EchoVerse – AI-Powered Audiobook Creation Tool

Overview

EchoVerse is an innovative AI-powered audiobook creation tool that transforms written content into engaging, natural-sounding audiobooks. The solution leverages cutting-edge language models to help authors, publishers, and listeners access immersive storytelling experiences with ease.

Key Features

Automatic conversion of text into high-quality audio narration. Customizable voices and tones for different genres and audiences. Multilingual support to reach global listeners. Seamless integration for authors and publishers to publish audiobooks quickly. Accessibility enhancement for visually impaired and busy readers.

Problem Solved

EchoVerse solves the problem of expensive and time-consuming audiobook production. Traditional audiobook creation often requires professional narrators, studios, and long editing cycles. With EchoVerse, content creators can instantly transform their writing into professional-quality audiobooks, making literature and learning resources more accessible and affordable worldwide.

How to Run the Code

1. Install the required dependencies:
pip install torch transformers

2. Save the script as echoverse_demo.py.

3. Run the script:
python echoverse_demo.py

4. The program will generate and display an AI-powered audiobook solution idea in the console.

Source Code

```
from transformers import AutoModelForCausalLM, AutoTokenizer, set_seed
import torch

def main():
    # ----------------------------
    # Model Setup
    # ----------------------------
    model_path = "ibm-granite/granite-3.3-2b-instruct"
    device = "cuda" if torch.cuda.is_available() else "cpu"

    print(f"Loading model on {device}...")
    # Load model
```

```
    model = AutoModelForCausalLM.from_pretrained(
        model_path,
        device_map="auto",
        torch_dtype=torch.bfloat16,
    )

    # Load tokenizer
    tokenizer = AutoTokenizer.from_pretrained(model_path)

    # --------------------------
    # Prompt (EchoVerse Idea)
    # --------------------------
    conv = [
        {
            "role": "user",
            "content": (
                "Design an innovative solution for EchoVerse – an AI-powered audiobook "
                "creation tool. Describe how it can transform written content into "
                "engaging audiobooks, highlight its unique features, and explain how it "
                "solves real-world problems for authors, publishers, and listeners."
            ),
        }
    ]

    # Convert input with chat template
    input_ids = tokenizer.apply_chat_template(
        conv,
        return_tensors="pt",
        return_dict=True,
        add_generation_prompt=True
    ).to(device)

    # Set reproducibility
    set_seed(42)

    # --------------------------
    # Generate Response
    # --------------------------
    print("\nGenerating EchoVerse Solution...\n")
    output = model.generate(
        **input_ids,
        max_new_tokens=1024,
    )

    # Decode
    prediction = tokenizer.decode(
        output[0, input_ids["input_ids"].shape[1]:],
        skip_special_tokens=True
    )

    # --------------------------
    # Display Final Output
    # --------------------------
    print("\n=== EchoVerse – AI Audiobook Solution ===\n")
    print(prediction)


if __name__ == "__main__":
    main()
```