

# GEN-AI HANDS-ON-1

**NAME:ARJUN S KANNUR**  
**SRN:PES2UG23CS090**

## Observation:

The transformers library acts as a bridge between our python code and AI models available on hugging face. It makes handling models simpler by using a function called pipeline(). Pipeline handles everything automatically. (preprocessing , inference and postprocessing)

We will be looking into two models - 1) distilGPT2  
2) gpt2

seed (42)

```
[4]: 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
```

### Step 3: Fast Model (distilgpt2)

Let's see how the smaller model performs.

```
[28]: # 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 examples to ...
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 ...
Generative AI is a revolutionary technology that is designed to work with existing AI systems. It has been developed by the University of California, Berkeley.
```

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

#### Step 4: Standard Model (gpt2)

Now let's try the standard model.

```
⌚ 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 examples to max length. Defaulting to 'longest_first' truncation.
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/generation#text-generation)
Generative AI is a revolutionary technology that allows users to build AI that can help solve complex problems. It brings together hundreds of different fields of knowledge and expertise to create something new and powerful.
The AI is a model of human intelligence, and has many aspects that are similar to artificial intelligence. It can learn from humans, and it can adapt to new situations.
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 is designed to work in different environments.
The AI is designed to be scalable and adaptable to different environments. It can be used to solve complex problems without relying on humans. It can be used in various industries.
The new AI is designed to work out problems that need to be solved in a way that is
```

#### 4.2 Question Answering

This task is **Extractive**. We provide a `context` (our text) and a `question`. The model highlights the answer within the text.

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

...
config.json: 100% [██████████] 473/473 [00:00<00:00, 35.4kB/s]
model.safetensors: 100% [██████████] 261M/261M [00:06<00:00, 27.3MB/s]
tokenizer_config.json: 100% [██████████] 49.0/49.0 [00:00<00:00, 3.67kB/s]
vocab.txt: 100% [██████████] 213k/213k [00:00<00:00, 6.29MB/s]
tokenizer.json: 100% [██████████] 436k/436k [00:00<00:00, 6.56MB/s]
Device set to use cpu
```

Let's ask about the risks mentioned in our text.

```
questions = [
    "What is the fundamental innovation of the Transformer?",
    "What are the risks of using Generative AI?"
]

for q in questions:
    res = qa_pipeline(question=q, context=text[:5000])
    print(f"\nQ: {q}")
    print(f"A: {res['answer']}")
```

Q: What is the fundamental innovation of the Transformer?  
A: to identify hidden patterns, structures, and relationships within the data

Q: What are the risks of using Generative AI?  
A: data privacy, intellectual property, and academic integrity

seed(70)

#### Step 3: Fast Model (distilgpt2)

Let's see how the smaller model performs.

```
prompt = "Generative AI is a revolutionary technology that"

# 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 examples to max length. Defaulting to 'longest_first' truncation.
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/generation#text-generation)
Generative AI is a revolutionary technology that advances the understanding of natural selection and the capacity for scientific research. As it turns out, the current AI technology is much more complex than we ever imagined.
```

In a study published in the Journal of Experimental Computational Biology (JETB), researchers from the University of Virginia, and researchers at University of Michigan, conducted in collaboration with the University of California Berkeley, found that distilGPT2 can generate text that is indistinguishable from human-written text in terms of quality and fluency.

distilGPT2 remains fast but inconsistent across different seed values.

#### Step 4: Standard Model (gpt2)

Now let's try the standard model.

```
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 examples to max length. Defaulting to 'longest_first' truncation.
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/generation#text-generation)
Generative AI is a revolutionary technology that enables intelligent AI to replace human agents. The AI is based on the concept of natural selection. Natural selection by natural selection is an important goal of these methods.
The goal of these methods is to eliminate the natural selection bias in our society, which is the natural bias. The natural bias is a result of the cognitive, behavioral and psychological processes that we are presented with a choice, we recognize that we are the ones that are being selected. However, when we are presented with a choice, they are also the ones that are being selected. If the choice is 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 selected.
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 selected.
What we should be doing is
```

Although it is readable , the phrasing and sentence order for seed value 70 is different in comparison to seed value 42. The model seemed to be better for seed value 42 as it has less repetition and the logical flow was better.

#### 4.2 Question Answering

This task is **Extractive**. We provide a `context` (our text) and a `question`. The model highlights the answer within the text.

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

Device set to use cpu

Let's ask about the risks mentioned in our text.

questions = [
    "What is the fundamental innovation of the Transformer?",
    "What are the risks of using Generative AI?"
]

for q in questions:
    res = qa_pipeline(question=q, context=text[:5000])
    print(f"\nQ: {q}")
    print(f"A: {res['answer']}")
```

```
Q: What is the fundamental innovation of the Transformer?
A: to identify hidden patterns, structures, and relationships within the data

Q: What are the risks of using Generative AI?
A: data privacy, intellectual property, and academic integrity
```

It remains the same for both the seed values

#### Conclusion:

There were noticeable changes with respect to change in seed value while comparing the two models (distilGPT2 and GPT2). However , summarization ,question answering and masked language modelling outputs did remain quite similar to each other indicating they are not much affected by the seed value.

