SUMMER OF INNOVATION

STORYFORGE

AI AVENGERS

ANMOL SHARMA (TEAM LEAD)
ADARSH GUPTA (MEMBER 2)
AMAN PUSHKAR (MEMBER 3)
MAHESH KRISHNAM (MEMBER 4)

INDIAN INSTITUTE OF TECHNOLOGY, DHARWAD

01

MODELS

GPT-2 for Story Generation

```
model_name="maheshkrishnam/promt_to_story"

# Load the model and tokenizer from Hugging Face
model = GPT2LMHeadModel.from_pretrained(model_name)
tokenizer = GPT2Tokenizer.from_pretrained(model_name)

# Load model directly
tokenizer = AutoTokenizer.from_pretrained("maheshkrishnam/promt_to_story")
model = AutoModelForCausalLM.from_pretrained("maheshkrishnam/promt_to_story")
```

SpeechT5 for Text-to-Speech

```
processor = SpeechT5Processor.from_pretrained("microsoft/speecht5_tts")
model = SpeechT5ForTextToSpeech.from_pretrained("microsoft/speecht5_tts")
vocoder = SpeechT5HifiGan.from_pretrained("microsoft/speecht5_hifigan")

# restriction of 600 characters for this tts model
inputs = processor(text=story, return_tensors="pt")

# load xvector containing speaker's voice characteristics from a dataset
embeddings_dataset = load_dataset("Matthijs/cmu-arctic-xvectors", split="validation")
speaker_embeddings = torch.tensor(embeddings_dataset[7306]["xvector"]).unsqueeze(0)

speech = model.generate_speech(inputs["input_ids"], speaker_embeddings, vocoder=vocoder)

sf.write("story.wav", speech.numpy(), samplerate=16000)
```

Stable Diffusion for Video Generation

```
model_id = "runwayml/stable-diffusion-v1-5"
```

```
# Adjusting durations to reach desired video length of 25-30 seconds
total_duration = sum(scene_durations)
target_duration = 25  # target video duration in seconds
factor = target_duration / total_duration
scene_durations = [duration * factor for duration in scene_durations]

# Ensure the total duration matches or slightly exceeds the target duration
current_duration = sum(scene_durations)
if current_duration < target_duration:
    scene_durations[-1] += target_duration - current_duration

# Generate the video without subtitles
output_video_path = 'story_video.mp4'
video_duration = create_video(images, scene_durations, output_video_path)

# Adding background music
def add_background_music(video_path, output_path, audio_path):
    video_clip = VideoFileClip(video_path)
    audio_clip = AudioFileClip(audio_path).subclip(0, video_duration)
    video_clip = video_clip.set_audio(audio_clip)
    video_clip.write_videofile(output_path, codec='libx264', audio_codec='aac')

# Add background music
background_music_path = 'story.wav'  # Path to background music file
output_video_with_music_path = 'final_video.mp4'
add_background_music(output_video_path, output_video_with_music_path, background_music_path)
print(f"Video with audio saved at {output_video_with_music_path}")
```

02

TRAINING OF MODELS

Fine-tuning GPT-2

The GPT-2 model was fine-tuned on the story dataset using the Adam optimizer with a learning rate of 0.01 for 5 epochs. The model was optimized for coherence and creativity in story generation.

Step	Training Loss
1000	3.366000
2000	3.191700
3000	3.095700
4000	3.019200
5000	2.985200
6000	2.939100
7000	2.916400

Fine-tuning SpeechT5

SpeechT5 was fine-tuned on the CMU Arctic dataset for 5 epochs, focusing on producing clear and natural speech outputs. The Adam optimizer with a learning rate of 3e-5 was used.

03

OUTPUT COMPARISON BEFORE AND AFTER FINE-TUNING

GPT-2 Story Generation

- Before Fine-tuning: Generated stories were less coherent and often deviated from the prompt.
- After Fine-tuning: The stories became more engaging and aligned closely with the given prompts.

```
Enter promt here : A group of children discover a dead body.
```

BEFORE FINE-TUNING

A group of children discover a dead body.

The group of children discover a dead body.

The group of children discover a dead

AFTER FINE-TUNING

```
The king of the country, who was watching the procession, suddenly
heard the sound of footsteps, and at first he did not know what it was. He
saw the bodies lying on the ground, but he could not see them, for
the bodies were lying where they had been, like dead people. The king
was terrified, as it seemed to him that some of them had crept into the
grave. But he made no objections; the children would not listen to
him. And so he waited patiently until at length the next morning the king was
caught by a terrible noise, which sounded like thunder. This terrible sound
soon became clearer and clearer, till at last the dead child was found,
and the frightened king called to his servants to come to the spot. There were
no guards at all, only the servants who came to pick him up. When the poor
king saw that he was safe
```

SpeechT5 Text-to-Speech

- Before Fine-tuning: Speech outputs were less natural and had noticeable artifacts.
- After Fine-tuning: The speech became more natural and pleasant to listen to, closely mimicking human speech.

Stable Diffusion Video Generation

- Before Training: Initial visual outputs were less coherent and lacked detail.
- After Training: Generated videos became more detailed and visually consistent with the story content.

THANK YOU !!

AI AVENGERS