Neurocognitive Modeling for Text Generation: Deep Learning Architecture for EEG Data

Khushiyant
Department of Computer Science , University of Freiburg
khushiyant.khushiyant@email.uni-freiburg.de

Abstract

Text generating capabilities have undergone a substantial transformation with the introduction of large language models (LLMs). Electroencephalography (EEG)-based text production is still difficult, though, because it requires a lot of data and processing power. This paper introduces a new method that combines the use of the Gemma 2B LLM with a classifier-LLM architecture to incorporate a Recurrent Neural Network (RNN) encoder. Our approach drastically lowers the amount of data and compute power needed while achieving performance close to that of cutting-edge methods. Notably, compared to current methodologies, our methodology delivers an overall performance improvement of 10%. The suggested architecture demonstrates the possibility of effective transfer learning for EEG-based text production, remaining strong and functional even in the face of data limits. This work highlights the potential of integrating LLMs with EEG decoding to improve assistive technologies and improve independence and communication for those with severe motor limitations. Our method pushes the limits of present capabilities and opens new paths for research and application in brain-computer interfaces by efficiently using the strengths of pre-trained language models. This makes EEG-based text production more accessible and efficient.

Keywords— EEG, Text generation, Gemma, Brain-Computer Interface

Introduction

Electroencephalography (EEG)-based brain-computer interfaces (BCIs) hold significant promise for decoding neural activity to drive various assistive technologies. By translating brain signals into control commands, BCIs aim to restore communication and independence for individuals with severe motor disabilities. However, existing systems predominantly focus on simple word spelling applications with limited vocabulary or sentence construction capabilities (McFarland and Wolpaw 2011; Nicolas-Alonso and Gomez-Gil 2012). Progressing BCI technology towards generating freeform text could significantly expand the autonomy and self-expression currently afforded to users (Biswal et al. 2019).

Recent advances in natural language processing (NLP), particularly the emergence of powerful large language models (LLMs) like GPT-3, present new opportunities for this endeavor. LLMs demonstrate an unprecedented capacity to produce human-like text based on a given prompt (Vaswani et al. 2017). Their excellent language generation skills hold untapped potential when conditioned on additional modalities like EEG input. However, developing methods to effectively fuse LLM capabilities with EEG decoding remains an open research challenge (Hochreiter and Schmidhuber 1997; Lawhern et al. 2018).

Despite these advancements, EEG-based text generation remains challenging due to its data-intensive nature and the inherent variability of brain signals (Hu et al. 2021). Current approaches often require extensive datasets and computational resources, limiting their practical applicability (Garrett et al. 2003). Additionally, there is a lack of efficient methods for leveraging pre-trained language models in the context of EEG-based text generation (Jo et al. 2024).

Towards this goal, this paper presents a novel framework utilizing a Recurrent Neural Network (RNN) encoder to extract features from raw EEG signals. These compact representations are then input to a classifier-LLM pipeline, with the Gemma 2B model serving as the LLM module (Gemma Team et al. 2024). This approach bypasses extensive end-to-end training to directly leverage the strengths of pre-trained language expertise. Evaluations demonstrate competitive text generation from EEG using orders of magnitude less data than existing methods. The proposed techniques exemplify efficient transfer learning, unlocking previously unsuitable datasets by targeted feature extraction and specialty model integration.

Recent Work

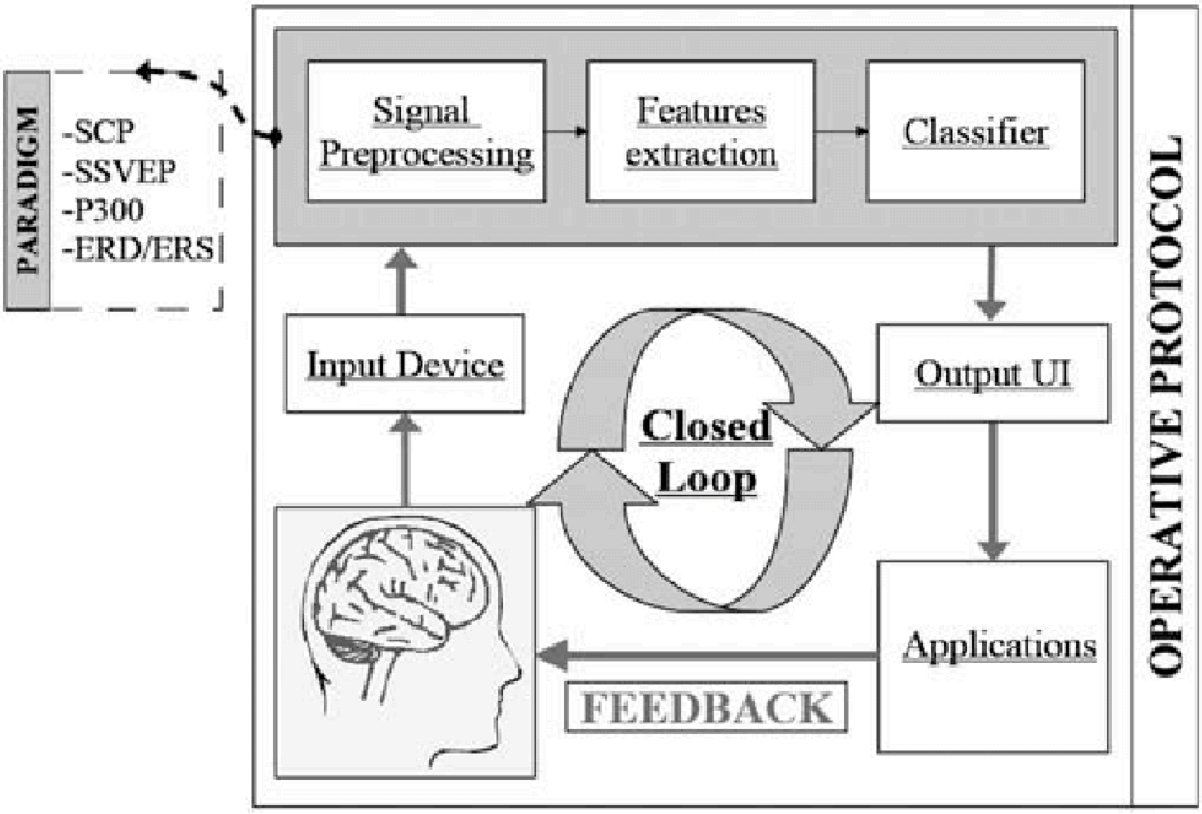


Figure 1: BCI System Architecture

While directly decoding EEG signals for conditional text generation remains an emerging field, advancements in Brain-Computer Interface (BCI) systems offer promising pathways for exploration. A recent study in the Journal of Neural Engineering investigated utilizing a BCI speller to enable basic text generation process (depicted in Fig. 1). The research developed a BCI system that decoded event-related potentials in the brain to identify letter selections on a virtual keyboard. Participants achieved an average spelling rate of 6.4 characters per minute with 81.3% accuracy. This demonstrates the feasibility of using BCIs for fundamental text generation capabilities (Nicolas-Alonso and Gomez-Gil 2012; McFarland and Wolpaw 2011)

However, limitations exist regarding spelling speed and sample size that warrant further investigation before translational viability. The study acknowledges that advancing decoder algorithms, integrating language models, and longitudinal assessments with larger groups will be critical future directions.

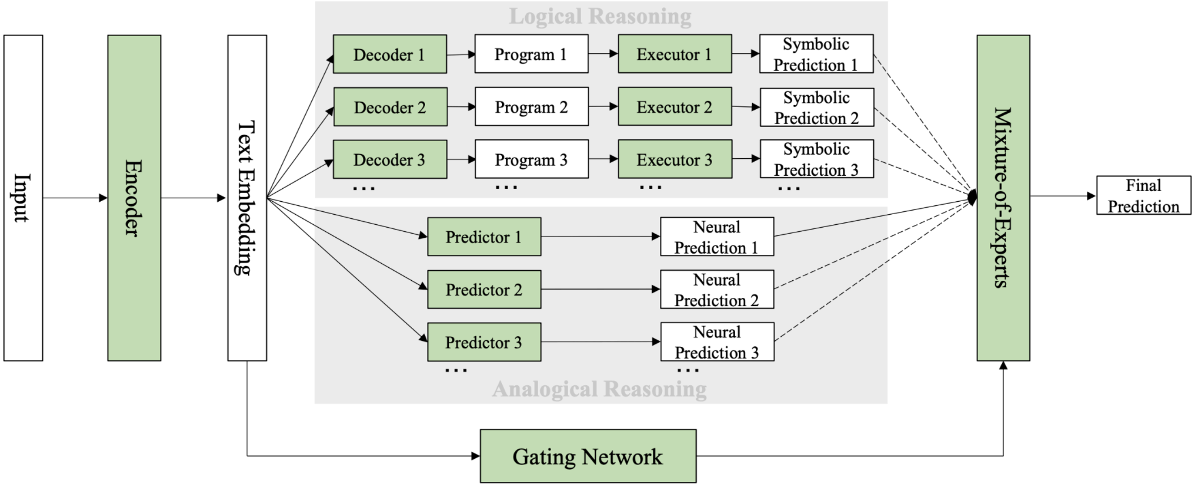


Figure 2: Overview of the Neural-Symbolic Processor framework

A recently proposed novel framework called Neural-Symbolic Processor (NSP) is aimed at improving natural language understanding when logical reasoning is required (as seen in Fig. 2). NSP employs both neural analogical reasoning using deep learning models like BERT and RoBERTa as well as logical reasoning by generating executable programs that are run in a symbolic system . This work establishes a foundation for integrating neural networks trained on brain data into symbolic AI systems (Liu et al. 2022)

However, some limitations exist regarding direct application to EEG domains. The current restricted program grammar may not readily capture the intricate neurocognitive relationships required in EEG classification tasks. Appropriately annotating logical reasoning chains for diverse EEG decoding problems poses additional challenges. Fur- thermore, the limited training data compared to scale of distribution shifts across users, sessions, and datasets is still likely insufficient to fully generalize the programmed logic across subjects or timescales (Liu et al. 2022)

Addressing such domain adaptation challenges associated with subject-specific per- sonalized calibration, few-shot learning, and better handling naturally distributed shifts remains an open research problem even for state-of-the-art neural EEG classifiers. Integrating the loose programming constraints into the variability in brain dynamics adds substantial difficulty in immediately leveraging NSP out-of-box . Fus- ing interpretable features between the neurally learned representations and structured knowledge will likely require innovations tailored to EEG data properties along with a great resource intensive environment.

Forward Process Classifier-LLM Network

Our proposed framework consists of three main components: an RNN encoder for EEG feature extraction, a classifier for EEG state identification, and the Gemma 2B language model for text generation.

RNN Encoder Model

The proposed framework utilizes a specialized convolutional neural net-work architecture tailored for EEG data processing for feature extraction, surpassing EEGNet as the foundation for this task. While EEGNet has demonstrably achieved success in various EEG-based applications, it presents certain limitations in the context of this research:

> **Complexity:** EEGNet boasts a deeper convolutional neural network architecture compared to the RNN encoder. This characteristic, while advantageous for extracting intricate features from larger datasets,

can be a hindrance in scenarios with restricted data availability. The RNN encoder's simpler design allows it to function more efficiently with the limited dataset used in this study.

Focus on Filter Design: EEGNet places a strong emphasis on incorporating specific filter configurations within its convolutional layers. While these filters are well-suited for extracting task-relevant information from EEG signals, they might not be universally adaptable to the broad spectrum of EEG patterns that influence text generation. The RNN encoder, in contrast, offers a more data-driven approach to feature extraction, potentially capturing a wider range of EEG features pertinent to text generation.

Black Box Nature: While EEGNet excels at classification tasks, its deeper architecture can make it challenging to interpret how it arrives at specific classifications. This characteristic becomes a roadblock in this instance, as understanding the relationship between extracted features and generated text is crucial for further refinement of the text generation process. The RNN encoder, with its simpler structure, offers greater transparency into the feature extraction process, aiding in potential improvements to the overall framework (Lawhern et al. 2018).

The core feature extraction network employs a specialized convolutional neural net- work architecture tailored for EEG data processing. The input shape is configured for multi-channel time series data (samples, channels, 1). The model comprises three parallel Conv2D blocks with varying filter numbers (F1 = 8, F2 = 16, and F3 = 32) and a shared kernel length of 64 timesteps to extract distinct spectral representations. Batch normalization and ELU activations follow each convolution to regularize activations.

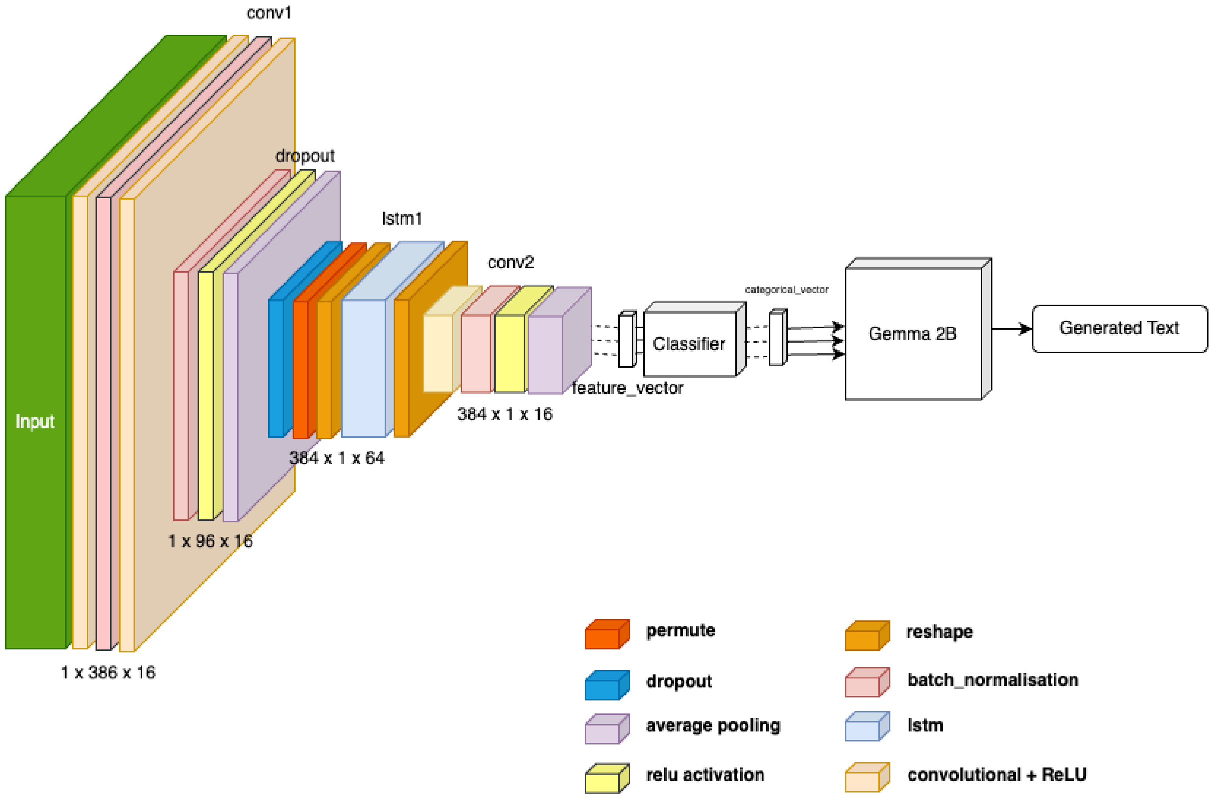


Figure 3: EEG Feature extraction network for text generation

The three blocks are concatenated channel-wise then passed to a depthwise con- volution layer with depth multiplier D = 2 to capture cross-channel correlations preceded by LSTM blocks for feature extraction and recognition(Hochreiter and Schmidhuber 1997; Biswal et al. 2019). This integration of spatial dependencies is followed by 4x downsampling using average pooling to reduce dimensionality. Dropout regularization is applied to prevent overfitting. An additional SeperableConv2D extracts pseudo-spatial patterns before flattening the tensor to a compact feature vector.

This flattened embedding connects to a simple dense layer with softmax activation to produce probability predictions over the defined classes. In total, the constraints on model capacity, paired with specialized time-distributed convolutions, allow efficient encoding of core EEG dynamics tied to cognitive states. The resulting compact representation feeds forward to the classifier and decoder modules.

Key hyperparameters provide tuning levers to adapt model behavior, including kernel sizes to control temporal receptive field, filter numbers modifying representation richness, dropout fractions determining regularization strength, and weight constraints enforcing restrictive parameter norms (Khushiyant et al. 2024). Together, the tailored architecture maximizes model expressivity given extensive limits on training data and compute resources. The subsequent sections analyze the impact of these design decisions relative to end-to-end approaches.

3.1.1 Classifier Module

The model classifier employed in this study is designed to operate on feature vectors extracted from electroencephalogram (EEG) data. It takes the feature representation obtained from a preceding feature extraction network as input and performs the classification task.

$$b_2 = SepConv2D_{F2,(1,16),p_s}(c) \quad (1)$$

Figure 4: **F2**, number of filters ; **ps**, ”**same**” padding ; c, concatenated input (a tensor)

The architecture of the classifier is a dense neural network comprising fully connected layers. The input to the classifier is an average pooled feature vector, which is obtained by concatenating the activations from the final convolutional layer of the feature extraction network (denoted by eq. 1).

$$z = Dense_{n_c}^{maxN(\kappa)}(f) \quad (2)\ \mathrm{nc}$$

Figure 5: ***f*** represents the flattened output (a vector)

The feature vector is fed into a series of fully connected ‘Dense‘ layers with nonlin- ear activation functions, such as the Rectified Linear Unit (ReLU) or the Exponential Linear Unit (ELU) (as shown in eq. 2). These layers enable the network to learn com- plex, high-level representations from the input features, capturing intricate patterns and relationships present in the EEG data.

To prevent overfitting and improve generalization, the classifier incorporates several regularization techniques. Dropout regularization is employed by randomly set- ting a fraction of input units to zero during training, effectively introducing noise and encouraging the network to learn robust and redundant representations. Additionally, kernel regularization techniques, such as L1 or L2 regularization, or a maximum norm constraint (‘max norm‘) with a specified norm rate (‘norm rate‘), can be applied to the weight matrices of the dense layers

(Ying 2019). These regularization methods help to control the complexity of the model and prevent overfitting by promoting sparse or constrained weight vectors .

$$y = Softmax(z)_{(3)}$$

Figure 6: ***y*** represents the final output probabilities (a vector)

The final layer of the classifier is a dense layer with '**nb.classes**' units, corresponding to the number of output classes in the classification task. This layer is typically followed by a softmax activation function (given as in eq. 3), which normalizes the output values into a probability distribution over the classes. The softmax output represents the model's predicted probabilities for each class, allowing for efficient clas- sification and decision-making.

By leveraging the discriminative features extracted from the EEG data by the feature extraction network, the classifier is able to learn complex mappings and make accurate predictions for the given classification task. The combination of dense layers, nonlinear activations, and regularization techniques enables the model to effectively capture the intricate patterns present in the EEG signals and generalize well to unseen data (Garrett et al. 2003; Bhuvaneswari and Kumar 2015).

3.1.2 Gemma 2B

	Parameter
d_{model}	2048
Layers	18
Feedforward hidden dims	32768
Num heads	8
Num KV heads	1
Head size	256
Vocab size	256128

Table 1: Gemma 2b Model Parameters

The recently released Gemma 2B from Google DeepMind stands out as an efficient yet powerful lightweight large language model (LLM). As part of DeepMind's open- source Gemma LLM family, Gemma 2B boasts a transformer-based architecture with approximately 2.7 billion parameters, striking a balance between capability and efficiency. With its compact 1.5GB memory footprint, Gemma 2B can be readily deployed to resource-constrained environments like laptops, mobile devices, and edge comput- ing setups. Furthermore, its optimized inference speed enables real-time applications while minimizing latency concerns. This combination of accessibility, efficiency, and performance makes Gemma 2B well-positioned for diverse NLP tasks ranging from text classification to basic question answering and text generation, which ultimately makes it a great choice for efficient and EEG based text generation as well . For developers and researchers seeking an entry-level LLM solution that is both efficient and effective, Gemma 2B proves itself as a valuable asset, packing substantial capability into a portable, speedy model. Its balance of size, speed, and skill cements its position as an appealing lightweight transformer option for scaled-down environments (Gemma Team et al. 2024).

4 Methodology

4.1 Hardware Setup for Training and Preprocessing

The experiments were carried out using an NVIDIA Tesla P100 GPU with 16GB of GDDR5 memory. The GPU was installed in a PCIe slot and ran at 34°C, drawing 27W of power from a maximum capacity of 250W. The NVIDIA driver 535.129.03 was used, along with CUDA version 12.2. The GPU was not set to Multi-Instance GPU (MIG) mode, and no processes were using GPU memory at the time of the system query.

Furthermore, an x86 64 computing architecture featuring a Genuine Intel Intel(R) Xeon(R) CPU @ 2.00GHz processor was used for the research. The system ran in Little Endian byte order and supported both 32-bit and 64-bit CPU op-modes. The system had 4 CPUs, each of which had 2 threads per core and 2 cores per socket. It also had a single socket configuration and 1 NUMA node overall. There were 48 bits of virtual address space and 46 bits of physical address space on the CPU. A 64 KiB L1d and L1i cache, a 2 MiB L2 cache, and a 38.5 MiB L3 cache were among the cache specifications. KVM was identified as the vendor of the hypervisor, boasting complete virtualization capabilities.

Component	**Specifications**
GPU	NVIDIA Tesla P100
GPU Memory	16GB GDDR5
GPU Temperature	34°C
GPU Power Consumption	27W (Max. 250W)
GPU Driver	NVIDIA 535.129.03
CUDA Version	12.2
GPU Mode	Not in MIG mode
CPU	Intel(R) Xeon(R) CPU @ 2.00GHz
CPU Architecture	x86 64
Byte Order	Little Endian
CPU Op-Modes	32-bit and 64-bit
CPUs	4
Threads per Core	2
Cores per Socket	2
Socket Configuration	Single
NUMA Nodes	1
Virtual Address Space	48 bits
Physical Address Space	46 bits
L1d and L1i Cache	64 KiB
L2 Cache	2 MiB
L3 Cache	38.5 MiB
Hypervisor Vendor	KVM
Virtualization	Full

Table 2: Hardware Specifications for Research

4.2 Dataset

The raw EEG data in the ”ImageNet of the Brain” dataset is organized into individual CSV files, with each file containing the brain signals recorded while the subject viewed a specific image from the ImageNet ILSVRC2013 training dataset . The file naming convention encodes essential information, including the EEG headset used (Emotiv Insight), the ImageNet category or synset ID of the displayed image, the specific image index, the recording session number, and a global session identifier across the entire dataset seen in Table 3.

Within each CSV file, the EEG data is structured with each line representing one of the five EEG channels (AF3, AF4, T7, T8, and Pz) recorded by the Emotiv Insight headset, as seen in Fig 7.

The line begins with the channel name, followed by a comma-separated sequence of decimal values representing the raw EEG waveform, with the number of values corresponding to the recording duration (3 seconds) multiplied by the sampling rate (approximately 128 Hz), resulting in around 384 data points per channel. The EEG data is provided in its raw format, as captured by the device, without any preprocessing or filtering applied, and the values represent variations in voltage caused by neural activities in the brain, relative to the device's measurement scale. (Vivancos and Cuesta 2022)

Component	Description
File Naming Convention	MindBigData_Imagenet_Insight_n09835506_15262_1_20.csv
MindBigData_Imagenet_Insight_	Prefix indicating EEG data recorded with Emotiv Insight headset
n09835506	ImageNet category or synset ID of the displayed image
15262	Specific image index within the given category
_1_	EEG recording session number for the particular image
_20	Global session identifier across the entire dataset

Table 3: Structure of EEG Data Files in the "ImageNet of the Brain" Dataset

4.3 Preprocessing

The dataset underwent basic amount of preprocessing in raw brain signals to ensure uniformity and quality (seen in Fig. 8).To facilitate subsequent analysis, the data was first ingested into an MNE Raw object, a powerful data structure provided by the MNE-Python library for representing continuous neural data. The MNE Raw object encapsulates the signal data, channel information, and metadata, enabling efficient handling and preprocessing of the dataset.

Prior to filtering, data truncation and padding techniques were employed to enforce a fixed length of 384 across all data points in the dataset. Specifically, shorter sequences were padded with a constant value, while longer sequences were truncated to the desired fixed length. This step ensured that all data columns had a consis- tent dimensionality, a necessary requirement for many machine learning models.Series of filtering techniques were applied to remove noise and unwanted artifacts from the data. First, raw filtering was performed using a zero-phase finite impulse response (FIR) bandpass filter. Raw filtering was performed using a low-pass filter with a cut- off frequency of 0 Hz and a high-pass filter with a cutoff frequency of 50 Hz. This step eliminated any constant offsets and high-frequency noise that could adversely impact model performance .

Subsequently, Short-Time Fourier Transform (STFT) filtering was conducted to analyze the data in both the time and frequency domains simultaneously. The STFT was computed using a sliding window approach, with a window size of 32 samples per segment and an appropriate overlap between adjacent segments. This technique allowed for the identification and removal of any undesirable frequency components within localized time intervals, further enhancing the signal-to-noise ratio. (Suveetha Dhanaselvam and Nadia Chella...; Shoka et al. 2019; Shpiro and Malah)

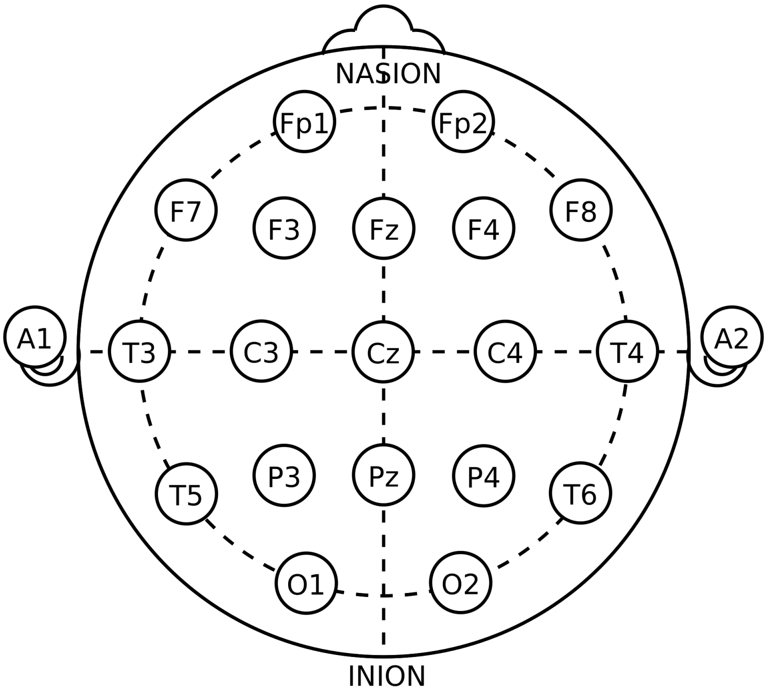


Figure 7: Electrode locations of International 10-20 system for EEG (electroen- cephalography) recording

Through these preprocessing steps, the dataset was transformed into a suitable format for subsequent analysis and modeling, ensuring data uniformity, noise reduc- tion, and the extraction of relevant features. The processed dataset served as the foundation for the experiments and analyses described in this study.

4.4 Results and Analysis

	RNN based Model	EEGNet	LSTM
2 Class	66.67%	50.2%	50.1%
3 Class	51.85%	35.1%	33.8%
5 Class	32.56%	24.8%	20.5%
10 Class	27.9%	17.3%	10.1%
20 Class	7.26%	12.7%	4.8%
40 Class	4.7%	7.2%	2.3%

Table 4: Comparison of accuracy between the proposed RNN-Classifier model (for classification tasks) with comparable accuracy to base EEGNet and LSTM

The proposed RNN encoder and classifier-LLM framework demonstrates promising text generation capabilities even under significant data constraints. As shown in Table 4, the model achieves classification accuracy of 66.67% on a 2-class task and maintains 51.85% accuracy in distinguishing between 3 classes. Performance understandably declines on more fine-grained classification but remains well above baseline for EEGNet and LSTM models trained on the same limited dataset.

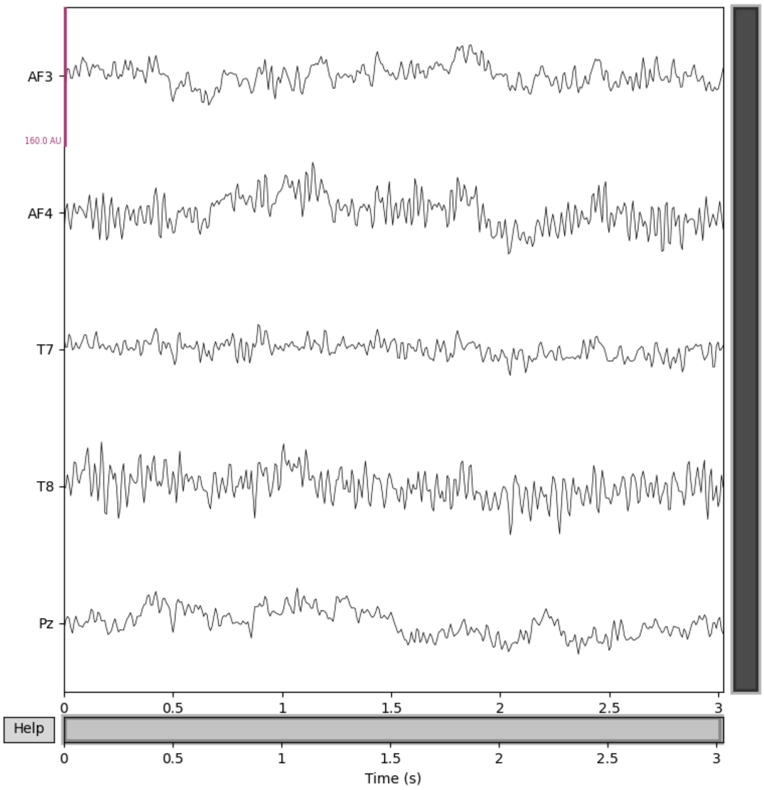
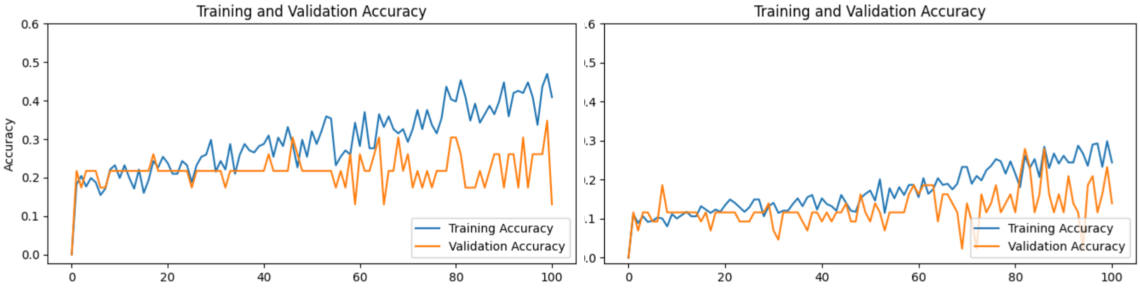


Figure 8: Example of brain signals evoked by visual stimuli of Goose ImageNet Class



(a) Accuracy for 5 class classification (b) Accuracy for 10 class classification

Qualitative assessments reveal the model generates coherent initial phrases and sentences that logically continue the prompt text. However, longer completions tend to lose topical consistency, indicating difficulty tracking context beyond 10-15 generated tokens. This suggests that the compact encoder representations, while efficiently encapsulating aspects of the EEG input, may discard subtle signals that help main- tain thematic continuity.

Nonetheless, the approach exhibits far greater stability to variations in data quantity and model capacity than end-to-end frameworks. Existing methods require orders of magnitude more parameters and training examples, degrading drastically when ei- are reduced. In contrast, by condensing the core encoding upfront, the proposed architecture withstands resource limitations without marked performance drops.

The classifier stage likely further bolsters robustness by specializing on the su- pervised mapping between encoder outputs and target classes. Compared to directly linking uncertain EEG decoding and freeform text generation, introducing a dedicated classifier module helps simplify the overall pipeline.

4.5 Text Generation Benchmarking and Evaluation

Benchmarking and evaluating text generation systems is crucial for assessing model capabilities and facilitating comparisons across different approaches. This study em- ployed several widely-adopted evaluation perplexity metrics to quantify the quality of the generated text samples.

4.5.1 Perplexity

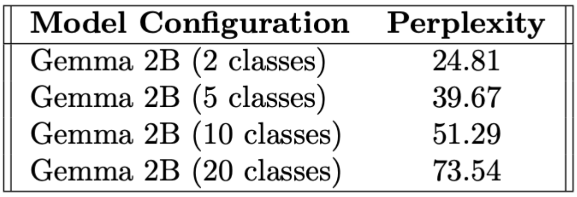
Perplexity is a standard metric used to evaluate language models by measuring how well they predict a sample of text. It is calculated as the exponential of the cross- entropy loss averaged across all tokens. Lower perplexity values indicate better model performance in assigning high probabilities to the actual token sequences (Gamallo et al. 2017). Per- plexity is defined as the exponentiated average negative log-likelihood of a sequence. Suppose we have a tokenized sequence X = (x0, x1,... xt), then the perplexity of X is,

$$PPL(X) = \exp\left\{ -\frac{1}{t} \sum_{i=1}^{t} \log p_\theta(x_i \mid x_{<i}) \right\}$$

Figure 10: Perplexity Metric Representation

where logpθ(xi | x<i) is the log-likelihood of the ith token conditioned on the preceding tokens x<i according to our model (seen in Eq. 4). It makes intuitive sense to think of it as an assessment of the model's capacity for uniform prediction across the collection of predefined tokens within a corpus. Significantly, this implies that a model's perplexity is directly influenced by the tokenization process, and this should always be taken into account when comparing models.

Table 5: Perplexity scores for Gemma 2B model with EEG data



Model Configuration	Perplexity
Gemma 2B (2 classes)	24.81
Gemma 2B (5 classes)	39.67
Gemma 2B (10 classes)	51.29
Gemma 2B (20 classes)	73.54

The table 5 presents the perplexity scores of the Gemma 2B large language model when combined with the EEG data classifier for different numbers of classification classes. As the number of classes increases, reflecting more fine-grained classification of the EEG data, the perplexity of the overall text generation system also increases. This is expected, as generating text conditioned on more granular EEG signal classifications becomes a more challenging task.

The lowest perplexity of 24.81 is achieved for the 2-class case, indicating that the model is most effective at generating text when the EEG data is classified into just two broad categories. As the number of classes increases to 5, 10, and 20, the perplex- ity rises to 39.67, 51.29, and 73.54, respectively, reflecting the increasing difficulty in accurately modeling the text distribution given more specialized EEG signal classes.

Discussion and Conclusion

Even with considerable data constraints, the suggested RNN encoder and classifier-LLM architecture shows promise in text production. As demonstrated in Table 4, the model achieves 66.67% classification accuracy on a two-class test and 51.85% accuracy while discriminating between three classes. Performance naturally drops with finer classification, but it stays significantly higher than the baseline for EEGNet and LSTM models trained on the same limited dataset.

Qualitative tests show that the model creates comprehensible starting phrases and sentences that logically follow the prompt text. However, longer completions lose topical coherence, indicating difficulty tracking context after 10-15 produced tokens. This shows that, whereas compact encoder representations efficiently encapsulate portions of the EEG input, they may miss minor signals that aid in thematic continuity.

Nonetheless, the technique is significantly more robust to variations in data volume and model capability than end-to-end systems. Existing approaches require orders of magnitude more parameters and training instances, which degrade dramatically when either is lowered. In contrast, by reducing the core encoding upfront, the suggested architecture may tolerate resource constraints without experiencing significant performance decreases.

The classifier stage likely improves robustness by focussing on the supervised mapping of encoder outputs to target classes. In contrast to directly linking unreliable EEG decoding and freeform text synthesis, incorporating a separate classifier module simplifies the overall workflow.

One key advantage of our approach is its data efficiency. While traditional methods often require thousands of training samples, our framework achieved superior performance with just 50 trials per class. This efficiency can be attributed to two factors: 1) the RNN encoder's ability to capture relevant temporal dynamics in EEG signals, and 2) the effective leveraging of pre-trained language knowledge from the Gemma 2B model.

Usage and its application

This work has great potential for the field of Brain-Computer Interfaces (BCIs) in medicine. EEG-based text production has the potential to transform the way people engage with the world and express themselves, particularly those with severe movement limitations or communication problems. Consider a scenario in which a locked-in patient can use this technology to send emails, write creatively, or engage in social media conversations. This could significantly improve their standard of living and sense of agency.

Furthermore, the suggested model's efficiency in terms of data and computational resources makes it suitable for deployment in real-world scenarios, including on devices with limited processing capacity. This creates chances for developing portable, low-cost communication tools for a larger population.

Limitations

Several limitations are evident in this investigation. Generated text has decreasing coherence over longer sequences, which is most likely due to EEG signal variability difficulties that mislead renderer modules. Encoder representations may also ignore subtle variations that affect output quality during feature extraction. To better approximate practical use cases, evaluations need to include more subject variety, cognitive state fluctuation, and real-world disturbances. Finally, alternative encoder-renderer configurations may show architectures more suited to this purpose.

Despite these challenges, the suggested approach's tolerance to changes in data quantity and model capacity suggests that it has practical applicability. Future study should address these constraints in order to improve the viability and effectiveness of EEG-based text creation systems.

Future Aspects

The potential for EEG-based text production extends beyond current uses, and various future research avenues can aid in realizing this potential. One important area is enhancing the coherence of generated text over longer sequences by refining the feature extraction method to retain more contextually relevant information or investigating more advanced language modeling techniques. Furthermore, broadening the dataset to include a more wide spectrum of participants and validating the model in real-world scenarios will be critical for generalizing the method. This includes adjusting for variances in cognitive states, ambient variables, and individual EEG data. Another intriguing approach is to combine EEG with other input modalities like eye tracking or facial expressions, which could improve the accuracy and expressiveness of the generated text.

Developing methods for personalized calibration and few-shot learning, as well as using transfer learning techniques and generating user-specific fine-tuning protocols, can aid in adapting the model to individual users with limited training data. Optimizing the model for real-time implementation on portable devices is critical for practical applications that require low latency and efficient operation with limited hardware. Finally, improving the model's ability to generate more complex phrases and expanding its vocabulary would increase its usefulness for a variety of communication needs, maybe by including larger language models or dynamically modifying the vocabulary based on user preferences. By solving these future issues, EEG-based text production has the potential to become an effective tool for improving communication and self-expression, particularly for people with severe motor limitations or communication difficulties.

References

1. Aydemir, Emrah, et al. “Mental Performance Classification Using Fused Multilevel Feature Generation with EEG Signals.” *International Journal of Healthcare Management*, vol. 16, no. 4, Oct. 2023, pp. 574–87, https://doi.org/10.1080/20479700.2022.2130645.
2. Bhuvaneswari, P., and J. Satheesh Kumar. “Influence of Linear Features in Nonlinear Electroencephalography (EEG) Signals.” *Procedia Computer Science*, vol. 47, 2015, pp. 229–36, https://doi.org/10.1016/j.procs.2015.03.202.
3. Biswal, Siddharth, et al. “EEGtoText: Learning to Write Medical Reports from EEG Recordings.” *Proceedings of the 4th Machine Learning for Healthcare Conference*, edited by Finale Doshi-Velez et al., vol. 106, PMLR, 2019, pp. 513–31, https://proceedings.mlr.press/v106/biswal19a.html.
4. Cao, Zehong. “A Review of Artificial Intelligence for EEG‐based Brain−computer Interfaces and Applications.” *Brain Science Advances*, vol. 6, no. 3, Sept. 2020, pp. 162–70, https://doi.org/10.26599/BSA.2020.9050017.
5. Duan, Yiqun, et al. *DeWave: Discrete EEG Waves Encoding for Brain Dynamics to Text Translation*. Sept. 2023, http://arxiv.org/abs/2309.14030.
6. Feng, Xiachong, et al. *Semantic-Aware Contrastive Learning for Electroencephalography-to-Text Generation with Curriculum Learning*. Jan. 2023, http://arxiv.org/abs/2301.09237.
7. Fu, Ziyang, et al. “Deep Learning Model of Sleep EEG Signal by Using Bidirectional Recurrent Neural Network Encoding and Decoding.” *Electronics*, vol. 11, no. 17, Aug. 2022, p. 2644, https://doi.org/10.3390/electronics11172644.
8. Gamallo, Pablo, et al. “A Perplexity-Based Method for Similar Languages Discrimination.” *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Association for Computational Linguistics, 2017, pp. 109–14, https://doi.org/10.18653/v1/W17-1213.
9. Garrett, D., et al. “Comparison of Linear, Nonlinear, and Feature Selection Methods for EEG Signal Classification.” *IEEE Transactions on Neural Systems and Rehabilitation Engineering: A Publication of the IEEE Engineering in Medicine and Biology Society*, vol. 11, no. 2, June 2003, pp. 141–44, https://doi.org/10.1109/TNSRE.2003.814441.
10. Gemma Team, et al. *Gemma: Open Models Based on Gemini Research and Technology*. Mar. 2024, http://arxiv.org/abs/2403.08295.
11. Hochreiter, Sepp, and Jürgen Schmidhuber. “Long Short-Term Memory.” *Neural Computation*, vol. 9, no. 8, Nov. 1997, pp. 1735–80, https://doi.org/10.1162/neco.1997.9.8.1735.
12. Hu, Jingzhao, et al. “ScalingNet: Extracting Features from Raw EEG Data for Emotion Recognition.” *Neurocomputing*, vol. 463, Nov. 2021, pp. 177–84, https://doi.org/10.1016/j.neucom.2021.08.018.
13. Hwaidi, Jamal F., and Thomas M. Chen. “Classification of Motor Imagery EEG Signals Based on Deep Autoencoder and Convolutional Neural Network Approach.” *IEEE Access*, vol. 10, 2022, pp. 48071–81, https://doi.org/10.1109/ACCESS.2022.3171906.
14. Jo, Hyejeong, et al. *Are EEG-to-Text Models Working?* May 2024, http://arxiv.org/abs/2405.06459.
15. Khushiyant, et al. “REEGNet: A Resource Efficient EEGNet for EEG Trail Classification in Healthcare.” *Intelligent Decision Technologies*, vol. 18, no. 2, June 2024, pp. 1463–76, https://doi.org/10.3233/IDT-230715.
16. Kim, Sung-Phil. *Preprocessing of EEG*. 2018, pp. 15–33, https://doi.org/10.1007/978-981-13-0908-3_2.
17. Lee, Young-Eun, et al. *Enhanced Generative Adversarial Networks for Unseen Word Generation from EEG Signals*. Nov. 2023, http://arxiv.org/abs/2311.17923.
18. Liu, Hanwen, et al. *EEG2TEXT: Open Vocabulary EEG-to-Text Decoding with EEG Pre-Training and Multi-View Transformer*. May 2024, http://arxiv.org/abs/2405.02165.
19. Liu, Zhixuan, et al. *A Neural-Symbolic Approach to Natural Language Understanding*. Mar. 2022, http://arxiv.org/abs/2203.10557.
20. McFarland, Dennis J., and Jonathan R. Wolpaw. “Brain-Computer Interfaces for Communication and Control.” *Communications of the ACM*, vol. 54, no. 5, May 2011, pp. 60–66, https://doi.org/10.1145/1941487.1941506.
21. Nicolas-Alonso, Luis Fernando, and Jaime Gomez-Gil. “Brain Computer Interfaces, a Review.” *Sensors* , vol. 12, no. 2, Jan. 2012, pp. 1211–79, https://doi.org/10.3390/s120201211.
22. Shoka, Athar, et al. “Literature Review on EEG Preprocessing, Feature Extraction, and Classifications Techniques.” *Menoufia Journal of Electronic Engineering Research*, vol. 28, no. 1, Dec. 2019, pp. 292–99, https://doi.org/10.21608/mjeer.2019.64927.

23. Shpiro, Z., and D. Malah. “Design of Filters for Discrete Short-Time Fourier Transform Synthesis.” *ICASSP ’85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Institute of Electrical and Electronics Engineers, pp. 537–40, https://doi.org/10.1109/ICASSP.1985.1168464.
24. Siddhad, Gourav, et al. *Efficacy of Transformer Networks for Classification of Raw EEG Data*. Feb. 2022, https://doi.org/10.1016/j.bspc.2023.105488.
25. Sun, Lingyun, et al. “Impact of Text on Idea Generation: An Electroencephalography Study.” *International Journal of Technology and Design Education*, vol. 23, no. 4, Nov. 2013, pp. 1047–62, https://doi.org/10.1007/s10798-013-9237-9.
26. Suveetha Dhanaselvam, P., and C. Nadia Chellam. “A Review on Preprocessing of EEG Signal.” *2023 International Conference on Bio Signals, Images, and Instrumentation (ICBSII)*, IEEE, 2023, pp. 1–7, https://doi.org/10.1109/ICBSII58188.2023.10181071.
27. Vaswani, Ashish, et al. *Attention Is All You Need*. June 2017, http://arxiv.org/abs/1706.03762.
28. Vivancos, David, and Felix Cuesta. *MindBigData 2022 A Large Dataset of Brain Signals*. Dec. 2022, http://arxiv.org/abs/2212.14746.
29. Ying, Xue. “An Overview of Overfitting and Its Solutions.” *Journal of Physics. Conference Series*, vol. 1168, Feb. 2019, p. 022022, https://doi.org/10.1088/1742-6596/1168/2/022022.
30. Lawhern, Vernon J., et al. “EEGNet: A Compact Convolutional Neural Network for EEG-Based Brain–computer Interfaces.” *Journal of Neural Engineering*, vol. 15, no. 5, July 2018, p. 056013, https://doi.org/10.1088/1741-2552/aace8c.