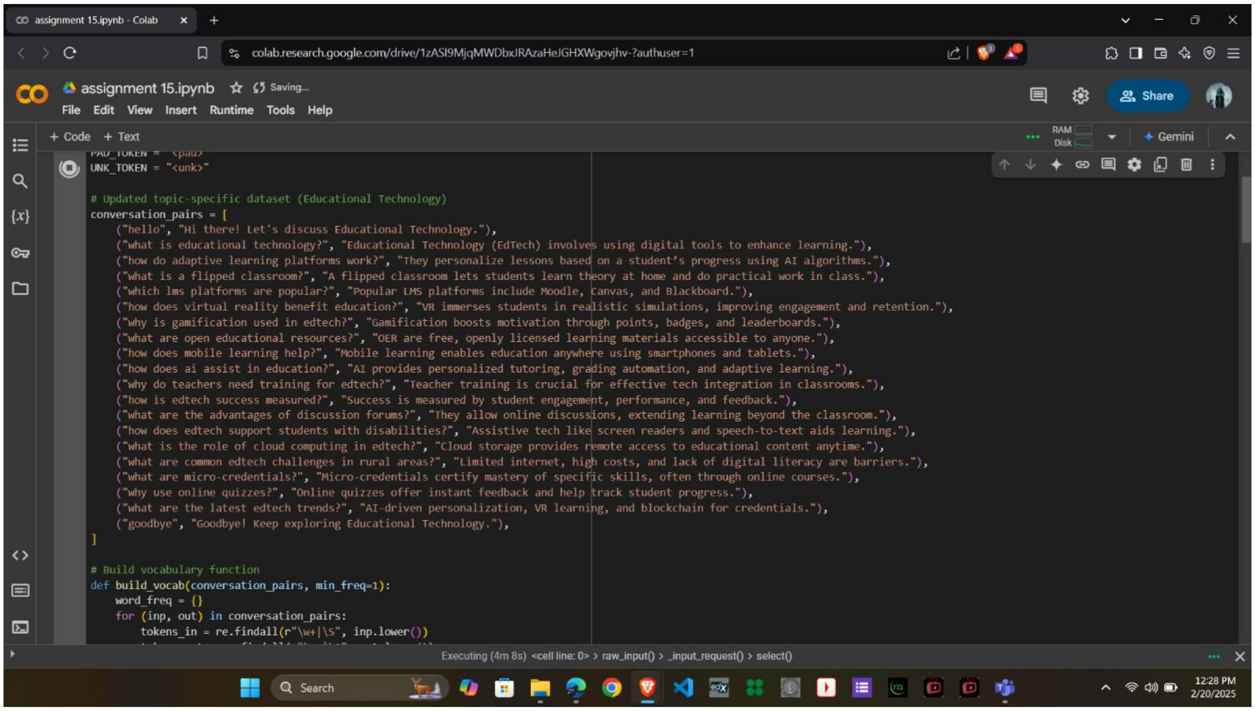
Assignment 15: Build a Topic-Specific Expert Chatbot

Step 1: Select a Topic:

I choose Educational Technology topic

Step 2: Generate my Dataset:


assignment 15.ipynb - Colab
colab.research.google.com/drive/1zASI9MjqMWDbxJRAzaHeJGHXWgovjhv-?authuser=1
assignment 15.ipynb
Saving...
File
Edit
View
Insert
Runtime
Tools
Help
Share
+ Code
+ Text
RAM
Disk
Gemini

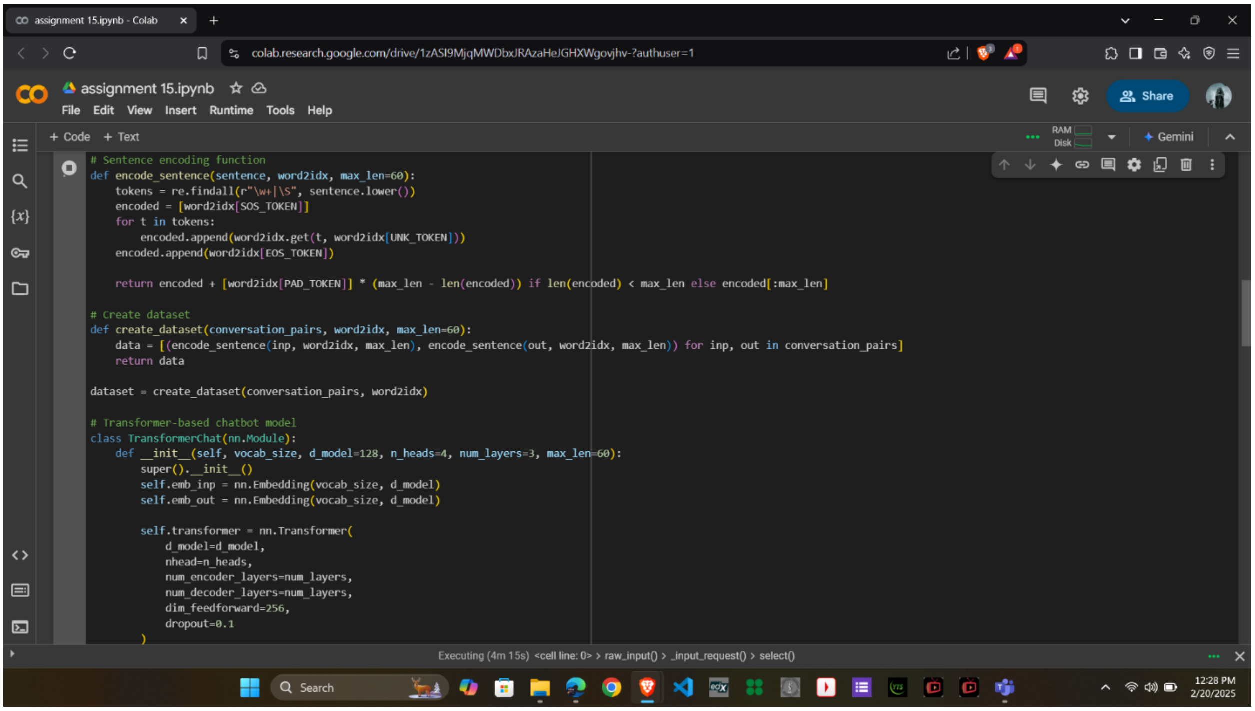
```
PAD_TOKEN = "<pad>"
UNK_TOKEN = "<unk>"

# Updated topic-specific dataset (Educational Technology)
conversation_pairs = [
    ("hello", "Hi there! Let's discuss Educational Technology."),
    ("what is educational technology?", "Educational Technology (EdTech) involves using digital tools to enhance learning."),
    ("how do adaptive learning platforms work?", "They personalize lessons based on a student’s progress using AI algorithms."),
    ("what is a flipped classroom?", "A flipped classroom lets students learn theory at home and do practical work in class."),
    ("which lms platforms are popular?", "Popular LMS platforms include Moodle, Canvas, and Blackboard."),
    ("how does virtual reality benefit education?", "VR immerses students in realistic simulations, improving engagement and retention."),
    ("why is gamification used in edtech?", "Gamification boosts motivation through points, badges, and leaderboards."),
    ("what are open educational resources?", "OER are free, openly licensed learning materials accessible to anyone."),
    ("how does mobile learning help?", "Mobile learning enables education anywhere using smartphones and tablets."),
    ("how does ai assist in education?", "AI provides personalized tutoring, grading automation, and adaptive learning."),
    ("why do teachers need training for edtech?", "Teacher training is crucial for effective tech integration in classrooms."),
    ("how is edtech success measured?", "Success is measured by student engagement, performance, and feedback."),
    ("what are the advantages of discussion forums?", "They allow online discussions, extending learning beyond the classroom."),
    ("how does edtech support students with disabilities?", "Assistive tech like screen readers and speech-to-text aids learning."),
    ("what is the role of cloud computing in edtech?", "Cloud storage provides remote access to educational content anytime."),
    ("what are common edtech challenges in rural areas?", "Limited internet, high costs, and lack of digital literacy are barriers."),
    ("what are micro-credentials?", "Micro-credentials certify mastery of specific skills, often through online courses."),
    ("why use online quizzes?", "Online quizzes offer instant feedback and help track student progress."),
    ("what are the latest edtech trends?", "AI-driven personalization, VR learning, and blockchain for credentials."),
    ("goodbye", "Goodbye! Keep exploring Educational Technology."),
]

# Build vocabulary function
def build_vocab(conversation_pairs, min_freq=1):
    word_freq = {}
    for (inp, out) in conversation_pairs:
        tokens_in = re.findall(r"\w+|\S", inp.lower())
```

Executing (4m 8s) <cell line: 0> > raw_input() > _input_request() > select()

Step 3: Adjust the Code Parameters:

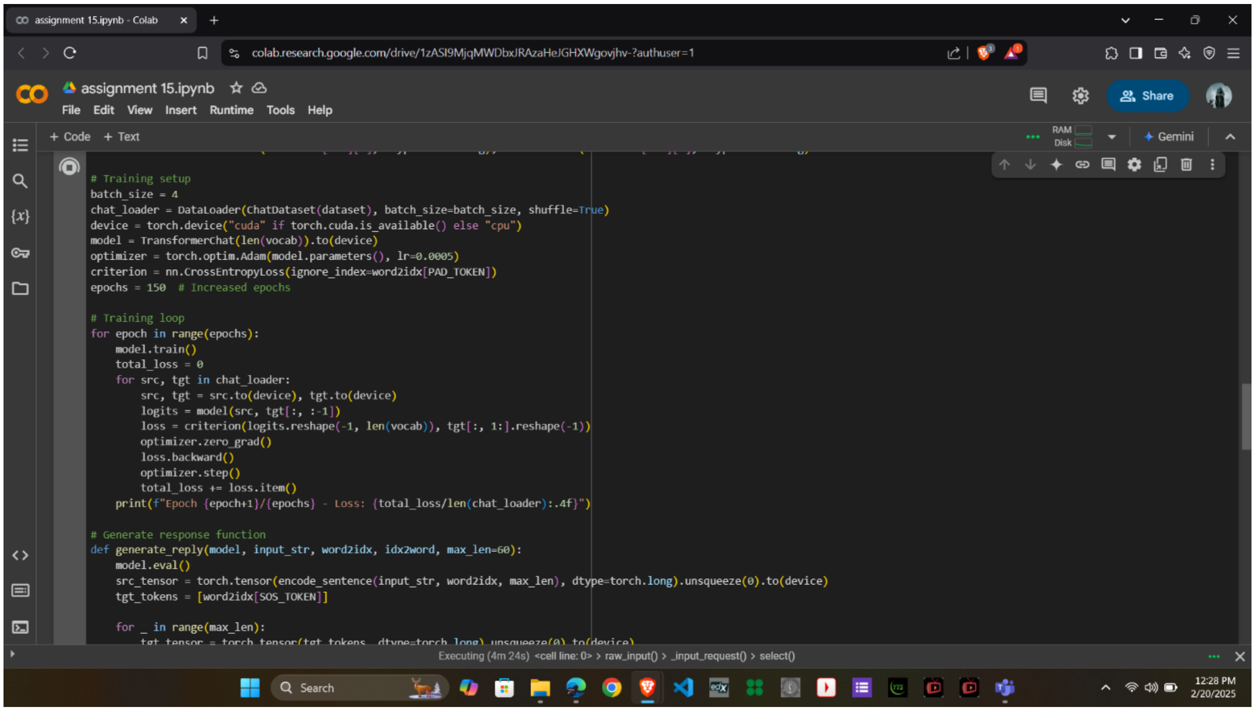
Max Length (max_len): 60

Training Epochs: 150


assignment 15.ipynb - Colab
colab.research.google.com/drive/1zASI9MjqMWDbxJRAzaHeJGHXWgovjhv-?authuser=1
assignment 15.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
Search
12:28 PM
2/20/2025

```
# Sentence encoding function
def encode_sentence(sentence, word2idx, max_len=60):
    tokens = re.findall(r"\w+|\S", sentence.lower())
    encoded = [word2idx[SOS_TOKEN]]
    for t in tokens:
        encoded.append(word2idx.get(t, word2idx[UNK_TOKEN]))
    encoded.append(word2idx[EOS_TOKEN])

    return encoded + [word2idx[PAD_TOKEN]] * (max_len - len(encoded)) if len(encoded) < max_len else encoded[:max_len]

# Create dataset
def create_dataset(conversation_pairs, word2idx, max_len=60):
    data = [(encode_sentence(inp, word2idx, max_len), encode_sentence(out, word2idx, max_len)) for inp, out in conversation_pairs]
    return data

dataset = create_dataset(conversation_pairs, word2idx)

# Transformer-based chatbot model
class TransformerChat(nn.Module):
    def __init__(self, vocab_size, d_model=128, n_heads=4, num_layers=3, max_len=60):
        super().__init__()
        self.emb_inp = nn.Embedding(vocab_size, d_model)
        self.emb_out = nn.Embedding(vocab_size, d_model)

        self.transformer = nn.Transformer(
            d_model=d_model,
            nhead=n_heads,
            num_encoder_layers=num_layers,
            num_decoder_layers=num_layers,
            dim_feedforward=256,
            dropout=0.1
        )
```

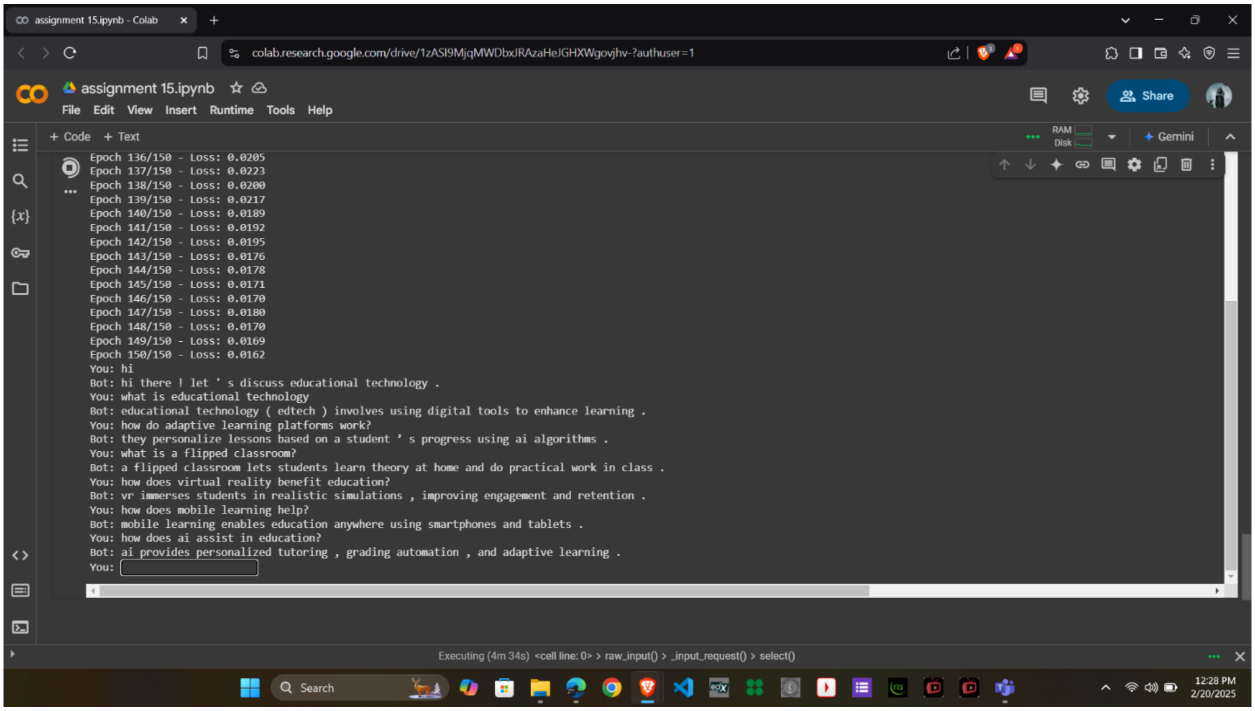
Executing (4m 15s) <cell line: 0> > raw_input() > _input_request() > select()


assignment 15.ipynb - Colab
colab.research.google.com/drive/1zASI9MjqMWDbxJRAzaHeJGHXWgovjhv-?authuser=1
assignment 15.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
Search
12:28 PM
2/20/2025

```
# Training setup
batch_size = 4
chat_loader = DataLoader(ChatDataset(dataset), batch_size=batch_size, shuffle=True)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = TransformerChat(len(vocab)).to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=0.0005)
criterion = nn.CrossEntropyLoss(ignore_index=word2idx[PAD_TOKEN])
epochs = 150  # Increased epochs

# Training loop
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for src, tgt in chat_loader:
        src, tgt = src.to(device), tgt.to(device)
        logits = model(src, tgt[:, :-1])
        loss = criterion(logits.reshape(-1, len(vocab)), tgt[:, 1:].reshape(-1))
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    print(f"Epoch {epoch+1}/{epochs} - Loss: {total_loss/len(chat_loader):.4f}")

# Generate response function
def generate_reply(model, input_str, word2idx, idx2word, max_len=60):
    model.eval()
    src_tensor = torch.tensor(encode_sentence(input_str, word2idx, max_len), dtype=torch.long).unsqueeze(0).to(device)
    tgt_tokens = [word2idx[SOS_TOKEN]]

    for _ in range(max_len):
        tgt_tensor = torch.tensor(tgt_tokens, dtype=torch.long).unsqueeze(0).to(device)
```

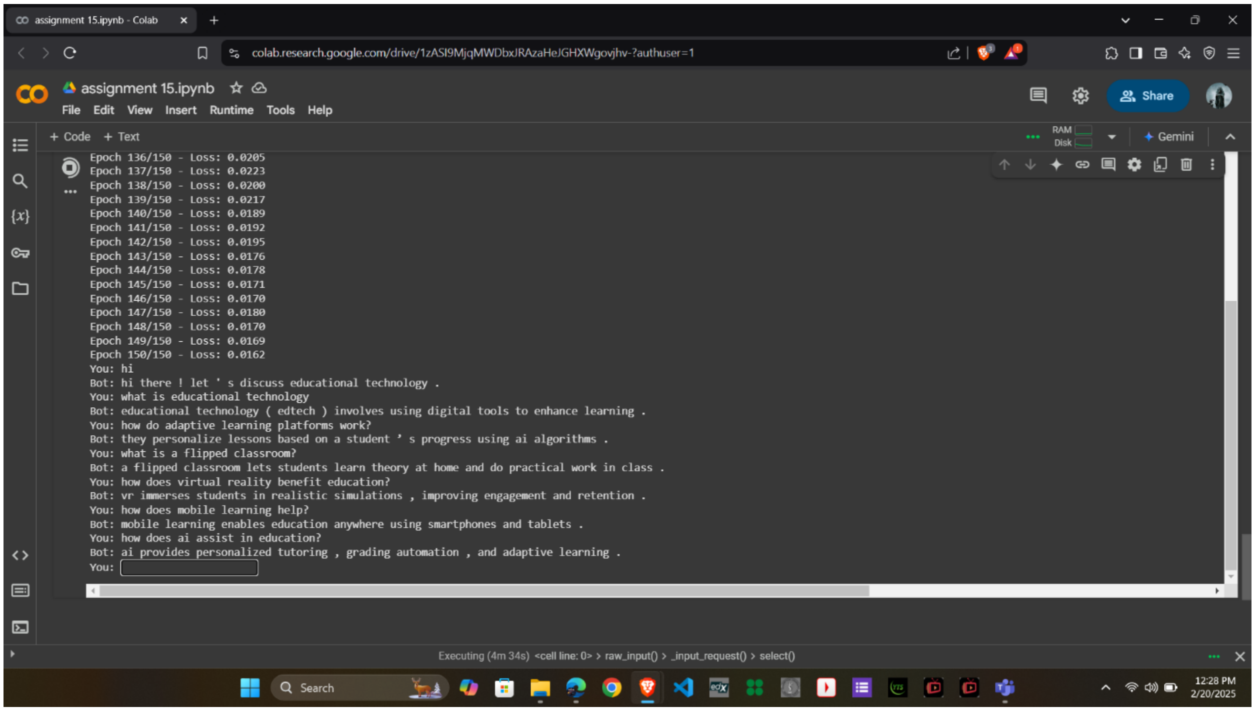
Executing (4m 24s) <cell line: 0> > raw_input() > _input_request() > select()

Step 4: Train my Chatbot:


assignment 15.ipynb - Colab
colab.research.google.com/drive/1zASI9MjqMWDbxJRAzaHeJGHXWgovjhv-?authuser=1
assignment 15.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Share
Gemini
Executing (4m 34s) <cell line: 0> > raw_input() > _input_request() > select()
12:28 PM
2/20/2025

```
Epoch 136/150 - Loss: 0.0205
Epoch 137/150 - Loss: 0.0223
Epoch 138/150 - Loss: 0.0200
Epoch 139/150 - Loss: 0.0217
Epoch 140/150 - Loss: 0.0189
Epoch 141/150 - Loss: 0.0192
Epoch 142/150 - Loss: 0.0195
Epoch 143/150 - Loss: 0.0176
Epoch 144/150 - Loss: 0.0178
Epoch 145/150 - Loss: 0.0171
Epoch 146/150 - Loss: 0.0170
Epoch 147/150 - Loss: 0.0180
Epoch 148/150 - Loss: 0.0170
Epoch 149/150 - Loss: 0.0169
Epoch 150/150 - Loss: 0.0162
You: hi
Bot: hi there ! let ' s discuss educational technology .
You: what is educational technology
Bot: educational technology ( edtech ) involves using digital tools to enhance learning .
You: how do adaptive learning platforms work?
Bot: they personalize lessons based on a student ’ s progress using ai algorithms .
You: what is a flipped classroom?
Bot: a flipped classroom lets students learn theory at home and do practical work in class .
You: how does virtual reality benefit education?
Bot: vr immerses students in realistic simulations , improving engagement and retention .
You: how does mobile learning help?
Bot: mobile learning enables education anywhere using smartphones and tablets .
You: how does ai assist in education?
Bot: ai provides personalized tutoring , grading automation , and adaptive learning .
You:
```

Step 5: Test and Evaluate:


assignment 15.ipynb - Colab
colab.research.google.com/drive/1zASI9MjqMWDbxJRAzaHeJGHXWgovjhv-?authuser=1
assignment 15.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Share
Gemini
Executing (4m 34s) <cell line: 0> > raw_input() > _input_request() > select()
12:28 PM
2/20/2025

```
Epoch 136/150 - Loss: 0.0205
Epoch 137/150 - Loss: 0.0223
Epoch 138/150 - Loss: 0.0200
Epoch 139/150 - Loss: 0.0217
Epoch 140/150 - Loss: 0.0189
Epoch 141/150 - Loss: 0.0192
Epoch 142/150 - Loss: 0.0195
Epoch 143/150 - Loss: 0.0176
Epoch 144/150 - Loss: 0.0178
Epoch 145/150 - Loss: 0.0171
Epoch 146/150 - Loss: 0.0170
Epoch 147/150 - Loss: 0.0180
Epoch 148/150 - Loss: 0.0170
Epoch 149/150 - Loss: 0.0169
Epoch 150/150 - Loss: 0.0162
You: hi
Bot: hi there ! let ' s discuss educational technology .
You: what is educational technology
Bot: educational technology ( edtech ) involves using digital tools to enhance learning .
You: how do adaptive learning platforms work?
Bot: they personalize lessons based on a student ’ s progress using ai algorithms .
You: what is a flipped classroom?
Bot: a flipped classroom lets students learn theory at home and do practical work in class .
You: how does virtual reality benefit education?
Bot: vr immerses students in realistic simulations , improving engagement and retention .
You: how does mobile learning help?
Bot: mobile learning enables education anywhere using smartphones and tablets .
You: how does ai assist in education?
Bot: ai provides personalized tutoring , grading automation , and adaptive learning .
You:
```

Report:

1. **Parameter Adjustments**

- **Max Sequence Length (max_len)**
    - Increased from **50 to 60** to handle longer educational answers without truncation.
- **Training Epochs**
    - Increased from **100 to 150 epochs** to allow the model to learn more effectively from the dataset.
- **Model Architecture Adjustments**
    - **Embedding Dimension (d_model)** increased from **64 to 128** for better feature extraction.
    - **Number of Transformer Layers (num_layers)** increased from **2 to 3**, improving learning depth.
    - **Number of Attention Heads (n_heads)** increased from **2 to 4** to allow the model to capture more complex relationships in text.
    - **Feedforward Layer (dim_feedforward)** increased from **128 to 256** for better representation learning.
- **Learning Rate Adjustment**
    - Decreased from **0.001 to 0.0005** to stabilize training and avoid overfitting.

2. Improved Training Process

- **Batch size increased from 2 to 4** for better generalization.
- **Cross-Entropy Loss (ignore_index=PAD_TOKEN)** ensures that padding tokens do not impact training.