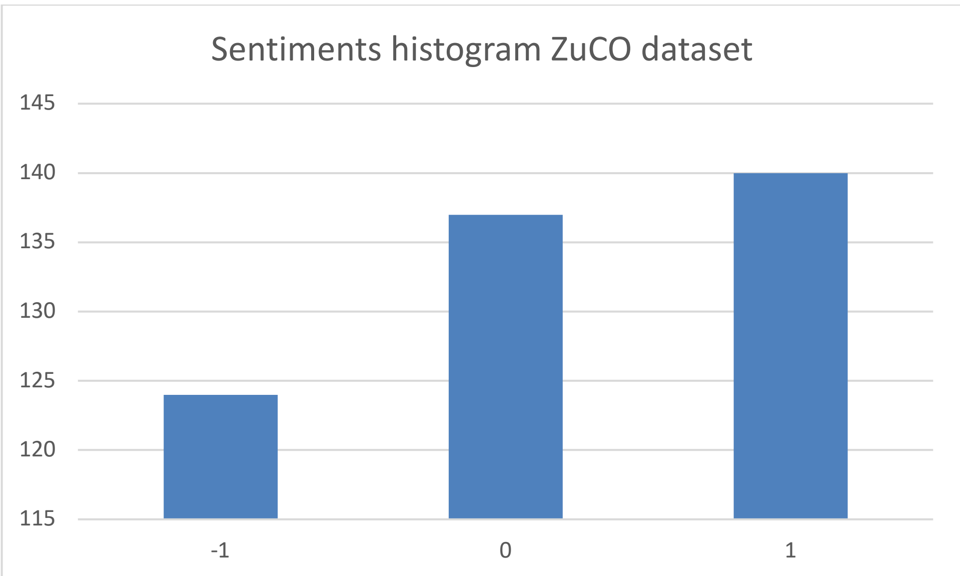

Figure 1-1: Histogram of sentiment labels of Zuco dataset

There is an imbalance in the dataset.

1.4.2. Stanford Sentiment Treebank (SST) dataset

The Stanford Sentiment Treebank is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in the language **(available as text and labels of sentiments)**. The corpus is based on the dataset introduced by Ketkar et al. in [17] and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser [18] and included 215,154 unique phrases from those parse trees, each annotated by three human judges. This new dataset allows us to analyze sentiment's intricacies and capture complex linguistic phenomena. Figure 1-2 shows how labeling is done, the count of many positive or negative words is calculated, and then the decision of sentiment is made based on the calculation results. There is no specific number of participants.

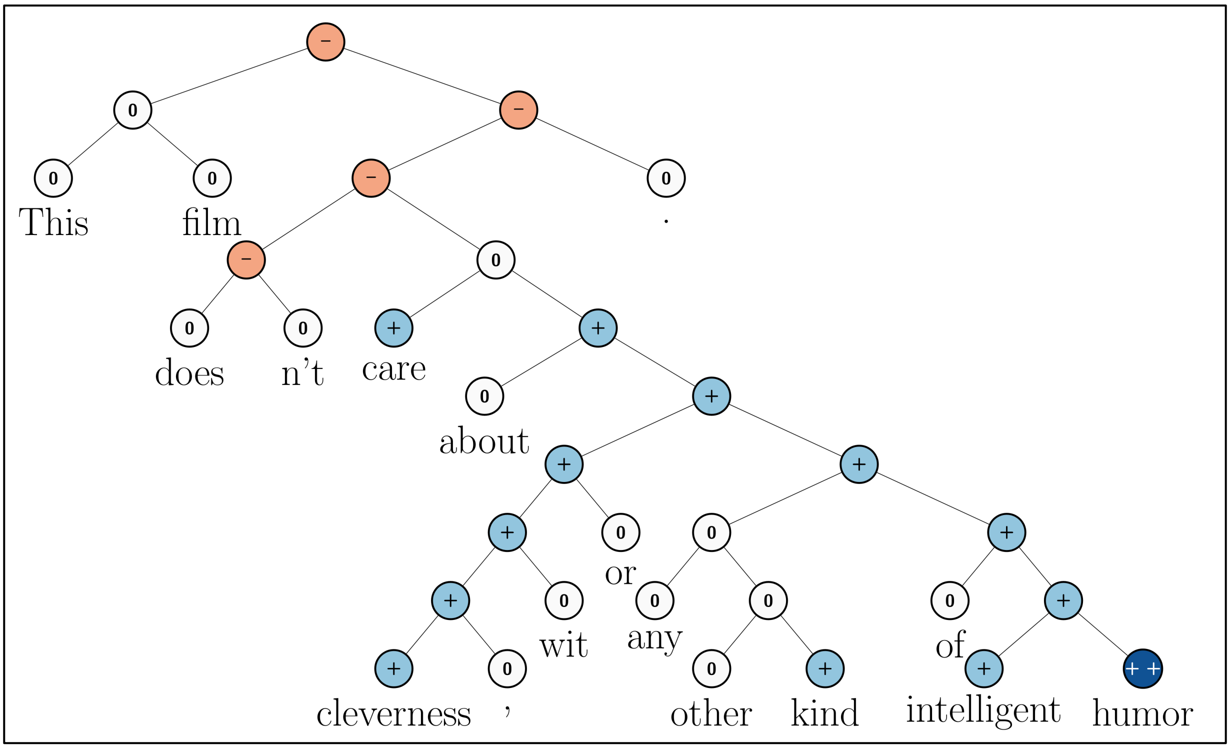

Figure 1-2: An Example of an annotated label of the SST dataset

As you can see, there are words such as 'cleverness,' 'intelligent,' and 'humor,' which are generally positive words. However, the negative word 'does not' in the text can make the overall sentiment negative. Sentiment analysis tasks, like Stanford Sentiment Treebank (SST), are critical for determining the sentiment of a text. For instance, they can help decide whether restaurant reviews are positive or negative.

Below are some examples that illustrate a range of sentiment, from positivity to negativity:

- This was the worst restaurant I have ever had the misfortune of eating at.
- The restaurant was a bit slow in delivering their food and didn't seem to use the best ingredients.
- This restaurant is pretty decent— its food is acceptable, considering the low prices.
- This is the best restaurant in the Western Hemisphere, and I will definitely be returning for another meal!

Based on these examples, sentiment analysis may seem like an easy task. However, many challenging nuances can make it difficult to analyze a phrase's sentiment accurately. Linguistic anomalies such as negation, sarcasm, and positive using negative terms are challenging for NLP models to handle. Take the following examples:

- I do not hate this restaurant. (Negation)
- I just love being served cold food! (Sarcasm)
- The food is unnervingly unique. (Negative words being positive)

As seen from these examples, it is not as easy as looking for words such as "hate" and "love." Instead, models must consider the context to identify these edge cases with nuanced language usage. With all the complexity necessary for a model to perform well, sentiment analysis is a complicated (and, therefore proper) task in NLP.
Compiling the dataset. For SST, the authors in [17] focused on movie reviews from Rotten Tomatoes. By scraping movie reviews, they ended 10,662 sentences, half negative and the other half positive. After converting all of the text to lowercase and removing non-English sentences, they used the Stanford Parser to split sentences into phrases, totaling 215,154. This dataset has/ sentiment clusters: very negative, negative, neutral, positive, and very positive. Table 1-2 shows examples with sentiments (Rates are divided in the range [0-1]).

Table 1-2: SST Sentiment dataset examples

Sentences	Sentiment	Rate
Gollum's performance is incredible	positive	0.70833
I know how to suffer and if you see this film you 'll know too	negative	0. 22222
I'm not exactly sure what this movie thinks it is about.	very negative	0.19444
I'm not suggesting that you actually see it, unless you're the kind of person who has seen every Wim Wenders film of the '70s	neutral	0. 52778
One of the best films of the year with its exploration of the obstacles to happiness	very positive	0. 96875

We can recover the five classes (very negative, negative, neutral, positive, and very positive) by mapping the positivity probability using the following cut-offs:

[0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1.0] for very negative, negative, neutral, positive, very positive, respectively. The number of very negative sentiments in the dataset is 11352, the number of negative sentiments is 43028, neutral ones are 119449, and the number of positive sentiments in the dataset is 50148. The number of very positive sentiments is 15255. Figure 1-3 shows a histogram of labels.

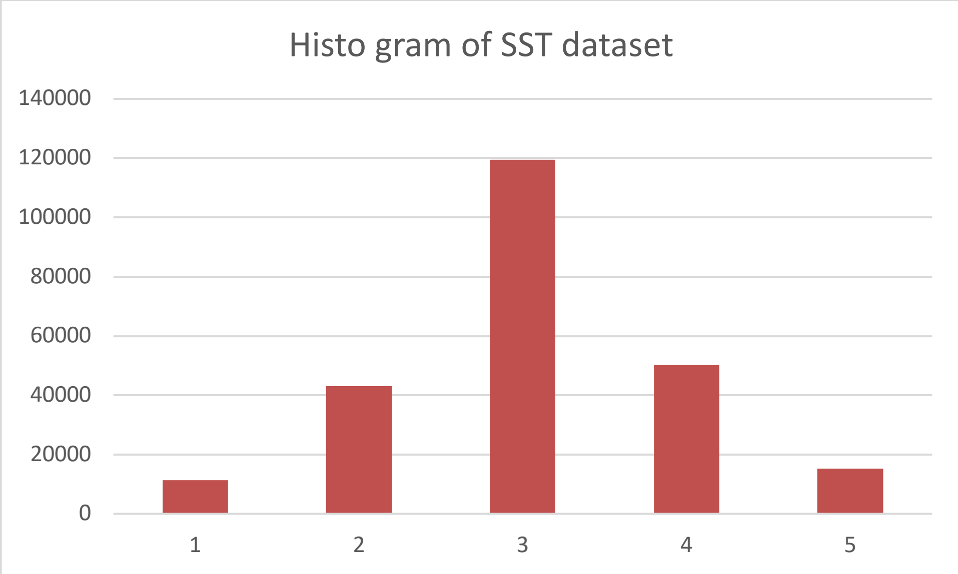

Figure 1-3: Histogram of Sentiments in the SST dataset

Figure 1-1 and Figure 1-3 show that both datasets are imbalanced.

1.5. Contributions

- EEG features to sentiment approach (tuning existing approaches)
- Text to Sentiment using Language models (tuning existing models)
- EEG to text using GPT2 to sentiments (new GPT approach with tuning)

1.6. Thesis organization

Through this presented thesis, chapter 2 will highlight state of the art in EEG conversion to texts or sentences, including a full description of all theoretical information about these ideas. Chapter 3 contains the developed methodology with specifications. Chapter 4. includes results, and the discussion will be illustrated. Conclusions and future scopes are mentioned at the end of the research.

2. Background and Literature Review

A brief and helpful introduction to EEG will take place in this section. Then, a more detailed summary of all of the techniques applied in the field of EEG, such as diagnosis and prediction, will be presented. In addition, the state of the art will be studied and analyzed more in depth.

2.1. Definition and applications of EEG

An electrogram of the brain's spontaneous electrical activity can be captured via electroencephalography (EEG). It has been demonstrated that the biosignal EEG's postsynaptic potentials of pyramidal neurons in the neocortex and allocortex. Using the International 10-20 system or variant, the EEG electrodes are often positioned along the scalp (commonly called "scalp EEG"). The term "intracranial EEG" is occasionally used to refer to electrocorticography, which involves surgically implanting electrodes. Visual inspection of the trace or quantitative EEG analysis are the two methods used most frequently in the clinical interpretation of EEG recordings [1]. Figure 2-1 shows an example of EEG signals with respect to time, and this example contains data as an image; the data points are not saved in the table. This is a type of EEG data where information is presented as an image. This needs image processing or computer vision tools.

Figure 2-1: Example of EEG

Electroencephalography (EEG) is a commonly used method for measuring and recording electrical activity generated by the brain; for example, Brain-Computer Interface (BCI) technology enables interaction between the human brain and devices, especially computers. This technology enables direct communication between the brain and a computer or other external devices by detecting and interpreting patterns of brain activity. BCI systems typically use EEG to detect neural activity and convert it into commands that can be used to control an external device, such as a computer cursor or a robotic arm. EEG measurements can also be used for other applications, such as detecting abnormal brain activity associated with certain medical conditions like epilepsy or sleep disorders. Research studies have explored the potential of using EEG-based BCI technologies in various fields, such as medicine, gaming, and accessibility for people with disabilities. However, many technical and ethical challenges must be overcome to make BCI more practical and widely accessible. Overall, EEG and BCI are interesting and active areas of research with a lot of potential for future development and applications [1].

The main aims of studying EEG (Electroencephalography) are to understand the brain's electrical activity better and use this knowledge to develop new applications and technologies. EEG is a non-invasive technique for measuring and recording the brain's electrical activity. It has been used in research to investigate brain function in various fields, from cognitive psychology to clinical neurology. Some of the specific aims of studying EEG include:

- To understand the neural mechanisms underlying cognitive processes such as attention, memory, and decision-making.
- To investigate the neural basis of neurological and psychiatric disorders such as epilepsy, Alzheimer's disease, and depression.
- To develop new methods for monitoring and diagnosing brain abnormalities and dysfunctions and evaluate treatments' effectiveness.
- To develop new applications and technologies based on EEG, such as Brain-Computer Interfaces (BCI), which allow individuals to interact with computers using their brain activity.

Overall, the study of EEG is an important and active area of research with a wide range of potential applications and benefits. By better understanding the brain's electrical activity, researchers can continue to make strides in improving human health and performance and developing innovative technologies.

Converting EEG signals to text refers to extracting meaningful information from the brain's electrical activity, as recorded by EEG, and converting it into a textual format that humans or machines can easily interpret and analyze.

Converting EEG signals to text involves several steps. First, raw EEG data is typically preprocessed to remove noise and artifacts and to enhance the signals of interest. After that, signal processing techniques, including filtering, feature extraction, and classification, are applied to the preprocessed data to identify specific patterns of brain activity.

One common application of converting EEG signals to text is developing brain-computer interfaces (BCIs); as mentioned earlier, this allows individuals to directly control external devices such as computers, prosthetics, or robotic systems through their brain activity. In this case, the EEG signal is processed in real-time to identify specific patterns of activity associated with different commands, such as moving a cursor or opening a robotic hand. These commands can then be translated into text-based instructions that the device can understand and execute.

Another application of converting EEG signals to text is in medical diagnosis and research. EEG data can be analyzed to identify abnormalities or patterns associated with different neurological or psychiatric conditions, such as seizures or depression.

Overall, converting EEG signals to text requires advanced signal processing techniques and machine learning algorithms and has the potential to revolutionize the way we interact with technology and understand the human brain.

2.2. Mapping EEG Features

As shown in Figure 2-2, a dataset is formed through the following cycle. While a person is reading a text, sensors measure EEG signals (these might be at the level of words or sentences) and monitor sentiments. In contrast, the read text is saved along with all other necessary and required details into a space so called a database.

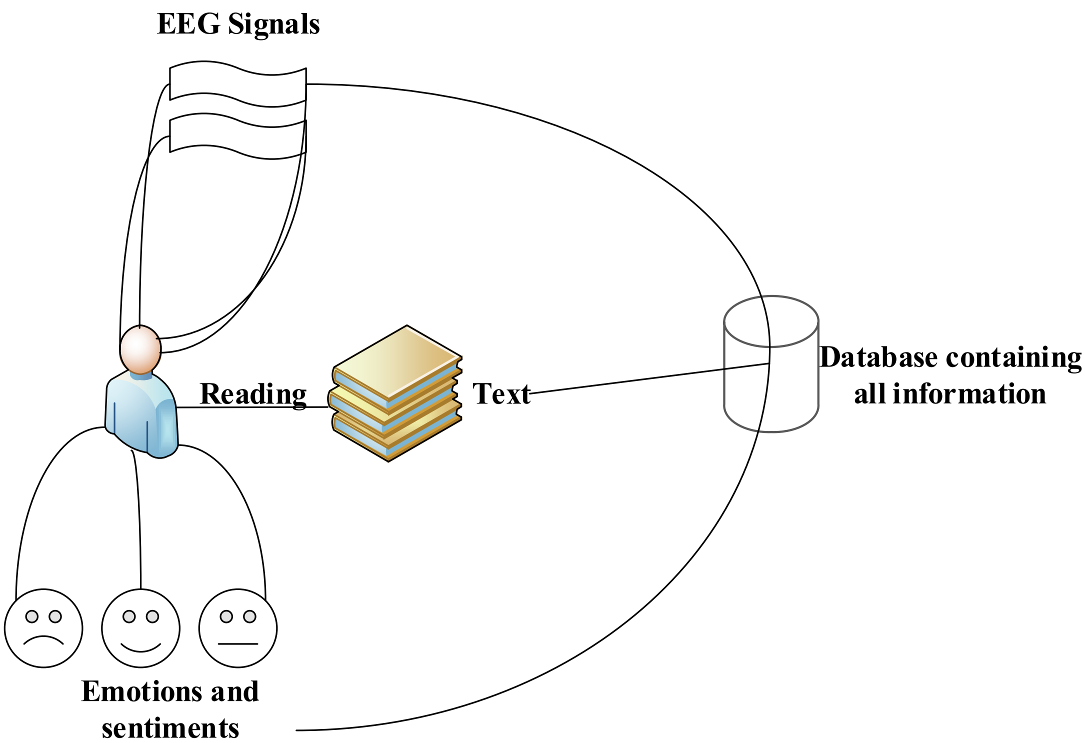

Figure 2-2: Gathering the dataset

All EEG signals are saved as features, while emotions are monitored and saved as labels. Text data must be converted and encoded for processing by machine learning algorithms. More details related to machine learning are presented in the following section (2-3). A detailed revision of EEG analysis will also be presented in the next chapter.

At the same time, a detailed revision of the EEG analysis will be presented in the next chapter.

2.3. Machine learning

Computer science, known as machine learning, has experienced tremendous growth in recent years. It is a branch of artificial intelligence that relies on creating statistical models and algorithms that enable computers to learn from data without being explicitly programmed. This methodology is employed in various industries, including marketing, banking, and healthcare. In this article, we will explore the evolution of machine learning, its various branches and sectors of application, as well as its connection to EEG readings.

2.3.1. History of Machine Learning:

Initial attempts to create machines that could learn from data were made in the 1950s when machine learning was seriously considered. Arthur Samuel, who created software that could play checkers and learn from their mistakes, was one of the early pioneers in this subject. This was a significant development since it showed that robots could be taught to carry out tasks based on experience. Over the ensuing decades, machine learning continued to advance as researchers created new approaches and algorithms for teaching computers to see patterns in data. Machine learning had a resurgence in popularity in the 1990s as a result of the development of the internet and the availability of vast amounts of data. This interest has persisted ever since.

2.3.2. Structures and Methods of Machine Learning:

Building mathematical models that can be taught to learn from data is the foundation of machine learning. Statistical algorithms that can spot patterns in data and anticipate the future are often the foundation of these models. Supervised learning, unsupervised learning, and reinforcement learning are the three primary categories of machine learning. By giving it labeled samples of such patterns, a computer may be trained to spot patterns in data through supervised learning. For instance, a machine might be taught to detect handwritten digits by being exposed to thousands of samples of labeled digits.

Unsupervised learning is the process of teaching a computer to spot patterns in data without giving it samples that have been labeled. This is frequently utilized when attempting to spot patterns that are not immediately obvious.

Training a computer to learn via trial and error is called reinforcement learning. This is frequently employed when a computer has to make conclusions based on insufficient or questionable data.

2.3.3. Branches of Machine Learning:

Many different subfields make up the significant topic of machine learning. Deep learning, computer vision, and natural language processing are some of the machine learning field's most crucial subfields.

A branch of machine learning called "deep learning" uses artificial neural networks as its foundation. By simulating the structure and operation of the human brain, these networks enable robots to learn from data in a manner similar to that of humans. Machine learning, known as "natural language processing, " aims to educate computers to comprehend and respond to human language. This is a crucial discipline since it has many uses in industries like Chabot and customer support.

Computer vision is a branch of machine learning that focuses on teaching machines to recognize and interpret visual information. This is important in applications such as self-driving cars, where machines must acknowledge and respond to objects in their environment.

2.3.4. Applications of Machine Learning:

There are various uses for machine learning in multiple industries, including marketing, finance, and healthcare. Among the most significant uses of machine learning are:

- Healthcare: The development of diagnostic tools, the prediction of patient outcomes, and the identification of prospective therapeutic targets all include the application of machine learning.
- Finance: Machine learning is employed in the financial sector to analyze vast volumes of data, anticipate stock values, spot fraud, and enhance risk management.
- Marketing: Machine learning is utilized in marketing to determine client preferences, tailor advertising, and enhance marketing strategies.

2.4. Deep Learning models

The structure and operation of the human brain, particularly the neural networks in charge of information processing, inspired the development of the branch of machine learning known as deep learning. Artificial neural networks, which are sophisticated mathematical models, are used in deep learning to discover patterns and correlations in data. Artificial neurons of numerous layers of linked nodes make up deep learning models. Each neuron processes one or more inputs using an activation function before producing an output. The data is gradually transformed across each layer as the output from one layer of neurons is provided to the next layer. The deep learning method modifies the weights and biases of each neuron in the network during training to maximize the model's performance on the given task. The difference between the model's anticipated and actual outputs, measured by a cost or loss function, is minimized to achieve this optimization [9][10].

Anomaly detection, natural language processing, recommender systems, and picture and audio recognition are just a few of the tasks that Deep Learning has been effectively used for. When enormous volumes of data are available for training, Deep Learning models frequently perform at the cutting edge. Deep Learning models, however, may be computationally demanding and need a lot of data and processing capacity to get accurate results.

Supervised and unsupervised learning are two learning methods used in machine learning (also known as deep learning).

Unsupervised learning algorithms are capable of working with raw, unlabeled data, which is frequently more readily available. In contrast, supervised learning algorithms require labeled data, which may be expensive and time-consuming [11].

2.4.1. Supervised Learning

A kind of machine learning called supervised learning includes learning from labeled data. In supervised learning, an algorithm is trained on a labeled dataset, where each data point is made up of a corresponding label (or output) and an input (or feature) vector. A mapping between inputs and outputs must be learned through supervised learning for the model to predict outputs for novel, unforeseen inputs correctly. Classification, regression, and object identification are just a few of the many tasks that may be accomplished with supervised learning. Logistic regression, decision trees, support vector machines (SVMs), and neural networks are a few types of supervised learning techniques [12].

2.4.2. Unsupervised Learning

Machine learning techniques such as unsupervised learning entail learning from unlabeled data. Unsupervised learning involves training an algorithm on a collection of inputs without labels or results. Unsupervised learning aims to find underlying patterns, structures, and relationships in the data. Unsupervised learning may be used for various problems, such as anomaly detection, dimensionality reduction, and grouping. Principal component analysis (PCA), autoencoders, and k-means clustering are examples of unsupervised learning methods [13].

2.5. Deep learning components

Deep learning components differ according to the input data; in the case of images, many components might be used. In the case of features such as structured data in tables such as EEG features, other elements might be used, as explained next.

2.5.1. Linear layers

This component creates a single layered-forwarded network with **n** inputs and **m** output. Mathematically, this module is designed to calculate the linear equation

$$Ax = o \tag{2.1}$$

Where **x** is input, **o** is output, and **A** is weight. In some cases, a trainable parameter is added to call bias. This is where the name 'Linear' came from. For example, let's take a case with n = 2 and m =1 then as shown in Figure 2-3, $x = [i_1, i_2]^T$, $A = [w_1, w_2]$

$$o = [w_1, w_2] * [i_1, i_2]^T + b \tag{2.2}$$

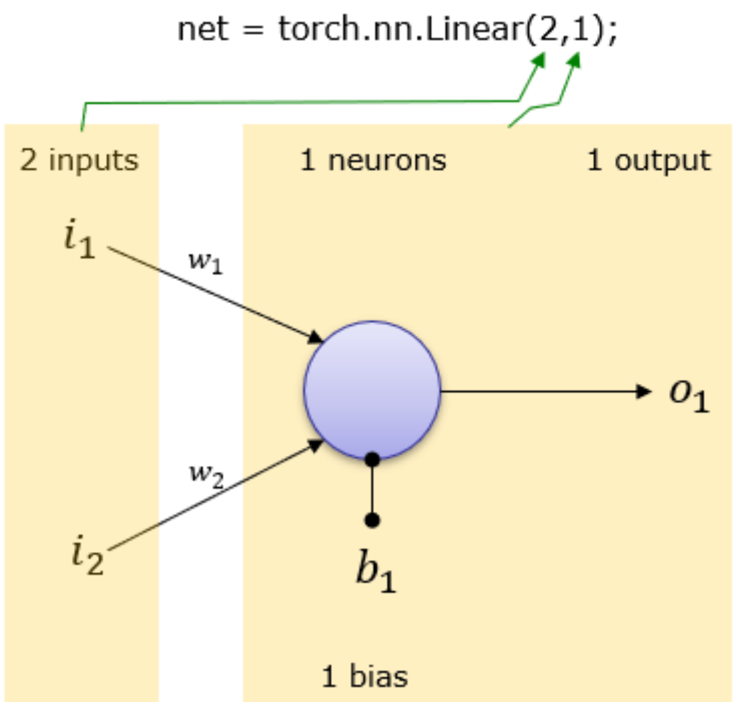

Figure 2-3: Example of Linear layer

The output of this layer might be required to be normalized or handled by an **activation function**.

2.5.2. Activating function

An Activation Function decides whether a neuron should be activated or not. This means that it will determine whether the neuron's input to the network is essential or not in the process of prediction using simpler mathematical operations. The famous and popular activation function is ReLU: Rectified Linear Unit; it is also known as non-linearity and is used after each input data sample. It is a non-linear process whose output is as in Figure 2-4 below (an example of activation function output): The ReLU method applies at the element and replaces all negative values with zero. The aim is to correct for nonlinearity in input data [14].

In Figure 2-4, the x-axis is the input of the activation function, which is ***o*** in the previous equation

$$o = [w_1, w_2] * [i_1, i_2]^T + b \qquad (2.3)$$

And y axis is the result of the activation function:

$$y = f(x) = f([w_1, w_2] * [i_1, i_2]^T + b) \qquad (2.4)$$

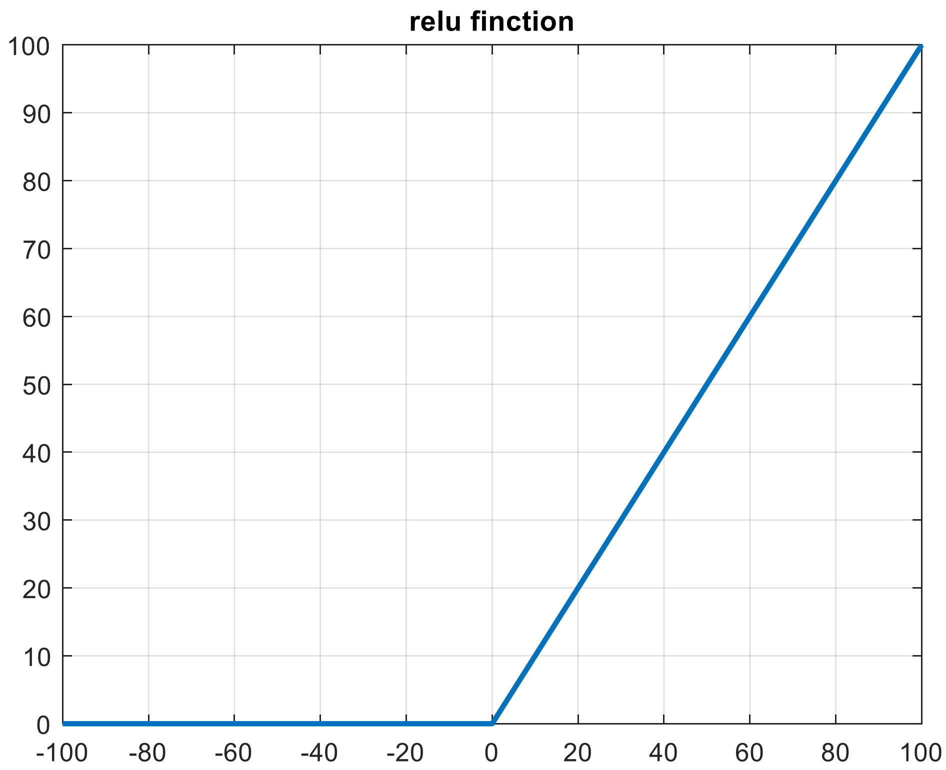

Figure 2-4: ReLu function

2.5.3. Dropout layer

A so-called dropout layer is added after a series of layers, and its task is to randomly exclude a percentage of neurons to reduce the possibility of retrofitting. The exclusion percentage is determined by a keep probability parameter [14] [15]. Figure

2-5 shows the effect of this layer on neurons where with dropout, some neurons are eliminated from working.

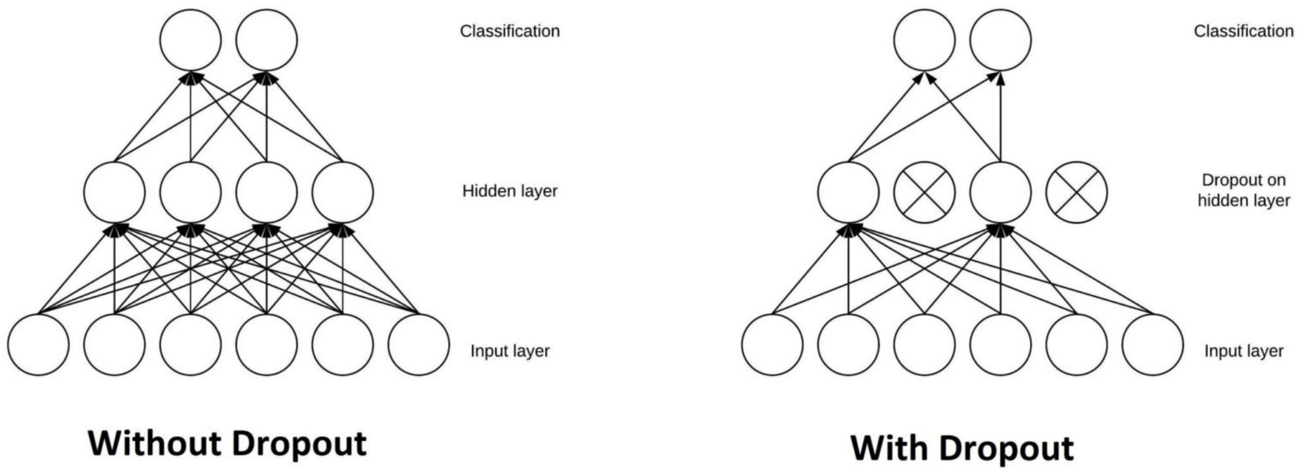

Figure 2-5: effect of dropout layer

2.5.4. Optimizers

The last layer of the system is connected to optimizers, which are algorithms or methods used to change the neural network's attributes, such as weights and learning rate, to reduce error. Optimizers are used to solve optimization problems by minimizing the cost of an error function.

We must specify the type of optimizer:

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Mini Batch Stochastic Gradient Descent (MB-SGD)
- SGD with momentum
- Nesterov Accelerated Gradient (NAG)
- Adaptive Gradient (AdaGrad)
- AdaDelta
- RMSprop
- Adam

More details about optimizers can be found in [16]. The optimizer needs to minimize a function as much as possible it can be. The loss function is defined according to the outputs; there are two types of outputs, numeric and categorical, and the loss function is determined according to that.

- LR: Learning rate

This parameter determines the learning speed; since there are trainable parameters in the system, they start with initial values, then their values are changed; the ratio of change in their values is the learning rate. Learning rate in some examples can be a trainable parameter.

2.6. Text representation

Text representation is a fundamental concept in NLP, and it refers to the process of representing any textual content or language corpus into a format that algorithms can comprehend. The current approaches to text representation involve using vectors and matrices that can produce machine-learning algorithms.

There are different ways to represent text, including:

- Bag-of-words model: This model represents text as a vector of word frequencies. Here, the frequency of each word in the text determines its weight, represented as a numerical value in the vector.
- Word embeddings: This model represents words using a vector space, where each word is mapped to a point in the space. Word embeddings capture the relationships between words, allowing algorithms to understand the language's meaning better.
- Character-based models: While word embeddings rely on meaning, character-based models represent text as sequences of letters, and algorithms can understand the patterns of sequences to generate new text or classify existing text.

Text representation is crucial in machine learning because it enables the algorithms to process human language, allowing them to model language data and perform various NLP tasks such as sentiment analysis, machine translation, and question answering.

Text representation requires an encoder in the roadmap to map each input token to an output vector, which can be extended to an infinite-length sequence. A simple word encoder is presented in Figure 2-6. Each word will have a unique value in the encoder, and each sentence might have a header and trailer.

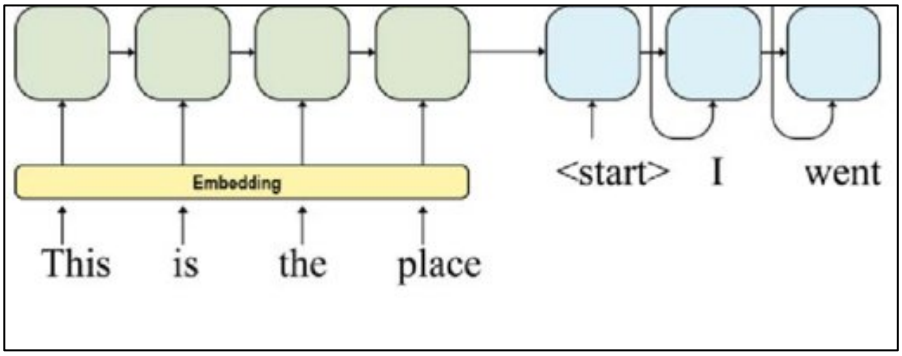

Figure 2-6: Simple Words encoder [5]

The development of words mapping has widely spread and improved, and the most known model for text representation is the BERT model.

2.6.1. BERT Model

The BERT (Bidirectional Encoder Representations from Transformers) model is a network design for natural language processing tasks such as sentence classification, question answering, and text generation. BERT is a pre-trained model, meaning that it is trained on large amounts of text data to encode knowledge of the English language in its parameters. The BERT model consists of a series of transfers comprised of multi-headed self-attention mechanisms and feed-forward layers. Self-attention mechanisms enable the model to encode bidirectional context dependencies in the input text, allowing it to have a deep understanding of the relationships between different words and phrases in a sentence. The feed-forward layer applies non-linearity to the output of the self-attention layer to create a deeper neural network [6].

The pre-training process for BERT involves training the model on large amounts of text data using two techniques: masked language modeling and next sentence prediction. Masked language modeling involves masking a portion of the input text during training and asking the model to fill in the masked section. This technique enables the model to learn about the relationships between different words and phrases in a sentence. Next sentence prediction involves training the model to predict whether two sentences are adjacent in a given input text. This technique enables the model to learn about the relationships between different sentences in a larger text corpus. Once pre-trained, the BERT model can be fine-tuned for various downstream tasks such as sentiment classification, entity recognition, and question answering. Fine-tuning involves further training the model on less task-specific data while keeping the pre-trained parameters fixed. By fine-tuning on a smaller amount of data, the model can generalize better to the task-specific data and achieve high accuracy [6]. Figure 2-7 shows two sentences with separation as input if BERT model, then how BERT handles them. First, there is a header and then a representation of two sentences.

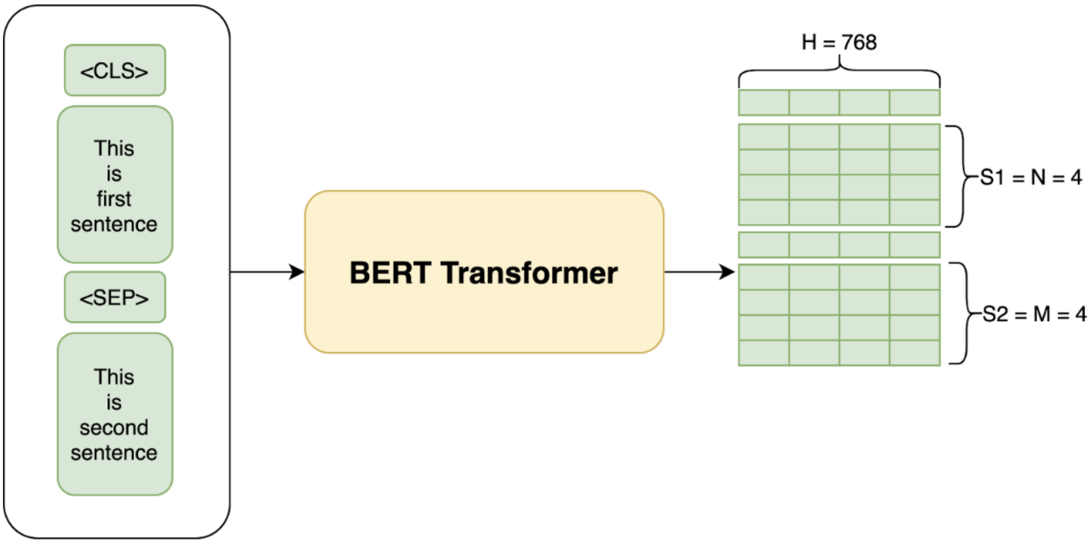

Figure 2-7: Example of BERT

N and M are the length of the representation of sentences, where H is the header and S1 and S2 are the first and second phrases, respectively. When working with any text-based dataset, this is necessary. The BERT model comes in a variety of forms, including [6]:

- BERT Base: A massive text corpus including 3.3 billion words served as the training set for this model, which includes 110 million parameters.
- BERT Large: This model was trained on a bigger text corpus with 11 billion words and includes 340 million parameters.
- RoBERTa: This modification of BERT performs better on several NLP tasks since it was trained on an even more extensive text corpus (160GB).
- ELECTRA: This updated version of BERT uses a fresh pre-training assignment to boost the model's effectiveness.
- ALBERT: This is a simplified version of BERT that achieves excellent performance on a variety of NLP tasks while using fewer parameters.

- DistilBERT: This quicker, more compact version of BERT performs as well as the more significant variants while having 40% fewer parameters.
- TinyBERT: Designed for usage on mobile and IoT devices, this is a more compact version of BERT.
- MobileBERT: This BERT variant is particularly resource-efficient and designed for usage on mobile devices.

Each of these versions differs in terms of the number of parameters, pre-training data, and specific architecture, with the aim of optimizing performance for different use cases while reducing memory and processing requirements.

2.6.2. BART Model

BART (Bidirectional and Auto-Regressive Transformer). Regarding natural language processing (NLP) tasks, BERT employs a transformer design with a multi-layer encoder, whereas BART is a sequence-to-sequence model. It is based on the encoder-decoder and transformer architectures used in BERT and GPT-2, respectively. BART has a bidirectional encoder and an auto-regressive decoder. The auto-regressive decoder is comparable to the decoder found in the GPT-2 transform. Considering to account for the previously created and the encoder output, it starts with the desired sequence. It creates a new token at each step. The BERT encoder is comparable to the bidirectional. Considering each token's on for the left and right each token, it provides a fixed-size representation for the input sequence. BART After being pre-trained on a sizable data corpus, as developed to perform several NLP tasks, including text creation, summarization, question answering, and machine translation, there may also be additional tasks that call for sequence-to-sequence models, such as dialogue generating and grammar checking [7][8].

BART has the advantage of producing multi-sentence outputs, which is helpful for jobs like summarization, where the output is frequently longer than a single phrase. Instead, BART may also produce high-quality abstractive summaries, which are summaries that encapsulate the main concepts of a text. Overall, BART is a flexible sequence-to-sequence with attained cutting-edge performance in some NLP tasks, especially those that require creating or summarizing text.

2.6.3. GPT

Generative Pre-trained OpenAI created the sophistic prediction and language prediction, Transformer 3, or GPT-2. With 175 billion parameters, it is one of the most extensive and intricate natural language processing (NLP) models currently available, outstripping other models, including its predecessor GPT-2.

GPT-2 uses unsupervised deep learning and the transformer neural network architecture to create human-like replies in response to input prompts. As a response to various fields, such as literature, science, and social media, among others, and can be trained on a lot of text data.

The GPT-2 model can comprehend and semantic various texts because it was trained on a vast corpus of web pages, books, and articles. It has multiple tasks, including sentiment analysis, question answering, language modeling, and summarization.

In many natural language processing tasks, such as content summarization, translation, and question answering, GPT-2 has demonstrated outstanding performance. It holds the record for several linguistic benchmarks, including Understanding Textual Entailment (the capacity to understand sentence relationships), SuperGLUE, and LAMBADA (a task for evaluating the models' capacity to predict missing words in a sentence).

One of GPT-2's key advantages is its capacity for common sense thinking, which enables it to provide contextually and logically consistent replies. This is made feasible by the model's extensive training data, which allows it to comprehend and produce phrases that defy grammatical norms.

Despite all of its successes, GPT-2 is not without flaws. One of the critical problems is that the model may provide biased results, which would raise ethical and fairness concerns. Additionally, bigger versions like GPT-2 are expensive and beyond reach for many people since they need specialized computational resources to function.

In conclusion, the GPT-2 language prediction model is highly sophisticated and has substantially advanced NLP. Its capacity for contextual understanding, common sense reasoning, and producing highly coherent sentences has significantly advanced language technology. Further study is still required to overcome its biases and limits to ensure its ethical and objective implementation across various fields.

2.6.4. Transformer XL

Transformers XL (extra-long) is a language model built using Google AI Language's transformer architecture. The capacity of this model to produce longer text sequences is greatly improved, and its capabilities are expanded over the original transformer model.

The model is made to handle the difficulty of producing longer text sequences than the transformer model's maximum length while keeping the generated text coherent and flowing. The length and coherence of the data the model could enter and process were constrained by the maximum sequence length in earlier transformer models.

It uses relative positional embeddings to implicitly incorporate the relative distances between words in a sequence without explicitly encoding them as additional features. This is one way that the Transformer XL model gets around this.

Transformer XL also suggests the concept of "recurrent transient layers," which enables the model to handle sequences block-wise, reducing the computing burden and allowing the model to produce more extensive sequences by tying together smaller blocOnesOne of its outstanding qualities is former XL's capacity to retain context and provide grammatically correct and logical replies a result, it is beneficial for producing lengthier, more intricate content, and composed of several layers of relationships.

The Transformer XL paradigm has been used in various context-based activities, including question-answering, summarization, and machine translation. It has been demonstrated to perform at the cutting edge on these tasks and is regarded as one of the most potent and precise large-scale models in NLP.

In conclusion, Transformer XL is an enhanced language model built on the transformer architecture that proposes several techniques, such as recurrent transient layers and relative positional Embedding, to address the problem of creating longer text sequences. Its capacity to remember context makes it very helpful in producing lengthy and cohesive text for various ones.

2.7. Sentiment Classification

The approach for sentiment prediction involves developing a predictive model that can determine the sentiment of a given text or document by analyzing its content, context, and tone. The primary objective of this approach is to accurately classify the sentiment of a given text as positive, negative, or neutral. Various strategies and algorithms are available for such sentiment prediction, including supervised and unsupervised learning techniques. Here, we'll go through the most popular methods for sentiment analysis:

Rule-based strategies: Rule-based techniques make use of a list of terms that are connected to both good and negative emotions, such as "happy," "sad," "angry," "fearful," and "surprised." These lists can be personally compiled or collected from accessible sources to the general public. A text's overall sentiment may be determined by counting the number of positive and negative words once the emotion-associated terms have been located. Although this method is relatively straightforward and quick to use, it might not be accurate, mainly when dealing with complicated texts or idiomatic idioms.

Lexicon-based techniques: Lexicon-based approaches are similar to rule-based approaches, but they extract sentiment from texts using sophisticated dictionaries or lexicons. Each word in these lexicons has an emotion score assigned to it, and together, they constitute a sizable collection of words, phrases, and idioms. The sentiment scores typically vary from negative to positive and are based on data annotated by humans. For instance, the AFINN lexicon is a well-known program that employs sentiment analysis using a collection of 2477 English words with corresponding sentiment ratings.

Techniques for machine learning: The most popular technique for sentiment analysis is machine learning algorithms. A labeled dataset with a sizable number of text instances divided into positive, negative, and neutral attitudes is used to train these algorithms. Using the patterns found in the training set of text data, the model may predict the sentiment of fresh and unexplored text data. Different machine learning methods can predict sentiment, such as Naive Bayes, Support Vector Machines (SVM), Random Forest, and Neural Networks.

In conclusion, sentiment analysis is crucial for many applications, including market research, product feedback analysis, and social media monitoring. Rule-based, lexicon-based, and machine learning-based methodologies can be used for sentiment prediction. The intricacy of the job, the available resources, and the performance requirements all play a role in selecting the best sentiment prediction method.

The technique of studying and figuring out a text or phrase's sentiment is called "text-to-sentiment analysis. " Sentiment analysis includes identifying and extracting subjective information from text, such as views, attitudes, emotions, and sentiments, using natural language processing (NLP) approaches. Text-to-sentiment analysis seeks to identify, categorize, and quantify the irrational information included in text data.

Different machine learning algorithms and methodologies can perform text-to-sentiment analysis, including rule-based approaches, unsupervised learning, and supervised learning. Through supervised learning, a model is trained to predict the sentiment in fresh, unlabeled data by using training data that has already been labeled and recognized the sentiment. Unsupervised learning algorithms may be used to evaluate vast volumes of unstructured text data to find patterns and subjects because they don't need labeled data.

For usage in text-to-sentiment analysis, various libraries and tools are available. These programs use pre-trained models and dictionaries to categorize text into positive, negative, or neutral feelings. Examples of well-known Python libraries that offer sentiment analysis features by implementing rule-based algorithms include TextBlob and Vader. In contrast, more sophisticated libraries, such as Hugging Face's Transformers library, can provide cutting-edge models that can classify text into a broader range of emotions or consider the language's context.

Text-to-sentiment analysis in business, marketing, social media analysis, and consumer insights are just a few examples. Insights into consumer happiness, brand reputation, and product feedback may be gained through analyzing customer reviews, feedback, and social media posts. Text-to-sentiment analysis may also be utilized to better serve customers by anticipatorily recognizing their issues and complaints.

The text-to-sentiment analysis does have certain restrictions, though. Accurately discerning the tone of sarcasm, irony, and other complicated forms of language that depend on context and cultural allusions is a considerable difficulty. The bias and inaccuracies in the training data and algorithms impact the sentiment analysis's accuracy.

The method of assessing and measuring the irrational information in text data is known as text-to-sentiment analysis. It includes classifying text into positive, negative, or neutral feelings using machine learning algorithms and libraries and has several uses in business, marketing, and consumer insights. Despite its shortcomings, text-to-sentiment analysis has advanced to the point where it can now provide insightful data while minimizing any biases or limits.

2.8. State of Art of converting EEG to Text

Ketkar [17], Has discussed using class connections to increase sentiment classification accuracy with rating scales. The article covers two strategies fusing se ties to raise the accuracy of sentiment categorization. It explains that class real refers to the connections between several classes or categories within a classification system. The first method is utilizing clustering or regression to put related classes together, enhancing accuracy and lessening data noise. In the second method, class connections in other datasets are learned from and used to improve performance on the present dataset using techniques like transfer learning or ensemble learning.

The authors stressed that applying express links to various rating scales can help increase sentiment categorization accuracy. These methods allow the model to learn from the between classes and enhance its performance on the current dataset.

Socher et al. [19], Authors examined the classification of sentiment and five classes of sentiments, starting from the text dataset. The method was a Recursive Neural Tensor Network; they first converted text to word vectors when fed into networks for training to predict sentiments, and then the accuracy of predicting sentiments was 80.7%.

Bird et al. [20] showed difficulty in appropriately categorizing motor imagery EEG signals (MI-EEG) for brain-computer interfaces (BCI) because of the signals' intrinsic noise and their' poor connection with brain activity. The authors provide a brand-new deep neural network-based learning architecture that concurrently acquires reliable high-level feature presentations from the underlying MI-EEG raw signals via low-dimensional dense embedding. An auto encoder layer and a combined convolutional recurrent neural network are used in the method to get rid of various artifacts like background activities. The suggested method is tested against existing state-of-the-art competitors, with findings showing that their method beats them with a classification accuracy of 95.53% on both a sizable public MI-EEG dataset and a small dataset gathered in the lab. The applicability of this proposed approach is further demonstrated with a practical BCI system for typing, which enables communication with the outside world by interpreting EEG signals.

Zhang et al. [21] presented the limits of the present brain-to-text technologies, which can only extract words from brain signals utilizing intrusive equipment and tiny restricted vocabularies. The paper suggests extending the issue to zero-shot phrase sentiment categorization on natural reading tasks and open vocabulary EEG-To-Text Sequence-To-Sequence decoding. The authors put forth a novel frame that uses language that models already undergone training, like BART, claiming that the human brain acts as a unique text encoder. Their method outperforms supervised baselines, scoring 40.1% BLEU-1 on EEG-to-text decoding, 77.4% F1 on zero-shot EEG-based ternary sentiment classification, and 27.5% F1 on Multi-Layer Perceptron EEG-based ternary sentiment classification. Given sufficient data, the model can handle data from various subjects and sources and shows excellent potential for a high-performance open vocabulary brain-to-text system, Ji [22]. The fundamentals of EEG and its viability for non-invasively recording brain activity are initially presented by the authors. Then, they go into how EEG can be used to identify emotion-related patterns in the amygdala, prefrontal cortex, and insula, giving information on how the brain processes natural and artificial emotions. The goes on to discuss several EEG characteristics and machine learning employed in sentiment perception, particularly for EEG classification issues, including Support Vector Machines (SVMs) or Artificial Neural Networks (ANN). The study also discusses several experiments that have employed EEG for sentiment perception, including identifying the mood of music, identifying the emotions elicited by images and videos, and observing people's moods as they interact with user interfaces. The authors highlight how different EEG features, such as frequency bands, event-related potentials (ERPs), and functional connectivity metrics, have been extracted to analyze temporal and spatial profiles of sentiment perception.

The authors also discuss how to expand the methods for capturing complex emotions like humor, sarcasm, and IR and how to use multimodal data sources to improve sentiment classification accuracy. Finally, they discuss EEG's drawbacks and potential directions in sentiment perception applications.

Ashraf et Razaq [23], Authors suggested a unique method for electroencephalography (EEG) signal-based sentiment identification. The suggested technique combines the one-dimensional convolutional neural network (1D-CNN) with bidirectional base autoregressive Softmax recurrent units with attention (BiBASRU-AT) models to improve sentiment identification accuracy. The authors first go through the difficulties and significance of sentiment identification using EEG data before outlining how the suggested model might address some of the drawbacks of earlier approaches. They clarify that the BiBASRU-AT model captures the temporal dependencies present in EEG data, while the 1D-CNN model is utilized to extract features from the EEG signals.

The design and training procedure of the suggested model are also covered in detail in the study. It starts with pre-processing the EEG data to reduce noise and raise the signal-to-noise ratio, then moves on to feature extraction with the 1D-CNN model. In order to capture the temporal relationships in the EEG signals and forecast the emotion of the input text, the features are fed into the BiBASRU-AT model. The authors used a publicly accessible dataset (DEAP) of EEG signals and accompanying sentiment labels to assess the proposed model. With an accuracy of 80.59%, the findings demonstrated that the suggested model performed better than various cutting-edge techniques for sentiment identification using EEG data.

In comparing feature extraction approaches, the authors also demonstrated that the suggested strategy performed better than the alternatives when utilizing the same dataset. They highlighted that the suggested method has significant applications in emotion identification and sentiment analysis, such as real-time prediction of user sentiment using EEG data in realistic circumstances.

In conclusion, a unique sentiment analysis approach was suggested using EEG data. With the addition of the 1D-CNN and BiBASRU-AT models, EEG data may be processed effectively while capturing the temporal relationships present in the signals. The findings demonstrate that the suggested model performs better than the state-of-the-art approaches and has substantial potential for real-world sentiment analysis and emotion identification applications.

Zhu et al. in [24], Authors suggested an approach for exploiting electroencephalography (EEG) data to identify emotions and identify stressful circumstances. To evaluate and categorize the EEG signals, the suggested technique combines the use of wavelet decomposition with machine learning methods. The authors first review the view of using EEG signals to detect stress and identify emotions and how the suggested technique may address some of these difficulties. According to their explanation, wavelet decomposition evaluates the EEG signals and extracts the pertinent characteristics. Support machines (SVM) and k-nearest neighbors (k-NN) are employed in machine learning algorithms to categorize the signals based on their features.

The design and training procedure of the suggested technique are further elaborated upon in the paper. The EEG signals are first subjected to noise removal before being subjected to feature extraction using wavelet decomposition. The SVM and k-NN algorithms are then given the characteristics to categorize the signals as being connected to various emotions and levels of stress. The authors gathered EEG data from several people experiencing varied levels of stress and emotion to assess the suggested approach. The findings demonstrated that using a wavelet of EEG signals associated with anxiety and emotions was accurate.

The suggested wavelet decomposition and machine learning methods strategy is contrasted with other methods of emotion detection from EEG methods in the study, emphasizing the benefits of wavelet decomposition and machine learning algorithms. The authors concluded that the suggested technique has significant implications for tracking people's stress and emotional states in real-world situations and enhancing their mental health.

Prior research on brain-to-speech decoding has successfully captured revelatory features by reconstructing spoken words from movements of the vocal tract [25][26]. In contrast, recent studies [27][28][29][30] have shown that language is the human brain's encoded language-dimensional semantic representations. Interestingly, they display comparable behavior by encoding pre-trained language models like BERT. These mode Bywords into contextualized semantic embeddings have these models ending transfer learning capabilities since they can significantly enhance a variety of natural language processing (NLP) tasks, such as sequence classification and text creation, by optimizing them on specific downstream tasks. Earlier research has demonstrated potential transfer learning skills in brain signal decoding using deep learning models despite the covariate shift in brain signal data due to intra- and inter-subject variability. Additionally, much research has looked into the relationship between brain signal decoding and NLP models by using brain signals to enhance NLP task performance. Or by exploiting NLP models to understand how the human brain encodes language.

Through this presented work, we utilize the existing linguistic knowledge gained by these models from large text corpora to perform an open vocabulary analysis task with scarce data to employ it in sentiment classification and test some approaches to representing models of vocals gained from EEG. Based on the hypothesis that the human brain functions as a special text encoder, we jointly fine-tune pre-trained language models with additional projection layers to achieve the goal of our work. Simply, an extension of this work has been done with results that outperformed the recent studies by tuning hyper-parameters of the algorithm of the learning process. Table 2-1 below summarizes and compares some commonly used related studies.

Table 2-1: Summarization of state of art of

Study	Contribution	Method	Accuracy	dataset
Bird et al. (2019) [20]	Sentiment prediction (+,0,-)	Random forest EEG to statistical features to sentiment	97.98	EEG dataset and Text dataset (readers)
Wang et al (2022) [22]	Sentiment prediction and text analysis	Many methods (BERT, BART, MLP, LSTM)	Best (79.7)	Recursive Neural Tensor Network Text-to-word vector to sentiment
Zhu et al (2023) [24]	Sentiment prediction	Recurrent network (EEG to sentiments)	90.24	EEG for people watching movies
Socher et al (2013) [19	Sentiment prediction (5 classes)	Recursive Neural Tensor Network Text-to-word vector to sentiment	80.7	Text dataset
Our approach	Sentiments (three classes)	Multi-layer perceptions, LSTM, and GPT2 (EEG to sentiments)	Best (73.9757) GPT is new	EEG data
	Sentiments (five classes)	BERT and BART models (text to sentiments)	BERT: 86.26 BART: 78.07	Text dataset

Through this presented work, a deeding on the work done by [21] with optimization and tuning for some parameters to improve the results of sentiment classifications (negative, neutral, and positive) through the EEG analysis. The developed method tunes some parameters to enhance results and use a new language model GPT.

3. OGPTSC Developed methodology

In this chapter, the developed model is presented with all the details about the model of EEG analysis and how to map EEG features used in general applications. OGPTSC is illustrated, which contains the main components:

- EEG data.
- Model to handle EEG data (different scenarios are tested)
- GPT linguistic model.
- Sentiment classification (Training and evaluation)

The final step is optimizing the ll process (Optimized GPT Sentiment Classification). EEG data needs to define spaces since EEG features, features map words, and final sentiments exist.

3.1. OGPTSC Spaces

To understand the OGPTSC approach, let us define some spaces:

- **E** space that contains **E**EG features
- **W** Space that contains **W**ords after Models (BERT, BART, or GPT)
- **S** Space that contains **S**entiments

Figure 3-1 shows how the mapping between **E** space and **W** space is met; suppose **e** is, namely, from **E** space, **w** is a word from **W** space, and **s** is a sentiment from **S** space.

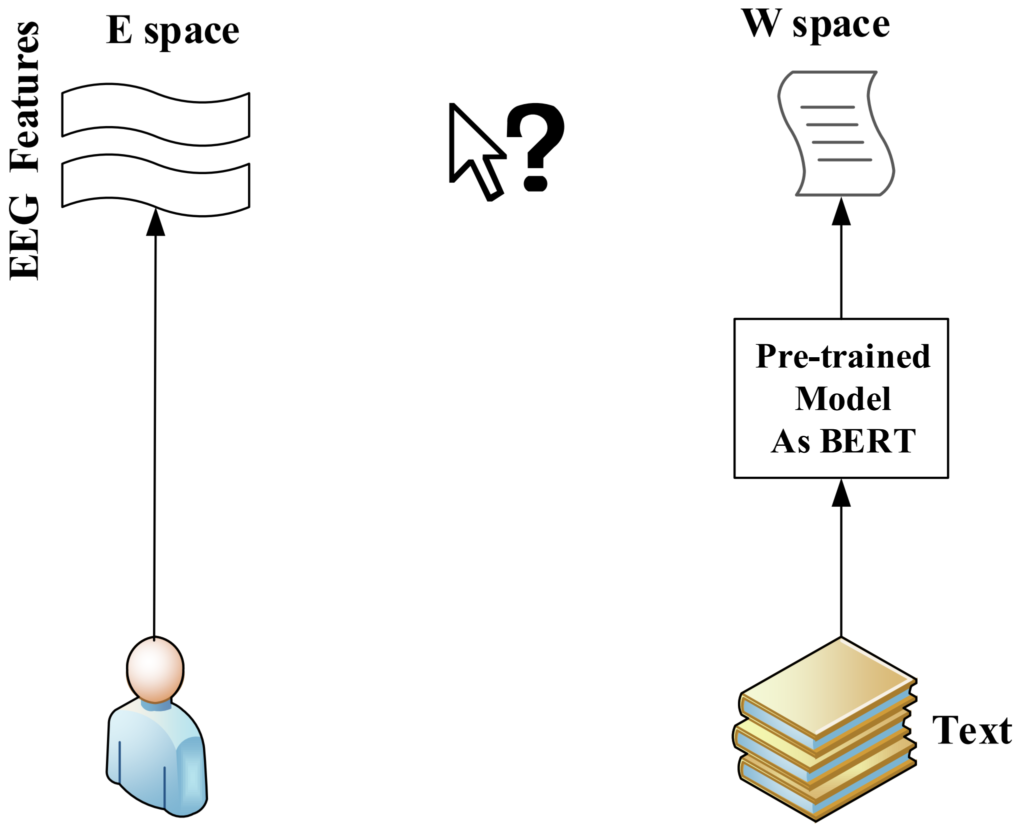

Figure 3-1: E space Vs W space

E space is obtained directly from the person that reads, and space is obtained from m read text. Word space means words or embedding representing words; that is why mode or others are needed to get **W** space. We define

$$< E, W >: e \rightarrow w \qquad (3.1)$$

$$< E, S >: e_m \rightarrow s: \quad where\ e_m is\ a\ group\ of\ e\ in\ E$$

The transitions $< E, S >$ can be achieved directly or indirectly:

$$< E, S > = < E, W > \ + < W, S > \qquad (3.2)$$

3.2. OGPTSC Text analysis

The flowchart Figure 3-1 presents our approach step by step. Our primary design, **E** space, is compared initially with the outcome of BERT Models; since **E** Space is features or coding of features (Like embedding), these codes must be compared or be trained to meet their match in **W** Space as shown in Figure 3-2. Through this approach, one may expect a high error ratio due to the fact that the EEG features may contain noises so the results could be affected by these additional factors.

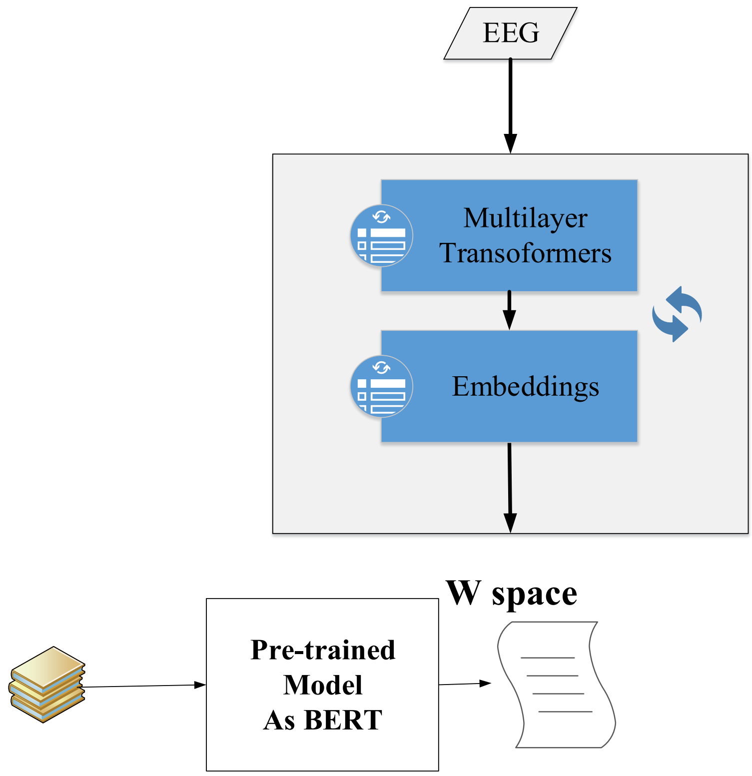

Figure 3-2: E space as Embedding

In Figure 3-2, the read text is needed during the training process to get W space that matches the **E** space obtained from EEG. **BERT** is just an example of a linguistic model for handling textile linguistics. Here, it is used as an encoder to code words. The main contribution here is that the GPT model will be used as a new model in the field of EEG. The GPT model, by its powerful features, will improve the overall process and cause differences. It is not just a new model but a model with many effective capabilities.

Therefore, another approach was considered, as shown in Figu. The advantage of this improved approach is that although errors will still occur, since the errors, in this case, will be in the form of words that can be filtered once the process is completed, The features from **E** swords will be converted to words at the end and then compared with the space. This enhanced approach is used for this presented thesis.

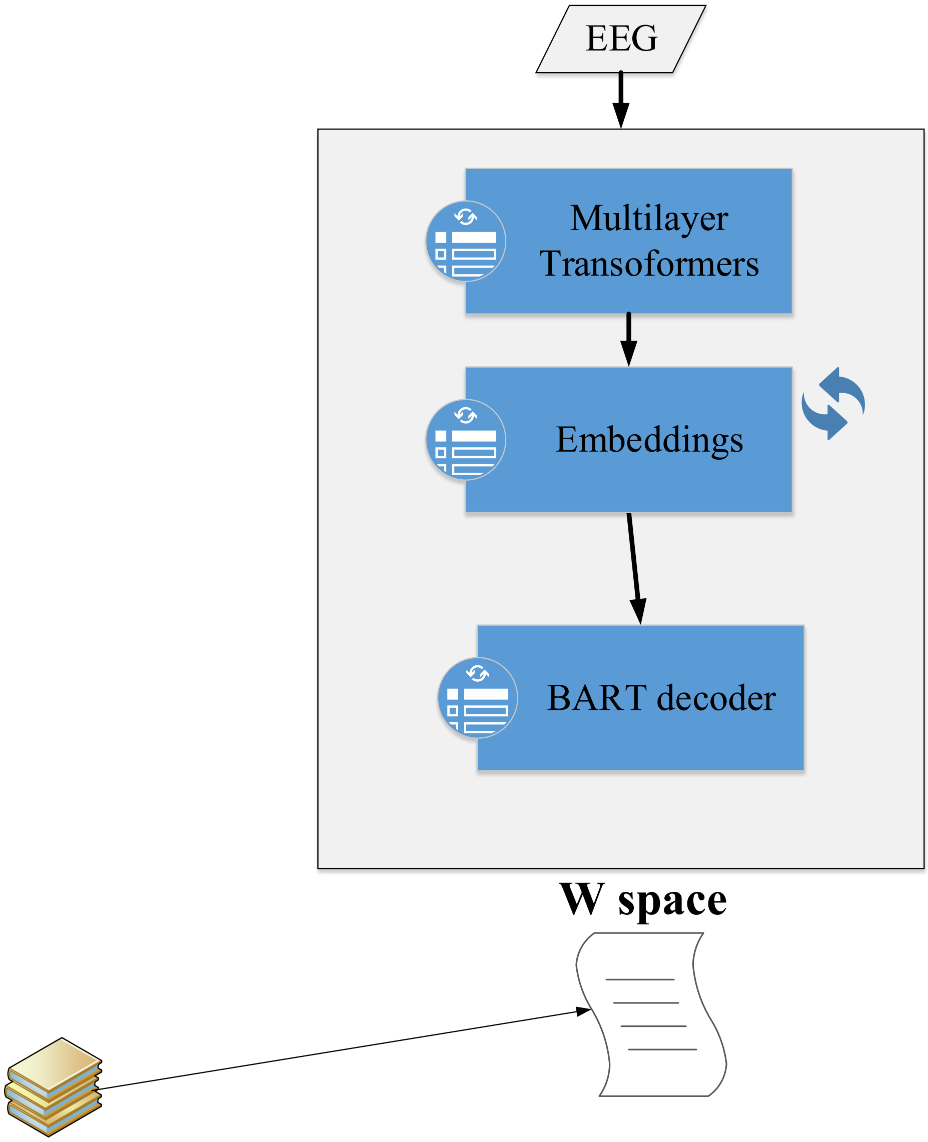

Figure 3-2The linguistic space as words

Lina's linguistic model here is used as a decoder to decode embedded predictions.

3.3. OGPTSC Sentiment Predictions

According to the text analysis, sentiment classification or prediction might be done depending on the type of EEG or their matches, as shown in Figure 3-3 and Figure 3-4.

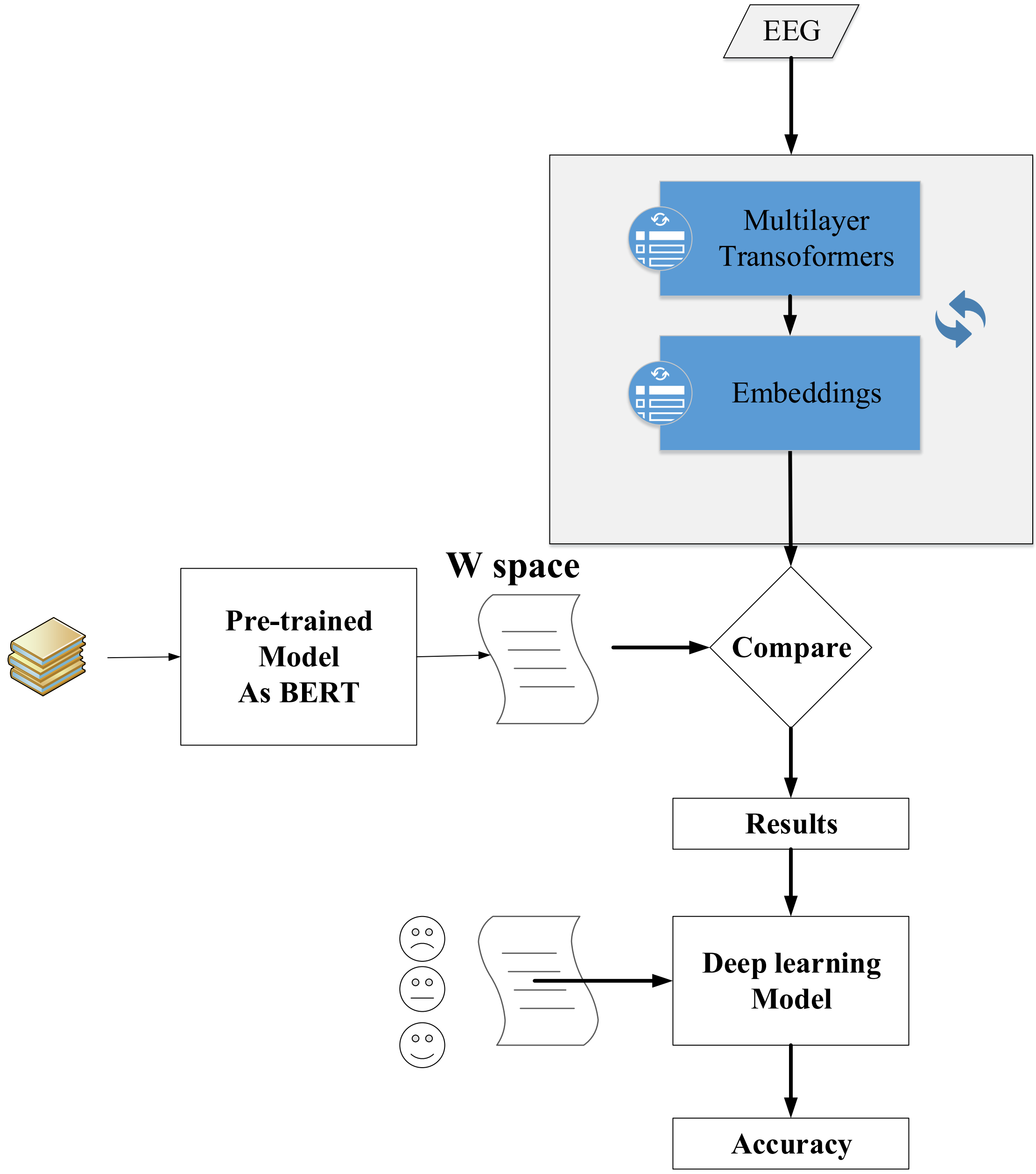

Figure 3-3: Sentiment Predictions with BERT

Figure 3-3 shows the matching between E space and W space and then the training process. Figure 3-4 shows a BERT is decoding without a matching process.

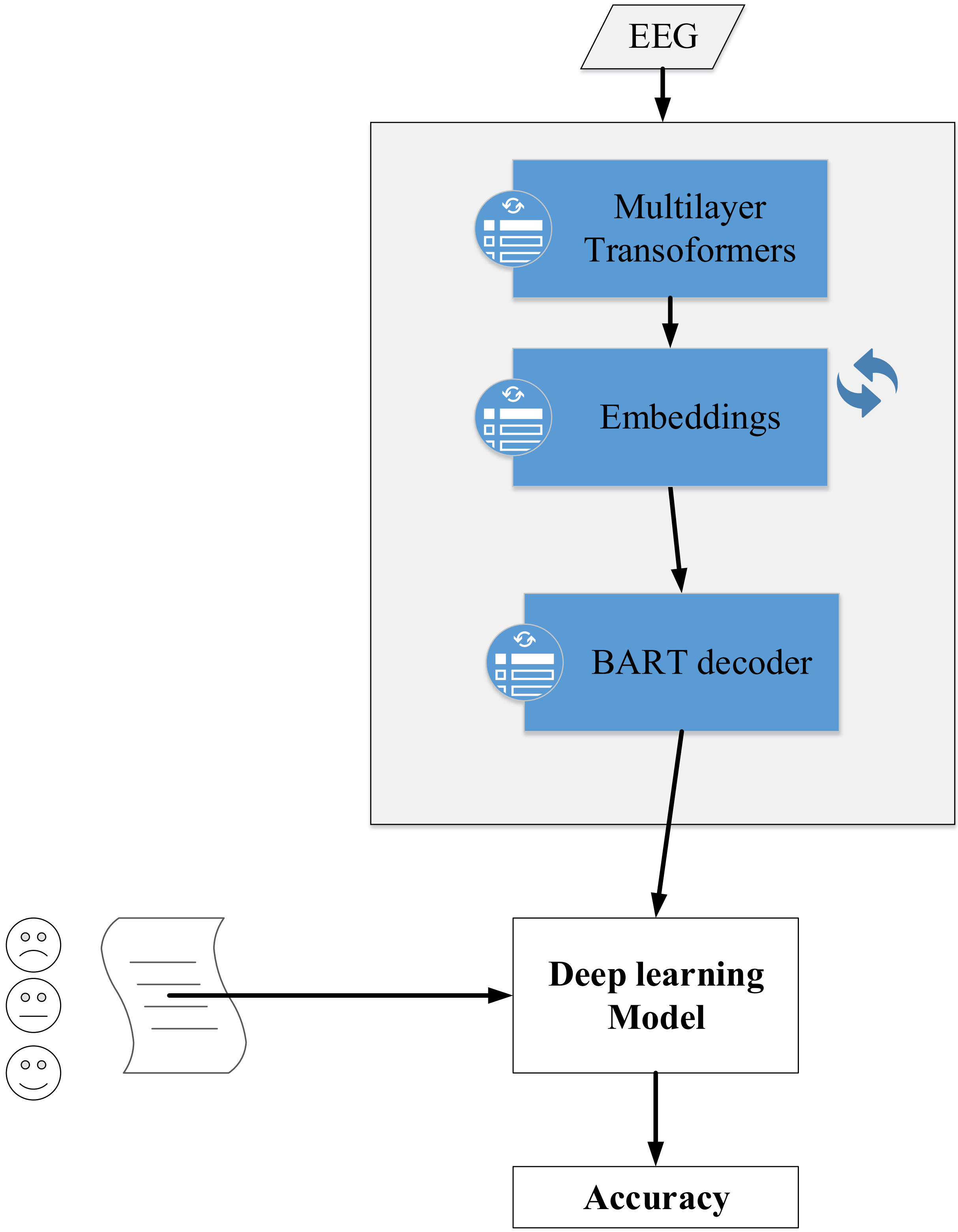

Figure 3-4: Sentiment Predictions with BART

3.3.1. Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a feedforward neural network consisting of multiple layers of interconnected nodes or neurons.

Within our chosen approach, MLP was used for sentiment classifications for EEG feature input and was trained on aggregated sentence-level EEG Sentiment pairs (**e, s**).

The MLP is a popular baseline model in deep learning and has been widely used for various tasks such as classification, regression, and prediction. The MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer contains a set of neurons, which take input from the previous layer and output to the next layer. The input to each neuron is a weighted sum of the inputs from the previous layer, which is then passed through an activation function [31].

The weights of the connections between the neurons in the MLP are learned through backpropagation, which involves adjusting the weights based on the difference between the predicted and actual output. MLPs are powerful models that can capture complex patterns in data and are highly flexible. However, they can also be prone to overfitting if the number of hidden layers and neurons is not properly selected and the dataset is small [32].

3.3.2. Binary Directional LSTM

This approach is used in implementation as an approach to be tuned, and it is a type of recurrent neural network (RNN) architecture that has been widely used for language processing tasks such as language modeling, part-of-speech tagging, named entity recognition, and sentiment analysis. The Bi-LSTM is an extension of the traditional LSTM architecture, which includes a feedback loop that allows the network to process input data sequences such as sentences or speech. The Bi-LSTM differs from the conventional LSTM in that it processes the input sequence in both forward and backward directions, which allows it to capture dependencies in the sequence that are not visible in a traditional LSTM [33].

The Bi-LSTM consists of two LSTM layers that process the input sequence in opposite directions. The output of the two layers is then concatenated and passed through a fully connected layer to produce the final output. The Bi-LSTM architecture is powerful because it can capture both forward and backward dependencies in the input sequence, which makes it well-suited for natural language processing tasks that require contextual understanding. The Bi-LSTM architecture has been shown to achieve state-of-the-art performance on various natural language processing tasks. However, it can also be computationally expensive and require considerable training data to perform well [34].

Bi-LSTM is a recurrent neural network architecture that processes the input sequence in both forward and backward directions. It has been widely used for natural language processing tasks and has been shown to achieve state-of-the-art performance. However, it can be computationally expensive and may require extensive training data to perform well.

3.4. Our main approaches

3.4.1. Approach 1 - EEG to sentiment

This approach starts from EEG features as input to the model, and the desired output is the sentiment labels. MLP and Binary-LSTM are used for this approach, as shown in Figure 3-6.

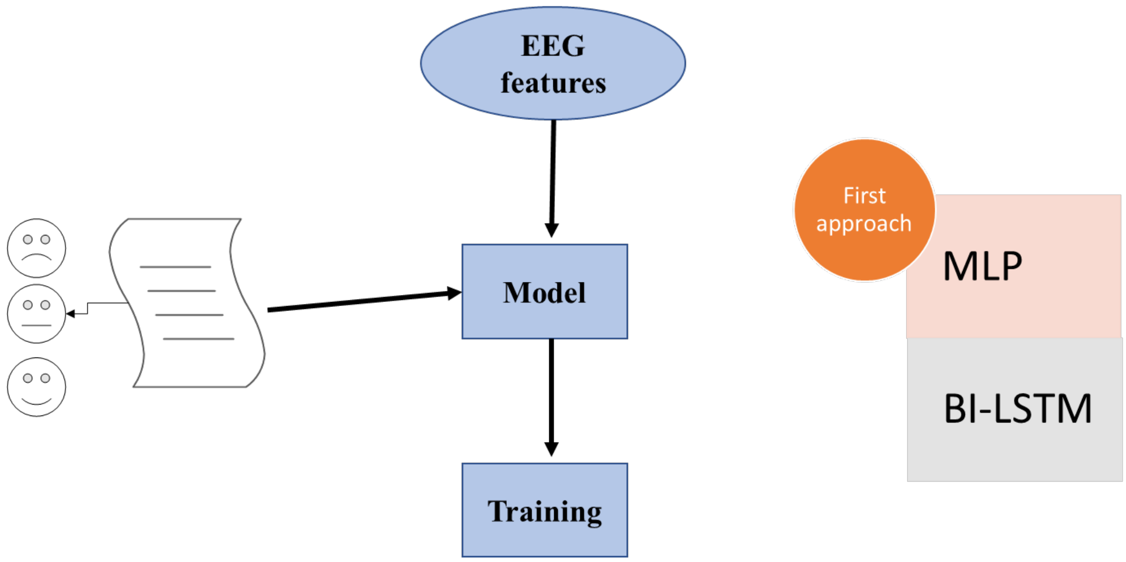

Figure 3-5: First approach (EEG features to Sentiments)

Using EEG features directly as input to be processed with layers I MLP or units of LSTM will help map these features to sentiments directly. Integrating EEG features as input provides a unique and innovative approach to understanding emotional states. This novel combination allows these models to capture subtle nuances in sentiment that traditional text-based sentiment analysis methods may overlook. The utilization of Multilayer Perceptions showcases the model's ability to learn complex relationships within the EEG features, enabling it to discern patterns associated with different sentiments. The nonlinear nature of MLPs makes them well-suited for capturing intricate dependencies within the EEG data, enhancing the model's capability to predict sentiments accurately.

Furthermore, incorporating Binary LSTM models enhances the temporal aspect of sentiment analysis. LSTM's ability to retain information over extended sequences is particularly relevant in capturing the dynamic nature of emotions. This temporal awareness contributes to the model's capacity to recognize evolving sentiments, making it more adept at handling nuanced emotional shifts over time. The significance of these models lies not only in their technical prowess but also in their potential applications. EEG-based sentiment analysis could find valuable use cases where sentiment analysis falls short, such as understanding emotional responses in real-time situations or assessing sentiments in a population with verbal expression.

3.4.2. Approach 2 - Text to sentiment using a Language model

This approach starts from the text as an input language model, such as BART/BERT models, and then feeds into the model, and the desired output is the sentiment labels; BERT and BART are used for this approach, as shown in Figure 3-7.

The utilization of BERT or BART for sequence classification further enhances the model's capability to discern sentiment in a highly context-aware manner. These transformer models capture long-range dependencies and contextual nuances, crucial for accurately interpreting the sentiment expressed in diverse and complex language structures. By training the model using a language model approach, we tap into the extensive pre-trained knowledge embedded in BERT or BART, fostering a more robust understanding of the subtleties of sentiment in natural language. This pre-training facilitates improved generalization to various domains and enhances the model's performance on sentiment analysis tasks, even when faced with previously unseen data.

The significance of this approach lies in its versatility and applicability across a wide range of text-based sentiment analysis scenarios. Whether applied to social media sentiment monitoring, customer feedback analysis, or any other domain where sentiment interpretation is pivotal, the model is a powerful tool for discerning and understanding the sentiment expressed in textual data. This contribution advances the state of the art in sentiment analysis and holds promise for practical applications in diverse real-world contexts.

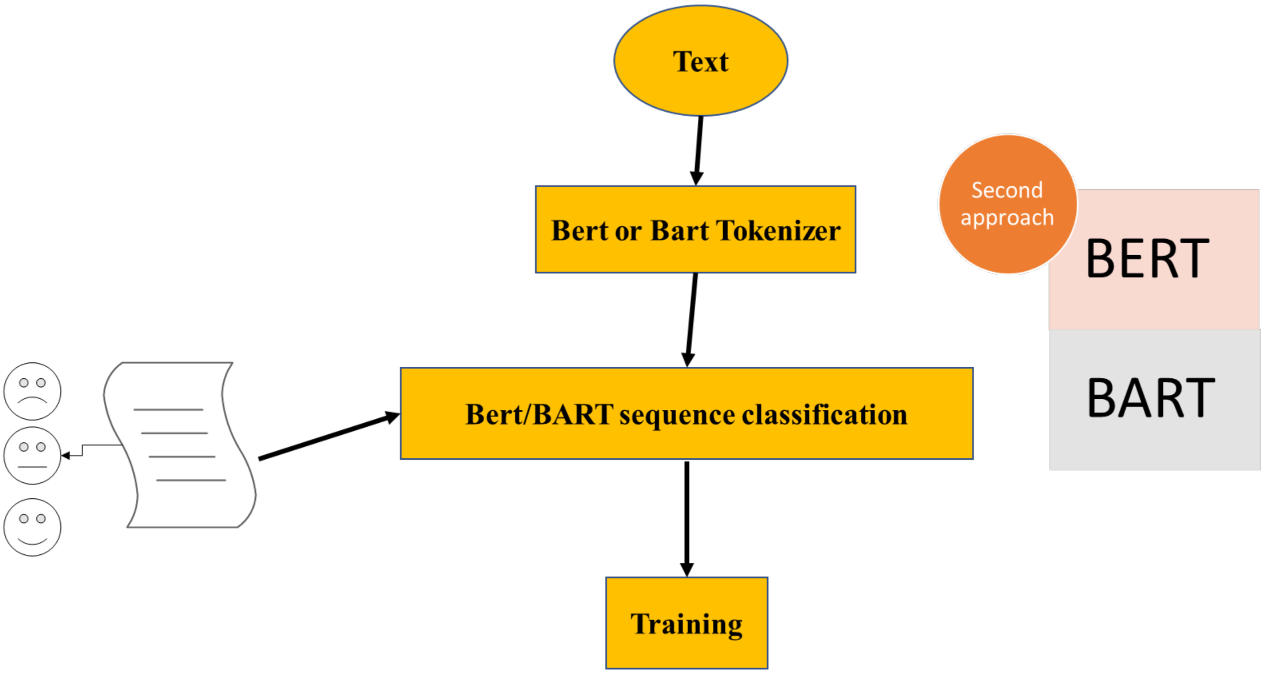

Figure 3-6: Second approach (TEXT to Sentiments by Language model)

3.4.3. Approach 3 - EEG to text to sentiment

This approach starts from EEG features as input to a GPT2 tokenizer, and the output is fed into the desired output, which is the sentiment labels, as shown in Figures 3-8.

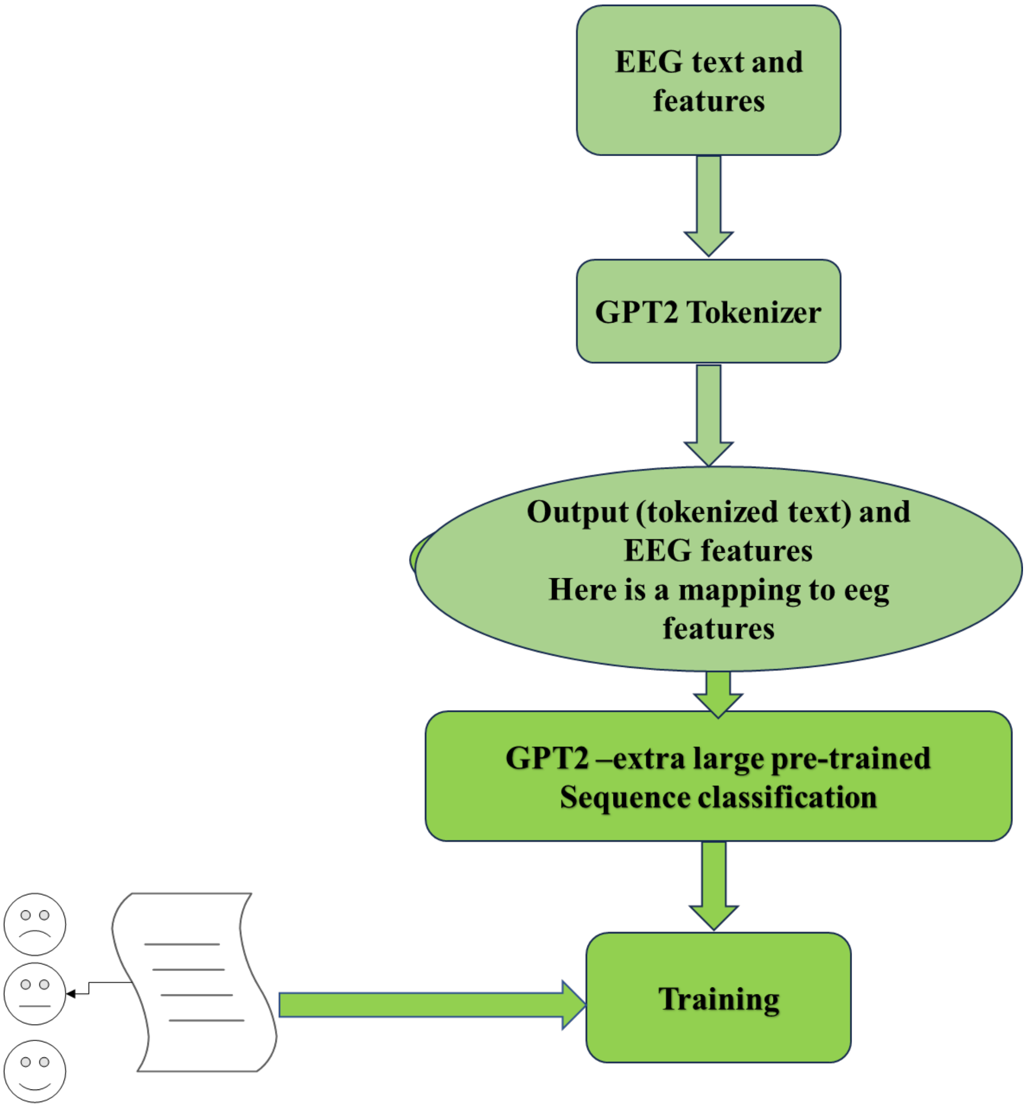

Figure 3-7: Third approach (EEG to Sentiments by GPT)

The unique sentiment analysis model, which begins with EEG features as input and utilizes a GPT-2 tokenizer for processing, represents an innovative fusion of physiological signals and natural language processing. This model marks a significant contribution to sentiment analysis by bridging the gap between physiological responses captured by EEG features and the nuanced understanding of sentiment provided by GPT-2.

The approach harnesses the contextual and semantic richness inherent in natural language by employing GPT-2, a powerful language model, as the tokenizer. This enables the model to interpret and process the transformed EEG data in a manner that goes beyond traditional signal analysis methods. The ability of GPT-2 to capture intricate language patterns ensures that the model can extract more nuanced sentiment information from the EEG input, thereby enhancing the depth and accuracy of sentiment predictions.

This approach is particularly noteworthy for its interdisciplinary nature, merging insights from neuroscience and natural language processing. The integration of EEG features allows the model to potentially capture subconscious or implicit emotional states that might not be overtly expressed in text. Consequently, your model offers a more holistic and comprehensive understanding of sentiment, making it well-suited for applications where both physiological and linguistic aspects contribute to the overall emotional context.

3.5. OGPTSC Tuning process

After explaining all the stages, some stages have hyper-parameters that are better for using a tuning method to choose their values. In our approach, suppose the hyper-parameter to be tuned is H and suppose the function to be maximized is the accuracy acc of training, so the fitness function is $1 - acc$; the tuning is shown in next equations:

$$\begin{gathered} L = 1 - acc \\ grad = L_{new} - L_{old} \\ H_{new} = H_{old} + \alpha * grad \end{gathered} \tag{3.3}$$

Where α is a constant term, it is the searching step; in this approach, it i$10^{a\ -4}$. L_{new} In the new loss calculated loss calculation, the gradient of loss is calculated at each step to monitor if the gradient is constant, then the model cannot be trained anymore; the ratio of grad is the change in the tuned parameter; if the grad is too large, then the tuned parameter will change with a high ratio, and if it is too small, the change of tuned parameter will be small. The loss is cross-entropy loss, as shown in equation (3.4).

$$\begin{gathered} loss = -\sum \log p(s) \qquad s{:}\,sentiment \\ p(s) = f(GPT2(h).W) \end{gathered} \tag{3.4}$$

$$f\ is\ activation\ funciton\ at\ last\ layer, h\ is\ the\ output\ of\ previous\ layer$$

$and\ W\ are\ the\ weights.$

The minus sign refers to log (1); it is normalized output. The tuned parameter was the learning rate, but it was tuned several times for different models and scenarios. The learning rate was tuned using the above equation.

At the start, a number generation needs to be found with the first value of the d parameter. This value is passed to the model, and the training stage occurs. The loss value is outputted from training to be compared with a small value (eps). These parameters are explained next.

There is no concern about loss when more prominent than 1, which could result from an imbalanced dataset. This means that some classes have significantly more samples than others, and the loss can be affected. The model might perform well on the majority class (resulting in high accuracy), but it may struggle with the minority class, leading to a higher loss for those samples.

The overall process is presented through the flow chart shown in Figure 3-8:

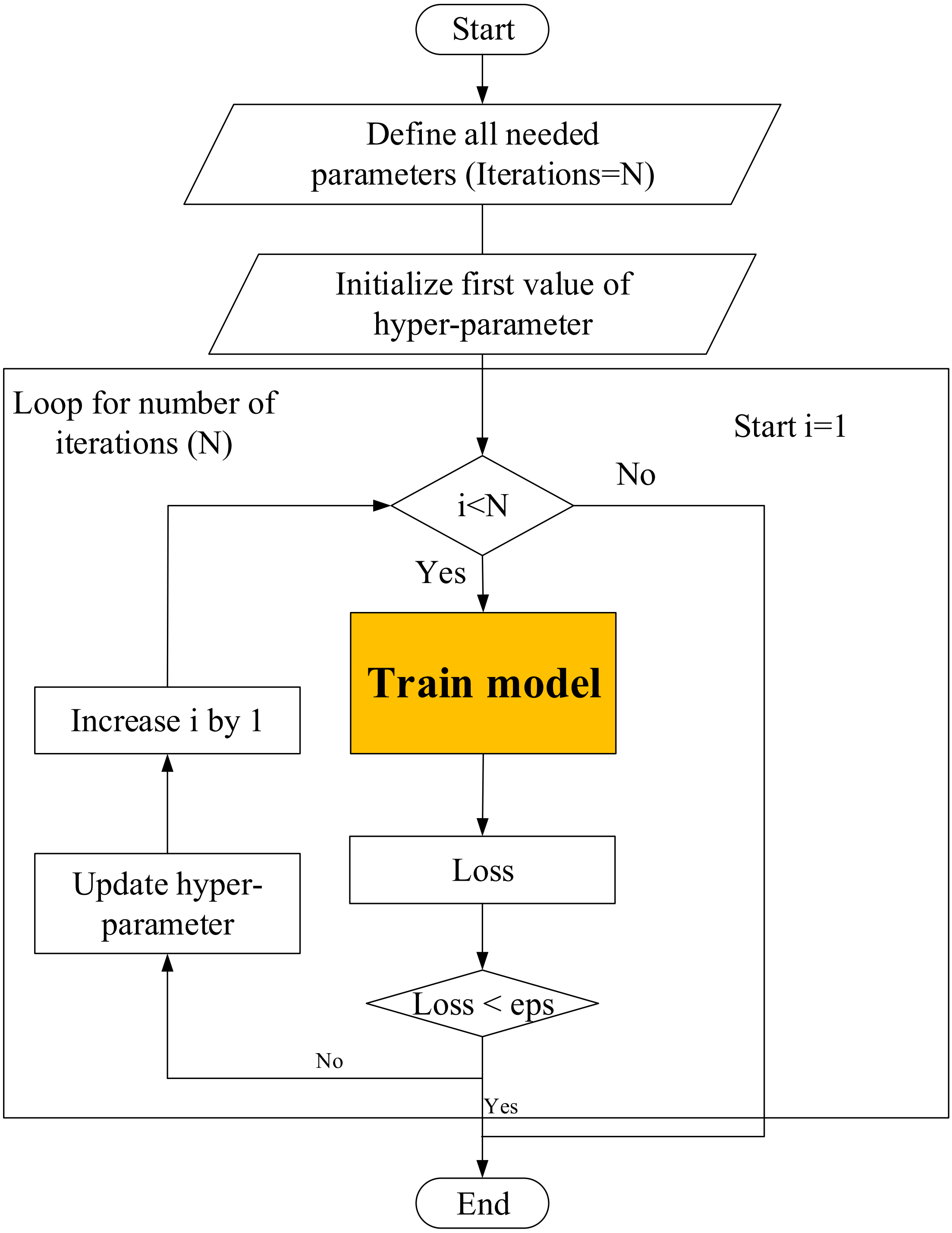

Figure 3-8: Flowchart of the tuning process

To apply the approach, a training model is shown in Figure 3-9

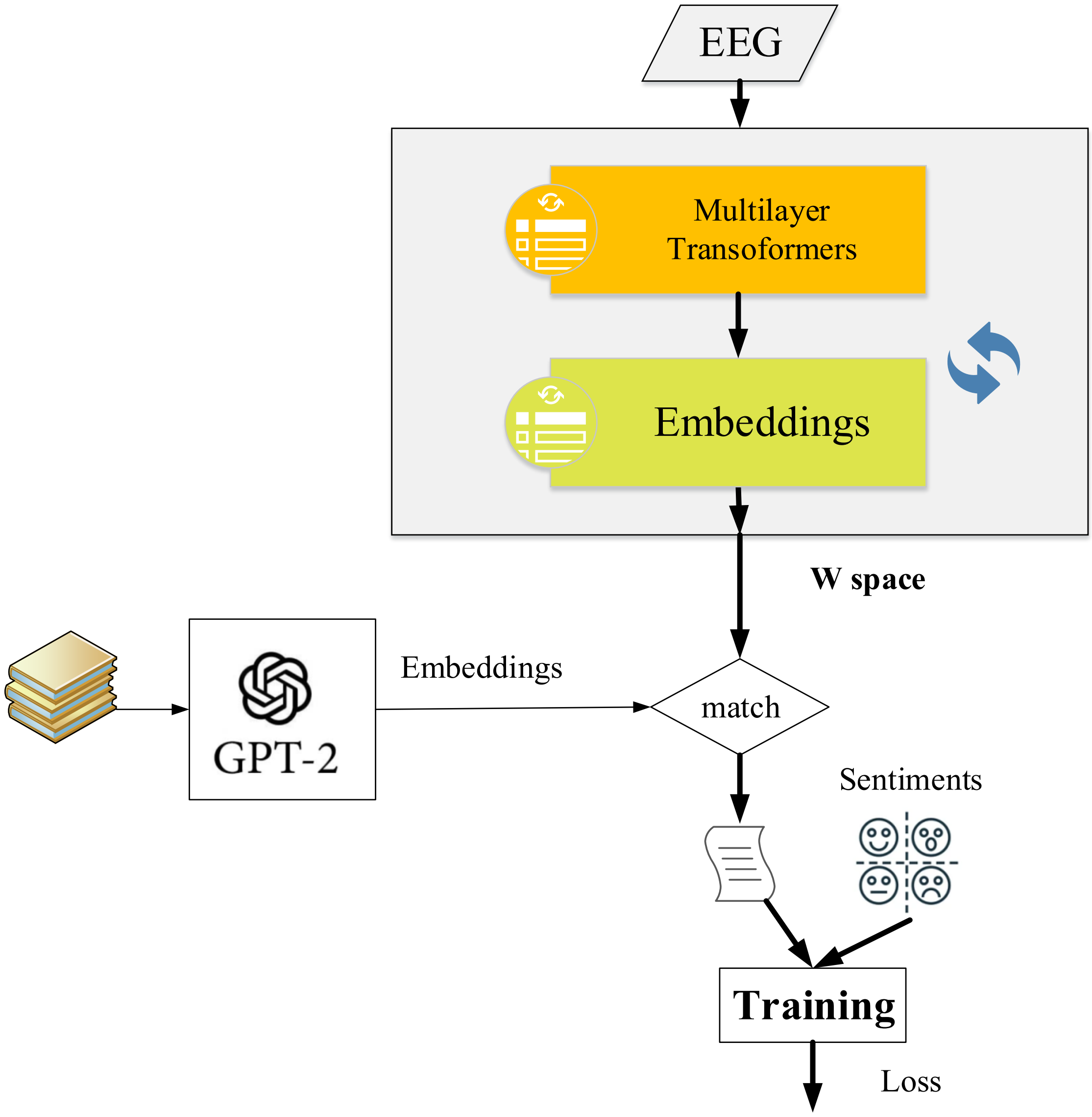

Figure 3-9: The train model

The procedure taken in this train model can be summarized through the following steps:

- First, all of the required parameters are to be initialized as follows:
  - N: number of iterations for the tuning process
  - Eps: very small value (optimally should be zero), but in general case, it will be like 0.001.
  - Batch size

- Model type: to handle EEG features
- The first value of the hyper-parameter to be tuned

- Define loop iterator I and test it each, comparing t ng to N.
  - If it is larger, then exit.
  - Else continue.
- Train model
- Get loss model
- Compare it with eps
  - If it is smaller, then exit
  - Else, continue to a new iteration of the loop
- Update hyper-parameter according to Eq.1
- Increase the iterator by 1
- Continue the loop until exit.

3.5.1. AdamW Optimizer

Optimizers were explained previously. Each training process needs an optimizer to minimize loss function and increase classification accuracy; Adam optimizer became the default method for training feed-forward and recurrent neural networks. Adam does not generalize as well as SGD with momentum when tested on diverse deep-learning tasks such as image classification, character-level language modeling, and constituency parsing. Adam lies in its dysfunctional implementation of weight decay. It limits the potential benefit of weight decay regularization because the weights do not decay multiplicatively. Adam might be outperformed by SGD with momentum because L2 regularization (a process that prevents Overfitting in the training process and improves the deep-learning Deep Learning models) is implemented in a suboptimal way in common deep-learning libraries. Adam leads to worse results than SGD with momentum (for which L2 regularization behaves as expected) [35][36].

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \nabla f_t(\theta_t) \qquad (3.5)$$

Where λ defines the rate of the weight decay per step and $\alpha \nabla f_t(\theta_t)$ is the t-th batch gradient to be multiplied by a learning rate α. For standard SGD, it is equivalent to standard L2 regularization. L2 regularization and weight decay regularization are equivalent to standard stochastic gradient descent when rescaled by the learning rate. However, this is not the case for Adam.

AdamW is a simple modification to recover the original formulation of weight decay regularization by decoupling the weight decay from the optimization steps taken for the loss function. AdamW decouples the optimal choice of weight decay factor from the learning rate setting for both standard SGD and Adam. It substantially improves Adam's generalization performance, allowing it to compete with SGD with momentum on image classification datasets. Many researchers have already adopted decoupled weight decay, and the community has implemented it in TensorFlow and PyTorch.

Adam generalizes substantially better with decoupled weight decay than with L2 regularization. This holds for various image recognition datasets (CIFAR-10 and ImageNet32x32); decoupled weight decay renders the optimal settings of the learning rate and the weight decay factor much more independent, easing hyperparameter optimization.

One fact that is often overlooked already for the simple case of SGD is that for the equivalence to hold, the L2 regularize λ' has to be set to $\lambda\ \alpha$, i.e., if there is an overall best weight decay value λ, the best value of λ' is tightly coupled with the learning rate α. In order to decouple the effects of these two hyperparameters, AdamW decouples the weight decay step [35][36].

Adam is different with L2 regularization in such a way that while with Adam, the sums of the gradient of the loss function and the gradient of the regularize (i.e., the L2 norm of the weights) are adapted, while with decoupled weight decay, only the gradients of the loss function are adapted (with the weight decay step separated from the adaptive gradient mechanism).

With L2 regularization, both types of gradients are normalized by their magnitudes, and therefore, weights x with large typical gradient magnitude s are regularized by a smaller relative amount than other weights. The learning rate of AdamW is tuned to achieve the best results, as shown in the next chapter.

4. Implementations and Results

4.1 Tools for Implementation

Python is a popular programming language for various tasks, including scientific computing, data analysis, and machine learning. Python can be run on a CPU, but running Python on a GPU can significantly accelerate performance for specific computationally intensive tasks, such as large-scale matrix operations or deep learning.

Python with GPU implementation typically involves using a library like TensorFlow, PyTorch, or CUDA to offload computationally intensive operations to the GPU. These libraries provide a high-level interface for users to define and train neural networks, which are then executed on the GPU.

Python with GPU was used for the implementation through this presented thesis, with a computer with a compatible GPU and the appropriate drivers installed.

The necessary libraries and frameworks are being installed to enable GPU acceleration. Using Python with GPU implementation can significantly speed up the training and execution of machine learning models, allowing the user to tackle more significant and complex problems than would be possible with CPU-based computation.

So, after introducing the tuning method, different models were tested and evaluated with this tuning method, which is explained next, and then a new approach to OGPTSC results was presented. All results are organized in a table and compared to related studies that worked on the same datasets.

4.1.1. <u>Simulation method</u>

The simulation method and coding were done in the terminal by calling Python scripts to be executed. The terminal printed the accuracies, and at the same time, they were plotted at the end. Needed libraries are installed. The terminal was used instead of guide and software because the terminal is faster. Object oriented was used, and many classes were to be called and used. The version of used libraries:

Cuda version: 11.8

Cudnn (The cuDNN library is a library optimized for CUDA containing GPU implementations): 8.2.4

Python version 3.10

Tensorflow version: 2. x

4.2 MLP for Sentiment Classification (SC)

This test was based on MLP, a Multi-Layer Perceptron baseline with three fully connected layers, ReLU activation, and one dropout layer. MLP is trained on aggregated sentence-level EEG Sentiment $< E, S >$ where E is features space and S is sentiments space. MLP is used to deal with EEG features.

4.2.1. Improve structure

There is no default structure for MLP; our unique structure has the following structure:

Fully connected (FC) layers were used because the model needs to learn complex nonlinear relationships between the input and output. However, the exact number of layers required and their sizes will depend on the problem's complexity and the dataset's size. This project used six FCs to balance capturing intricate patterns and preventing overfitting. The choice of six FC layers was based on experimentation and empirical evidence, ensuring sufficient capacity to capture the desired level of complexity without excessive model complexity.

Unlike the output layer, dropout layers are used after every FC layer to prevent overfitting. Dropout randomly sets a proportion of the input units to 0 during training, which helps to reduce the co-adaptation of units and forces the network to learn more robust features. Using a dropout rate of 0.2 after the first FC layer, 0.3 after the second FC layer, and 0.4 after the third and fourth FC layers.

Batch normalization (BN) layers are used after every FC layer to normalize the inputs to the activation function. BN reduces the impact of input changes to each layer and can speed up training and improve performance. The BN layer is used after every

FC layer except the last one, as the output layer does not require BN. Figure 4-1 shows the improved structure.

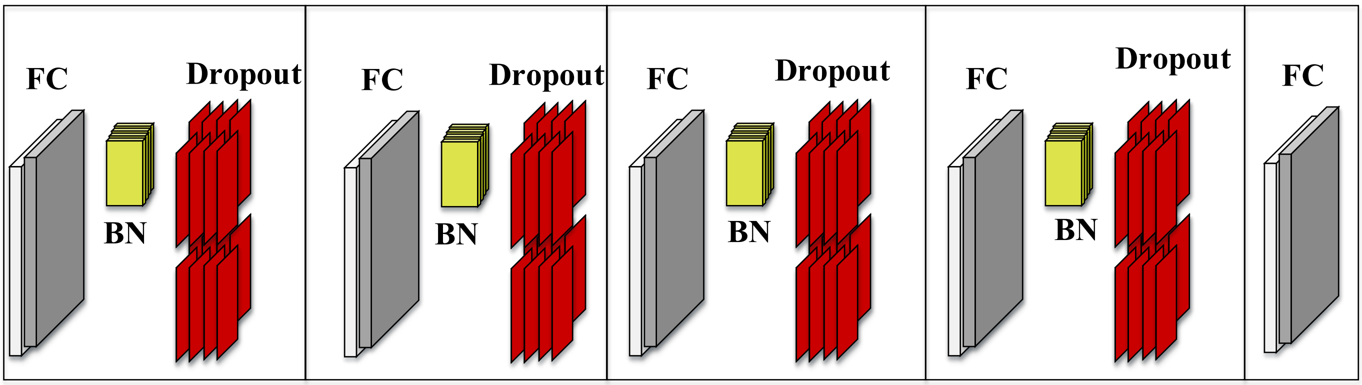

Figure 4-1: Suggested MLP structure

This structure depends on BN, which might have one or multiple dimensions; in our case, one dimension is required since the dataset is a one-dimensional type, and the last layer needs only fully connected layers. The changes in structure are adding BN after FC and increasing the number of FC layers.

In the initial configuration, the neural network consisted of three layers, each comprising fully connected (FC) units and dropout layers, each having 256 neurons. Subsequently, the network architecture was enhanced by expanding to five layers and incorporating batch normalization (BN). These five layers were structured with 512-512-1024-1024 neurons in sequence. Notably, the final layer was tailored to accommodate a specific number of neurons corresponding to the quantity of distinct labels in the classification task.

4.2.2. Test settings

The test settings, as initial, are given in Table 4-1.

Table 4-1: Settings for sentiment classification using MLP

Parameter	Value
Learning rate	0.00001
Batch size	64
Epochs	25
Input Dataset	ZuCo

- Test learning rate: it is shown in Table 4-2.

Table 4-2: Results of sentiment classification using MLP

Learning rate (batch size=32, epochs =25)	Entropy Loss	Accuracy
0.00001	1.100578	0.293
0.0001	1.094448	0.33
0.001	1.076439	0.445
0.01	1.079039	0.46
0.1	1.100433	0.42
0.05	**1.087933**	**0.47**

- Changing batch size: after selecting the Learning rate as 0.05, batch size was changed as follows [8, 16, 32, and 64], but the accuracy was not changed.
- Changing epochs also does not affect the accuracy.

The curve of accuracy is shown below:

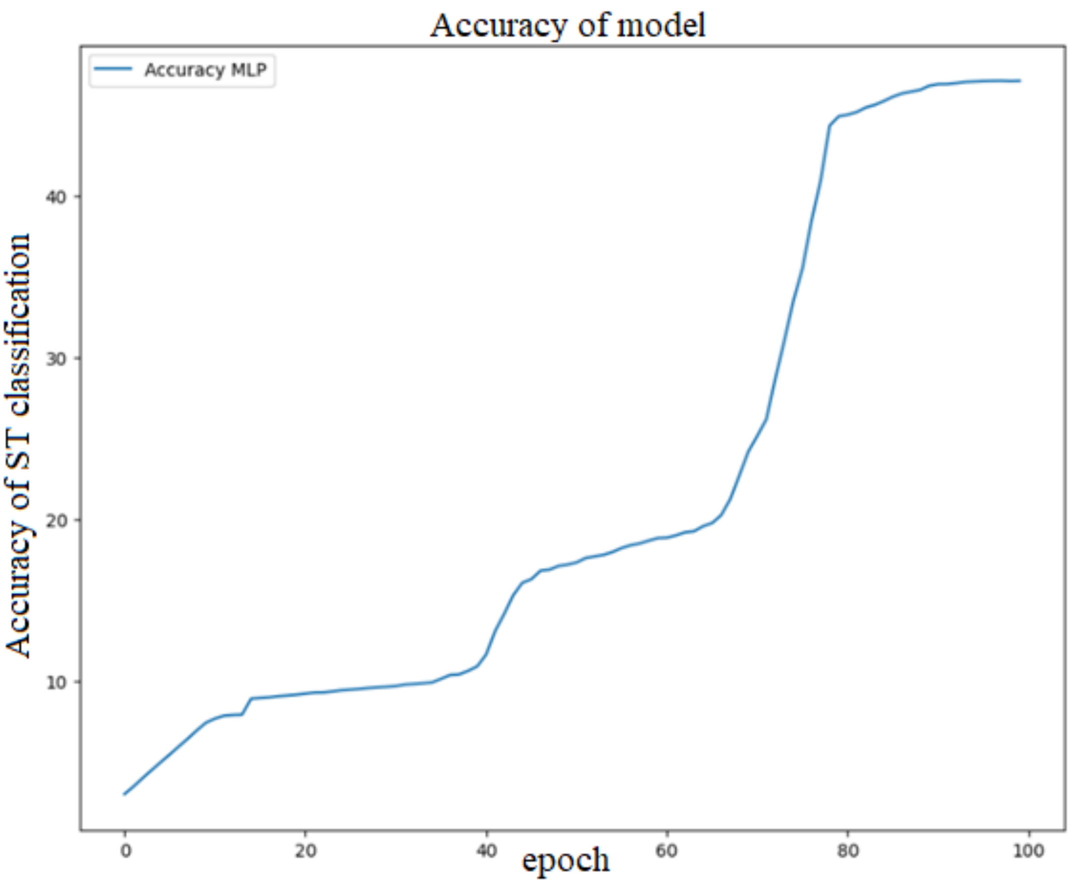

Figure 4-2: Accuracy of my MLP model

Using BN helped mitigate overfitting and solve some problems in first initializations. A specific number of layers, many layers with the type of data, might eliminate features, and a few layers won't result in valuable features.

4.3. Bi LSTM for Sentiment Classification (SC)

This test was based on LSTM. A Bi-directional-Long Short-Term Memory (BL-LSTM) baseline with four stacked LSTM layers and a single-layer classification head.

4.3.1. Improve structure

The chosen structure for this model was as follows:

- Layer to adapt input of EEG space to LSTM input (pack_padded_sequence)
- LSTM layer
- Output layer (pad_packed_sequence)

- Average layer

4.3.2. Test settings

The initial test setting is given in Table. 4-3.

Table 4-3: Settings for sentiment classification using LSTM

Parameter	Value
Learning rate	0.0001
Batch size	64
Epochs	25
Input Dataset	ZuCo

- The test learning rate is shown in Table 4-4:

Table 4-4: Results of sentiment classification using LSTM

Learning rate (batch size=32, epochs =20)	Entropy Loss	Accuracy
0.00005	1.084680	0.404
0.0001	1.072350	0.43
0.0005	1.065818	0.469
0.001	**1.068527**	**0.472**
0.005	1.064697	0.466
0.00075	1.065886	0.471

- Changing batch size: after selecting the Learning rate as 0.001, batch size was altered as follows [8, 16, 32, and 64], but the accuracy was not changed.
- Changing epochs also does not affect the accuracy.
- The curve of accuracy is shown below:

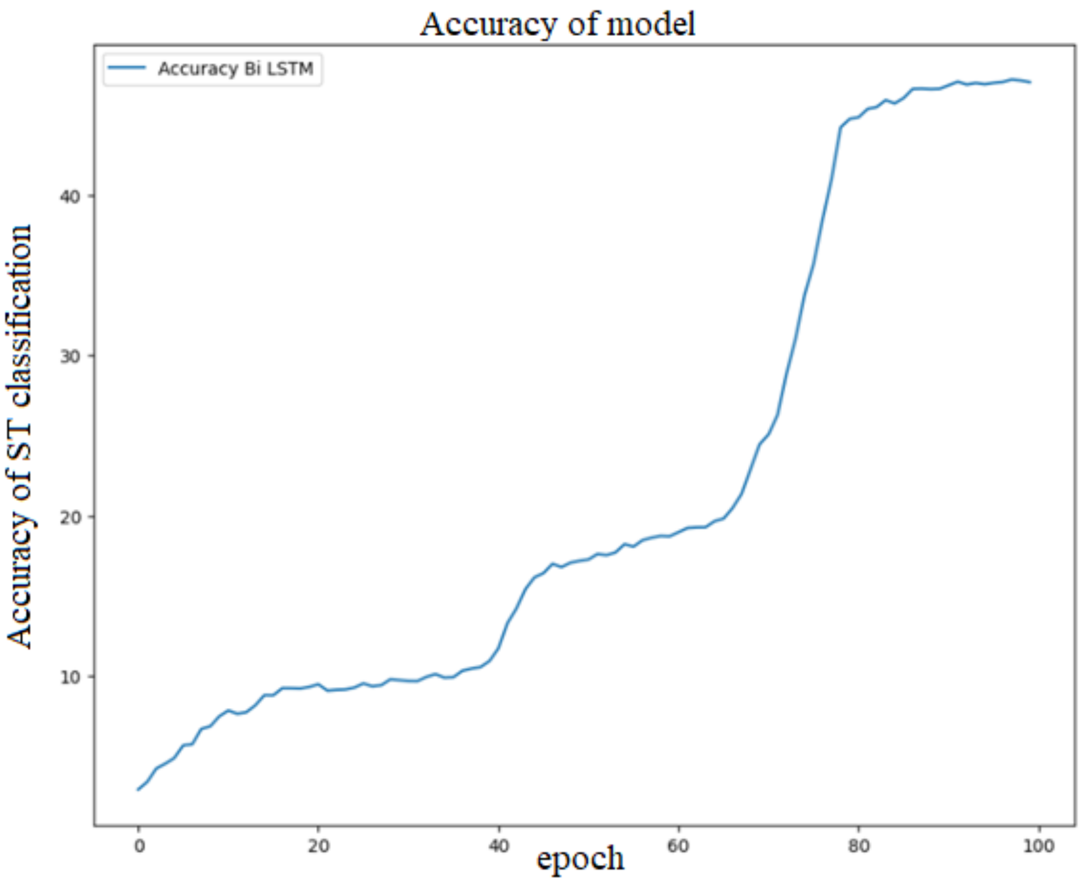

Figure 4-3: Accuracy of my Bi-LSTM model

4.4. BART for Sentiment Classification (SC)

This test was based on BART. BART is used to deal with EEG features.

4.4.1. Test settings

The test setting as initial is given in Table 4-5:

Table 4-5: Settings for sentiment classification using BART

Parameter	Value
Learning rate	0.0001
Batch size	8
Epochs	20
Input Dataset	SST

- The test learning rate is shown in Table 4-6:

Table 4-6: Results of sentiment classification using BART

Learning rate (batch size=32, epochs =25)	Entropy Loss	Accuracy
0.0001	0.412222	0.845
0.001	0.460060	0.80
0.0005	**0.632222**	**0.86**
0.00075	1.078325	0.416
0.0002	1.100286	0.34

The best value was 0.0005

- Changing batch size: after selecting the Learning rate as 0.0005, batch size was altered as follows [8, 16, 32, and 64], but the accuracy was not changed.
- Changing epochs also does not affect the accuracy.

The curve of accuracy is shown in Figure 4-4:

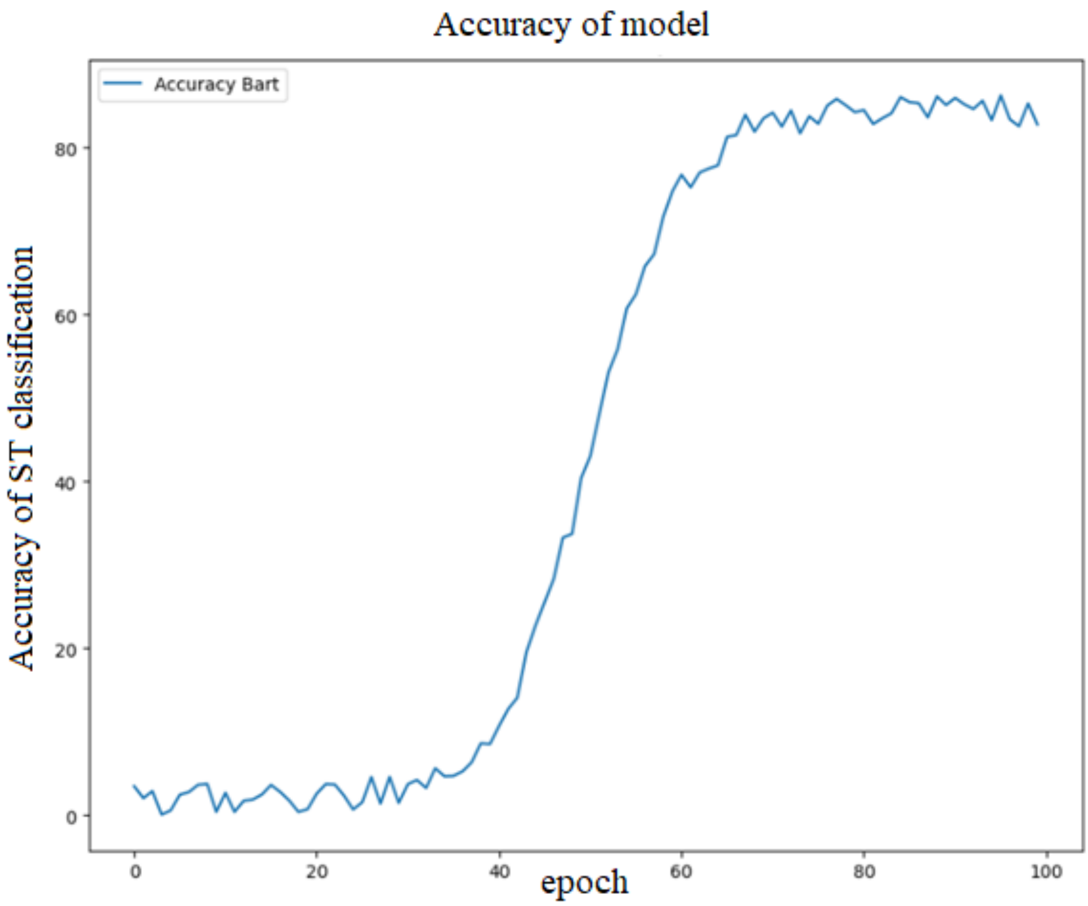

Figure 4-4: Accuracy of my model BART

4.5. BERT for Sentiment Classification (SC)

This test was based on BERT, which deals with EEG features.

4.5.1. <u>Test settings</u>

The test setting as initial is given in Table 4-7:

Table 4-7: Settings for sentiment classification using BERT

Parameter	Value
Learning rate	0.0001
Batch size	8
Epochs	20
Input Dataset	SST

- The test learning rate is given in Table 4-8:

Table 4-8: Results of sentiment classification using BERT

Learning rate (batch size=32, epochs =25)	Entropy Loss	Accuracy
0.0001	0.633474	0.737
0.001	**0.910009**	**0.78**
0.01	1.100596	0.34
0.0005	0.795847	0.778

The best value was 0.001

- Changing batch size: after selecting the Learning rate as 0.001, batch size was altered as follows [8, 16, 32, and 64], but the accuracy was not changed.
- Changing epochs also does not affect the accuracy.

The curve of accuracy is shown in Figure 4-5:

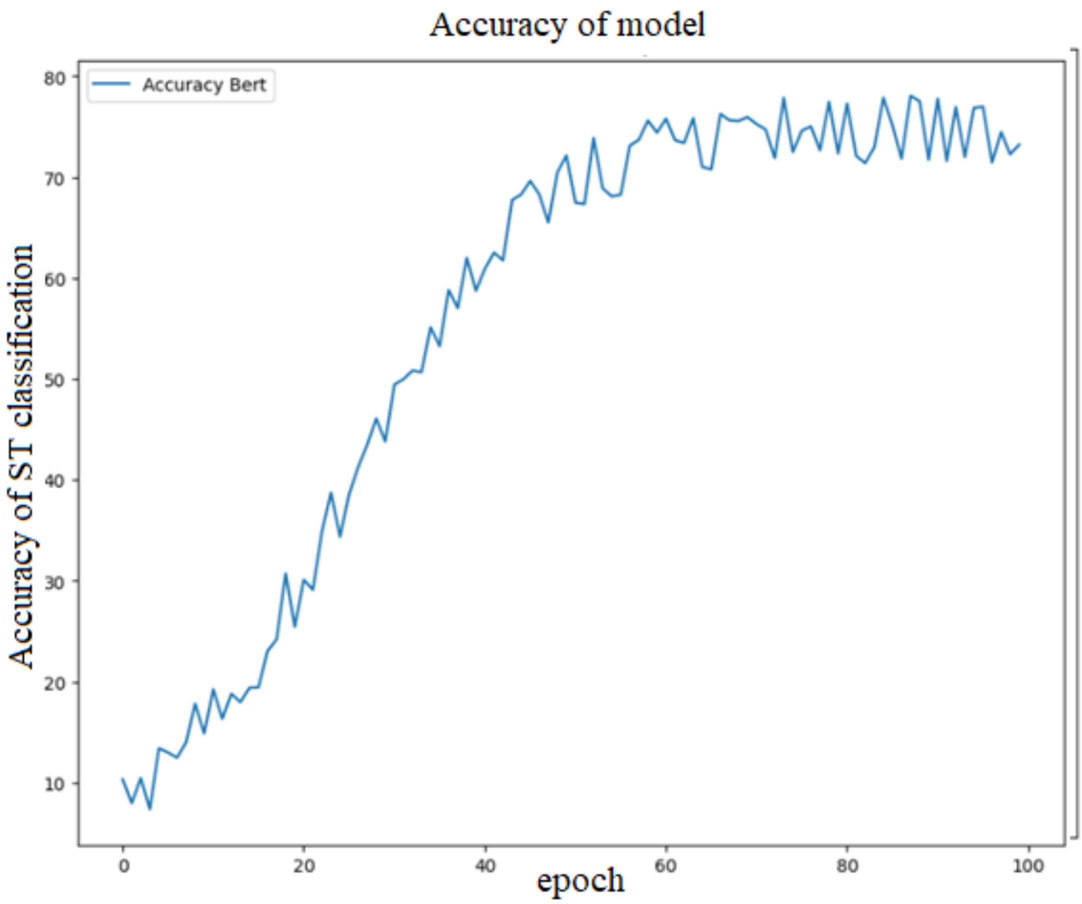

Figure 4-5: Accuracy of my model Bert

4.6. GPT-2 for Sentiment Classification (SC)

Previous models were tested and evaluated, and we tuned them by an optimization process to choose the learning rate. In addition, GPT was used as a new approach, replacing BERT and BART models with GPT and with the tuning process. The test setting as initial is given in Table 4-9:

Table 4-9: Settings for sentiment classification using GPT2

Parameter	Value
Learning rate	0.0001
Batch size	8
Epochs	20
Input Dataset	ZuCo

The test learning rate is given in Table 4-10:

Table 4-10: Results of sentiment classification using BERT

Learning rate (batch size=32, epochs =25)	Entropy Loss	Accuracy
0.0001	0.883346	0.677
0.001	0.940009	0.64
0.0003	0.822596	0.68
0.0006	**0.71027**	**0.73**

The best value was 0.0006

- Changing batch size: after selecting the Learning rate as 0.001, batch size was altered as follows [8, 16, 32, and 64], but the accuracy was not changed.
- Changing epochs also does not affect the accuracy.

The curve of accuracy when applying GPT is shown in Figure 4-6:

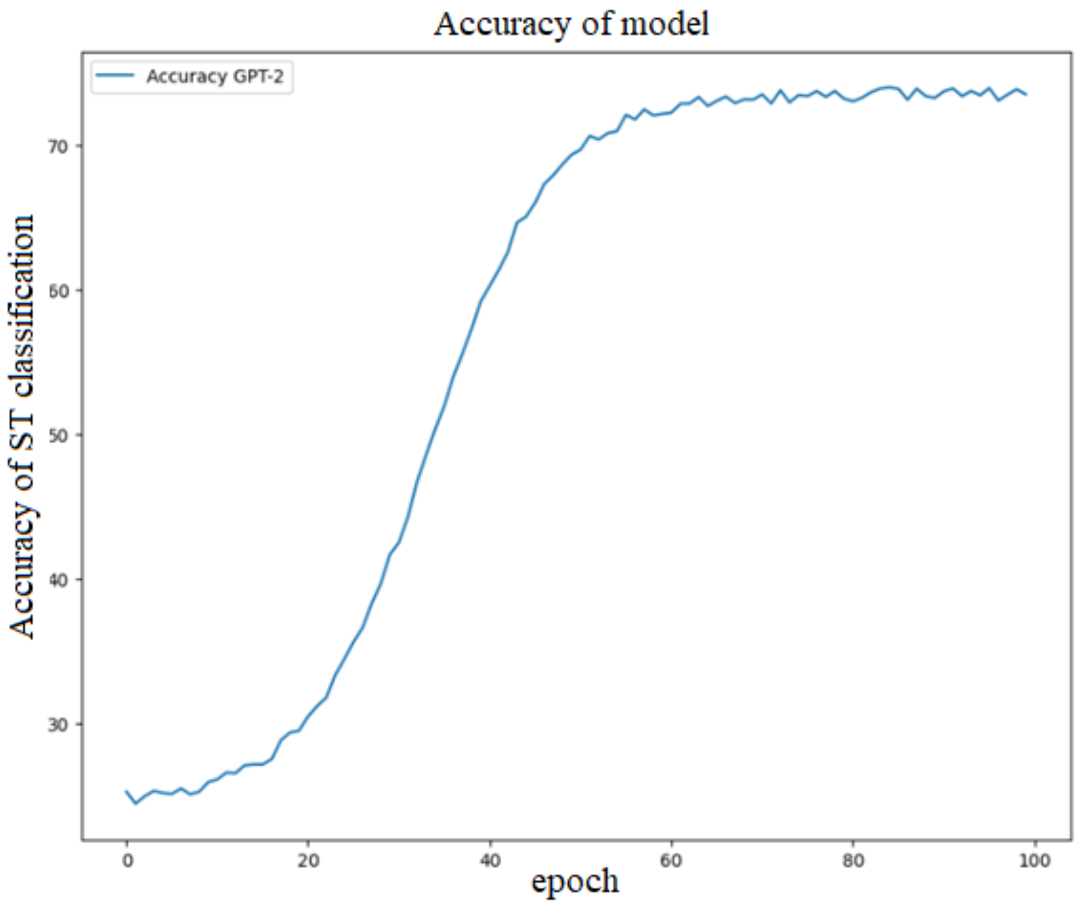

Figure 4-6: Accuracy of my model GPT-2

GPT achieved high accuracy when it was used as a tool and when mapping EEG features with text; this will add more features (not only text features). Additionally, the high space of GPT of words and strings

4.7. Results comparisons

Our results for all the above cases are compared against those achieved by Zha Wang et al. in [22] and Socher in [19], as shown in Table 4-11, The firms that in enhancements and approaches outperformed state-of-the-art in these fields and those being done on the same dataset. The main improvements can be summarized as follows:

- Using GPT-2 as a language model in the field of EEG.
- Improve the structure of MLP with the addition of optimizing and tuning processes.

- Improve the structure of Bi-LSTM with the addition of optimizing and processing.
- Tuning process of BERT and BART models

Table 4-11: Results Comparison

	Flow	**Approach**	**Accuracy**		
			Contributions	Wang et al (2022) [22]	Socher et al (2013) [19]
MLP	EEG to sentiments	Approach 1 (Tuning and new structure of MLP)	0.47	0.318	—
LSTM			0.472	0.309	
GPT2	EEG to text analysis of sentiments	Approach 3 (new and tuning)	0.739	0.553 (EEG to text using BART model to sentiment)	
BART	Text to sentiments	Approach 2 (tuning)	0.86	0.797	0.807 (text to word vector to sentiment)
BERT			0.78	0.75	

The calculated accuracy is the accuracy from the model itself; other accuracy metrics can be calculated, such as recall, precision, and f1-score. To define them, we need to define the following terms:

Consider a scenario with a designated target label, denoted as "1," and a predicted label.

True Positive (TP): The model correctly predicts the target label; the target and predicted are equals (1).

False Positive (FP): The model incorrectly predicts the target label; predicted label=1, while the target label does not equal 1.

Let us now consider a scenario with a designated target label, "0," and a predicted label.

True Negative (TN): The model correctly predicts the target label; the target and predicted are equal (0).

False Negative (FN): The model incorrectly predicts the target label; predicted label=0 while the target label is not equal 0

Suppose positive sentiment is 1, then negative and neutral sentiment will be 0 for the calculations of these four terms.

Precision: Precision measures the accuracy of the positive predictions made by the model. It quantifies the proportion of accurate positive predictions among all optimistic predictions made by the model.

$$Precision = \frac{TP}{TP + FP}$$

Recall: also known as sensitivity, is the proportion of actual positive cases the model correctly identifies. It is similar to the true positive rate.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: is the harmonic mean of precision and recall. It balances precision and recall, especially when there is an imbalance between positive and negative cases.

$$F1_{Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Table 4-12 represents the accuracy metrics precision, recall, and f1-score of our approaches:

Table 4-12: precision, recall and f1-score of all approaches

Method	Accuracy	Precision	Recall	F1-score
MLP (3 classes)	0.47	0.471 0.472 0.46	0.468 0.471 0.462	0.4694 0.4715 0.4649
LSTM (3 classes)	0.472	0.47 0.47 0.469	0.475 0.466 0.461	0.4734 0.4684 0.4649
GPT (3 classes)	0.739	0.738 0.736 0.735	0.738 0.735 0.735	0.738 0.735 0.735

BART (5 classes)	0.86	0.853 0.855 0.866 0.862 0.861	0.843 0.855 0.863 0.861 0.855	0.8479 0.855 0.8645 0.8615 0.8579
BERT (5 classes)	0.78	0.767 0.777 0.769 0.768 0.754	0.777 0.767 0.779 0.766 0.761	0.7719 0.7719 0.7739 0.7669 0.7574

Precision measures how well the model avoids false positives. Recall measures how well the model captures all actual positive instances. The F1-Score balances precision and recall and is useful when we want a single metric that considers both false positives and false negatives.

Explanation of the results:

1. Using GPT-2 as a Language Model in EEG Analysis:

GPT-2, a widely recognized and influential language model, was initially designed for natural language processing tasks. However, its application in EEG analysis represents a groundbreaking approach. By leveraging its transformer-based architecture, GPT-2 can be fine-tuned to handle EEG data and perform sentiment classification tasks.

The high accuracy achieved with GPT-2 in EEG analysis can be attributed to several factors:

- Representation Learning: GPT-2's transformer architecture excels in learning hierarchical representations from sequential data, making it adept at capturing complex patterns in EEG signals. This enables the model to extract relevant features crucial for sentiment classification.
- Large Pre-trained Model: GPT-2 is pre-trained on a vast corpus of diverse text data, providing a rich understanding of language and context. This pre-training significantly boosts its ability to recognize and interpret EEG data.
- Transfer Learning: The ability to fine-tune GPT-2 for EEG analysis through transfer learning allows the model to adapt its knowledge to the specific domain.

This knowledge transfer from pre-training to the EEG sentiment classification task enhances its performance and accuracy.

- Contextual Information: GPT-2's transformer-based approach captures contextual information from the EEG data, enabling it to grasp the temporal dependencies and correlations within the sequences. This contextual understanding contributes to better sentiment classification.

2. Improving MLP Structure with Optimization and Tuning:

The Multilayer Perceptron (MLP) is a simple yet powerful neural network architecture. Enhancing its structure and applying optimization and tuning techniques substantially improves its EEG sentiment classification performance.

- Enhanced Capacity: Optimizing the structure of the MLP involves increasing the number of layers and neurons or using more advanced activation functions. This increases the model's capacity to learn complex patterns, improving accuracy.
- Regularization: To prevent overfitting, the tuning process incorporates regularization techniques such as dropout and weight decay. This ensures that the MLP generalizes well to unseen EEG data, contributing to its high accuracy.
- Hyperparameter Tuning: Tuning hyperparameters like learning rates, batch sizes, and optimization algorithms help the MLP converge efficiently and find better local minima in the loss landscape. This fine-tuning optimizes its performance.

3. Improving Bi-LSTM Structure with Optimization and Tuning:

Bidirectional Long Short-Term Memory (Bi-LSTM) is a popular recurrent neural network (RNN) variant well-suited for sequence-to-sequence tasks like EEG analysis. Improving its structure and applying optimization and tuning techniques lead to increased accuracy in sentiment classification.

- Bidirectional Context: Bi-LSTM captures both forward and backward temporal dependencies in the EEG data, providing a more comprehensive context for sentiment classification. This bidirectional information enables the model to make more informed decisions.

- Parameter Tuning: Adjusting the number of LSTM layers, hidden units, and dropout rates during the tuning process allows the Bi-LSTM to find the optimal configuration for the EEG sentiment classification task.
- Gradient Clipping: During training, gradient clipping can be applied to prevent exploding gradients, ensuring smoother convergence and better generalization.

4. Tuning Process for BERT and BART Models:

BERT (Bidirectional Encoder Representations from Transformers) and BART (Bidirectional and Auto-Regressive Transformers) are state-of-the-art transformer-based models widely used in natural language processing tasks. Fine-tuning them for EEG sentiment classification yields remarkable accuracy.

- Domain Adaptation: The tuning process adapts BERT and BART models to the EEG sentiment classification domain, enabling them to learn EEG-specific patterns and nuances.
- Task-specific Head: Adding a task-specific classification head on top of the pre-trained BERT and BART models can be tailored to the sentiment classification task, leading to higher accuracy.
- Learning Rate Schedules: Adjusting learning rate schedules during tuning can enhance the models' convergence and stabilize training, improving sentiment classification performance.

In summary, the high accuracy achieved in these approaches can be attributed to the combination of powerful pre-trained models, transfer learning, optimized model architectures, and fine-tuning techniques. These elements collectively equip the models to handle EEG data, extract relevant features, and make accurate sentiment classifications. The research's success lies in the seamless integration of advanced techniques from natural language processing and EEG analysis, culminating in state-of-the-art results for sentiment classification.

5. Conclusions and Future Works

5.1. Conclusion

In this research, we embarked on an ambitious quest to unravel the mysteries of EEG features and their representation in language. Our efforts culminated in a comprehensive and illuminating study, unlocking the potential of linguistic models in EEG analysis. With meticulous precision, we delved into the intricacies of EEG feature representation in Word space, a novel approach that bridges the gap between brain signals and linguistic comprehension. Our exploration led us to examine various linguistic models, carefully selecting and implementing them to harness their strengths in the EEG context. At the heart of our innovation lies the cutting-edge "Optimized Chat Generative Pre-training EEG to Sentiment Classification (OGPTSC)" model. OGPTSC emerged as a true trailblazer through rigorous development and relative optimization, setting new standards in EEG-based sentiment classification. Yet, we did not stop there. Famed models such as BERT, BART, and Multilayer Perception were put through meticulous tuning processes, extracting their latent potential and identifying the optimal learning rates for maximum performance.

Determined to leave no stone unturned, we subjected each model to rigorous testing, evaluating their prowess in processing and representing EEG features. These linguistic models honed through our dedication, served as the bedrock for sentiment predictions and estimations, ushering in a new era of EEG-driven sentiment analysis. Our research journey encompassed not one but two diverse datasets, providing a robust and holistic evaluation of the efficacy of our methodology. Our approach proved its mettle in the crucible of real-world EEG data, delivering results that surpassed related studies on the same datasets.

In light of the far-reaching implications of EEG technology, our work emerges as an indispensable guide for harnessing its power across various fields. From healthcare to human-computer interaction, emotion recognition, and beyond, our research equips the scientific community with the tools to decode EEG signals and unleash their potential. A cornerstone of our success lies in the simplicity and elegance of our tuning process, which remarkably enhanced the overall performance of our approach. This pragmatic and effective technique exemplifies the fusion of innovative thinking and practicality in pursuit of excellence.

In closing, this work symbolizes a transformative milestone in neuroscience and natural language processing. Our exploration, experimentation, and innovation journey has brought a paradigm shift in EEG analysis and sentiment classification. With a newfound understanding of EEG features in linguistic space and the power to wield any linguistic model, we stand at the forefront of a new era in the convergence of brain signals and language. As we forge ahead, the impact of our research resonates across disciplines, igniting the torch of progress and inspiring future breakthroughs in the fascinating world of EEG-driven technologies.

5.2.Future works

This work can be updated in the future for real-time applications. For instance, it can be used for those who face significant difficulty in digesting and understanding the meaning of words through other's speech by recognition of the sentiments of the speech. In addition, it can be updated by using different methods or improving the existing techniques and mathematical algorithms.

References

[1] Olejniczak, P. (2006). Neurophysiologic basis of EEG. Journal of clinical neurophysiology, 23(3), 186-189.

[2] Aggarwal, S., & Chugh, N. (2022). Review of machine learning techniques for EEG based brain computer interface. Archives of Computational Methods in Engineering, 1-20.

[3] Socher, R., Bauer, J., Manning, C. D., & Ng, A. Y. (2013, August). Parsing with compositional vector grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 455-465).

[4] Hollenstein, N., Troendle, M., Zhang, C., & Langer, N. (2019). ZuCo 2.0: A dataset of physiological recordings during natural reading and annotation. arXiv preprint arXiv:1912.00903.

[5] Roy, S., Hossain, S. I., Akhand, M. A. H., & Siddique, N. (2018, December). Sequence modeling for intelligent typing assistant with Bangla and English keyboard. In 2018 International Conference on Innovation in Engineering and Technology (ICIET) (pp. 1-6). IEEE.

[6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2016). Bert: Bidirectional encoder representations from transformers.

[7] Mohammed, A. H., & Ali, A. H. (2021, July). Survey of BERT (bidirectional encoder representation transformer) types. In *Journal of Physics: Conference Series* (Vol. 1963, No. 1, p. 012173). IOP Publishing.

[8] Kang, T., Lee, H., Choe, B., & Jung, K. (2021, July). Entangled bidirectional encoder to autoregressive decoder for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1657-1661).

[9] Alokla, A., Gad, W., Nazih, W., Aref, M., & Salem, A. B. (2022). Pseudocode Generation from Source Code Using the BART Model. *Mathematics*, *10*(21), 3967.

[10] Learning, D. (2020). Deep learning. High-dimensional fuzzy clustering.

[11] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[12] Berry, M. W., Mohamed, A., & Yap, B. W. (Eds.). (2019). Supervised and unsupervised learning for data science. Springer Nature.

[13] Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. National science review, 5(1), 44-53.

[14] James, G., Witten, D., Hastie, T., Tibshirani, R., James, G., Witten, D., ... & Tibshirani, R. (2021). Unsupervised learning. An introduction to statistical learning: with applications in R, 497-552.

[15] Ide, H. and Kurita, T., 2017, May. Improvement of learning for CNN with ReLU activation by sparse regularization. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2684-2691). IEEE

[16] Sainath, T.N., Vinyals, O., Senior, A. and Sak, H., 2015, April. Convolutional, long short-term memory fully connected deep neural networks. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4580-4584). IEEE

[17] Ketkar, N., 2017. Introduction to keras. In Deep learning with Python (pp. 97-111). Apress, Berkeley, CA.

[18] Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization wiconcerningating scales. arXiv preprint cs/0506075.

[19] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642).

[20] Bird, J. J., Ekart, A., Buckingham, C. D., & Faria, D. R. (2019, April). Mental emotional sentiment classification with an eeg-based brain-machine interface. In *Proceedings of the International Conference on Digital Image and Signal Processing (DISP'19)*.

[21] Zhang, X., Yao, L., Sheng, Q. Z., Kanhere, S. S., Gu, T., & Zhang, D. (2018, March). Converting your thoughts to texts: Enabling brain typing via deep feature learning of EEG signals. In *2018 IEEE international conference on pervasive computing and communications (PerCom)* (pp. 1-10). IEEE.

[22] Wang, Z., & Ji, H. (2022, June). Open vocabulary electroencephalography-to-text decoding and zero-shot sentiment classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 36, No. 5, pp. 5350-5358).

[23] Ashraf Kiyani, I., & Razaq, A. (2022). A Comprehensive Review on Sentiment Perception Using Electroencephalography (EEG). SN Computer Science, 3(3), 245.

[24] Zhu, F., Wang, J., Ding, W., Xie, T., & Han, Q. (2023, February). Sentiment recognition model of EEG signals combined with one-dimensional convolution and BiBASRU-AT. In Journal of Physics: Conference Series (Vol. 2425, No. 1, p. 012020). IOP Publishing.

[25] Hole, K., & Anand, D. (2022, October). Detection of stress and emotion recognition using EEG signal. In AIP Conference Proceedings (Vol. 2555, No. 1, p. 050007). AIP Publishing LLC.

[26] Makin, J. G.; Moses, D. A.; and Chang, E. F. 2020. Machine translation of cortical activity to text with an encoder–decoder framework. Nature Neuroscience, 23(4): 575–582

[27] Anumanchipalli, G. K.; Chartier, J.; and Chang, E. F. 2019. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753): 493–498.

[28] Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880

[29] Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. Advances in Neural Information Processing Systems

[30] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020.*

[31] Goh, G.; Cammarata, N.; Voss, C.; Carter, S.; Petrov, M.; Schubert, L.; Radford, A.; and Olah, C. 2021. Multimodal neurons in artificial neural networks. *Distill*, 6(3): e30

[32] Zheng, H., Wang, G., & Li, X. (2022). Swin-MLP: A strawberry appearance quality identification method by Swin Transformer and multi-layer perceptron. Journal of Food Measurement and Characterization, 16(4), 2789-2800.

[33] Jiang, W., Tang, H., Zhao, Y., Liu, Z., & Yang, C. (2019, July). Distribution transformers connectivity recognition based on multi-layer perceptron classifier. In 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) (pp. 1482-1486). IEEE.

[34] Alsaadi, H. S., Hedjam, R., Touzene, A., & Abdessalem, A. (2020, February). Fast binary network intrusion detection based on matched filter optimization. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT) (pp. 195-199). IEEE.

[35] Pandey, R., & Singh, J. P. (2023). BERT-LSTM model for sarcasm detection in code-mixed social media post. Journal of Intelligent Information Systems, 60(1), 235-254.

[36] Wright, L., & Demeure, N. (2021). Ranger21: a synergistic deep learning optimizer. arXiv preprint arXiv:2106.13731.

[37] Chen, X., Liang, C., Huang, D., Real, E., Liu, Y., Wang, K. & Le, Q. V. (2022). Evolved Optimizer for Vision. In First Conference on Automated Machine Learning (Late-Breaking Workshop).