Brain2Text: A Comparison Study of T5 Transformers and XPhoneBert for the Translation of Phonemes to Text

December 7th, 2025

https://github.com/hyeonggeonkim-berkeley/mids-w266-fall2025-vishnu-geon

Hyeong Geon Kim
hyeonggeonkim@berkeley.edu

Vishnu Gorur
vishnu.gorur@gmail.com

Abstract

Individuals with severe speech impairments often retain language comprehension but cannot produce speech. Recent brain-computer interface (BCI) research suggests that neural activity correlates more strongly with phonemes than with words, motivating a two-stage decoding pipeline: neural signals → phonemes → text. This project focuses on the second stage. We investigate whether a lightweight sequence-to-sequence transformer, T5-small, can translate phoneme sequences into coherent English sentences. T5's performance is also compared against a custom Seq2Seq model. Both models are trained on paired phoneme sentence data and evaluated using BLEU, Word Error Rate (WER), and qualitative error analysis. Results show that both approaches capture phonetic structures and produce fluent, semantically accurate text.

Introduction

Millions of people live with speech impairments caused by stroke, traumatic brain injury, or neurodegenerative disease such as aphasia[1]. Although many retain full cognitive language ability, they cannot articulate speech due to damaged motor pathways[1]. Brain-computer interfaces (BCIs) offer a path toward restoring communication by decoding intended speech directly from neural activity.

A growing body of work shows that neural recordings encode information at a **phonemic** level[9]. This motivates the use of a two-stage decoding pipeline:

1. Predict phonemes from neural activity.
2. Convert predicted arpabet (English ASCII) phonemes into coherent sentences.

While attempts were made for the first phase of this pipeline, most ended in failure. This project focuses on the second stage. Classical phoneme-to-text methods, such as n-gram models or WFSTs, struggle with long-range dependencies and produce limited fluency, especially when phoneme sequences are noisy[4]. Furthermore, most research has been done on Text-to-Phoneme conversion due to the necessity of text-to-speech models.

Transformer-based models, however, excel at sequence transduction and have recently shown strong results in grapheme-to-phoneme [5] and International Phonetic Alphabet; IPA-to-text tasks[8].

The particular class of Transformers this paper focuses on are seq2seq models. The first approach leverages the popular pre-trained T5-small model from Huggingface. The second model uses a pre-trained encoder but a custom decoder.

We train and evaluate these models on a supervised phoneme-to-sentence dataset, comparing its performance to simple baselines using BLEU, WER, and qualitative analysis. Our goal is to assess whether a compact transformer can serve as an effective component in real-time speech-restoration pipelines.

Background

Recent neural approaches demonstrate that transformer models can effectively process phonological inputs. G2P models based on T5 show that pretrained text-to-text transformers outperform rule-based or recurrent architectures in handling phoneme-grapheme relationships[8]. Other work has applied sequence-to-sequence models to IPA-to-English translation or trained phoneme-level transformers (e.g., T5lephone) for spoken language understanding, providing evidence that phonemic representations are compatible with modern transformer architectures[8].

In parallel, BCI research shows that cortical activity encodes speech at a phonemic or articulatory level. Studies by Moses et al.[9] and Anumanchipalli et al.[10] demonstrate that predicted phoneme sequences serve as effective intermediates for decoding intended speech from neural data[9].

The models implemented in this paper were intended for a competition hosted on Kaggle by Nicholas Card[11]. The goal of the competition was to build a pipeline for decoding neural signals into coherent sentences. The baseline approach used a RNN architecture to model the temporal nature of the neural signals collected during trials. This was framed as a multi-class classification problem, with CTC loss, as there were 41 defined classes of phonemes. To validate the predictions of the model, they used a greedy beam search decoding algorithm to convert the predicted logits to phonemes.

In the second phase, the baseline model used a n-gram (5-gram) model with beam search decoding for their language modeling problem.

Phoneme to words by nature is more of a mapping problem than a predictive problem. There is a fundamental dataset used in these problems called the CMUdict which has approximately 134,000 mappings of words to its corresponding phoneme sequence[12]. However, in the case of BCI, there is a high chance of noise in the generated phonemes making the mapping less clear and thus causing a deterministic mapping to result in <UNK> tokens. As such, these lines of work suggest that language modeling techniques are the most reliable method in BCI settings.

Methods

All models were tested with at least 2 final metrics: BLEU score and a Word Error Rate (WER). Whilst WER was the primary evaluation metric in the baseline pipeline, we decided to use a Bleu score in addition to WER. Given that the meaning behind the words is more impactful in a Brain-to-text translation, the Bleu score holds more weight. To still stay true to the competition, we still tracked and reported the WER.

The data for this project was provided on Kaggle by Card for everyone to download. The raw neural data had been preprocessed and normalized ready for analysis. The predicted phonemes, however, were only accessible through the Redis server database. We extracted the predictions with their corresponding ground truth sentence labels and stored them in an external csv for ease of access during experimentation.

Baseline Model

The baseline model acquired from the neuroprosthetics lab via Card et al used a custom 5gram language model requiring 300 GB of dedicated ram plus rescoring via Facebook OPT6.7b containing 6.7b billion parameters. Finetuning was done via the neuroprosthetics lab via 2 RTX 4090s over the course of multiple weeks.
A 1gram model was run to ensure that their pipeline is reproducible. However, due to hardware limitations, the baseline statistics were instead gathered from Card lab's output files via their 5gram model.

Neural Signal Modeling

While not covered in detail in this paper, this section highlights the main attempts at

adapting architectures from other applications to this problem. The two most promising approaches at getting some semblance of comparable decoding to the baseline decoder were the Conformer architecture from Automatic Speech Recognition (ASR)[13] and the RNN-Transformer architecture[14] . In the Conformer approach, it was clear that convolutional 1D layers were not working for un-separable time series. This was something Gated Recurrent Unit layers (GRU) in the RNN was able to capture. After the Conformer approach failed, we tried to use GRU layers with the Conformer [15] to see if that was enough but it did not work.
Our final attempt at this problem was adapting GRU layers to the RNN-transformer, adjusting their code to handle my input dimensions and GRU layers. Unfortunately this did not work as expected. After these iterations, we decided to focus our efforts on Phoneme-to-text as it had a lower barrier to experimentation and there was more groundwork done before us.

Phoneme to Sentences Modeling

T5 Model

The first successful model in our experiments was a pretrained T5-small sequence-to-sequence model, which was fine-tuned using the provided phoneme-to-text dataset. This approach was motivated by prior literature demonstrating the effectiveness of sequence-to-sequence models, including recurrent neural networks, for mapping variable-length input sequences to text outputs (e.g., encoder-decoder LSTM models in early sequence-to-sequence translation work). Building on this foundation, we framed phoneme decoding as a translation problem, where noisy phoneme sequences serve as the input “language” and English text as the output. This framing allows the model to leverage T5’s strong English-language pretraining while learning to map symbolic phoneme inputs to coherent sentences.

Before settling on T5-small, we evaluated multiple variants of T5-based and related models. In total, three notable candidate modeling approaches were explored:

ByT5 (byte-level T5)
We initially experimented with a pretrained ByT5 model released as part of the T5lephone framework, which is designed to process phonemicized text and byte-level representations. ByT5 operates directly on UTF-8 bytes rather than subword tokens[6], making it theoretically robust to unconventional input symbols such as phonemes. However, in practice, this model failed to converge to meaningful outputs under our training setup. Training was computationally expensive due to substantially longer byte-level sequences, and we encountered hardware and memory limitations during fine-tuning. Additionally, the relatively small size of our training dataset (approximately 2,000 samples) likely limited the model’s ability to learn stable alignments at the byte level. As a result, this approach was abandoned.

Wav2Vec-style representations
We also investigated the use of wav2vec-based models[2], which have been used in prior work (including T5lephone) to extract speech representations for downstream language modeling. However, wav2vec models are pretrained on raw audio and are optimized for learning acoustic representations, rather than operating on symbolic phoneme sequences. Because our inputs consisted of phonemes rather than audio, this approach proved ill-suited to our data modality and did not yield meaningful results.

T5-small (subword-level)

Ultimately, we adopted T5-small, a subword-based encoder-decoder Transformer pretrained primarily on English text[3]. Compared to byte-level modeling, T5-small offered significantly lower computational cost, more stable training dynamics, and effective utilization of English-language priors during decoding. When fine-tuned on phoneme-to-text pairs, this model consistently converged and produced coherent sentence-level outputs, making it the most practical and effective choice for our task.

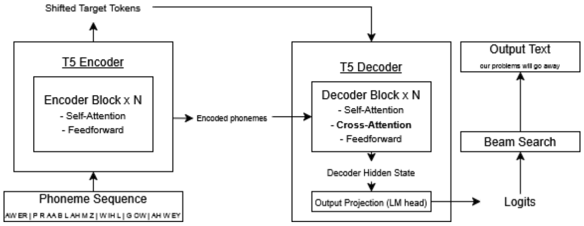


Figure 1. T5-based phone-to-text model architecture

Custom Seq2Seq Model

The other model we built was also an encoder-decoder model. However instead of using a pretrained model, like T5, we conducted a literature review for either pre-trained encoders or pre-trained decoders. The idea behind this approach was to build some sort of hybrid seq2seq model leveraging at-least one pre-trained component.

The first attempt was replicating the LSTM encoder-decoder model architecture described in the following paper[16]. The paper indicated that this architecture worked well for them but after implementation, the model trained very slowly. As a result, it was not a viable architecture for us to iterate and experiment with under the time limits.

Another failed attempt was trying to fine tune OpenAI's Whisper ASR model[17]. The motivation behind this approach was to leverage their decoder and customize their encoder to our task. The challenge was that the encoder was built for a mel spectrogram embedding dimension (3000), whereas the phonemes were a dimension (80) due to the binning of the neural signals. Interpolating the phonemes input dimensions to match allowed for the same data to be fed through the encoder. However the model struggled to learn anything as reflected in its training loss. As a result, the Whisper approach was quickly scrapped in favor of another pre-trained encoder.

The model that found success in this task was an encoder-decoder model that utilized a pre-trained encoder from Huggingface. The encoder, XPhoneBERT, was pre-trained on 330M phoneme sequences[18]. The decoder, on the other hand, was built using the structure of the decoder in the seminal 'Attention is all you need' paper[19]. The outputs from the decoder were also decoded using a beam search algorithm[20]. Importantly, during training, the encoder was frozen during training and it was only the embeddings and decoder that were unfrozen.

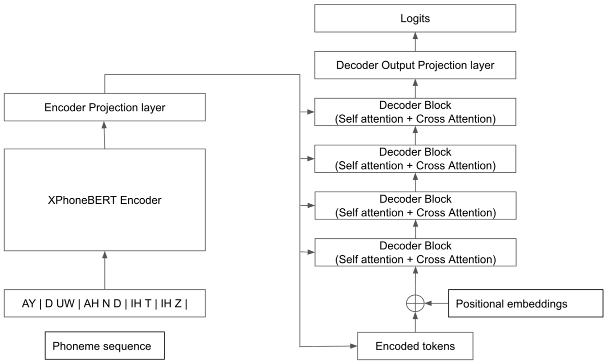

Logits
Decoder Output Projection layer
Decoder Block
(Self attention + Cross Attention)
Decoder Block
(Self attention + Cross Attention)
Decoder Block
(Self attention + Cross Attention)
Decoder Block
(Self attention + Cross Attention)
Positional embeddings
Encoded tokens
Encoder Projection layer
XPhoneBERT Encoder
AY | D UW | AH N D | IH T | IH Z |
Phoneme sequence

Figure 2: XPhoneBERT-Decoder Model architecture.

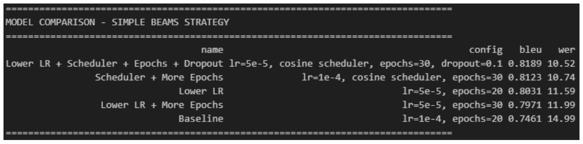
Results and Experiments

5-gram Baseline

As aforementioned, the 5-gram model could not be run due to hardware limitations. However, Card et al provided the output dataset including their respective expected outputs via a .pkl file. From these outputs we were able to gather the baseline statistics of a BLEU score of 0.9179 and a WER score of 4.34%.

T5 Model

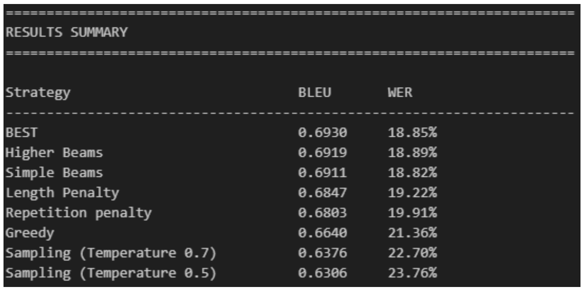
3 main processes for finetuning experimentation were involved when running the T5 model: T5 training arguments, beam configuration, and validating on an unseen test dataset.



```
==============================================================================
MODEL COMPARISON - SIMPLE BEAMS STRATEGY
==============================================================================
                                   name                                              config   bleu   wer
Lower LR + Scheduler + Epochs + Dropout lr=5e-5, cosine scheduler, epochs=30, dropout=0.1 0.8189 10.52
                 Scheduler + More Epochs            lr=1e-4, cosine scheduler, epochs=30 0.8123 10.74
                                Lower LR                              lr=5e-5, epochs=20 0.8031 11.59
                  Lower LR + More Epochs                              lr=5e-5, epochs=30 0.7971 11.99
                                Baseline                              lr=1e-4, epochs=20 0.7461 14.99
==============================================================================
```

Fig 3. T5 Training Experiments

Fig 3 shows the increase in BLEU score over iterations of T5 architecture experimentation. Though evaluation metrics increased with the addition of epochs, dropouts, changes in learning rates, etc., there is, however, a pattern of diminishing return. Once achieving a 0.80 Bleu score, one additional configuration was run to try and mitigate overfitting by adding light dropouts.

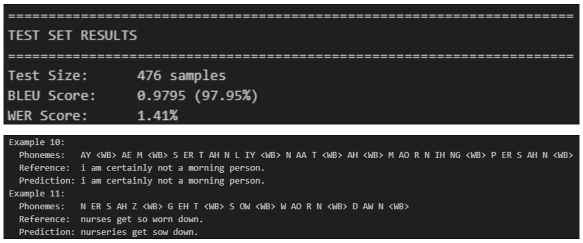


```
======================================================================
RESULTS SUMMARY
======================================================================

Strategy                              BLEU        WER
----------------------------------------------------------------------
BEST                                  0.6930      18.85%
Higher Beams                          0.6919      18.89%
Simple Beams                          0.6911      18.82%
Length Penalty                        0.6847      19.22%
Repetition penalty                    0.6803      19.91%
Greedy                                0.6640      21.36%
Sampling (Temperature 0.7)            0.6376      22.70%
Sampling (Temperature 0.5)            0.6306      23.76%
```

Fig 4. T5 Beam Search Configuration Testing

Beam config experimentation was performed which had an interesting final conclusion. Typically a beam search with more options or a more robust configuration would be expected to perform better. However, in this case, the best solution was to select just a relatively moderate number of beams. This may be due to the T5 model already doing much of the work expected of the beam, and therefore doubling up on the filter may be the cause of the drop in quality for more complex beam operations.



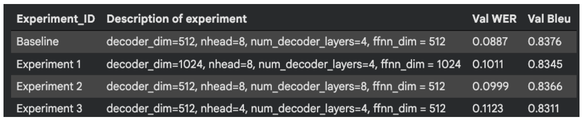
```
======================================================================
TEST SET RESULTS
======================================================================
Test Size:      476 samples
BLEU Score:     0.9795 (97.95%)
WER Score:      1.41%
```

```
Example 10:
  Phonemes:   AY <WB> AE M <WB> S ER T AH N L IY <WB> N AA T <WB> AH <WB> M AO R N IH NG <WB> P ER S AH N <WB>
  Reference:  i am certainly not a morning person.
  Prediction: i am certainly not a morning person.
Example 11:
  Phonemes:   N ER S AH Z <WB> G EH T <WB> S OW <WB> W AO R N <WB> D AW N <WB>
  Reference:  nurses get so worn down.
  Prediction: nurseries get sow down.
```

Fig 5. T5 Test Set Results

Finally, the best combination of T5 and beam was run on a Test set to see results. Though the evaluation metrics came to an astonishing Bleu score of 0.9795 and a WER of 1.41%, this may be skewed due to the random selection of 476 samples being relatively "easy" and we should take into account the quality of the validation results from Figure 4.

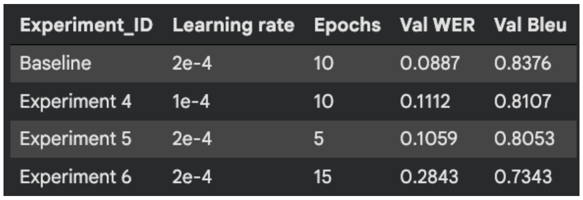
XPhoneBERT-Decoder Model



Experiment_ID	Description of experiment	Val WER	Val Bleu
Baseline	decoder_dim=512, nhead=8, num_decoder_layers=4, ffnn_dim = 512	0.0887	0.8376
Experiment 1	decoder_dim=1024, nhead=8, num_decoder_layers=4, ffnn_dim = 1024	0.1011	0.8345
Experiment 2	decoder_dim=512, nhead=8, num_decoder_layers=8, ffnn_dim = 512	0.0999	0.8366
Experiment 3	decoder_dim=512, nhead=4, num_decoder_layers=4, ffnn_dim = 512	0.1123	0.8311

Fig 6. Decoder architecture experiments

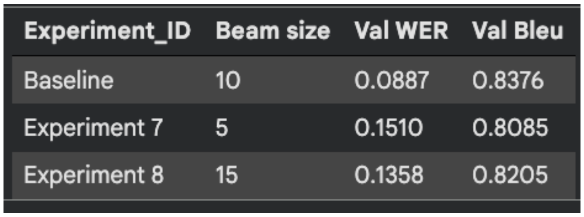
The decoder model levers to tune for a better performance were the number of decoder layers, attention heads and dimensions of the layers themselves. The decoder embedding size was required to be 768 as that was the dimensions set by the XPhoneBERT encoder. Figure 6 illustrates the impact of changing the decoder model's architecture. From these experiments, none of the changes to dimensions of the NNs, layers or attention head made a significant difference over the baseline.



Experiment_ID	Learning rate	Epochs	Val WER	Val Bleu
Baseline	2e-4	10	0.0887	0.8376
Experiment 4	1e-4	10	0.1112	0.8107
Experiment 5	2e-4	5	0.1059	0.8053
Experiment 6	2e-4	15	0.2843	0.7343

Fig 7. Training paradigm experiments

Based on the training experiments, the most noticeable degradation in performance comes from training for 15 epochs with a learning rate fixed at $2x10^{-4}$. This is likely an indication of overfitting to the training dataset and failing to generalize to the unseen examples in the Val dataset.



Experiment_ID	Beam size	Val WER	Val Bleu
Baseline	10	0.0887	0.8376
Experiment 7	5	0.1510	0.8085
Experiment 8	15	0.1358	0.8205

Fig 8. Beam size experiments

The beam size experiments showcase an interesting trend. Intuitively, by increasing the beam size there's a higher possibility of choosing the right next token during decoding and so the WER should be reduced, as expected. However, both changes to beam size resulted in marginal decrease in WER and Bleu score on the unseen data. In (Vaswani et al., 2017) their ideal beam size was 4 and given that our decoder architecture was adapted from their paper, the difference in decoding is likely due to the encoder specialized for our Phoneme-to-text. All these experiments for hyperparameter tuning suggest that the original baseline model is the optimal model as none of the other models showed any noticeable improvement. With a final avg Val WER of 0.08 and avg Val Bleu of 0.8376, the model is able to generalize to the unseen data and make competitive predictions with the other approaches in this space.

Conclusion

After conducting all our Phoneme-to-text experiments, it was clear that the original intended baseline that used the simplest n-gram model performed as good if not better than our best models, based on the WER. This could be due to the fact that these are very short phoneme sequences being translated to sentences (avg length 5 - 15 words). If they are very short then attention might not make too much of a difference. A 5 gram model would be capturing large chunks of the sentence making it sufficient for this purpose. As expected, the 1-gram model, which lacks contextual modeling beyond individual phonemes, performed substantially worse, achieving a WER of 47.85%.
Our experiments show that there are still many orthogonal methods of modeling Phoneme-to-text as seen in the field of ASR and in machine translation. In this paper we showcase the breadth of possible approaches for this task. Additionally the neural signal decoding problem is a far harder task that requires domain knowledge and much more time as it is a niche academic topic. While we thought that Speech Recognition architectures would work for Neural signals, we learned that speech is 1-Dimensional unlike Neural signals making it unable to fully capture dependencies.

Going forward, we believe that the decoded phonemes from neural signals can be improved with more sophisticated approaches than we tested. Similarly, we believe that our approaches can be improved to capture even random words in a phoneme-to-text to sentences.

Appendix

Contributions

This project consisted of a team of 2: Hyeong Geon Kim and Vishnu Gorur. Writing of the report was shared to split work amongst both members, with more focus on the individual's locus.
Hyeong Geon Kim - Ran the initial baseline model, created data ingest script to scrape the hdf5 files to train on, gathered baseline statistics, created T5 models.
Vishnu Gorur - Tested various approaches for the Neural signal problem based on existing literature. Also tested various ASR approaches to Phoneme-to-text modeling before settling on a XPhonebert encoder-decoder model with custom encoder.

References

1. https://www.asha.org/practice-portal/clinical-topics/aphasia/
2. Baevski, A. *et al.* (2020a) *WAV2VEC 2.0: A framework for self-supervised learning of speech representations*, *arXiv.org*. Available at: https://arxiv.org/abs/2006.11477
3. Raffel, C. *et al.* (2023) *Exploring the limits of transfer learning with a unified text-to-text transformer*, *arXiv.org*. Available at: https://arxiv.org/abs/1910.10683
4. Li, K. (2021) *ADAPTATION, CONTEXT-AWARE MODELING AND RESCORING METHODS FOR NEURAL LANGUAGE MODELS IN AUTOMATIC SPEECH RECOGNITION*, *jscholarship*. Available at: https://jscholarship.library.jhu.edu/server/api/core/bitstreams/cf1e80aa-8228-426b-9e3e-6b41f588079d/content
5. Yolchuyeva, S., Németh, G. and Gyires-Tóth, B. (2020) *Transformer based grapheme-to-phoneme conversion*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2004.06338
6. Xue, L. *et al.* (2022) *BYT5: Towards a token-free future with pre-trained byte-to-byte models*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2105.13626
7. Graper, Zane. (2025). IPA-to-Text Transformation Using a Sequence-to-Sequence T5 Model. 10.13140/RG.2.2.10871.28324.
8.Hsu, C.-J. *et al.* (2022) *T5lephone: Bridging speech and text self-supervised models for spoken language understanding via phoneme level T5*, *arXiv.org*. Available at: https://doi.org/10.48550/arXiv.2211.00586.
9. Anumanchipalli, G.K., Chartier, J. & Chang, E.F. Speech synthesis from neural decoding of spoken sentences. *Nature* **568**, 493–498 (2019). https://doi.org/10.1038/s41586-019-1119-1
10. Moses, D.A. *et al.* (2021) 'Neuroprosthesis for decoding speech in a paralyzed person with anarthria', *New England Journal of Medicine*, 385(3), pp. 217–227. doi:10.1056/nejmoa2027540.
11. https://www.kaggle.com/competitions/brain-to-text-25/data
12. http://www.speech.cs.cmu.edu/cgi-bin/cmudict
13. 'Conformer: Convolution-augmented Transformer for Speech Recognition.' Gulati A. et al., 2020
14. R-Transformer: Recurrent Neural Network Enhanced Transformer. Wang Z. et al., 2019
15. 'Conformer-based Ultrasound-to-Speech Conversion' Ibrahimov I. et al., 2025
16. 'A systematic comparison of grapheme-based vs. phoneme-based label units for encoder-decoder-attention models' Zeineldeen M. et al., 2021

17. Whisper: ‘Robust Speech Recognition via Large-Scale Weak Supervision’ Radford A. et al., 2022
18. ‘XPhoneBERT: A Pre-trained Multilingual Model for Phoneme Representations for Text-to-Speech’ Nguyen L. et al., 2023
19. ‘Attention is all you need’ Vaswani A. et al., 2017
20. https://www.baeldung.com/cs/beam-search