EduBot: AI-Powered Educational Assistant

# Abstract

EduBot is an AI-powered educational assistant designed to support learners by answering questions, summarizing content, explaining key computer science concepts, and generating multiple-choice questions (MCQs). Leveraging powerful transformer models from Hugging Face, EduBot provides an interactive console-based interface, combining simplicity with the strength of natural language processing. The backend is implemented in Python using the `transformers` library, making it an effective learning aid for both students and educators.

# 1. Introduction

The integration of AI into education has revolutionized the learning experience, making knowledge more accessible and personalized. EduBot addresses common educational needs such as quick concept explanation, automatic summarization of text, question answering, and MCQ generation. Its primary goal is to simulate a virtual tutor experience that is available 24/7.

# 2. Literature Review

AI in education has been explored through adaptive learning systems, intelligent tutoring systems (ITS), and AI chatbots. Notable works include:

- Duolingo: A language learning platform using AI to personalize content.

- Socratic by Google: Uses NLP for homework help.

- Transformers by Hugging Face: Offer state-of-the-art pre-trained models for various NLP tasks like QA and summarization.

EduBot stands on the shoulders of such systems, utilizing publicly available transformer models to create a lightweight educational assistant.

# 3. Methodology

EduBot’s core architecture is based on command-line user interactions driven by a Python program. It performs:

- Concept explanation using predefined dictionaries.

- Text summarization using the `sshleifer/distilbart-cnn-12-6` model.

- Question answering using the `distilbert-base-cased-distilled-squad` model.

- MCQ generation with predefined questions and answers.

# 4. Technology Involved

- Programming Language: Python 3

- Libraries Used: transformers (Hugging Face), torch (optional)

- Platform: Google Colab / Any local Python environment

- Models: distilbert-base-cased-distilled-squad (Q&A), sshleifer/distilbart-cnn-12-6 (Summarization)

# 5. Block Diagram

[Block Diagram Placeholder: Will be added in the final formatted version]

# 6. Complete Python Backend Code

from transformers import pipeline

qa\_pipeline = pipeline("question-answering")

summarizer = pipeline("summarization", model="sshleifer/distilbart-cnn-12-6")

concepts = {

"os": "An operating system (OS) manages computer hardware and software resources and provides services for programs.",

"data structure": "A data structure enables efficient access and modification of data.",

"recursion": "Recursion solves a problem by solving smaller instances of the same problem."

}

sample\_text = """

Python is an interpreted, high-level, general-purpose programming language.

Its design emphasizes code readability and supports multiple paradigms.

"""

def explain\_concept():

topic = input("Enter a topic (e.g., OS, Data Structure): ").lower()

print("📘 Explanation:", concepts.get(topic, "Sorry, I don't have info on that."))

def summarize\_topic():

print("📚 Summary:")

print(summarizer(sample\_text, max\_length=50, min\_length=25, do\_sample=False)[0]['summary\_text'])

def ask\_question():

question = input("Ask a question: ")

print("❓ Answer:")

print(qa\_pipeline({'question': question, 'context': sample\_text})['answer'])

def generate\_mcq():

question = "What is Python primarily used for?"

options = ['Low-level programming', 'Database management', 'Web and software development', 'Network configuration']

correct = 2

print("📝 MCQ:")

print(question)

for i, option in enumerate(options):

print(f"{i+1}. {option}")

print("✅ Answer:", options[correct])

while True:

print("\n🤖 EduBot Options:\n1. Explain Concept\n2. Summarize Topic\n3. Ask a Question\n4. Generate MCQ\n5. Exit")

choice = input("Choose: ")

if choice == '1':

explain\_concept()

elif choice == '2':

summarize\_topic()

elif choice == '3':

ask\_question()

elif choice == '4':

generate\_mcq()

elif choice == '5':

break

else:

print("Invalid choice.")

# 7. Results and Discussion

EduBot successfully performs all functionalities with high reliability. The transformer models provide accurate and concise outputs for both QA and summarization tasks. The command-line interface is simple yet effective for user interaction.

\*Screenshots from Colab interface can be added here in the final Word document.\*

# 8. Conclusion and Future Scope

Conclusion:

EduBot demonstrates the integration of state-of-the-art NLP models into a lightweight educational assistant. It provides a foundational framework that can be extended for various domains and subjects.

Future Scope:

- Add speech-to-text and text-to-speech capabilities.

- Integrate a GUI using Tkinter or a web-based frontend.

- Include real-time feedback and adaptive learning features.

- Expand domain coverage using external APIs or custom datasets.

# 9. References and Bibliography

1. Hugging Face Transformers Documentation - https://huggingface.co/docs

2. DistilBERT Model - https://huggingface.co/distilbert/distilbert-base-cased-distilled-squad

3. DistilBART Summarizer - https://huggingface.co/sshleifer/distilbart-cnn-12-6

4. Python Official Documentation - https://docs.python.org/3/

5. Google Colab Platform - https://colab.research.google.com