GEN AI

Lab-2

NAME- Dhrushaj Achar

SRN- PES2UG23CS171

CLASS- CSE 6C

Category 6: Creative Writing & Content (Expansion on Generation)

Project Title- Joke Punchline Generator

Goal: Create a system that takes a joke setup as input (e.g., "Why did the chicken cross the road?") and uses AI to generate a funny punchline to complete the joke.

Technology: Text generation using Hugging Face's `pipeline('text-generation')

Abstract:

This project implements an AI-powered joke punchline generator that uses natural language processing to complete joke setups. The system leverages the GPT-2 language model through Hugging Face's Transformers library to generate creative and contextually relevant punchlines. Users can input any joke setup, and the AI will automatically generate a completion, making it useful for entertainment, creative writing, or understanding how language models work with humour.

What I understood from this project:

- Text Generation Pipeline: Hugging Face provides a simple pipeline('text-generation') interface that abstracts away the complexity of loading and using pre-trained language models.
- GPT-2 Model: GPT-2 is a transformer-based language model trained on a large corpus of text. It can predict what comes next in a sequence, making it suitable for completing joke setups.

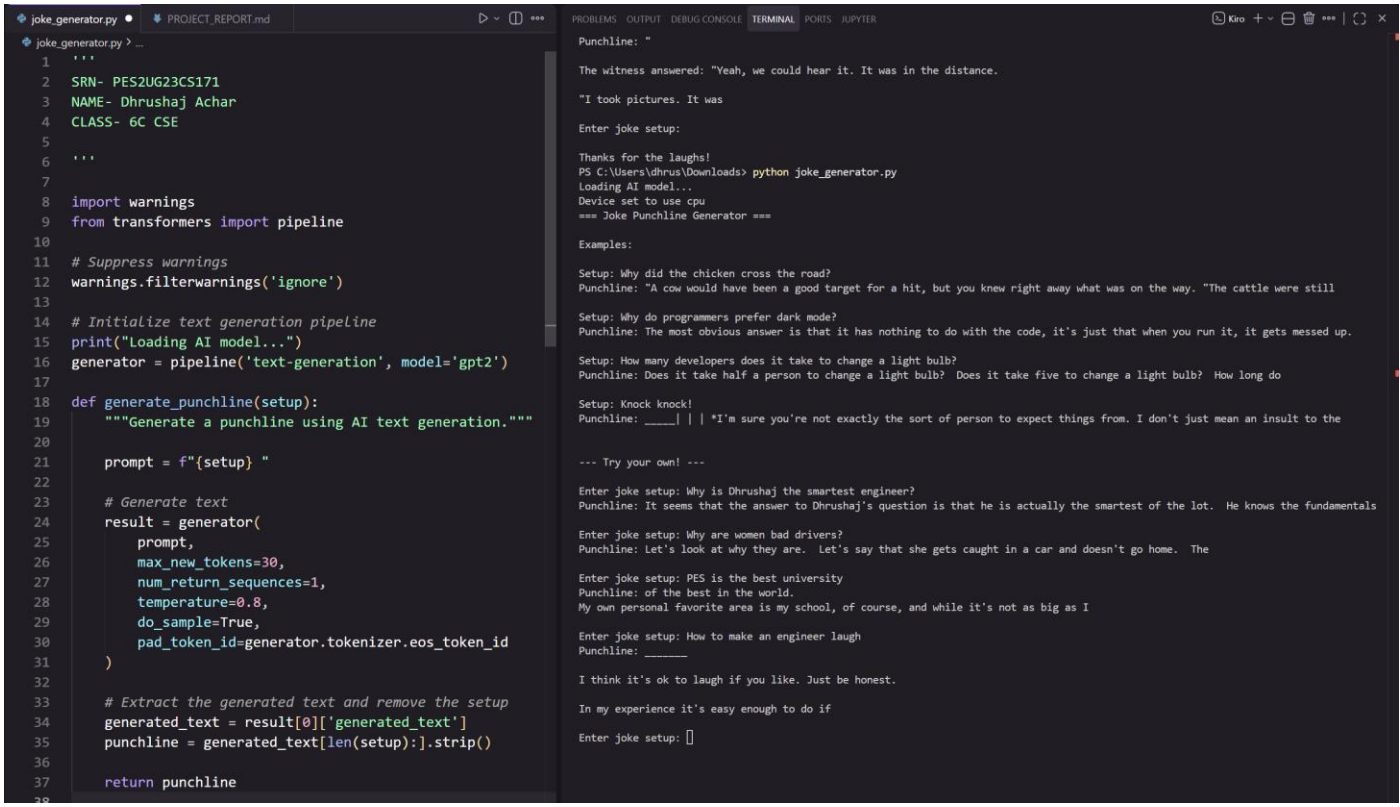
- Parameters Matter: The quality of generated text depends on parameters like:
- max_new_tokens: Controls how long the generated punchline will be
- temperature: Controls randomness (higher = more creative, lower = more predictable)

- do_sample: Enables sampling for more diverse outputs

Core Functionality:

- Loads the GPT-2 model using Hugging Face's Transformers library
- Takes joke setups as input (either from predefined examples or user input)
- Generates AI-powered punchlines using text generation
- Provides an interactive mode for users to test their own joke setups

OUTPUT:



Technical Implementation:

- Used pipeline('text-generation', model='gpt2') as the core AI engine
- Configured generation parameters for optimal joke completion
- Implemented warning suppression for cleaner output
- Added error handling for graceful exit (Ctrl+C)

User Experience:

- Displays example jokes with AI-generated punchlines on startup
- Provides an interactive loop where users can input custom joke setups
- Clean, formatted output showing both setup and generated punchline

How It Works

- The program initializes by loading the GPT-2 model (downloads on first run)
- When given a joke setup, it treats it as a text prompt
- The AI model analyses the context and generates a continuation
- The generated text is extracted and presented as the punchline
- Users can try multiple jokes in the interactive mode

Challenges Faced

- Warning Messages: Initially, the model generated warnings about conflicting parameters. I resolved this by using max_new_tokens instead of max_length and suppressing unnecessary warnings.
- Model Size: The GPT-2 model is about 500MB, so the first run requires downloading it.
- Output Quality: Text generation can sometimes produce unexpected results, so parameter tuning was necessary.

Future Improvements

- Add support for larger models (GPT-2 Medium/Large) for better joke quality
- Implement fine-tuning on a joke dataset for more humorous outputs
- Add a rating system to evaluate punchline quality
- Create a web interface for easier access

Conclusion

This project successfully demonstrates the application of AI text generation for creative purposes. By using the Hugging Face Transformers library and the GPT-2 model, I was able to create a functional joke punchline generator that showcases how modern NLP models can understand context and generate human-like text. The project helped me understand the practical implementation of text generation pipelines and the importance of parameter tuning in AI applications.