



# COPS Summer of Code 2025

Intelligence Guild

*Club Of Programmers, IIT (BHU) Varanasi*

---

**Sequence Learning and Attention Mechanisms**  
11 – 17 june 2025

---

*Official IG Website:* <https://cops-iitbhu.github.io/IG-website/>

*All deadlines are strict. No extensions will be granted.*

# Introduction

COPS Summer of Code (CSOC) is a flagship initiative under the Club Of Programmers, IIT (BHU) Varanasi, with all verticals contributing through focused tracks. This document embarks the journey of deep learning and contains the contents of ANN.

Modules will be released from time to time. **Adhere strictly to deadlines.** Submissions will be evaluated on approach, technical correctness, and clarity. The most technically accurate solution may not necessarily be the one chosen; clarity of thought and a well-reasoned approach will be valued more.

## Communities

All communication for the programme will be conducted strictly via [Discord](#). Do not reach out through other channels. Resources and updates will be posted on [Github](#), and all notifications will be made via Discord.

## Final Report

A concise report may be submitted along with your final assignment. While **not mandatory**, it may strengthen your overall evaluation. Reports must be written in  $\text{\LaTeX}$  and submitted in PDF format only. We are not interested in surface-level descriptions — focus strictly on your analysis, approach, and reasoning. The report itself constitutes the final assignment. No additional files are to be submitted. Refer to the Assignment section for details.

## Contact Details

In case of any doubts, clarifications, or guidance, you can contact one of us. We request that you stick to Discord as the preferred mode of communication for all the questions that you have as it will also benefit others. However, you can reach out to us through other means in case we fail to respond on Discord.

- Tejbir Panghal - 9034705165
- Sakshi Kumar - 8073247266

# Resources

In this module, we're stepping into the next frontier of NLP. Moving beyond traditional RNN-based architectures. You'll explore the Sequence-to-Sequence (Seq2Seq) framework, understand the power of attention mechanisms, and finally dive into the revolutionary world of Transformers.

Just a quick note before we jump into the resources, don't feel restricted to only these! If any topic feels unclear or if you're curious to explore further (which we highly recommend if you have the time), feel free to consult other resources. These are simply the ones we found helpful and worth sharing.

## Seq to Seq

- This video by statquest is a great start to know about seq to seq architecture. [Seq to seq explained](#).
- A video about encoder decoder, that dives deep into seq to seq : [Encoder Decoder](#)
- Check out these blog posts as well:
  - [Exploring Seq2Seq](#)
  - [Understanding Encoders-Decoders with an Attention-based mechanism](#)

## Transformers

Now meet the architecture that changed everything in NLP - Transformers. Transformers leverage self-attention to understand context better and faster.

- “[Attention is All You Need](#)” - this groundbreaking 2017 paper by Google introduced the Transformer architecture, sparking a revolution in NLP and paving the way for models like BERT and GPT. First paper for y'all(read it after you have gone through resources)
- This [playlist by Sir Mitesh Khapra](#) explains all the componets of transformer very well(Videos 1-16 are enough)
- If you still having a hard time understanding some individual components of Transformer, here are resources for some parts.
  - [Self Attention](#)
  - [Understanding and Coding the Self-Attention Mechanism of Large Language Models](#)
  - [The math behind Attention: Keys, Queries, and Values matrices](#)
  - [Positional Encoding in Transformers](#)
  - [Positianal encoding blog](#)
  - [Transformers blog](#). Long but brilliant intuition of self attention and math explanation.

- Here is the [Hugging face guide](#) on transformers. This will teach you how to use Hugging face library in nlp tasks. For now introduction, models and preprocessing sections will be enough.

## Additional resources

- [Essential Guide to Transformer Models in Machine Learning](#)
- [The annotated transformers](#)
- [Understanding Encoder And Decoder LLMs](#)
- [Fine tuning for any kind of transformer](#)

The task will be updated soon. In the meantime, please go through the provided materials. )

## Assignment: Neural Machine Translation(X to Y language):

### Objective

Build a sequence-to-sequence model to perform machine translation. Start simple with an encoder-decoder, then extend it using attention mechanism. At last implement a transformer-based version.

### Project Tasks

Build an encoder-decoder model using LSTMs or GRUs.

#### Part 1: Seq2Seq without Attention

- Build an encoder-decoder model using LSTMs or GRUs.

#### Part 2: Add Attention

- Integrate Bahdanau or Luong attention into your encoder-decoder model.
- Show how attention improves performance or convergence.

#### Part 3: Transformer

- Use pretrained transformer models from Hugging Face for translation.
- Fine-tune as needed on your dataset.

### Performance Metrics

- BLEU Score

## Evaluation and Analysis

### 1.3.1 Performance Metrics

Compare the BLEU score for all three parts):

### 1.3.2 Bonus: Attention Lens Challenge:

Choose 3–5 translation examples and visualize the attention weights from your encoder-decoder model with attention (Part 2). Then reflect on:

- Where is your model attending while generating each word?
- Are there any surprising patterns or failures in focus?

## Submission Guidelines

- Create a GitHub repository named `<roll_number>-CSOC-IG` (e.g., 23014019-CSOC-IG)
- Repository organization:
  - A folder named Seq to seq containing all source code implementations
  - The final report in PDF format, authored using  $\text{\LaTeX}$

Everything must be in the github repo itself.

- Submit the repository link via the provided Google Form [here](#)
- **Note:** The report constitutes the primary assignment submission. No additional files are required

## Final Remarks

Ensure that your submission reflects a clear understanding of the concepts and methodologies applied. Focus on the analytical aspects and the rationale behind your implementations. We look forward to your insightful contributions.

***Adios, and keep learning!***