EEGtoText: Learning to Write Medical Reports from EEG Recordings

Siddharth Biswal SBISWAL7@GATECH.EDU
School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA, USA

Cao Xiao CAO.XIAO@IQVIA.COM
Analytics Center of Excellence
IQVIA
Cambridge, MA, USA

M. Brandon Westover MWESTOVER@MGH.HARVARD.EDU
Department of Neurology
Massachusetts General Hospital
Boston, MA, USA

Jimeng Sun JSUN@CC.GATECH.EDU
School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA, USA

Abstract

Electroencephalography (EEG) is widely used in hospitals and clinics for the diagnosis of many neurological conditions. Such diagnoses require accurate and timely clinical reports to summarize the findings from raw EEG data. In this paper, we investigate whether it is possible to automatically generate text reports directly from EEG data. To address the challenges, we proposed `EEGtoText`, which first extracted shift invariant and temporal patterns using stacked convolutional neural networks and recurrent neural networks (RCNN). These temporal patterns are used to classify key phenotypes including EEG normality, sleep, generalized and focal slowing, epileptiform discharges, spindles, vertex waves and seizures. Based on these phenotypes, the impression section of the EEG report is generated. Next, we adopted a hierarchical long short-term memory network(LSTM) that comprises of paragraph-level and sentence-level LSTMs to generate the detail explanation of the impression. Within the hierarchical LSTM, we used an attention module to localize the abnormal areas in the EEG which provide another explanation and justification of the extracted phenotypes.

We conducted large-scale evaluations on two different EEG datasets Dataset1 (n=12,980) and TUH (n=16,950). We achieved an area under the ROC curve (AUC) between .658 to .915 on phenotype classification, which is significantly higher than CRNN and RCNN with attention. We also conducted a quantitative evaluation of the detailed explanation, which achieved METEOR score .371 and BLEU score 4.583. Finally, our initial clinical reviews confirmed the effectiveness of the generated reports.

1. Introduction

Electroencephalography (EEG) in the form of multivariate time series are widely used in hospitals for the diagnosis of neurological conditions including seizure disorders, sleep disorders, and brain disorders(Nuwer, 1997). Typically neurologists will visually inspect EEG signals that measure the brain activity to reflect the condition of the brain (Bagic et al., 2011), and then compose text report to narrate the abnormal patterns (i.e., *EEG phenotypes*) and detailed explanation of those phenotypes. The clinical report writing is cumbersome and labor intensive. Moreover, it requires a thorough knowledge and extensive experience in understanding the EEG phenotypes, and how they evolve over time, and their correlations with target diseases (Organization et al., 2004). To alleviate the limitation of manual report writing, we investigate the approach of automatically generating EEG reports. This task involves solving the following challenges.

- *Capturing the complex patterns of the data.* The complexity of pattern extraction from raw EEG data includes encoding variable length EEG records, handling pattern shifts, and capturing temporal patterns.

- *Generate structured reports.* EEG reports are structured hierarchically, with high-level impressions that summarize the key phenotypes, and a detailed description of each impression.

- *Focused and interpretable narration.* Reports need to provide intuitive understanding to facilitate clinical decision making. For example, it is preferred that the report can connect these output impressions and descriptions to raw EEG patterns.

Although there are a number of image/video captioning (Vincent et al., 2008; Xu et al., 2015b) or medical image reporting generation methods (Jing et al., 2017b; Li et al., 2018) being proposed, they tend to perform poorly on clinical time series report generation due to two primary reasons. First, they often take fixed size data (e.g., images) as input and generate short sentences (e.g., image caption), whereas clinical reports often consist of multiple paragraphs based on variable-length input data(EEG). Second, clinical reports need to align with clinical guidelines and have special structures, which cannot be acquired by simply applying the aforementioned methods.

To fill in the gap, we propose `EEGtoText`, a framework that learns to generate hierarchically structured EEG reports given variable-length EEG recording as input. In particular, `EEGtoText` is carefully designed as follows.

1. *High-capacity encoding.* `EEGtoText` first extracts features using stacked convolutional neural networks and recurrent neural networks for capturing shift invariant and temporal patterns, respectively. These features were then used to generate key phenotypes via a multi-label classification module. The module outputs multi-label keyword such as "abnormal", "sleep", "generalize theta slowing" etc.

2. *Structured report generation with hierarchical LSTM.* `EEGtoText` adopts a hierarchical long-short term memory networks (LSTM) that comprises of paragraph-LSTM and sentence-LSTM to generate the longer-form parts of the medical report (details and impression).

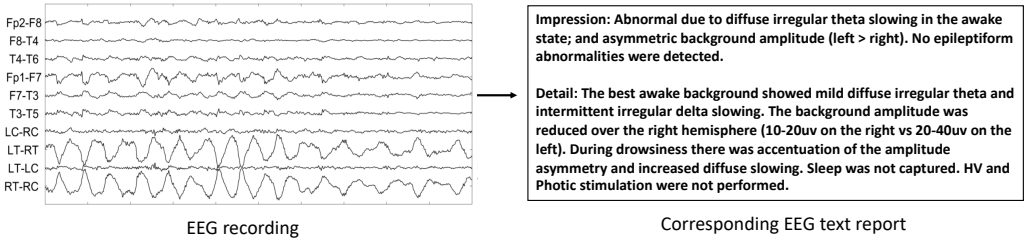

Figure 1: An example of medical report generation from EEG recording. The left box shows EEG recording and right box show corresponding EEG text report. We propose our `EEGtoText` framework which take an EEG input and produce text reports as shown in the right side of the diagram.

3. *Focused and interpretable narration with attention module.* Within the hierarchical LSTM, `EEGtoText` has an attention module to localize the regions in the detailed text description and abnormal areas in raw EEG that explain the corresponding phenotypes.

- **Clinical Relevance**: Our proposed framework `EEGtoText` works towards the goals of helping a cumbersome and labor intensive EEG report writing task. `EEGtoText` can extract different EEG phenotypes and generate impression and details sections of the EEG reports. This will also be highly beneficial in low-resource clinical settings where there is a shortage of expert clinicians.
- **Technical Significance**: In this work, our framework `EEGtoText` can understand variable length input such as EEG to produce multiple short paragraphs using an attention based encoding of input features and hierarchical LSTM based decoders. Using the combination of phenotype classification and detail generation, our method is able to outperform all other baselines to produce high-quality EEG text reports.

We conduct extensive empirical studies to demonstrate the efficiency of `EEGtoText` on two large-scale EEG datasets. In the evaluation of phenotype classification, our results show the `EEGtoText` outperformed the second best approach (CRNN with Attention) by 3.7% in Spindle phenotype and by 1.4% on average across all tasks. In the evaluation of detailed explanation, `EEGtoText` achieved METEOR score 0.398 and BLEU score 4.532. We also perform qualitative evaluation of generated EEG reports by expert clinicians. This shows that our proposed framework `EEGtoText` has the potential to able to generate high quality EEG text reports from EEG datasets.

2. Related Work

Deep Learning in EEG Analysis Recent years, the analysis of EEG signals has gone beyond traditional two step EEG feature engineering and classification (Lin et al., 2008; Alomari et al., 2014; Shoeb and Guttag, 2010; Al-Fahoum and Al-Fraihat, 2014) to end-to-end learning. Deep neural networks have been applied to both raw EEG signals and spectrogram representations (Bashivan et al., 2015; Tibor Schirrmeister et al., 2017; Schirrmeister et al., 2017; Biswal et al., 2017; Thodoroff et al., 2016) or on clinical texts (Maldonado

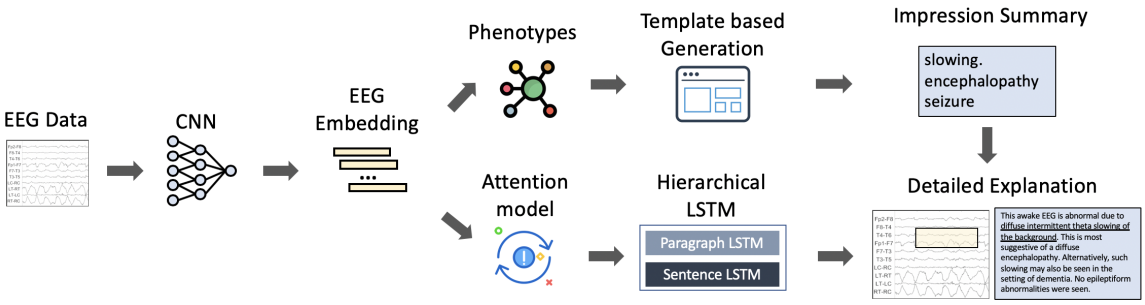

Figure 2: An overview of the proposed framework for generating EEG text report from EEG sample. We use a CNN to create feature vectors from the EEG sample. These feature vectors are used for phenotype classification and these phenotypes are used in Impression generation. These feature vectors are passed to the attention module to generate a final context vector for details decoder. The decoder generates text reports from the encoded representation

et al., 2017) and demonstrated promising results in various health analytics tasks. While careful feature engineering has led to good performances, it has been difficult to extend these methods to wide variety of EEG related tasks.

Caption Generation There has been a large body of works that build connection between visual data (e.g., images or videos) and textual data (e.g., summary or caption) in computer vision domain(Vinyals et al., 2015; Xu et al., 2015b; Rennie et al., 2017; Zhou et al., 2017). For example, image caption generation using recurrent neural networks with attention (Xu et al., 2015b), or dense caption generation with considering region-of-interest of the image (Johnson et al., 2016). Some authors have shown that it is possible to generate video captions by encoding video features using either a recurrent encoder (Donahue et al., 2015; Venugopalan et al., 2015; Xu et al., 2015a) or an attention model (Yao et al., 2015). Some works have focused on dense video captioning task where event are captioned in the video (Zhou et al., 2018).

Medical Text Generation There has been increased interest in generating reports from medical data such as images in the past years. There are two primary works in this subdomain of medical image captioning which use X-ray images to produce short descriptions (Li et al., 2018; Jing et al., 2017a; Li et al., 2019). However, these methods cannot be easily extended to time series(EEG) text report generation due to a few different reasons. The primary difference is that EEG recordings are of variable length input, which poses different challenges compared to X-ray images. Also, EEG reports usually contains two different sections "Impression section" and "Detail section" which range from two to three paragraphs which is longer compared to X-ray text reports.

Table 1: Notations used in `EEGtoText`

Notation	Definition
$\mathbf{D}^{(i)} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{I}$	i-th EEG sample, $i = 1, 2...I$
$\mathbf{X}^{(i)} \in \mathbb{R}^{C \times T}$	i-th EEG sample, where i=1,2... I samples, we denote it further as X ignoring i
$\mathbf{Y}^{(i)} = (\mathbf{S}_1^i, \mathbf{S}_2^i, \ldots, \mathbf{S}_J^i)$	i-th paragraph/report sample, where i=1,2... I samples, we denote it further as Y ignoring i
$\mathbf{x}_i$	i-th EEG epoch, where $i = 1, 2, \ldots, T$ samples
$\mathbf{f}_i$	i-th EEG feature vector extracted from i-th EEG epoch, where $i = 1, 2, \ldots, T$ samples
$\mathbf{S}_j$	j-th sentence in report, where $j = 1, 2, \ldots, J$ samples, we denote it further as S_m ignoring i
$\mathbf{g}_t^{(p)}$	t-th attention vector for t-th EEG epoch and p-th pass over attention module
$\mathbf{M}^{(p)}$	pth pass over the attention module

3. Methods

3.1. Data and Task Definitions

Data We denote EEG samples as $\mathbf{D}^{(i)} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{I}$. Here $\mathbf{X}^{(i)} \in \mathbb{R}^{C \times T}$ is the EEG records for subject i, where C is number of electrodes and T is the number of discretized time steps per recording. We represent the text report to be generated as a sequence of sentences $\mathbf{Y}^{(i)} = (\mathbf{S}_1^i, \mathbf{S}_2^i, \ldots, \mathbf{S}_J^i)$ and each sentence is a sequence of words and V is the complete vocabulary of all words in the EEG reports.

Task Given an EEG sample $\mathbf{X}^{(i)}$, our goal is to generate an EEG text report consisting of a sequence of sentences $\mathbf{Y}^{(i)} = (\mathbf{S}_1^i, \mathbf{S}_2^i, \ldots, \mathbf{S}_J^i)$ to narrate the patterns and findings in $\mathbf{X}^{(i)}$. This is done by maximizing the conditional probability of output sentences given input EEG record and is parameterized by context vector $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} = \arg\min \sum logP(\mathbf{Y}^{(i)}|\mathbf{X}^{(i)}, \boldsymbol{\theta})$.

3.2. The `EEGtoText` Framework

In this section, we propose `EEGtoText` framework that can generate structured medical reports given clinical EEG time series data of a patient. Given multivariate time series EEG records $\mathbf{x}_i$ as input, `EEGtoText` firstly encodes them into feature vectors $\mathbf{f}_1 \ldots \mathbf{f}_T$, then predicts different impression phenotypes (i.e., themes of EEG patterns such normal, seizure etc) based on the encoded EEG feature vectors $\mathbf{f}_i$. Next These impression phenotypes are be passed as input into a template based method to generate the impression section of the report which provides high-level structures of the EEG recording. The feature vectors generated are combined with the representation phenotype representations to generate context vectors. Then the framework uses these context vectors to generate the detailed explanation using hierarchical (paragraph- and sentence-) RNN 2. Below are the detailed description of

each module.

Data Encoding Given EEG records as input, we first embed them into feature vectors $\mathbf{f}_1 \dots \mathbf{f}_T$ using neural networks. In particular, an EEG sample $\mathbf{X}$ consists of multiple EEG epochs, each at length 1 minute. We represent these EEG epochs as $\mathbf{X} = \mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_T$ and embed each $\mathbf{x}_i$ into a feature vector $\mathbf{f}_i$ such that

$$\mathbf{f}_i = \mathbb{CNN}(\mathbf{x}_i). \tag{1}$$

In this work, we adopt a CNN with convolutional-max pooling blocks for processing the EEG data into feature space. We use Rectified Linear Units(ReLUs) activation function for these convolutional networks, and with batch normalization (Szegedy et al., 2013).

Attention based neural networks have been successful in directing neural network to focus on specific parts of input to produce accurate classification and high dimentional output. Inspired by the design of dynamic memory networks (Xiong et al., 2016), we designed our architecture to perform multiple passes over the attention module to create the final context vector. This final context vector is passed to the decoder module. In this attentional EEG encoder module, we process the context vector along with the keywords produced by the EEG classification module. Intuitively, $\mathbf{f}_1 \dots \mathbf{f}_T$ can be considered as all the information contained in the EEG and multi pass attention model extract facts or information from the the $\mathbf{f}_1 \dots \mathbf{f}_T$ to create the context vector $\mathbf{m}^{(p)}$.

These EEG feature vectors are multiplied with another representation $\mathbf{q}$ from keywords $\mathbf{K}$ generated in the EEG keyword classification module, where $\mathbf{q}$. The module for classifying the EEG vector into Keywords $\mathbf{K}$ is described in next section. $\mathbf{q}$ is representation of the keywords predicted and provides understanding of important patterns present in the EEG data. Our intuition is that these keywords provide the required guidance for generating context vector for detailed explanation generation.

We first calculate an attention vector g_i^p using EEG encoder feature vectors F $=\mathbf{f}_1 \dots \mathbf{f}_T$ and embedding vector generated from keywords K. As noted this is for p-th pass over memory module.

$$\mathbf{z}_i^{(p)} = [\mathbf{f}_i \odot \mathbf{q}; \mathbf{f}_i \odot \mathbf{m}^{(p-1)}] \tag{2}$$

$$\mathbf{Z}_i^{(p)} = \mathbb{MLP}(z_i^{(p)}) \tag{3}$$

$$g_i^{(p)} = \frac{\exp(\mathbf{Z}_i^p)}{\sum_{k=1}^{M_i} \exp(\mathbf{Z}_k^{(p)})} \tag{4}$$

where $\odot$ is the element-wise product and ; represents concatenation of the vectors. Since this is the episodic memory, we go over multiple passes over the attention module. We denote the passes over the attention module using superscript (p). After obtaining $g_i^{(p)}$, we use a soft attention mechanism to obtain $c^{(p)}$. This soft attention mechanism performs a weighted summation of EEG feature vectors $\mathbf{f}_i$ with corresponding attention gates g_i^p.

$$c^{(p)} = \sum \mathbf{f}_i g_i^p \tag{5}$$

We use multiple passes of update to produce the output of this module $\mathbf{m}^{(p)}$. This memory module gets updated with newly constructed context vector $\mathbf{c}^{(p)}$, producing $\mathbf{m}^{(p)}$.

We use a gated recurrent units(GRU) to compute this context vector $\mathbf{c}^{(p)}$. The input to the GRU are current context vector $\mathbf{c}^{(p)}$ and $\mathbf{m}^{(p-1)}$. The episodic memory $\mathbf{m}^{(p)}$ for pass p is computed using this following equation.

$$\mathbf{m}^{(p)} = \mathbb{GRU}(\mathbf{c}^{(p)}, \mathbf{m}^{(p-1)}) \tag{6}$$

Impression Phenotype Classification In this module, we use the feature vectors learned by the EEG encoder $\mathbf{f}_1 \dots \mathbf{f}_T$ to produce the impression phenotypes associated with the EEG recording. These phenotypes K are passed to the template to fill in the template to produce the impression section of the report. We denote these impression phenotypes as K which are further used in the memory modules using a different representation. The phenotypes which are used as labels this work are (1) Normal (2) sleep (3) Drowsiness (4) Generalized Slowing (5) Focal Slowing (6) Epileptiform discharges (7) Spindles (8) Vertex Waves (9) Seizures. We use the EEG feature vectors f_t produced by EEG feature encoder as as input for fully connected layer to produce the impression phenotype K.

$$P(\mathbf{K}) = \sigma(\mathbb{MLP}(\mathbf{f}_i)) \tag{7}$$

where σ indicates sigmoid function.

As mentioned earlier, EEG text reports contains two primary sections e.g., "Impression section", "Details section". Impression section contains a summary of the report, so it is usually very concise and contains specific keywords. We have taken a template based approach to generate the Impression section of the EEG report. This template was extracted manually by reviewing EEG text reports. The impression phenotypes K predicted in this module is used with the templates to create this "Impression section" of the report.

Detailed Explanation Generation We take an approach where we stack a paragraph generator module on top of a sentence generator module. The paragraph generator is responsible for capturing the inter-sentence dependencies and guides the sentence generator. The sentence generator is built using a combination of LSTM for language model, multi-layer perceptron of integrating information and an attention model. The paragraph generator is responsible for deciding the numbers of sentences and W-dimensional topic vector for each of these sentences. Given a topic vector for a sentence, the sentence generator generates words for that sentence.

Paragraph Generator: The paragraph generator is a LSTM with initial hidden and cell states set to zero. It receives EEG feature vector which is processed using the memory module. So the input for the paragraph generator is $\mathbf{m}^{(p)}$, and in turn produces a sequence of hidden states $\boldsymbol{h}_j \in \mathbb{R}^H$. These hidden states are each for sentence in the paragraph. First, a linear projection from $\boldsymbol{h}_j$ and logistic classifier produce a distribution p_j to decide continue or stop. This decides if jth sentence is the last sentence of the paragraph. Then we pass the hidden state $\boldsymbol{h}_j$ through a two layer MLP to produce a topic vector $\mathbf{O}_j \in \mathbb{R}^P$. This topic vector $\mathbf{O}_j$ is passed to sentence generator as the input.

Sentence Generator: This sentence generator module is another long short term memory(LSTM) network with hidden size H =512, which takes a topic vector $\mathbf{O}_j$ as input passed

from the paragraph generator. A $START$ token is also passed with topic vector to the start input. The subsequent inputs are learned embeddings(hidden layer output) for the words. At each time-step the hidden state of the last LSTM layer is used to predict the distribution over the words in the vocabulary. We produce an END token to indicate the end of the sentence. At the end of we concatenate the sentences to form the paragraph.

3.3. Training and Inference

Our training loss $l(x, y)$ for the sample (x, y) is cross-entropy term on the final output word distribution and ground truth words.

$$J(Y|X;\boldsymbol{\theta}) = -\sum_{t=1}^{N} p_t(Y_t|X;\boldsymbol{\theta}) \tag{8}$$

where $p_t(Y_t)$ is the probability of observing the correct word Y_t at time t. The loss is minimized with respect to parameters in the set $\boldsymbol{\theta}$ which are the parameters of the end to end model including CNN encoder, memory module and decoder module. At inference time, we sample from the from the decoder using the feature representation encoded using CNN encoder and memory module.

4. Experiments

In this section, we describe our experiments and show that `EEGtoText` is able to generate text reports from EEG recordings. We evaluate the results using both quantitative and qualitative measures. We first introduce the two different datasets(Dataset1 and TUH) used in our experiments and then discuss different experiments and analysis performed in evaluation `EEGtoText`.

Table 2: Dataset Statistics

	Dataset1 Data	TUH Data
Number of Patients	10,890	10,865
Number of EEG Samples	12,980	16,950
Total EEG length	4,523 hours	3,452 hours
Total number of Tokens	755,019	542,765

4.1. Data

We conducted experiments using the following datasets. More details about data are provided in table 2.

1. **Dataset1 EEG Report Data set**: The dataset was collected at large medical center. It contains 12,980 deidentified EEG recordings paired with text reports, which were collected over four years of time.

2. **TUH EEG Report Data set**: We also evaluated our methods using the dataset from Temple University Hospital EEG corpus (Obeid and Picone, 2016). It contains 16,950 sessions from 10,865 unique subjects.

EEG Report Preprocessing For EEG text reports, it contains multiple sections as impression, patient history, comparison, details section, EKG analysis. Here we only focus on impression and detail sections since they outline patient conditions and are the target for our report generation task. In addition, we also process the reports by tokenizing and converting to lower-cases. We remove some tokens from the corpus if the frequency is less than two.

4.2. Experimental Setup

We implemented `EEGtoText` in PyTorch 1.0 (Paszke et al., 2017).We use Adam (Kingma and Ba, 2014) with batch size of 64 samples. We use a machine equipped with Intel Xeon e5-2640, 256GB RAM, eight Nvidia Titan-X GPU and CUDA 8.0. While training the models, we use batch size of 128 and ADAM as the optimization method. For Adam to optimize all models and the learning rate is selected from [2e-3, 1e-3, 7.5e-4] and β_1 is selected from [0.5, 0.9]. We train all models for 500 epochs. We start to half the learning rate every 2 epochs after epoch 30. We used 10% of the dataset as a validation set for tuning hyperparameters of each model. We provide more details about hyperparameters in table 7 in supplementary section. We provide the model configuration information in table 7. We searched for different model parameters using random search method.

Baselines: We compare `EEGtoText` with the following baselines.

1. **Mean-pooling(MP)**: We use CNN to extract features for different EEG segments and combined using mean pooling. This mean pooled feature vector is passed to an 2 layered LSTM to produce the text reports for EEG samples. (Venugopalan et al., 2014).

2. **S2VT**: We apply a sequence to sequence model which reads CNN outputs using an LSTM and uses another LSTM to produce text reports.(Venugopalan et al., 2015)

3. **Temporal Attention Network(TAM)**: In this model, we use CNN to learn EEG features, and then pass them to a decoder equipped with temporal attention which allows focusing on different EEG segments to produce the text report (Yao et al., 2015).

4. **Soft Attention(SA)** In this methods, we use a soft attention mechanism to allow the decoder to be able to focus on EEG feature representations(Bahdanau et al., 2014).

Evaluation Metrics

To evaluate the phenotype classification component, we use area under the receiver operating characteristic curve **ROC-AUC** and area under the Precision-Recall Curve **PR-AUC**. To evaluate report generation quality, we use BLEU, METEOR, and CIDEr which are commonly used to evaluate image/video caption generation tasks. Since METEOR is always better than BLEU and ROUGE in terms of consistency with human judgement (Kilickaya et al., 2016), here we consider METEOR as our primary metric for evaluating EEG text generation.

- **METEOR(M)**: Metric for Evaluation of Translation with Explicit Ordering (METEOR) metric is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision (Banerjee and Lavie, 2005). Higher METEOR scores are considered better in different NLP tasks.

- **CIDEr**: CIDEr measures the consensus between candidate model generated description and the reference sentences (Vedantam et al., 2015). Similar to METEOR score higher CIDEr score is considered better.
- **BLEU**: BLEU score is defined as the geometric mean of n-gram precision scores multiplied by a brevity penalty for short sentence(Papineni et al., 2002). Brevity penalty gives a penalty on sentences which are shorter than the reference thus it prevents shorter sentences from obtaining higher score. This score ranges from 0-100% and higher is considered better.

4.3. Experiment 1: Impression Phenotype classification

We first evaluate the effectiveness in predicting target EEG keywords or phenotypes. In clinical setting, these phenotypes can provide very quick insight into neurological state of the patient. Here we describe the following EEG phenotypes.

- **Normality**: This defines if the EEG recording was considered as normal or abnormal.
- **Sleep**: It indicates if any sleep pattern was observed during the EEG recording duration.
- **Generalized Slowing (Gen. Slowing)**: Generalized slowing indicates if there was any observed slowing of EEG pattern. Slowing is generally associated with different diffuse encephalopathies.
- **Focal Slowing**: It indicates focal dysfunction found in the EEG recording.
- **Epileptiform Discharges(Epi Discharges)**: Epileptiform discharges indicates the presence of different spikes, sharp waves, triphasic waves, lateral periodic discharges(LPD), generalized periodic discharges(GPD), generalized spike and wave patterns.
- **Drowsiness**: Drowsiness indicates very slow frequency of 0.25 to 1.0 Hz in the frontal and lateral frontal channels seen in the EEG.
- **Spindles**: Spindles are bursts of oscillatory brain activity generated in the reticular nucleus of the thalamus that occur during sleep.
- **Vertex Waves**: Vertex waves are seen in sleep stages I and II. These are usually seen as focal sharp transients in the EEG recording.
- **Seizure**: Seizures indicate any presence of seizure like activities in the EEG recording.

The baseline models in evaluating phenotype prediction components are listed below.

- **CRNN**: In this baseline model, a RNN is used to process all feature vectors $f_1 \dots f_T$ from CNN encoder. We take a the final step of the RNN and pass it through two fully connected layers to produce the classification results.
- **CRNN with attention**: Based on aforementioned CRNN model, it has additional attention mechanism attached to the RNN layer to produce the hidden layer outputs.

Table 3: **Classification Performances for Phenotype classification task on Dataset1 Test set**

	ROC-AUC			PR-AUC		
	`EEGtoText`	CRNN	CRNN w/ Att	`EEGtoText`	CRNN	CRNN w/ Att
Normality	**0.915**	0.884	0.893	**0.891**	0.869	0.855
Sleep	**0.849**	0.803	0.825	**0.824**	0.771	0.781
Gen Slowing	**0.763**	0.743	0.756	**0.775**	0.710	0.728
Focal Slowing	0.684	0.678	**0.686**	**0.673**	0.665	0.671
Epi Discharges	**0.798**	0.765	0.786	**0.751**	0.731	0.743
Drowsiness	**0.748**	0.728	0.742	**0.753**	0.712	0.731
Spindles	**0.668**	0.626	0.631	**0.642**	0.614	0.634
Vertex Waves	**0.658**	0.643	0.637	**0.628**	0.614	0.626
Seizure	**0.794**	0.761	0.778	**0.775**	0.728	0.745

Table 4: **Classification Performances for Phenotype classification task on TUH Test set**

	ROC-AUC			PR-AUC		
	`EEGtoText`	CRNN	CRNN w/ Att	`EEGtoText`	CRNN	CRNN w/ Att
Normality	**0.878**	0.843	0.858	**0.871**	0.845	0.853
Sleep	**0.810**	0.773	0.784	**0.801**	0.751	0.789
Gen Slowing	**0.751**	0.701	0.726	0.732	0.715	**0.749**
Focal Slowing	**0.612**	0.581	0.591	**0.621**	0.572	0.602
Epi Discharges	**0.734**	0.704	0.725	**0.718**	0.695	0.716
Drowsiness	**0.753**	0.731	0.739	**0.748**	0.738	0.744
Spindles	**0.631**	0.608	0.619	0.621	0.592	**0.623**
Vertex Waves	**0.614**	0.582	0.603	0.601	0.571	**0.615**
Seizure	**0.743**	0.705	0.726	**0.739**	0.712	0.721

`EEGtoText` has the phenotype classification module described earlier in the paper which classifies the EEG recording into distinct phenotype categories. We describe the results of this classification experiment in table 3.

Both Tables 3 and 4 show that `EEGtoText` outperforms the baselines in phenotype classification tasks for most of the labels. The performance of `EEGtoText` varies across different phenotypes quite a bit and it can be attributed to the distribution of the labels for these specific phenotypes. We also observed that with larger sample size `EEGtoText` is able to perform better compared other baselines. With more data for those specific phenotypes such as Spindles, Vertex Waves, Focal Slowing, the overall performance will become better.

4.4. Experiment 2: Impression Section Generation

The impression section includes a few sentences to summarize the overall EEG recording for the specific patient. Here we evaluate the capability of `EEGtoText` in impression generation based on the METEOR, CIDEr, BLEU scores. The obtained results are described in table 5.

We can readily see that `EEGtoText` outperforms all baseline models across all datasets except TAM method in TUH dataset for METEOR and BLEU@2 metric. The performance

Table 5: **Automatic evaluation results on Dataset1 and TUH Testset for Impression Section generation task. First row presents performance on Dataset1 and second row on TUH data set.**

Method	METEOR	CIDEr	B@1	B@2	B@3	B@4
MP	0.323	0.367	0.714	0.644	0.563	0.443
	0.345	0.363	0.645	0.578	0.459	0.361
S2VT	0.325	0.319	0.741	0.628	0.529	0.462
	0.361	0.364	0.724	0.613	0.543	0.438
TAM	0.382	0.334	0.749	0.668	0.581	0.378
	0.369	0.381	0.714	**0.647**	0.492	0.461
SA	0.394	0.348	0.684	0.629	0.568	0.472
	0.353	0.341	0.736	0.619	0.519	0.420
`EEGtoText`	**0.428**	**0.384**	**0.759**	**0.704**	**0.596**	**0.492**
	0.358	**0.455**	**0.758**	0.639	**0.611**	**0.483**

gap between `EEGtoText` and closest baselines is around 3.5% for METEOR metric. This experiment confirms that `EEGtoText` is able to generate high quality impression sections of the EEG recordings better than other methods. Since we use a template based method to generate the EEG Impression section, these results indicate that using the inherent structure present in the report helps our method to produce high qulaity impression section of the EEG report.

4.5. Experiment 3: Detail Description Generation

Detail sections of EEG reports usually contain granular details of the EEG recording in terms of important neurological events happening during the recording. The event level details create challenges for the task. To evaluate its quality, we leverage the same set of metrics as in evaluating impression generation. The results are summarized in table 6.

Our results indicate that our model performs better than the baselines, showing that our proposed framework is able to generate meaningful details section text from EEG recording. We achieve METEOR score of 0.371 and 0.381 on dataset1 and TUH dataset which is better than all other baselines. Similarly, we also outperform other methods in terms of BLEU@4 scores. In order to generate more specific detailed description, in future work. we would attempt to train our models at detailed events level.

4.6. Experiment 4: Qualitative Clinician Evaluation

In order to understand usefulness of our models for clinical practice we evaluated the results by expert neurologist. We provided the experts with samples with ground truth report and generated text report presented side by side. In this setup, we measure two metrics for the generated texts. Expert neurologists are asked to provide two of the following scores.

- Quality score: This score is to evaluate overall quality of the generated report. The possible range of this score is 0-5.

Table 6: **Automatic evaluation results on Dataset1 and TUH Testset for Detail Description Generation Task. First row presents performance on Dataset1 and second row on TUH data set.**

Method	METEOR	CIDEr	B@1	B@2	B@3	B@4
MP	0.216	0.326	0.638	0.567	0.442	0.349
	0.291	0.338	0.669	0.581	0.422	0.317
S2VT	0.228	0.365	0.542	0.521	0.418	0.378
	0.217	0.385	0.475	0.421	0.384	0.328
TAM	0.319	0.378	0.642	0.586	0.541	0.432
	0.364	0.392	0.662	0.562	0.465	0.362
SA	0.275	0.321	0.562	0.525	0.468	0.427
	0.261	0.326	0.583	0.534	0.443	0.352
EEGtoText	**0.371**	**0.473**	**0.794**	**0.751**	**0.648**	**0.583**
	0.381	**0.457**	**0.832**	**0.736**	**0.626**	**0.512**

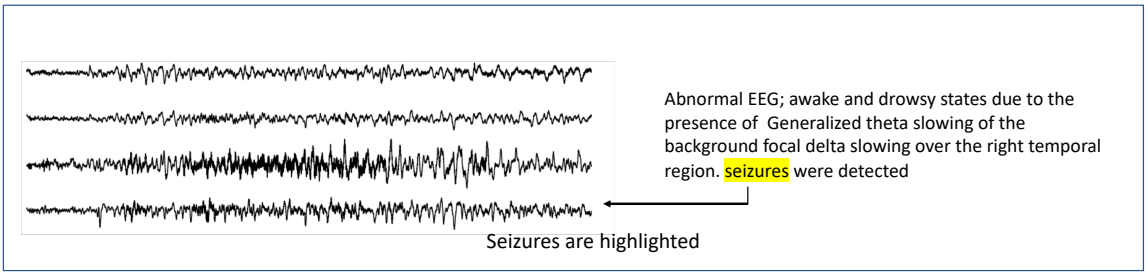

Figure 3: Examples showing EEG plot with generated report. In this plot, we highlight the location with Seizures being attention module as shown in the diagram. Due to lack of space we only 4 EEG channels 'FP1-F3','F3-C3','C3-P3','P3-O1'.

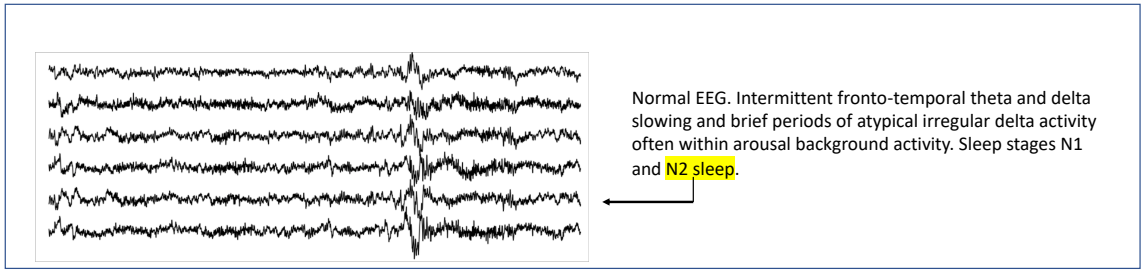

Figure 4: This example shows another EEG plot with generated report. In this plot, we highlight the location with sleep phenotype being highlighted by module as shown in the diagram.

- Agreement score: This score indicates the agreement of the labels(keywords) present in the generated report with the ground truth report. Similar to quality score, the range of this score is 0-5 too.

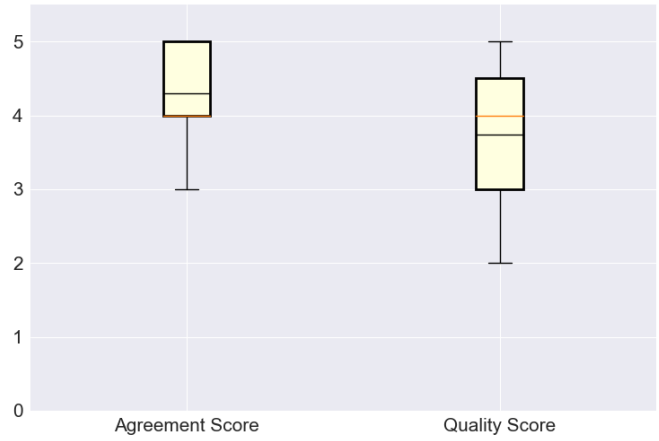

Figure 5: Plot showing results of qualitative evaluation of generated reports by clinicians

These two metrics provide indication that these generated report and classification outputs are useful for clinical purposes. We obtained average agreement score of 4.16 and average quality score of 3.75 in our clinical evaluation experiment as shows in figure 5.

5. Conclusion

In this work, we presented `EEGtoText`, a framework for understanding and generating EEG text reports given EEG recording as inputs. In our extensive experimental evaluation, we showed that `EEGtoText` performed well in phenotype classification tasks. We have also showed that `EEGtoText` is able to generate impression and details section of the EEG reports. As we have showed that `EEGtoText` can generate detailed EEG reports, in future, we plan to extend `EEGtoText` to capture more granular details present in the EEG reports to produce higher quality reports such as location, frequency of different patterns. Finally, our proposed `EEGtoText` with it's capacity to generate text reports can help aid in the neurology clinical workflow for understanding and diagnosis of neurological conditions from EEG recordings.

References

Amjed S Al-Fahoum and Ausilah A Al-Fraihat. Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains. *ISRN neuroscience*, 2014, 2014.

Mohammad H Alomari, Emad A Awada, Aya Samaha, and Khaled Alkamha. Wavelet-based feature extraction for the analysis of eeg signals associated with imagined fists and feet movements. *Computer and Information Science*, 7(2):17, 2014.

Anto I Bagic, Robert C Knowlton, Douglas F Rose, John S Ebersole, ACMEGS Clinical Practice Guideline (CPG) Committee, et al. American clinical magnetoencephalography society clinical practice guideline 1: recording and analysis of spontaneous cerebral activity. *Journal of Clinical Neurophysiology*, 28(4):348–354, 2011.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Pouya Bashivan, Irina Rish, Mohammed Yeasin, and Noel Codella. Learning representations from eeg with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*, 2015.

Siddharth Biswal, Joshua Kulas, Haoqi Sun, Balaji Goparaju, M Brandon Westover, Matt T Bianchi, and Jimeng Sun. Sleepnet: automated sleep staging system via deep learning. *arXiv preprint arXiv:1707.08262*, 2017.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

Baoyu Jing, Pengtao Xie, and Eric Xing. On the automatic generation of medical imaging reports. *arXiv preprint arXiv:1711.08195*, 2017a.

Baoyu Jing, Pengtao Xie, and Eric P. Xing. On the automatic generation of medical imaging reports. *CoRR*, abs/1711.08195, 2017b. URL http://arxiv.org/abs/1711.08195.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. *arXiv preprint arXiv:1612.07600*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Christy Y Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. Knowledge-driven encode, retrieve, paraphrase for medical image report generation. *arXiv preprint arXiv:1903.10122*, 2019.

Yuan Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. Hybrid retrieval-generation reinforced agent for medical image report generation. In *Advances in Neural Information Processing Systems*, pages 1537–1547, 2018.

Yuan-Pin Lin, Chi-Hong Wang, Tien-Lin Wu, Shyh-Kang Jeng, and Jyh-Horng Chen. Support vector machine for eeg signal classification during listening to emotional music. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 127–130. IEEE, 2008.

R. Maldonado, TR. Goodwin, and SM. Harabagiu. Memory-augmented active deep learning for identifying relations between distant medical concepts in electroencephalography reports. In *AMIA Joint Summits on Translational Science proceedings.*, pages 156–165, 2017.

Marc Nuwer. Assessment of digital eeg, quantitative eeg, and eeg brain mapping: report of the american academy of neurology and the american clinical neurophysiology society. *Neurology*, 49(1):277–292, 1997.

Iyad Obeid and Joseph Picone. The temple university hospital eeg data corpus. *Frontiers in neuroscience*, 10:196, 2016.

World Health Organization et al. Neurology atlas 2004. *URL www. who. int/mentalhealth/neurology/neurology_atlas_review_references. pdf*, 2004.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

Ali H Shoeb and John V Guttag. Application of machine learning to epileptic seizure detection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 975–982, 2010.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Pierre Thodoroff, Joelle Pineau, and Andrew Lim. Learning robust features using deep learning for automatic seizure detection. In *Machine learning for healthcare conference*, pages 178–190, 2016.

Robin Tibor Schirrmeister, Lukas Gemein, Katharina Eggensperger, Frank Hutter, and Tonio Ball. Deep learning with convolutional neural networks for decoding and visualization of eeg pathology. *arXiv preprint arXiv:1708.08012*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406, 2016.

Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. A multi-scale multiple instance video description network. *arXiv preprint arXiv:1505.05914*, 2015a.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015b.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

Luowei Zhou, Chenliang Xu, Parker Koch, and Jason J Corso. Watch what you just said: Image captioning with text-conditional attention. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 305–313. ACM, 2017.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.

6. Supplement

In the following table we have provided information about different hyperparameters which are used in our model and experiments.

Table 7: Information about different parameters used in the model

Parameter	**Values**
Number of layers in CNN	9
Initial Learning rate	0.1
Activation Function	ReLU
Pooling Layer	Max-Pooling
Paragraph LSTM dimensions	1128
sentence LSTM dimensions	256
dropout information	0.15
regularization information	L1L2(0.01, 0.01)