

# Artificial Intelligence (AI101B)

## Even Semester Session

### 2024-25

## Language Translator

## Using Sequence

## Model

Minakshi Tomar 202410116100119

Jatin Gupta 202410116100094

Mukul Dhiman 202410116100126

Jitendra Kumar 202410116100095

**Project Supervisor:**

**Mr. Apoorv Jain**


**(Assistant Professor)**

# Introduction


- Language is a powerful tool for communication.
- Machine translation bridges the language gap.
- Sequence models like MarianMT offer high-quality multilingual translation.
- This project uses Hugging Face Transformers with Gradio for a user-friendly interface.



# Objective

- Translate English text to multiple languages (Hindi, French, Spanish, German).
  - Utilize pretrained transformer models (MarianMT).
  - Provide a simple, interactive web interface using Gradio.
  - Showcase application of AI in real-world communication challenges.
- 

# Methodology

1. **Model Selection:** Used MarianMT models for multilingual translation.
  2. **Tokenizer & Model Loading:** Loaded specific models based on language choice.
  3. **Text Preprocessing:** Tokenized input using MarianTokenizer.
  4. **Translation:** Generated translated output using MarianMTModel.generate.
  5. **User Interface:** Built interactive front-end using Gradio.
- 



# Model Development

- ▶ **Language Options:**

- English to Hindi
- English to French
- English to Spanish
- English to German

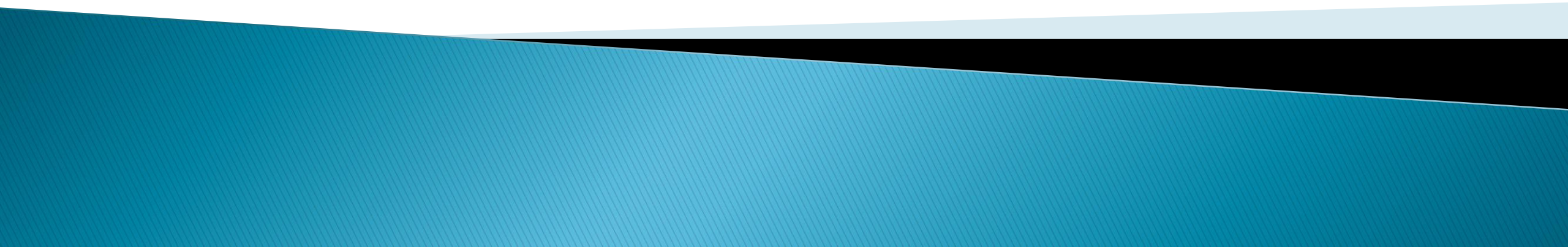
- ▶ **Libraries Used:**

- transformers, torch, gradio

- ▶ **Process:**

Input Text → Tokenizer → Sequence Model → Decoder → Translated Output

# Model Evaluation

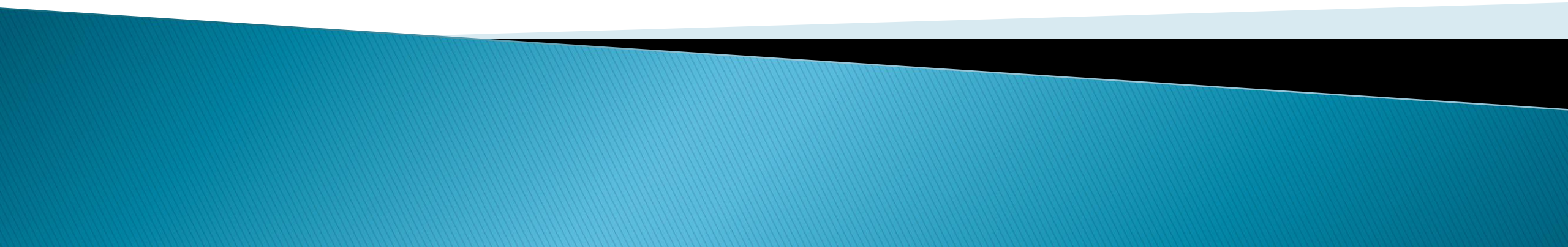
- All models used are pre-trained on large, open translation datasets.
  - Evaluation Metrics (in original research):
    - BLEU Score
    - Accuracy in translation tasks
  - Real-time testing with user input via Gradio.
- 

# Code Implementation

```
def translate(text, language_choice):  
    model_name = language_models[language_choice]  
    tokenizer = MarianTokenizer.from_pretrained(model_name)  
    model = MarianMTModel.from_pretrained(model_name)  
    tokenized_text = tokenizer(text, return_tensors="pt", padding=True)  
    translation = model.generate(**tokenized_text)  
    return tokenizer.decode(translation[0], skip_special_tokens=True)
```


- Interactive Interface created with gr.Interface.

# Future Enhancements


- Add more language pairs and bidirectional translation.
  - Add speech-to-text input and text-to-speech output.
  - Improve UI with language detection and translation history.
  - Deploy model as a web app using Hugging Face Spaces or Flask.
- 



# Outcomes & Findings

- Successfully implemented real-time text translation.
  - High accuracy with pretrained models from Hugging Face.
  - User-friendly Gradio interface makes translation accessible.
  - Demonstrated practical use of sequence models in NLP.
- 

# Conclusion

- ▶ Machine translation simplifies cross-language communication.
  - ▶ Sequence models like MarianMT are powerful and efficient.
  - ▶ This project integrates model, UI, and usability.
  - ▶ Opens doors for multilingual AI applications in real life.
- 

THANK YOU