Name: Ananya Shet

SRN: PES2UG23CS065

Section : A

Unit 1 Hands-on: Generative AI & NLP Fundamentals

1. Introduction & Setup

We're setting up the environment and loading the data we'll work with. Pipeline abstracts all the complicated model steps.

Introduces **Hugging Face**, pipeline(), **os** for file handling and **nltk** for traditional NLP.

2. Generative AI: Dumb vs. Smart Models

Shows the difference between **small/faster models** (distilgpt2) vs **standard/larger models** (gpt2). Uses set_seed() to get reproducible outputs.

Smaller models are faster but less coherent and large models are more relevant , better.

Set seed(42)

3. NLP Fundamentals: Under the Hood

We go deeper into how AI **actually processes text**.

3.1 Tokenization

- Splits text into pieces (tokens) that the model can understand.
- Converts tokens into numbers (IDs).

3.2 POS Tagging

POS tagging helps the model understand **grammar and sentence structure**

3.3 Named Entity Recognition (NER)

Extracts named entities like names, organizations, dates. Uses a BERT model fine-tuned for NER.

4. Advanced Applications: Comparative Analysis

4.1 Summarization

Different models trade off speed vs accuracy. Fast models are okay for quick summaries, large models for precise summaries.

4.2 Question Answering

Uses an **extractive QA model**. Gives a context and a question; model finds the answer in text.

4.3 Masked Language Modeling

Uses BERT to fill in a missing word ([MASK]) in a sentence**.**

Step 3: Fast Model (`distilgpt2`)

Let's see how the smaller model performs.

```
# Initialize the pipeline with the specific model
fast_generator = pipeline('text-generation', model='distilgpt2')

# Generate text
output_fast = fast_generator(prompt, max_length=50, num_return_sequences=1)
print(output_fast[0]['generated_text'])
```

```
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncation strategy. If you er
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/mair
Generative AI is a revolutionary technology that enables us to learn and adapt our intelligence, our intelligence and our lives.
```

Let's see what the model thinks Generative AI creates.

```
masked_sentence = "The goal of Generative AI is to create new [MASK]."
preds = mask_filler(masked_sentence)

for p in preds:
    print(f"{p['token_str']}: {p['score']:.2f}")
```

```
applications: 0.06
ideas: 0.05
problems: 0.05
systems: 0.04
information: 0.03
```

Set seed(10)

```
# Generate text
output_fast = fast_generator(prompt, max_length=50, num_return_sequences=1)
print(output_fast[0]['generated_text'])
```

```
/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
secret `HF_TOKEN` does not exist in your Colab secrets.
uthenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
will be able to reuse this secret in all of your notebooks.
se note that authentication is recommended but still optional to access public models or datasets.
rnings.warn(
g.json: 100% 762/762 [00:00<00:00, 27.1kB/s]
el.safetensors: 100% 353M/353M [00:04<00:00, 123MB/s]
ration_config.json: 100% 124/124 [00:00<00:00, 4.75kB/s]
izer_config.json: 100% 26.0/26.0 [00:00<00:00, 710B/s]
b.json: 100% 1.04M/1.04M [00:00<00:00, 1.64MB/s]
es.txt: 100% 456k/456k [00:00<00:00, 767kB/s]
izer.json: 100% 1.36M/1.36M [00:00<00:00, 2.07MB/s]
ce set to use cuda:0
cation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncation strategy. If you encod
ing `pad_token_id` to `eos_token_id`:50256 for open-end generation.
`max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/main_cl
rative AI is a revolutionary technology that lets you create a single intelligent, intelligent, intelligent machine and create a world where you can become more intelligent and smarter.
```

Let's see what the model thinks Generative AI creates.

[32] ✓ 0s

```
masked_sentence = "The goal of Generative AI is to create new [MASK]."
preds = mask_filler(masked_sentence)

for p in preds:
    print(f"{p['token_str']}: {p['score']:.2f}")
```

```
applications: 0.06
ideas: 0.05
problems: 0.05
systems: 0.04
information: 0.03
```

Set Seed(55)

43] ✓ 3s

```
# Initialize the pipeline with the specific model
fast_generator = pipeline('text-generation', model='distilgpt2')

# Generate text
output_fast = fast_generator(prompt, max_length=50, num_return_sequences=1)
print(output_fast[0]['generated_text'])
```

```
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncation strategy. If you en
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/main
Generative AI is a revolutionary technology that enables a wide range of AI systems to be applied to the world. It's a revolutionary technology that enables a wide range of AI systems to be applied to the world. It's a revo
```

[32] ✓ 0s

```
masked_sentence = "The goal of Generative AI is to create new [MASK]."
preds = mask_filler(masked_sentence)

for p in preds:
    print(f"{p['token_str']}: {p['score']:.2f}")
```

```
applications: 0.06
ideas: 0.05
problems: 0.05
systems: 0.04
information: 0.03
```

Difference between distilgpt2 and gpt2

GPT-2:

- Slower and uses more memory
- Better quality text and more coherent
- More parameter and deep layer

DistilGPT-2

- Smaller and faster
- Uses less RAM and runs well on low end systems
- Slightly lower text quality than GPT-2