# CREATOR: Synthetic Neurolinguistic EEG Data Generator

## Abstract

Recent works have demonstrated the potential of Electroencephalography (EEG) data to develop models for downstream Natural Language Process (NLP) tasks. However, the shortage of sufficiently large EEG datasets for NLP has created a training bottleneck, hindering the advancement of research in this area. With contributing factors associated with conducting an EEG-based user study such as cost, time, the difficulty of obtaining high-quality EEG data, the hardware requirements, and specialised skill set, as well as the potential of capturing sensitive/personal data, it is crucial to explore alternative avenues to increase the size of the existing EEG datasets without the collection of new data samples. To do so, this paper proposes generating synthetic EEG samples via augmenting pre-existing EEG datasets for NLP to capture the semantics of textual input. To study our proposed approach, we develop **CREATOR**, syntheti**C** neu**R**olinguistic **E**EG d**A**ta genera**TOR**, which leverages Generative Adversarial Networks (GANs) conditioned on text input to produce synthetic EEG samples for data augmentation. We investigated the effectiveness of SENSE on three popular EEG-Text paired NLP tasks i.e. Named Entity Recognition, Sentiment Classification, and EEG-To-Text decoding. Our findings demonstrate that the synthetic EEG samples generated via CREATOR resulted in a statistically significant increase in the overall performance of each task by 6.6%, 6.1%, and 7.9% respectively. Furthermore, the findings of our extensive ablation analysis reveal that the enhancement in performance is attributable to incorporating CREATOR EEG samples into task datasets. Our proposed CREATOR model serves as a first step in the production of synthetic EEG data samples for downstream NLP tasks facilitating the training of large and more complex Neurolinguistic models.

## 1 Introduction

Electroencephalography (EEG) data has recently garnered significant attention within the Natural Language Processing (NLP) field due to several prior works demonstrating its capabilities for enhancing downstream NLP tasks [21, 10]. By affixing an array of electrodes to subjects' scalp, EEG can capture the brain's electrical activity, capturing patterns that correlate with different mental states, such as attention [24], relaxation [11], or specific linguistic processes [8]. These patterns, when decoded accurately, can provide valuable insights into cognitive processes and can be harnessed to improve various NLP tasks by providing further contextual information [21, 10]. The use of EEG data is favoured for these tasks over alternative neuroimaging techniques such as Functional magnetic resonance imaging (fMRI) due to its high temporal resolution, portability and non-invasive application [2]. However, one of the most significant limitations of EEG data is the scarcity of adequately sized open-source datasets for the training and development of new models and components specific to the decoding of EEG signals [14]. This likely arises from numerous compounding factors relating to the creation of new EEG datasets, such as the financial cost associated with user studies, the time required to capture data from numerous participants, the various hardware components to ensure the EEG data is high quality, a specialised skill set to facilitate the collection of the data, as well as the potential to collect sensitive and personal information inherent with neurophysiological data [4, 12].

Preprint. Under review.

Hence, any means that can alleviate the scarcity of open-source EEG datasets while reducing the need for further data collection would be beneficial to research relating to the application of EEG data, which this work aims to do.

One approach to address data scarcity is the use of EEG augmentation techniques. Prior works have demonstrated the success of traditional techniques such as temporal jittering [6] or electrode permutation [22] for the augmentation of EEG datasets. More recent works have incorporated deep learning approaches to synthesising new data samples, specifically the use of Generative Adversarial Networks (GANs) [15, 14] which have demonstrated their capabilities to produce high-quality data that when used to augment target datasets has been observed to benefit model performance. However, these augmentation strategies do not take into consideration the neurolinguistic relationship between synthetic EEG samples and their corresponding text representation which reduces their effectiveness for augmenting downstream NLP task datasets.

As such, this work aims to address these issues by providing a new EEG augmentation strategy to maintain the neurolinguistic relationship between synthetic EEG samples and their corresponding text representation. Hence, we formulate three key research questions: **(1)** Is it possible to generate synthetic EEG samples that capture the neurolinguistic characteristics of the brain perceiving text? **(2)** What granularity of EEG generation is best suited for downstream NLP tasks? **(3)** How do we decide which samples EEG samples to generate for augmenting downstream NLP datasets? To address these research questions, we introduce **CREATOR** - syntheti**C** neu**R**olinguistic **E**EG d**A**ta genera**TOR**, which utilises GANs conditioned on text embeddings to produce synthetic EEG samples that aim to capture the underlying neurolinguistic characteristics of the original EEG samples. We assess synthetic EEG sample quality by augmenting datasets for three key EEG-related NLP tasks: Named Entity Recognition, Sentiment Classification, and EEG-To-Text decoding. We investigated how conditioning the model at word, contextual, and sentence levels influences task performance, along with strategies for selecting EEG segments to augment.

## 2 Related Work

### 2.1 GAN Augmentation

A primary application of the GAN model is the generation of high-quality synthetic data; this makes it ideal for the augmentation of datasets, wherein learning the distribution of $\mathcal{X}_\mathrm{r}$ it can generate new samples that closely resemble real data instances. Furthermore, the flexibility of GANs allows for the generation of diverse and realistic samples across different modalities and domains. Whether it's images, text, audio, or EEG, GANs can be adapted to learn the respective data distributions and generate synthetic samples that align with the desired properties. This strategy has been successfully applied for augmenting datasets that have improved the performance of downstream tasks such as EEG-based emotion recognition [27, 15]. However, to the best of our knowledge, no work has explored the capabilities of GANs to produce semantically similar synthetic samples for the augmentation of EEG NLP tasks such as Named Entity Recognition, Sentiment Analysis, and EEG-To-Text decoding.

### 2.2 EEG and NLP

Previous works have explored the use of EEG in conjunction with NLP for tasks such as sentiment analysis, named entity recognition, and direct EEG-to-text translation. By combining EEG data with traditional text-based features, studies have shown improvements in detecting emotional states from text. For instance, prior works have demonstrated that integrating EEG features with textual data significantly boosts the accuracy of sentiment classification and named entity recognition models [10, 21]. Furthermore, EEG-to-text decoding represents one of the most ambitious applications, aiming to translate brain activity directly into written language. This technology holds immense potential for assisting individuals with communication impairments, offering a means to express thoughts and ideas without the need for traditional input devices. Several methodologies have been proposed for this purpose. For example, researchers have employed deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [28], to decode EEG signals into corresponding text sequences, with more recent works leveraging pre-trained large language models (LLMs) [25, 5].

## 3 Preliminaries

### 3.1 Generative Adversarial Networks

The Generative Adversarial Networks (GANs) first introduced by [7] comprise of two models, the generator *G* and the discriminator *D*. Both of these models are engaged in an adversarial game where the generator aims to produce realistic synthetic samples given a noise variable $z$ as input (prior distribution). The discriminator then estimates the probability of a given sample originating from the original (real) data distribution $\mathcal{X}_{\mathrm{r}}$ or the synthetic (fake) data distribution $\mathcal{X}_{\mathrm{G}}$. The training process involves minimising the generator's loss, $\mathcal{L}_{\mathrm{G}}$, which measures the discrepancy between the discriminator's predictions on generated samples and the 'real' label, and maximising the discriminator's loss, $\mathcal{L}_{\mathrm{D}}$, which captures the difference between its predictions on real and fake data. These objectives are expressed through adversarial loss functions for the generator and discriminator, which can be seen in equations 1 and 2, respectively. The adversarial process can be represented by a minimax game between *G* and *D*, represented in equation 3. Where *z* is a random noise vector sampled from the prior distribution $p_z(z)$, $\mathbb{E}_{z\sim p_z(z)}$ denotes the expectation over the noise distribution $p_z(z)$, $\mathbb{E}_{x\sim p_{\text{data}}(x)}$ is the expectation over the real data distribution, and $\theta_g$,$\theta_d$ represent the generator and discriminator parameters respectively.

$$\mathcal{L}_{\mathrm{G}} = -\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z;\theta_g);\theta_d))] \tag{1}$$

$$\mathcal{L}_{\mathrm{D}} = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x;\theta_d)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z;\theta_g);\theta_d))] \tag{2}$$

$$\min_G \max_D \mathcal{V}(D,G)\mathcal{V}(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

Since their introduction, the GAN has suffered from training instability due to Kullback-Leibler (KL) [1] and Jensen-Shannon (JS) [16] divergence being insufficient for providing gradients to the generator [26]. During training minimising the loss of the generator is equivalent to minimising JS or KL divergence between $\mathcal{X}_{\mathrm{r}}$ and $\mathcal{X}_{\mathrm{G}}$, where it is highly unlikely that there will be a non-existence overlap between the two, regardless of their distance, as a result the generator can exploit this and converge to a solution that does not accurately represent the true data distribution [27]. To combat the training instability, [1] proposed the Wasserstein-GAN with gradient penalty (WGAN-GP), which uses the Wasserstein Distance (a.k.a Earth Movers Distance) in place of JS divergence and the gradient penalty (GP). During training the WGAN provides a more meaningful and stable metric for measuring the discrepancy between the generated and real data distributions compared to JS divergence. Furthermore, the GP regularises the discriminator by penalising the norm of its gradients with respect to interpolated points along the straight lines between pairs of real and generated samples. This penalty encourages the discriminator to produce gradients that are not only informative but also smooth, thus promoting a more stable training process.

### 3.2 Conditional Generation

Traditional GANs use a random noise vector, *z*, to generate synthetic samples [7]. However, this often results in a lack of control over the generated data's characteristics. In text-paired EEG synthesis, this lack of control poses challenges, as traditional GANs struggle to capture the nuanced relationship between textual inputs and corresponding EEG segments due to the complexity of EEG signals and detailed semantic content of text. To address this, conditional GANs (cGANs) offer a more effective solution [15, 23]. By incorporating conditioning information, such as textual input, cGANs provide explicit guidance to the generator, enhancing the synthesis of text-paired EEG samples. This approach has been demonstrated in text-to-image generation [20], where a textual description *t* is encoded using a text encoder $\phi$ as $\phi(t)$ and concatenated with the latent noise vector *z* before being fed to the generator network. In our work, we adopt a similar approach. We encode word tokens $\phi(\mathcal{W})$ of the corresponding EEG segment sequence $\mathcal{E}_{\mathrm{r}}$ and concatenate them with the noise vector $\phi(\mathcal{W}) \oplus z$. This combined input is then used to condition the generation of synthetic EEG samples $\mathcal{E}_{\mathrm{f}}$.

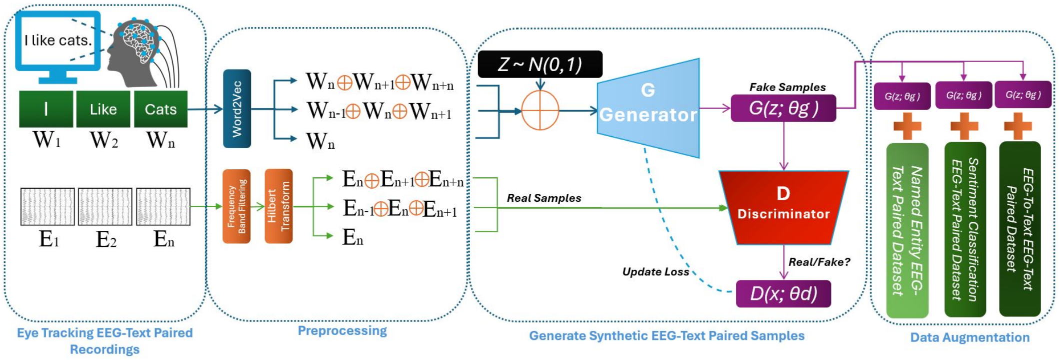

Figure 1: Overview of Augmentation Process using Generative Adversarial Networks

# 4 Approach

## 4.1 Task Formulation

Given a paired sequence of EEG features and word tokens $\langle\ \mathcal{E}_{\text{r}}, \mathcal{W}\ \rangle$ the aim is to generate synthetic EEG features $\mathcal{E}_{\text{f}}$ that capture the underlying neurolinguistic characteristics present in the original EEG sample $\mathcal{E}_{\text{r}}$ whilst reading the corresponding word token $\mathcal{W}$. We explored three methods of generating synthetic EEG features $\mathcal{E}_{\text{f}}$ given the original sequence of EEG and word tokens $\langle\ \mathcal{E}_{\text{r}}, \mathcal{W}\ \rangle$: Word, Contextual, and Sentence-Level. Moreover, we explored the four strategies for the augmentation of downstream EEG NLP datasets: Random, TF-IDF-Low, TF-IDF-Medium, and TF-IDF-High.

## 4.2 Generation Level

In order to address research question **(2)**, investigating how the granularity of a text sequence affects synthetic EEG sample generation we designed three distinct approaches (see Figure 1). **Word-Level:** Here the embedded word token $\phi(\mathcal{W}_n)$ is concatenated with the noise vector $\phi(\mathcal{W}_n) \oplus\ z$, where it is then provided to *G*. This would then produce synthetic EEG sample $\mathcal{E}_{\text{f}n}$ matching the dimensions of $\mathcal{E}_{\text{r}n}$. **Contextual-Level:** This approach takes into consideration the words surrounding an EEG segment. For a given real EEG sample $\mathcal{E}_{\text{r}n}$ its corresponding word token embedding would become $\phi(\mathcal{W}_{\text{n-1}}) \oplus \phi(\mathcal{W}_{\text{n}}) \oplus \phi(\mathcal{W}_{\text{n+1}})$. The word embeddings would be concatenated with $z$ and provided to *G* to generate a synthetic EEG sample $\mathcal{E}_{\text{f}n}$ matching the original dimensions of $\mathcal{E}_{\text{r}n}$. For any EEG segments positioned at the start or end of their sentences where there would be no prior or proceeding words, they were padded with zeros equal to the size of the embedding dimension. **Sentence-Level:** This approach includes the entire sentence sequence of EEG features and word token samples. This is done by concatenating all of the real EEG features in a given sequence $\mathcal{E}_{\text{r}n} \oplus \mathcal{E}_{\text{r}n+1} \oplus ... \oplus \mathcal{E}_{\text{r}n+n}$ along with the corresponding word token embeddings $\phi(\mathcal{W}_{\text{n}}) \oplus \phi(\mathcal{W}_{\text{n+1}}) \oplus ... \oplus \phi(\mathcal{W}_{\text{n+n}})$. If the EEG feature and word token sentences were below the maximum sentence length they would be padded with zeros up to the maximum sentence length. The combined word embedding would be concatenated with $z$ and provided to *G* as input to produce a synthetic EEG sample matching the dimensions of the combined real EEG samples.

## 4.3 Augmentation Strategy

When working with NLP datasets it is unlikely that every word within the vocabulary will be evenly represented. As such, randomly sampling from the dataset to determine which words should be used to produce new synthetic EEG samples may not yield optimal results. Therefore, we formulated three augmentation strategies to investigate the impact of augmenting words and sentences with varying occurrences within the dataset, see Figure 1. The Term Frequency-Inverse Document Frequency (TF-IDF) score was calculated for each word in the corpus using the formula $\text{TF-IDF}(w) = \text{TF}(w) \times \log\left(\frac{N}{\text{DF}(w)}\right)$, where $\text{TF}(w)$ is the term frequency of word $w$, $\text{DF}(w)$ is the document frequency of $w$, and $N$ is the total number of documents. We then divided the words into three equal portions based on their TF-IDF scores: Low, Medium, and High thresholds. During inference, for both word-level and contextual-level augmentation, the TF-IDF score of the target EEG segment was compared against the current augmentation strategy range. If it fell within this range, a synthetic EEG segment was added back into the dataset. For sentence-level augmentation, we computed the sum of TF-IDF scores for each word in the sentence, and if this sum fell within the current strategy range, the sentence was considered for augmentation. This approach allows us to systematically explore the effects of augmenting words and sentences with different frequency characteristics.

### 4.4 Network Architecture

To evaluate the generative capabilities of synthetic EEG samples, we explored the capabilities of two prominent GAN architectures: DCGAN and WGAN-GP. Both models share a common network architecture but differ in training methods. The discriminator ($\mathcal{D}$) comprises a series of 2D convolutional layers with instance normalization and Leaky ReLU activation, progressively increasing filters while downsampling spatial dimensions. A fully connected layer reduces feature maps to a single value per sample, followed by a sigmoid activation for real vs. fake determination. The generator ($\mathcal{G}$) consists of three 2D convolutional layers with batch normalisation and Leaky ReLU activations, gradually reducing filters to one. The final synthetic output is obtained via a hyperbolic tangent activation. To condition the generator with textual embedding $\phi(\mathcal{W})$ using Word2Vec [3], we implemented a variation of the architecture that concatenates the latent noise input $z$ with $\phi(\mathcal{W})$. Results show that WGAN-GP consistently outperforms DCGAN in augmentation tasks, thus all results presented are from the WGAN-GP model.

## 5 Experimental Set-up

#### 5.0.1 Data

This study incorporates the ZuCo 1.0 and 2.0 datasets, capturing text and EEG features during Normal Reading (NR) and Task-Specific Reading (TSR) modes. Textual content is drawn from movie reviews and Wikipedia articles. EEG data are gathered via a 128-channel system at 500Hz, filtered between 0.1Hz and 100Hz; post-preprocessing, 105 channels remain. Word-level EEG features are generated by synchronising raw EEG data with eye-tracking fixation points on each word. Fixations recorded during reading aid in aligning EEG data with text. Aligned EEG segments are segmented into eight frequency bands: theta1 (4–6Hz), theta2 (6.5–8 Hz), alpha1 (8.5–10 Hz), alpha2 (10.5–13 Hz), beta1 (13.5–18Hz), beta2 (18.5–30 Hz), gamma1 (30.5–40 Hz), and gamma2 (40–49.5 Hz), resulting in an EEG segment with dimensions (105,8) corresponding to each word. The dataset was then split into a train set (80%), validation set (10%), and test set (10%). For training a 5-fold cross-validation strategy was applied for each experiment dataset as well as ablation scenarios, as such the results in the tables are an average across each fold. To determine if the results were statistically significant relative to the baseline model results a t-test was applied with a p-value = 0.05.

### 5.1 Implementation details

The model, implemented using PyTorch [17], was trained on a server equipped with an NVIDIA Tesla A100 GPU boasting 40GB of memory. Losses for both the generator and discriminator were tracked across epochs spanning 10, 20, 40, 80, 100, 120, and 140. The checkpoint at epoch 100 was chosen for data augmentation, as the model's loss reached a plateau beyond this point. Word embedding dimensions for word-level, contextual, and sentence-level generation were set to 50, 150, and 2850, respectively. The learning rates for both the discriminator and generator were fixed at $2e^{-5}$ throughout training, with a batch size of 64. Random noise input $z$ was sampled from a normal Gaussian distribution within the range [0, 1] as a single vector of size 100.

### 5.2 Ablation Analysis

To assess the impact of dataset augmentation on task performance, we propose two ablation analyses. In the first scenario, "Noise," we investigate whether the EEG samples generated contribute to any performance increases observed in downstream use cases. Here, augmentation proceeds as usual, but instead of generating a new synthetic EEG sample $\mathcal{E}_f$, we obtain a random noise EEG sample $\mathcal{E}_{rnd}$ with matching dimensions by sampling from a normal Gaussian distribution within the range [0, 1]. By comparing the "Noise" results with those of the synthetic EEG augmentation, we gain insights into the effectiveness of our generation. In the second scenario, "Misalignment," we examine whether the synthetic EEG samples contain semantic information that correlates with their target labels. Augmentation proceeds as normal, generating a new synthetic EEG sample. However, before appending the sample to the task dataset, the EEG sample $\mathcal{E}_f$ is randomly assigned an incorrect label from the vocabulary. We expect that the performance of misaligned generated EEG samples will be lower than that of correctly labelled EEG samples.

## 6 Use case 1 : Named Entity Recognition

**Models.** The first use case we applied our augmentation approach to was an EEG Named Entity Recognition dataset. For this task we used named entity labels[1] to identify entities in the training, validation, and test datasets, categorising them into person, organisation, and location types. Due to the lack of an open-source Named Entity Recognition EEG classifier, we developed a Bidirectional Long Short Term Memory (BLSTM) classifier. BLSTM models have previously shown success in Named Entity Recognition and EEG classification [21, 10].

**Experimental Procedure.** To determine the optimal augmentation scale for Named Entity Recognition, we incrementally augmented synthetic data by 5%, 10%, 15%, 20%, and 25%, appending it to the original training dataset. These values were chosen in particular as during preliminary experiments we observed continuous performance drop off from an augmentation size of 25% up to 100%.

**Evaluation Metric.** Classification accuracy was used to evaluate the baseline and augmented performance. Classification accuracy, defined as the ratio of the sum of true positives ($TP$) and true negatives ($TN$) to the total number of examples ($N$), was used to evaluate both the baseline and augmented performance. Mathematically, the classification accuracy $A$ can be expressed as $A = \frac{TP+TN}{N}$, where $TP$ is the number of true positive predictions, $TN$ is the number of true negative predictions, and $N$ is the total number of examples. This metric provides a straightforward measure of the classifier's performance by indicating the proportion of correctly classified instances out of the entire dataset.

### 6.1 Results

Examining the results from Table 1 highlights that the Contextual Generation method with the TF-IDF-Medium Augmentation strategy and a 10% augmentation size achieved the highest performance, averaging an accuracy of 0.617 (0.018), which is a 0.066 improvement over the baseline. Performance trends across different augmentation sizes indicate average improvements over the baseline of 0.039, 0.046, 0.039, and 0.023 for augmentation sizes of 5%, 10%, 15%, and 20% respectively. Conversely, a 25% augmentation size resulted in a 0.003 decrease below the baseline. Improvements over the baseline for Word, Contextual, and Sentence Generation-Levels were 0.027, 0.036, and 0.023 respectively. Analysing the performance of each Augmentation Strategy reveals improvements over the baseline: Random (0.037), TF-IDF-Low (0.011), TF-IDF-Medium (0.041), and TF-IDF-High (0.025). From the ablation scenario NER-Noise, the average score across all augmentation strategies and sizes is 0.514 (0.02), with the highest score of 0.545 achieved using TF-IDF-Medium at 5% augmentation, 0.06 lower than the baseline. Average scores for augmentation sizes of 5%, 10%, 15%, 20%, and 25% are 0.542, 0.529, 0.516, 0.504, and 0.483, respectively. In the NER-Misalign scenario, the highest score was 0.568 using Contextual-Level Generation with TF-IDF-Medium at 10% augmentation, a 0.017 increase over the baseline. For Word, Contextual, and Sentence Generation-Levels, average scores decreased by 0.021, 0.012, and 0.024, respectively.

## 7 Use case 2 : Sentiment Classification

**Experimental Procedure.** The second use case involved the augmenting of a sentiment classification EEG dataset. To do so we utilised a BLSTM model initially designed for Named Entity Recognition (Section 6). Sentiment labels for the train, validation, and test datasets were derived from a pre-trained RoBERTa-Large model fine-tuned for sentiment classification of English text [9]. This model provided probabilities for sentences being positive or negative. We applied the same augmentation

[1] https://github.com/DS3Lab/ner-at-first-sight

Table 1: The performance of the augmentation at each Generation-Level (Gen-Level), Augmentation Strategy (Aug-Strat), and Augmentation Size (Aug-Size) for sentiment (SEN) and named entity (NER) classification. Includes the performance of the Noise and Misalignment (Misalign) ablation. Values in parenthesis are the performance % change compared to the baseline and * denotes statistically significant results compared to baseline 0% augmentation size.

| Gen-Level | Aug-Strat | Aug-Size | SEN-Accuracy | SEN-Noise | SEN-Misalign | NER-Accuracy | NER-Noise | NER-Misalign |
|---|---|---|---|---|---|---|---|---|
| N/A | N/A | 0% | 0.755 | 0.755 | 0.755 | 0.551 | 0.551 | 0.551 |
| Word | Random | 5% | 0.791 (4.8) | 0.731 (-3.2) | 0.772 (2.3) | 0.597 (8.3)* | 0.541 (-1.8) | 0.549 (-0.4) |
| | | 10% | 0.793 (5.0) | 0.729 (-3.4) | 0.773 (2.4) | 0.602 (9.3)* | 0.529 (-4.0) | 0.554 (0.5) |
| | | 15% | 0.801 (6.1)* | 0.713 (-5.6) | 0.783 (3.7) | 0.592 (7.4)* | 0.515 (-6.5) | 0.544 (-1.3) |
| | | 20% | 0.789 (4.5) | 0.693 (-8.2)* | 0.769 (1.9) | 0.578 (4.9) | 0.504 (-8.5)* | 0.53 (-3.8) |
| | | 25% | 0.784 (3.8) | 0.676 (-10.5)* | 0.765 (1.3) | .568 (3.1) | 0.487 (-11.6)* | 0.52 (-5.6)* |
| | TF-IDF-Low | 5% | 0.787 (4.2)* | 0.734 (-2.8) | 0.762 (0.9) | 0.565 (2.5) | 0.539 (-2.2) | 0.517 (-6.2)* |
| | | 10% | 0.791 (4.8) | 0.728 (-3.6) | 0.771 (2.1) | 0.574(4.2) | 0.527 (-4.4) | 0.526 (-4.5)* |
| | | 15% | 0.784 (3.8) | 0.711 (-5.8)* | 0.767 (1.6) | 0.573 (4.0) | 0.516 (-6.4) | 0.525 (-4.7)* |
| | | 20% | 0.777 (2.9) | 0.699 (-7.4)* | 0.757 (0.3) | 0.559 (1.5) | 0.509 (-7.6)* | 0.511 (-7.3)* |
| | | 25% | 0.768 (1.7) | 0.687 (-9.0)* | 0.748 (-0.9) | 0.519 (-5.8) | 0.483 (-12.3)* | 0.471 (-14.5)* |
| | TF-IDF-Medium | 5% | 0.79 (4.6)* | 0.731 (-3.2) | 0.77 (2.0) | 0.601 (9.1)* | **0.545 (-1.1)** | 0.553 (0.4) |
| | | 10% | 0.802 (6.2)* | 0.726 (-3.8) | 0.782 (3.6) | 0.605 (9.8)* | 0.532 (-3.4) | 0.557 (1.1) |
| | | 15% | 0.805 (6.6)* | 0.717 (-5.0)* | 0.785 (4.0) | 0.597 (8.3)* | 0.519 (-5.8) | 0.549 (-0.4) |
| | | 20% | 0.782 (3.6) | 0.705 (-6.6)* | 0.762 (0.9) | 0.582 (5.6)* | 0.506 (-8.2)* | 0.534 (-3.1) |
| | | 25% | 0.783 (3.7) | 0.698 (-7.5)* | 0.763 (1.1) | 0.563 (2.2) | 0.489 (-11.3)* | 0.515 (-6.5)* |
| | TF-IDF-High | 5% | 0.787 (4.2)* | **0.741 (-1.9)** | 0.767 (1.6) | 0.592 (7.4)* | 0.541 (-1.8) | 0.544 (-1.3) |
| | | 10% | 0.797 (5.6)* | 0.729 (-3.4) | 0.777 (2.9) | 0.601 (9.1)* | 0.528 (-4.2) | 0.553 (0.4) |
| | | 15% | 0.804 (6.5)* | 0.713 (-5.6)* | 0.786 (4.1) | 0.586 (6.4) | 0.517 (-6.2) | 0.538 (-2.4) |
| | | 20% | 0.781 (3.4) | 0.702 (-7.0)* | 0.761 (0.8) | 0.576 (4.5) | 0.498 (-9.6)* | 0.528 (-4.2) |
| | | 25% | 0.776 (2.8) | 0.694 (-8.1)* | 0.752 (-0.4) | 0.548 (-0.5) | 0.474 (-14.0)* | 0.5 (-9.3)* |
| Contextual | Random | 5% | 0.793 (5.0) | 0.731 (-3.2) | 0.773 (2.4) | 0.601 (9.1)* | 0.541 (-1.8) | 0.553 (0.4) |
| | | 10% | 0.799 (5.8) | 0.729 (-3.4) | 0.779 (3.2) | 0.611 (10.9)* | 0.529 (-4.0) | 0.563 (2.2) |
| | | 15% | 0.807 (6.9)* | 0.713 (-5.6) | 0.787 (4.2) | 0.602 (9.3)* | 0.515 (-6.5) | 0.554 (0.5) |
| | | 20% | 0.794 (5.2) | 0.693 (-8.2)* | 0.774 (2.5) | 0.589 (6.9) | 0.504 (-8.5)* | 0.541 (-1.8) |
| | | 25% | 0.786 (4.1) | 0.676 (-10.5)* | 0.766 (1.5) | 0.567 (2.9) | 0.487 (-11.6)* | 0.519 (-5.8) |
| | TF-IDF-Low | 5% | 0.79 (4.6)* | 0.734 (-2.8) | 0.77 (2.0) | 0.592 (7.4)* | 0.539 (-2.2) | 0.544 (-1.3) |
| | | 10% | 0.797 (5.6) | 0.728 (-3.6) | 0.779 (3.2) | 0.589 (6.9)* | 0.527 (-4.4) | 0.541 (-1.8) |
| | | 15% | 0.789 (4.5)* | 0.711 (-5.8)* | 0.769 (1.9) | 0.598 (8.5)* | 0.516 (-6.4) | 0.55 (-0.2) |
| | | 20% | 0.782 (3.6) | 0.699 (-7.4)* | 0.762 (0.9) | 0.571 (3.6) | 0.509 (-7.6)* | 0.523 (-5.1)* |
| | | 25% | 0.77 (2.0) | 0.687 (-9.0)* | 0.75 (-0.7) | 0.526 (-4.5) | 0.483 (-12.3)* | 0.478 (-13.2)* |
| | TF-IDF-Medium | 5% | 0.799 (5.8)* | 0.731 (-3.2) | 0.779 (3.2) | 0.602 (9.3)* | **0.545 (-1.1)** | 0.553 (0.4) |
| | | 10% | 0.805 (6.6)* | 0.726 (-3.8) | 0.785 (4.0) | **0.617 (12.0)*** | 0.532 (-3.4) | **0.568 (3.1)** |
| | | 15% | 0.811 (7.4)* | 0.717 (-5.0)* | 0.791 (4.8) | 0.608 (10.3)* | 0.519 (-5.8) | 0.559 (1.5) |
| | | 20% | 0.796 (5.4)* | 0.705 (-6.6)* | 0.776 (2.8) | 0.597 (8.3)* | 0.506 (-8.2)* | 0.548 (-0.5) |
| | | 25% | 0.788 (4.4) | 0.698 (-7.5)* | 0.768 (1.7) | 0.578 (4.9) | 0.489 (-11.3)* | 0.529 (-4.0) |
| | TF-IDF-High | 5% | 0.787 (4.2)* | **0.741 (-1.9)** | 0.762 (0.9) | 0.592 (7.4)* | 0.541 (-1.8) | 0.544 (-1.3) |
| | | 10% | 0.797 (5.6)* | 0.729 (-3.4) | 0.777 (2.9) | 0.605 (9.8)* | 0.528 (-4.2) | 0.557 (1.1) |
| | | 15% | 0.804 (6.5)* | 0.713 (-5.6)* | 0.783 (3.7) | 0.601 (9.1)* | 0.517 (-6.2) | 0.553 (0.4) |
| | | 20% | 0.781 (3.4) | 0.702 (-7.0)* | 0.761 (0.8) | 0.565 (2.5) | 0.498 (-9.6)* | 0.517 (-6.2)* |
| | | 25% | 0.776 (2.8) | 0.694 (-8.1)* | 0.756 (0.1) | 0.538 (-2.4) | 0.474 (-14.0)* | 0.49 (-11.1)* |
| Sentence | Random | 5% | 0.796 (5.4)* | 0.731 (-3.2) | 0.774 (2.5) | 0.594 (7.8)* | 0.541 (-1.8) | 0.546 (-0.9) |
| | | 10% | 0.808 (7.0)* | 0.729 (-3.4) | 0.788 (4.4) | 0.600 (8.9)* | 0.529 (-4.0) | 0.552 (0.2) |
| | | 15% | 0.813 (7.7)* | 0.713 (-5.6) | 0.793 (5.0)* | 0.589 (6.9)* | 0.515 (-6.5) | 0.541 (-1.8) |
| | | 20% | 0.802 (6.2)* | 0.693 (-8.2)* | 0.781 (3.4) | 0.572 (3.8) | 0.504 (-8.5) * | 0.524 (-4.9) |
| | | 25% | 0.788 (4.4) | 0.676 (-10.5)* | 0.768 (1.7) | 0.558 (1.3) | 0.487 (-11.6) * | 0.51 (-7.4)* |
| | TF-IDF-Low | 5% | 0.79 (4.6)* | 0.734 (-2.8) | 0.773 (2.4) | 0.562 (2.0) | 0.539 (-2.2) | 0.514 (-6.7)* |
| | | 10% | 0.802 (6.2)* | 0.728 (-3.6) | 0.782 (3.6) | 0.571 (3.6) | 0.527 (-4.4) | 0.523 (-5.1)* |
| | | 15% | 0.805 (6.6)* | 0.711 (-5.8)* | 0.785 (4.0) | 0.569 (3.3) | 0.516 (-6.4) | 0.506 (-8.2)* |
| | | 20% | 0.791 (4.8)* | 0.699 (-7.4)* | 0.771 (2.1) | 0.552 (0.2) | 0.509 (-7.6)* | 0.482 (-12.5)* |
| | | 25% | 0.775 (2.6) | 0.687 (-9.0)* | 0.755 (0.0) | 0.514 (-6.7) | 0.483 (-12.3)* | 0.546 (-0.9) |
| | TF-IDF-Medium | 5% | 0.806 (6.8)* | 0.731 (-3.2) | 0.786 (4.1) | 0.598 (8.5)* | **0.545 (-1.1)** | 0.551 (0.0) |
| | | 10% | 0.812 (7.5)* | 0.726 (-3.8) | 0.792 (4.9)* | 0.603 (9.4)* | 0.532 (-3.4) | 0.54 (-2.0) |
| | | 15% | **0.815 (7.9)*** | 0.717 (-5.0)* | **0.796 (5.4)*** | 0.592 (7.4)* | 0.519 (-5.8) | 0.526 (-4.5)* |
| | | 20% | 0.794 (5.2)* | 0.705 (-6.6)* | 0.774 (2.5) | 0.578 (4.9) | 0.506 (-8.2)* | 0.509 (-7.6)* |
| | | 25% | 0.771 (2.1) | 0.698 (-7.5)* | 0.751 (-0.5) | 0.561 (1.8) | 0.489 (-11.3)* | 0.538 (-2.4) |
| | TF-IDF-High | 5% | 0.794 (5.2)* | **0.741 (-1.9)** | 0.772 (2.3) | 0.590 (7.1)* | 0.541 (-1.8) | 0.545 (-1.1) |
| | | 10% | 0.805 (6.6)* | 0.729 (-3.4) | 0.784 (3.8) | 0.597 (8.3)* | 0.528 (-4.2) | 0.541 (-1.8) |
| | | 15% | 0.809 (7.2)* | 0.713 (-5.6)* | 0.789 (4.5) | 0.584 (6.0)* | 0.517 (-6.2) | 0.524 (-4.9) |
| | | 20% | 0.791 (4.8)* | 0.702 (-7.0)* | 0.773 (2.4) | 0.572 (3.8) | 0.498 (-9.6)* | 0.505 (-8.3)* |
| | | 25% | 0.784 (3.8) | 0.694 (-8.1)* | 0.761 (0.8) | 0.542 (-1.6) | 0.474 (-14.0)* | 0.536 (-2.7) |

scaling range used for Named Entity Recognition (Section 6). To evaluate the performance the same classification accuracy metric used in Section 6 was employed.

### 7.1 Results

The results in Table 1 show the highest average accuracy of 0.815 (0.014), a 0.06 improvement over the baseline, achieved with Sentence-Level Generation using TF-IDF-Medium at 15% augmentation. At each augmentation size 5%, 10%, 15%, 20%, and 25% they improve over the baseline by 0.037, 0.045, 0.048, 0.033, and 0.024, respectively. Word, Contextual, and Sentence Generation-Levels show improvements of 0.033, 0.037, and 0.042, respectively. Examining each Augmentation Strategy, Random, TF-IDF-Low, TF-IDF-Medium, and TF-IDF-High improve baseline performance by 0.041, 0.031, 0.042, and 0.036, individually. In the Sent-Noise scenario, the average score across all strategies and sizes is 0.712 (0.017), with the highest score of 0.741 using TF-IDF-High at 5% augmentation, 0.014 lower than the baseline. Average scores for augmentation sizes of 5%, 10%, 15%, 20%, and 25% are 0.734, 0.728, 0.713, 0.699, and 0.688, respectively. In the Sent-Misalign scenario, the highest score was 0.796 using Sentence-Level Generation with TF-IDF-Medium at 15% augmentation, a 0.041 increase over the baseline. Average scores for Word, Contextual, and Sentence Generation-Levels each increased by 0.013, 0.017, and 0.022.

# 8 Use case 3 : EEG-To-Text

**Models.** To evaluate the performance of our synthetic samples for EEG-to-Text decoding, we implemented the Open Vocabulary Electroencephalography-to-Text Decoding model[2] [25] as a baseline for augmentation, as to the best of our knowledge it is the most recent, open source model available. The implemented model was trained without the use of teacher forcing [19].

**Experimental Procedure.** During dataset augmentation, we the dataset with sizes increasing incrementally from 25%, 55%, 75%, to 100%.

**Evaluation Metrics.** BLEU-1 [18] and ROUGE-1-F [13] scores were used to evaluate baseline and augmented performance. **BLEU-1** measures the precision of unigrams between the generated and reference text. It is calculated as:

$$B = \frac{\text{Number of matching unigrams}}{\text{Total number of unigrams in generated text}}$$

**ROUGE-1-F** measures the F1 score based on unigram overlap between the generated and reference text. It is calculated as:

$$R = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

where precision is the ratio of matching unigrams in the generated text to the total number of unigrams in the generated text, and recall is the ratio of matching unigrams to the total number of unigrams in the reference text.

## 8.1 Results

From Table 8.1, the highest BLEU-1 and Rouge-1f scores of 0.167 (0.022) and 0.156 (0.019) were achieved using Sentence-Level Generation with TF-IDF-Medium at 55% augmentation, improving by 0.079 and 0.02 over the baseline. Average BLEU-1 scores for augmentation sizes of 25%, 55%, 75%, and 100% are 0.128, 0.158, 0.127, and 0.117, respectively, while average Rouge-1f scores are 0.121, 0.148, 0.136, and 0.125. Across all Augmentation Strategies, Random, TF-IDF-Low, TF-IDF-Medium, and TF-IDF-High achieved average BLEU-1 scores of 0.133, 0.131, 0.135, and 0.132, and average Rouge-1f scores of 0.131, 0.130, 0.134, and 0.134. In the ablation scenario, Noise BLEU-1 and Rouge-1f scores averaged 0.115 (0.003) and 0.128 (0.002), with highest scores of 0.119 and 0.134 using TF-IDF-High and Random at 75% augmentation. In Sent-Misalign results, average BLEU-1 scores decreased by 0.007, 0.004, and 0.002 for Word, Contextual, and Sentence Generation-Levels, with Rouge-1f decreasing by 0.006 across all levels.

# 9 Discussion and Conclusion

The results presented in Sections 6 and 7 indicate that the optimal augmentation sizes range from 10% to 15%, with performance increasing up to these sizes and declining thereafter. For EEG-To-Text decoding (Section 8), the optimal augmentation size appears to be 55%, addressing research question **(2)**. For Named Entity Recognition (Section 6), Contextual-Level generation is the most effective augmentation method. This effectiveness might be due to the shorter length of named entity sequences, which benefit from contextual information not available at the Word-Level but do not fully utilise Sentence-Level context, addressing research question **(2)**. Similarly, Sentence-Level

[2] https://github.com/MikeWangWZHL/EEG-To-Text

Table 2: The performance of the augmentation at each Generation-Level (Gen-Lvel), Augmentation Strategy (Aug-Strat), and Augmentation Size (Aug-Size) for EEG-To-Text Decoding. Includes the performance of the Noise and Misalignment (Misalign) ablation. Values in parenthesis are the performance % change compared to the baseline and * denotes statistically significant results compared to baseline 0% augmentation size.

| Gen-Level | Aug-Strat | Aug-Size | BLEU-1 | Rouge-1-f | Noise BLEU-1 | Noise Rouge-1-f | Misalign BLEU-1 | Misalign Rouge-1-f |
|---|---|---|---|---|---|---|---|---|
| N/A | N/A | 0% | 0.088 | 0.136 | 0.088 | 0.136 | 0.088 | 0.136 |
| Word | Random | 25% | 0.125 (42.0) | 0.121 (-11.0) | 0.116 (31.8) | 0.129 (-5.1) | 0.077 (-12.5) | 0.131 (-3.7) |
| | | 55% | 0.154 (75.0)* | 0.145 (6.6) | 0.117 (33.0) | 0.132 (-2.9) | 0.106 (20.5) | 0.134 (-1.5) |
| | | 75% | 0.124 (40.9) | 0.135 (-0.7) | **0.121 (37.5)** | **0.134 (-1.5)** | 0.076 (-13.6) | **0.136 (0.0)** |
| | | 100% | 0.116 (31.8) | 0.127 (-6.6) | 0.115 (30.7) | 0.13 (-4.4) | 0.068 (-22.7) | 0.132 (-2.9) |
| | TF-IDF-Low | 25% | 0.121 (37.5) | 0.122 (-10.3) | 0.114 (29.5) | 0.127 (-6.6) | 0.073 (-17.0) | 0.129 (-5.1) |
| | | 55% | 0.152 (72.7)* | 0.137 (0.7) | 0.119 (35.2) | 0.129 (-5.1) | 0.104 (18.2) | 0.131 (-3.7) |
| | | 75% | 0.125 (42.0) | 0.132 (-2.9) | 0.12 (36.4) | 0.126 (-7.4) | 0.077 (-12.5) | 0.128 (-5.9) |
| | | 100% | 0.117 (33.0) | 0.124 (-8.8) | 0.112 (27.3) | 0.122 (-10.3) | 0.069 (-21.6) | 0.124 (-8.8) |
| | TF-IDF-Medium | 25% | 0.124 (40.9) | 0.122 (-10.3) | 0.113 (28.4) | 0.131 (-3.7) | 0.076 (-13.6) | 0.133 (-2.2) |
| | | 55% | 0.158 (79.5)* | 0.148 (8.8) | 0.118 (34.1) | 0.13 (-4.4) | 0.11 (25.0) | 0.132 (-2.9) |
| | | 75% | 0.126 (43.2)* | 0.137 (0.7) | 0.119 (35.2) | 0.127 (-6.6) | 0.078 (-11.4) | 0.129 (-5.1) |
| | | 100% | 0.117 (33.0) | 0.129 (-5.1) | 0.115 (30.7) | 0.121 (-11.0) | 0.069 (-21.6) | 0.123 (-9.6) |
| | TF-IDF-High | 25% | 0.12 (36.4) | 0.121 (-11.0) | 0.111 (26.1) | 0.132 (-2.9) | 0.072 (-18.2) | 0.134 (-1.5) |
| | | 55% | 0.154 (75.0)* | 0.144 (5.9) | 0.112 (27.3) | 0.133 (-2.2) | 0.106 (20.5) | 0.135 (-0.7) |
| | | 75% | 0.126 (43.2)* | 0.136 (0.0) | 0.119 (35.2) | 0.126 (-7.4) | 0.078 (-11.4) | 0.128 (-5.9) |
| | | 100% | 0.114 (29.5) | 0.125 (-8.1) | 0.114 (29.5) | 0.123 (-9.6) | 0.066 (-25.0) | 0.125 (-8.1) |
| Contextual | Random | 25% | 0.126 (43.2) | 0.122 (-10.3) | 0.116 (31.8) | 0.129 (-5.1) | 0.078 (-11.4) | 0.131 (-3.7) |
| | | 55% | 0.159 (80.7)* | 0.15 (10.3) | 0.117 (33.0) | 0.132 (-2.9) | 0.111 (26.1) | 0.134 (-1.5) |
| | | 75% | 0.127 (44.3) | 0.138 (1.5) | **0.121 (37.5)** | **0.134 (-1.5)** | 0.079 (-10.2) | 0.136 (0.0) |
| | | 100% | 0.12 (36.4) | 0.129 (-5.1) | 0.115 (30.7) | 0.13 (-4.4) | 0.072 (-18.2) | 0.132 (-2.9) |
| | TF-IDF-Low | 25% | 0.127 (44.3)* | 0.125 (-8.1) | 0.114 (29.5) | 0.127 (-6.6) | 0.079 (-10.2) | 0.129 (-5.1) |
| | | 55% | 0.154 (75.0)* | 0.147 (8.1) | 0.119 (35.2) | 0.129 (-5.1) | 0.106 (20.5) | 0.131 (-3.7) |
| | | 75% | 0.122 (38.6) | 0.135 (-0.7) | 0.12 (36.4) | 0.126 (-7.4) | 0.074 (-15.9) | 0.128 (-5.9) |
| | | 100% | 0.117 (33.0) | 0.125 (-8.1) | 0.112 (27.3) | 0.122 (-10.3) | 0.069 (-21.6) | 0.124 (-8.8) |
| | TF-IDF-Medium | 25% | 0.129 (46.6)* | 0.125 (-8.1) | 0.113 (28.4) | 0.131 (-3.7) | 0.081 (-8.0) | 0.133 (-2.2) |
| | | 55% | 0.161 (83.0) | 0.153 (12.5)* | 0.118 (34.1) | 0.13 (-4.4) | 0.113 (28.4) | 0.132 (-2.9) |
| | | 75% | 0.13 (47.7)* | 0.14 (2.9) | 0.119 (35.2) | 0.127 (-6.6) | 0.082 (-6.8) | 0.129 (-5.1) |
| | | 100% | 0.122 (38.6) | 0.131 (-3.7) | 0.115 (30.7) | 0.121 (-11.0) | 0.074 (-15.9) | 0.123 (-9.6) |
| | TF-IDF-High | 25% | 0.126 (43.2)* | 0.123 (-9.6) | 0.111 (26.1) | 0.132 (-2.9) | 0.078 (-11.4) | 0.134 (-1.5) |
| | | 55% | 0.158 (79.5)* | 0.151 (11.0)* | 0.112 (27.3) | 0.133 (-2.2) | 0.106 (20.5) | 0.135 (-0.7) |
| | | 75% | 0.127 (44.3)* | 0.138 (1.5) | 0.119 (35.2) | 0.126 (-7.4) | 0.075 (-14.8) | 0.128 (-5.9) |
| | | 100% | 0.117 (33.0) | 0.127 (-6.6) | 0.114 (29.5) | 0.123 (-9.6) | 0.069 (-21.6) | 0.125 (-8.1) |
| Sentence | Random | 25% | 0.135 (53.4) | 0.117 (-14.0) | 0.116 (31.8) | 0.129 (-5.1) | 0.081 (-8.0) | 0.131 (-3.7) |
| | | 55% | 0.163 (85.2)* | 0.149 (9.6) | 0.117 (33.0) | 0.132 (-2.9) | 0.114 (29.5) | 0.134 (-1.5) |
| | | 75% | 0.131 (48.9) | 0.131 (-3.7) | **0.121 (37.5)** | **0.134 (-1.5)** | 0.082 (-6.8) | 0.136 (0.0) |
| | | 100% | 0.12 (36.4) | 0.118 (-13.2) | 0.115 (30.7) | 0.13 (-4.4) | 0.071 (-19.3) | 0.132 (-2.9) |
| | TF-IDF-Low | 25% | 0.131 (48.9)* | 0.116 (-14.7) | 0.114 (29.5) | 0.127 (-6.6) | 0.082 (-6.8) | 0.129 (-5.1) |
| | | 55% | 0.163 (85.2)* | 0.147 (8.1) | 0.119 (35.2) | 0.129 (-5.1) | 0.114 (29.5) | 0.131 (-3.7) |
| | | 75% | 0.129 (46.6) | 0.131 (-3.7) | 0.12 (36.4) | 0.126 (-7.4) | 0.08 (-9.1) | 0.128 (-5.9) |
| | | 100% | 0.116 (31.8) | 0.121 (-11.0) | 0.112 (27.3) | 0.122 (-10.3) | 0.067 (-23.9) | 0.124 (-8.8) |
| | TF-IDF-Medium | 25% | 0.138 (56.8)* | 0.12 (-11.8) | 0.113 (28.4) | 0.131 (-3.7) | 0.089 (1.1) | 0.133 (-2.2) |
| | | 55% | **0.167 (89.8)*** | **0.156 (14.7)*** | 0.118 (34.1) | 0.13 (-4.4) | **0.116 (31.8)** | 0.132 (-2.9) |
| | | 75% | 0.135 (53.4) | 0.137 (0.7) | 0.119 (35.2) | 0.127 (-6.6) | 0.084 (-4.5) | 0.129 (-5.1) |
| | | 100% | 0.121 (37.5) | 0.119 (-12.5) | 0.115 (30.7) | 0.121 (-11.0) | 0.07 (-20.5) | 0.123 (-9.6) |
| | TF-IDF-High | 25% | 0.134 (52.3) | 0.126 (-7.4) | 0.111 (26.1) | 0.132 (-2.9) | 0.083 (-5.7) | 0.134 (-1.5) |
| | | 55% | 0.162 (84.1)* | 0.153 (12.5)* | 0.112 (27.3) | 0.133 (-2.2) | 0.111 (26.1) | 0.135 (-0.7) |
| | | 75% | 0.131 (48.9) | 0.142 (4.4) | 0.119 (35.2) | 0.126 (-7.4) | 0.08 (-9.1) | 0.128 (-5.9) |
| | | 100% | 0.118 (34.1) | 0.129 (-5.1) | 0.114 (29.5) | 0.123 (-9.6) | 0.067 (-23.9) | 0.125 (-8.1) |

Generation is optimal for both Sentiment Analysis (Section 7) and EEG-To-Text augmentation (Section 8). The longer sequences in these tasks allow the generator to produce higher quality synthetic samples. Across all tasks, TF-IDF-Medium emerges as the optimal strategy. It potentially balances the distribution of underrepresented words in the vocabulary more effectively than TF-IDF-Low or TF-IDF-High, addressing research question **(3)**. The results also suggest that noise introduction reduces overall model performance. Misalignment of synthetic EEG samples with text labels decreases augmentation effectiveness, while correctly aligned synthetic EEG samples enhance performance. This indicates that semantically aligned synthetic samples are crucial, addressing research question **(1)**.

In conclusion, this paper addresses the scarcity of extensive EEG datasets for NLP by introducing CREATOR, a GAN-based model that generates synthetic EEG samples conditioned on text input. Our approach significantly enhances performance in Named Entity Recognition, Sentiment Classification, and EEG-to-Text decoding tasks, with improvements of 6.6%, 6.1%, and 7.9%, respectively, due to the integration of CREATOR-generated samples. This study represents an initial step towards leveraging synthetic EEG data for training advanced neurolinguistic models, offering a promising solution to data collection challenges in EEG-based NLP research. Future work will focus on refining CREATOR's performance by incorporating more complex text embeddings, such as Bidirectional Encoder Representations from Transformers (BERT), and using techniques like contrastive learning to align these embeddings with EEG signals, thereby producing synthetic samples that more accurately capture semantics.

## References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[2] Tonio Ball, Markus Kern, Isabella Mutschler, Ad Aertsen, and Andreas Schulze-Bonhage. 2009. Signal quality of simultaneously recorded invasive and non-invasive EEG. *Neuroimage* 46, 3 (2009), 708–716.

[3] Kenneth Ward Church. 2017. Word2Vec. *Natural Language Engineering* 23, 1 (2017), 155–162.

[4] Marcos Del Pozo-Banos, Jesús B Alonso, Jaime R Ticay-Rivas, and Carlos M Travieso. 2014. Electroencephalogram subject identification: A review. *Expert Systems with Applications* 41, 15 (2014), 6537–6554.

[5] Yiqun Duan, Jinzhao Zhou, Zhen Wang, Yu-Kai Wang, and Chin-Teng Lin. 2023. Dewave: Discrete eeg waves encoding for brain dynamics to text translation. *arXiv preprint arXiv:2309.14030* (2023).

[6] Zijun Gao, Lingbo Li, and Tianhua Xu. 2023. Data augmentation for time-series classification: An extensive empirical study and comprehensive survey. *arXiv preprint arXiv:2310.10060* (2023).

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[8] Jacek Gwizdka, Rahilsadat Hosseini, Michael Cole, and Shouyi Wang. 2017. Temporal dynamics of eye-tracking and EEG during reading and relevance decisions. *Journal of the Association for Information Science and Technology* 68, 10 (2017), 2299–2312.

[9] Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. 2023. More than a Feeling: Accuracy and Application of Sentiment Analysis. *International Journal of Research in Marketing* 40, 1 (2023), 75–87. https://doi.org/10.1016/j.ijresmar.2022.05.005

[10] Nora Hollenstein, Maria Barrett, Marius Troendle, Francesco Bigiolli, Nicolas Langer, and Ce Zhang. 2019. Advancing NLP with cognitive language processing signals. *arXiv preprint arXiv:1904.02682* (2019).

[11] Gregg D Jacobs and Richard Friedman. 2004. EEG spectral analysis of relaxation techniques. *Applied psychophysiology and biofeedback* 29 (2004), 245–254.

[12] Ahmad Khodayari-Rostamabad, James P Reilly, Gary Hasey, Hubert Debruin, and Duncan MacCrimmon. 2010. Diagnosis of psychiatric disorders using EEG data and employing a statistical decision model. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 4006–4009.

[13] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[14] Yun Luo and Bao-Liang Lu. 2018. EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2535–2538. https://doi.org/10.1109/EMBC.2018.8512865

[15] Yun Luo and Bao-Liang Lu. 2018. EEG data augmentation for emotion recognition using a conditional Wasserstein GAN. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2535–2538.

[16] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute* 334, 2 (1997), 307–318.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[18] Matt Post. 2018. A call for clarity in reporting BLEU scores. *arXiv preprint arXiv:1804.08771* (2018).

[19] Shaik Rafi and Ranjita Das. 2021. RNN encoder and decoder with teacher forcing attention mechanism for abstractive summarization. In *2021 IEEE 18th India council international conference (INDICON)*. IEEE, 1–7.

[20] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *International conference on machine learning*. PMLR, 1060–1069.

[21] Yuqi Ren and Deyi Xiong. 2021. CogAlign: Learning to align textual neural representations to cognitive language processing signals. *arXiv preprint arXiv:2106.05544* (2021).

[22] Cédric Rommel, Joseph Paillard, Thomas Moreau, and Alexandre Gramfort. 2022. Data augmentation for learning predictive models on EEG: a systematic comparison. *Journal of Neural Engineering* 19, 6 (2022), 066020.

[23] Kaleb E Smith and Anthony O Smith. 2020. Conditional GAN for timeseries generation. *arXiv preprint arXiv:2006.16477* (2020).

[24] Yayat Sudaryat, Jatmika Nurhadi, and Rosita Rahma. 2019. Spectral topographic brain mapping in EEG recording for detecting reading attention in various science books. *Journal of Turkish Science Education* 16, 3 (2019), 440–450.

[25] Zhenhailong Wang and Heng Ji. 2022. Open vocabulary electroencephalography-to-text decoding and zero-shot sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5350–5358.

[26] Lilian Weng. 2019. From gan to wgan. *arXiv preprint arXiv:1904.08994* (2019).

[27] Aiming Zhang, Lei Su, Yin Zhang, Yunfa Fu, Liping Wu, and Shengjin Liang. 2021. EEG data augmentation for emotion recognition with a multiple generator conditional Wasserstein GAN. *Complex & Intelligent Systems* (2021), 1–13.

[28] Xiang Zhang, Lina Yao, Quan Z Sheng, Salil S Kanhere, Tao Gu, and Dalin Zhang. 2018. 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)*. IEEE, 1–10.