

GENERATIVE AI AND ITS APPLICATIONS

Unit 1 Hands-on: Generative AI & NLP Fundamentals

NAME:Akshata Udagane

SRN:PES2UG23CS043

SEM:6

SECTION:A

Hugging Face

Its an website where we get AI model on it . We can find use download even we can test AI models -especially models for text , images speech etc

Difference between distilgpt2 and main gpt2

- 1.Distilgpt2 is trained on small size of dataset where as main gpt2 is trained on huge dataset
- 2.speed is faster for distilgpt2 than gpt2 / low latency
- 3.Distilgpt2 is cheaper than gpt2 as memory use is low compared to gpt2
- 4.Distilgpt2 is less accurate than gpt2 as Distilgpt2 is trained on less training dataset

Tokenization

Text is broken in tokens and as models won't understand English so further this is converted into numerical

POS(part-of-speech)

This is important because it provides grammatical structure and to understand ambiguity like which is verb and noun , particular word sometimes behave as noun and sometimes verb . So this Pos helps to understand grammar.

NER(Named Entity Recognition)

It finds names in text and says what it is,example 'Raju studies at PES '

NER tags would be

Raju : Person

Pes : Organization

So the model learns who is person , which is place organization etc

Seed Value

This is number just to generate random number so that it produces same random results everytime we run.

```
set_seed(42)

distilgpt2
```

... /usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
config.json: 100% [██████████] 762/762 [00:00<00:00, 9.96kB/s]
model.safetensors: 100% [██████████] 353M/353M [00:03<00:00, 182MB/s]
generation_config.json: 100% [██████████] 124/124 [00:00<00:00, 3.00kB/s]
tokenizer_config.json: 100% [██████████] 26.0/26.0 [00:00<00:00, 828B/s]
vocab.json: 100% [██████████] 1.04M/1.04M [00:00<00:00, 4.55MB/s]
merges.txt: 100% [██████████] 456k/456k [00:00<00:00, 16.5MB/s]
tokenizerjson: 100% [██████████] 1.36M/1.36M [00:00<00:00, 5.28MB/s]
Device set to use cpu
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly tr
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
Generative AI is a revolutionary technology that is designed to work with existing AI systems. It has been developed by the Universi

The research team led by Professor Daniel Kranz, from the University of California, Berkeley, has developed a program to learn how to
The research team developed the program to learn how to use the AI to improve the performance of the software. It has been developed
The research team developed the program to learn how to use the AI to improve the performance of the software. It has been developed

gpt2

```
*** config.json: 100% [665/665] [00:00<00:00, 32.3kB/s]
model.safetensors: 100% [548M/548M] [00:03<00:00, 321MB/s]
generation_config.json: 100% [124/124] [00:00<00:00, 9.10kB/s]
tokenizer_config.json: 100% [26.0/26.0] [00:00<00:00, 1.93kB/s]
vocab.json: 100% [1.04M/1.04M] [00:00<00:00, 4.62MB/s]
merges.txt: 100% [456k/456k] [00:00<00:00, 4.37MB/s]
tokenizer.json: 100% [1.36M/1.36M] [00:00<00:00, 9.65MB/s]

Device set to use cpu
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate the generated text.
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
Generative AI is a revolutionary technology that allows users to build AI that can help solve complex problems. It brings together h
The AI is a model of human intelligence, and has many aspects that are similar to artificial intelligence. It can learn from humans,
It is the main driving force behind the new Artificial Intelligence, and the AI is very important to the success of AI. The new AI i
The AI is designed to be scalable and adaptable to different environments. It can be used to solve complex problems without relying o
The new AI is designed to work out problems that need to be solved in a way that is
```

Now seed value is 70

```
set_seed(70)
```

distilgpt2

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 cpu
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate the generated text.
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
Generative AI is a revolutionary technology that can transform the way humans interact with the world as we know it.
```

gpt2

```
▶ smart_generator = pipeline('text-generation', model='gpt2')

output_smart = smart_generator(prompt, max_length=50, num_return_sequences=1)
print(output_smart[0]['generated_text'])

*** Device set to use cpu
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate the generated text.
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
Generative AI is a revolutionary technology that enables intelligent AI to replace human agents. The AI is based on the concept of n
The goal of these methods is to eliminate the natural selection bias in our society, which is the natural bias is
When we are presented with a choice, we recognize that we are the ones that are being selected. However, when we are presented with
In a free society, where we have choices about whether we are good or bad, that is, if we are good or bad, we are at risk for being
What we should be doing is
```

Seed Value when changed from 45 to 70

When the seed value was changed from 45 to 70, both models produced different continuations of the same prompt. This shows that seed controls the randomness of text generation. DistilGPT-2, being a smaller distilled model, produced shorter and simpler sentences, while GPT-2 generated longer and more detailed text with better coherence. The important point is that even with the same seed values, both models did not produce identical outputs, because seeds only reproduce randomness within a model, not across different models.

Explanation of each cell

Cell1: `from transformers import pipeline, set_seed, GPT2Tokenizer`

Meaning: Loads the main tools from Hugging Face needed for text generation and tokenization.

Cell2:

```
import os  
import nltk
```

Meaning: os helps read files, nltk helps with classical NLP tasks.

Cell3: `file_path = "unit 1.txt"`

Meaning: Tells Python where our study material file is.

Cell4:

```
with open(file_path, "r", encoding="utf-8") as f:  
    text = f.read()
```

Meaning: Reads the text file into a Python string variable.

Cell5: `print(text[:500])`

Meaning: Shows first 500 characters for sanity check.

Cell6: **set_seed(42)**

Meaning:Fixes randomness so text output is repeatable.

Cell7: **prompt = "Generative AI is a revolutionary technology that"**

Meaning:Sets the starting sentence the model will continue.

Cell8: **fast_generator = pipeline('text-generation', model='distilgpt2')**

Meaning:Loads the smaller DistilGPT-2 model for generation.

Cell9:

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

Meaning:Generates text & prints it.

Cell10: **smart_generator = pipeline('text-generation', model='gpt2')**

Meaning:Loads the full GPT-2 model.

Cell11:

```
output_smart = smart_generator(prompt, max_length=50, num_return_sequences=1)
print(output_smart[0]['generated_text'])
```

Meaning:Generates text again, but with better model.

Cell12: **tokenizer = GPT2Tokenizer.from_pretrained("gpt2")**

Meaning:Loads the tokenizer used by GPT-2.

Cell13: **sample_sentence = "Transformers revolutionized NLP."**

Meaning:Use a simple example to show how tokenization works

Cell14:

```
tokens = tokenizer.tokenize(sample_sentence)
```

```
print(tokens)
```

Meaning:Breaks sentence into tokens.

Cell15:

```
token_ids = tokenizer.convert_tokens_to_ids(tokens)
```

```
print(token_ids)
```

Meaning:Maps tokens into numbers.

Cell16:

```
pos_tags = nltk.pos_tag(nltk.word_tokenize(sample_sentence))
```

```
print(pos_tags)
```

Meaning:Labels each word with part-of-speech.

Cell17:

```
ner_pipeline = pipeline("ner", model="dbmdz/bert-large-cased-finetuned-conll03-english", aggregation_strategy="simple")
```

Meaning:Loads a BERT model trained to detect entities.

Cell18: **entities = ner_pipeline(text[:1000])**

Meaning:Extracts named entities from the first 1000 characters.

Cell19:

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

```
res_fast = fast_sum(transformer_section, max_length=60, min_length=30, do_sample=False)
```

Meaning:Ask fast model to summarize the paragraph.

Cell20:

```
smart_sum = pipeline("summarization", model="facebook/bart-large-cnn")
```

```
res_smart = smart_sum(transformer_section, max_length=60, min_length=30,
do_sample=False)
```

Meaning: Ask strong model to summarize same text better

Cell21:

```
qa_pipeline = pipeline("question-answering", model="distilbert-base-cased-distilled-
squad")
```

```
res = qa_pipeline(question=q, context=text[:5000])
```

Meaning: Give text + ask a question → model extracts answer from text

Cell 22:

```
mask_filler = pipeline("fill-mask", model="bert-base-uncased")
```

```
preds = mask_filler(masked_sentence)
```

Meaning: Hide a word and let model guess what fits