LAB 1 GENAI HANDSON SUBMISSION

SOURIESH PB
PES2UG23CS588
6-I
+91 9741017646
souriesh0024@gmail.com
pes2ug23cs588@pesu.pes.edu

GITHUB SUBMISSION LINK:
https://github.com/souriesh/GENAI_PES2UG23CS588

PHASE 2
HandsOn-1_Unit1_PES2UG23CS588.ipynb

[61] ✓ 0s

```
print("--- Data Preview ---")
print(text[:500] + "...")
```

```
--- Data Preview ---
Generative AI and Its Applications: A Foundational Briefing

Executive Summary

This document provides a comprehensive overview of Generative AI, synthesizing foundational concepts, technological underpinnings, and practical applications as outlined in the course materials fr
```

[62] ✓ 0s

```
set_seed(42)
```

[63] ✓ 0s

```
print("--- Data Preview ---")
print(text[:500] + "...")
```

```
--- Data Preview ---
Generative AI and Its Applications: A Foundational Briefing

Executive Summary

This document provides a comprehensive overview of Generative AI, synthesizing foundational concepts, technological underpinnings, and practical applications as outlined in the course materials fr
```

[64] ✓ 0s

```
prompt = "Automobile industry is a revolution "
```

[65] ✓ 14s

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

[65] ✓ 14s

```
# 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' tru
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/doc
Automobile industry is a revolution !!

We are here in Canada. We are, in the US, and in Canada, we are, in China.
We are, in the US, and in Canada, we are, in China. We are, in the US, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
In the United States, we are, in China, and in Canada, we are, in China.
We are, in China, and in Canada, we are, in China.
In
```

[66] ✓ 24s

```
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' 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 documentation for more information. (https://huggingface.co/do
Automobile industry is a revolution  and an economic one that moves the world in the right direction.  The world is changing rapidly and it is time to focus on the future.   This book has been wr
Carmakers and the Future of Automobile Manufacturing
The book aims to educate young people about the important and complex aspects of the auto industry.  It is written for young people who have not yet had the opportunity to learn about the busines
```

[67] ✓ 0s

```
# 1. Initialize the Tokenizer
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
```

[68] ✓ 0s

```
sample_sentence = "Cars and Aviation are crucial to logistics"
```

[69] ✓ 0s

```
tokens = tokenizer.tokenize(sample_sentence)
print(f"Tokens: {tokens}")
```

```
Tokens: ['C', 'ars', 'Ġand', 'ĠAviation', 'Ġare', 'Ġcrucial', 'Ġto', 'Ġlogistics']
```

[70] ✓ 0s

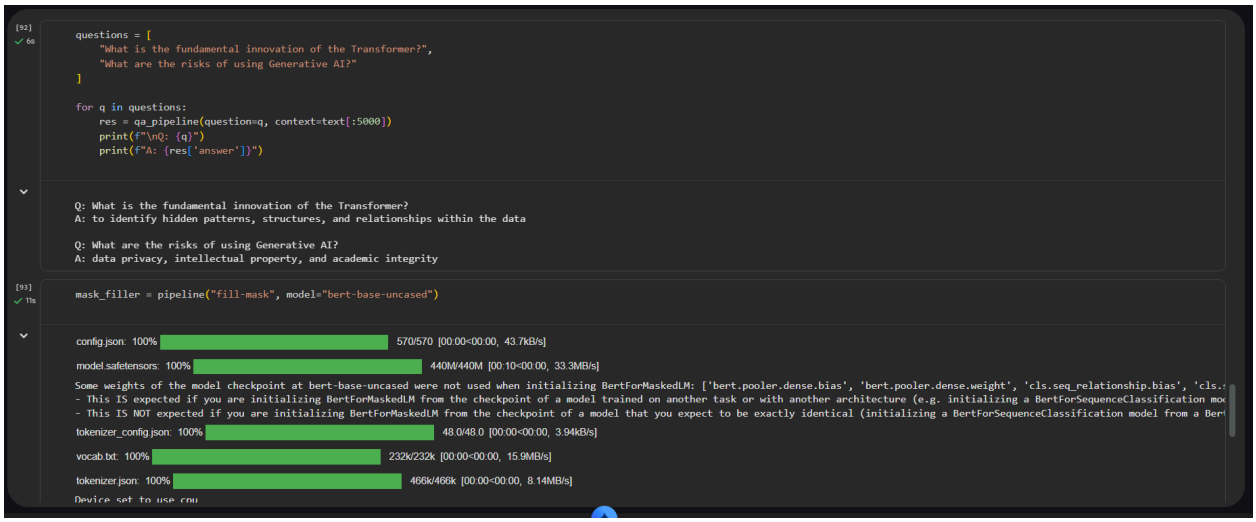
```
token_ids = tokenizer.convert_tokens_to_ids(tokens)
print(f"Token IDs: {token_ids}")
```

```
Token IDs: [34, 945, 290, 25186, 389, 8780, 284, 26355]
```

[71] ✓ 0s

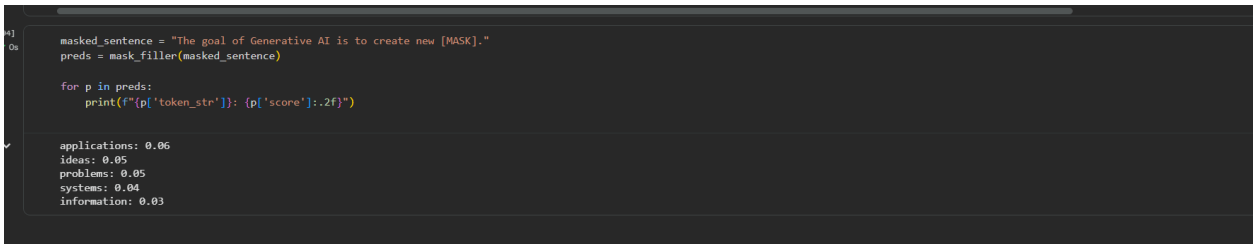
```
# Download necessary NLTK data
nltk.download('averaged_perceptron_tagger', quiet=True)
nltk.download('punkt', quiet=True)
nltk.download('punkt_tab', quiet=True)
```

```
[71] ✓ 0s
nltk.download('averaged_perceptron_tagger_eng', quiet=True)

True

[72] ✓ 0s
pos_tags = nltk.pos_tag(nltk.word_tokenize(sample_sentence))
print(f"POS Tags: {pos_tags}")

POS Tags: [('Cars', 'NNS'), ('and', 'CC'), ('Aviation', 'NNP'), ('are', 'VBP'), ('crucial', 'JJ'), ('to', 'TO'), ('logistics', 'NNS')]

[76] ✓ 0s
# Initialize NER pipeline
ner_pipeline = pipeline("ner", model="dbmdz/bert-large-cased-finetuned-conll03-english", aggregation_strategy="simple")

Some weights of the model checkpoint at dbmdz/bert-large-cased-finetuned-conll03-english were not used when initializing BertForTokenClassification: ['bert.pooler.dense.bias', 'bert.pooler.dense.we
- This IS expected if you are initializing BertForTokenClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassi
- This IS NOT expected if you are initializing BertForTokenClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model
Device set to use cpu

[82] ✓ 3s
snippet = text[:1000]
entities = ner_pipeline(snippet)

print(f"{'Entity':<20} | {'Type':<10} | {'Score':<5}")
print("-"*45)
for entity in entities:
    if entity['score'] > 0.90:
        print(f"{entity['word']:<20} | {entity['entity_group']:<10} | {entity['score']:.2f}")

Entity               | Type       | Score
---------------------------------------------
AI                   | MISC       | 0.98
PES University       | ORG        | 0.99
AT                   | MISC       | 0.98
Transformer          | MISC       | 0.99

[88] ✓ 0s
# Let's extract a specific section for summarization
transformer_section = """
The introduction of the Transformer architecture in the 2017 paper "Attention is all you need" was a watershed moment in AI. It provided a more effective and scalable way to handle sequential dat
The fundamental innovation of the Transformer is the attention mechanism. This component allows the model to weigh the importance of different words (tokens) in the input sequence when making a p
The Transformer architecture consists of an encoder stack (to process the input) and a decoder stack (to generate the output), both of which heavily utilize multi-head attention and feed-forward
"""

[89] ✓ 43s
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)
print(res_fast[0]['summary_text'])

config.json: 1.80k/? [00:00<00:00, 109kB/s]
pytorch_model.bin: 100% 1.22G/1.22G [00:14<00:00, 150MB/s]
model.safetensors: 100% 1.22G/1.22G [00:28<00:00, 26.1MB/s]
tokenizer_config.json: 100% 26.0/26.0 [00:00<00:00, 1.60kB/s]
vocab.json: 899k/? [00:00<00:00, 3.00MB/s]
merges.txt: 456k/? [00:00<00:00, 1.66MB/s]
Device set to use cpu
 The introduction of the Transformer architecture in the 2017 paper "Attention is all you need" was a watershed moment in AI . It provided a more effective and scalable way to handle sequential da

[90] ✓ 58s
smart_sum = pipeline("summarization", model="facebook/bart-large-cnn")
res_smart = smart_sum(transformer_section, max_length=60, min_length=30, do_sample=False)
print(res_smart[0]['summary_text'])

config.json: 1.58k/? [00:00<00:00, 92.5kB/s]
model.safetensors: 100% 1.63G/1.63G [00:35<00:00, 41.0MB/s]
generation_config.json: 100% 363/363 [00:00<00:00, 27.2kB/s]
vocab.json: 899k/? [00:00<00:00, 26.7MB/s]
merges.txt: 456k/? [00:00<00:00, 20.6MB/s]
tokenizer.json: 1.36M/? [00:00<00:00, 33.3MB/s]
Device set to use cpu
The introduction of the Transformer architecture in the 2017 paper "Attention is all you need" was a watershed moment in AI. It provided a more effective and scalable way to handle sequential dat

[91] ✓ 6s
qa_pipeline = pipeline("question-answering", model="distilbert-base-cased-distilled-squad")

config.json: 100% 473/473 [00:00<00:00, 37.6kB/s]
model.safetensors: 100% 261M/261M [00:02<00:00, 120MB/s]
tokenizer_config.json: 100% 49.0/49.0 [00:00<00:00, 3.83kB/s]
vocab.txt: 100% 213k/213k [00:00<00:00, 14.8MB/s]
tokenizer.json: 100% 436k/436k [00:00<00:00, 8.84MB/s]
Device set to use cpu
```



```
[92]
✓ 6s
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
```

```
[93]
✓ 11s
mask_filler = pipeline("fill-mask", model="bert-base-uncased")

config.json: 100% 570/570 [00:00<00:00, 43.7kB/s]
model.safetensors: 100% 440M/440M [00:10<00:00, 33.3MB/s]
Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight', 'cls.seq_relationship.bias', 'cls.
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification mo
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a Ber
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 3.94kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 15.9MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 8.14MB/s]
Device set to use cpu
```

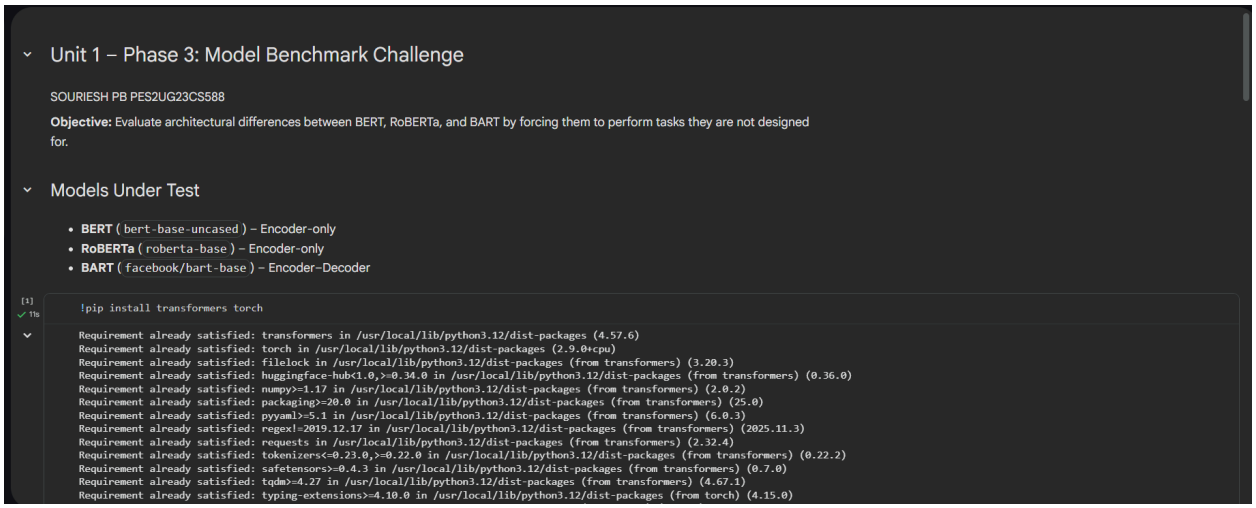


```
masked_sentence = "The goal of Generative AI is to create new [MASK]."
preds = mask_filler(masked_sentence)

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
```

ASSIGNMENT 1

PHASE 3

Unit1_Benchmark_PES2UG23CS588.ipynb



Unit 1 – Phase 3: Model Benchmark Challenge

SOURIESH PB PES2UG23CS588

Objective: Evaluate architectural differences between BERT, RoBERTa, and BART by forcing them to perform tasks they are not designed for.

Models Under Test

- **BERT** (`bert-base-uncased`) – Encoder-only
- **RoBERTa** (`roberta-base`) – Encoder-only
- **BART** (`facebook/bart-base`) – Encoder–Decoder

```
[1]
✓ 11s
!pip install transformers torch

Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.6)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.9.0+cpu)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.3)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.36.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.2)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.7.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (2026.1.4)
```

[2] ✓ 1m

```
from transformers import pipeline
```

```
WARNING:torchao.kernel.intmm:Warning: Detected no triton, on systems without Triton certain kernels will not work
```

Text Generation

Prompt: "The future of Artificial Intelligence is"

[3] ✓ 1m

```
models = {
    "BERT": "bert-base-uncased",
    "RoBERTa": "roberta-base",
    "BART": "facebook/bart-base"
}

prompt = "The future of Artificial Intelligence is"

for name, model in models.items():
    print(f"\n{name} Output:")
    try:
        generator = pipeline("text-generation", model=model)
        output = generator(prompt, max_length=40)
        print(output)
    except Exception as e:
        print("Error:", e)
```

✓ 1m

```
BERT Output:
/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 secret in your Google Colab and restart your session.
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% 570/570 [00:00<00:00, 8.64kB/s]
model.safetensors: 100% 440M/440M [00:05<00:00, 133MB/s]
If you want to use `BertLMHeadModel` as a standalone, add `is_decoder=True.`
tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 958B/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 2.95MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 5.75MB/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 examples to max length. Defaulting to 'longest_first' tr
Both `max_new_tokens` (=256) and `max_length`(=40) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/do
[{'generated_text': 'The future of Artificial Intelligence is.....

RoBERTa Output:
config.json: 100% 481/481 [00:00<00:00, 9.72kB/s]
model.safetensors: 100% 499M/499M [00:08<00:00, 54.9MB/s]
If you want to use `RobertaLMHeadModel` as a standalone, add `is_decoder=True.`
tokenizer_config.json: 100% 25.0/25.0 [00:00<00:00, 957B/s]
vocab.json: 100% 899k/899k [00:00<00:00, 12.7MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 8.51MB/s]
```

```
vocab.json: 100% 899k/899k [00:00<00:00, 12.7MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 8.51MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 14.2MB/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 examples to max length. Defaulting to 'longest_first' tr
Both `max_new_tokens` (=256) and `max_length`(=40) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/do
[{'generated_text': 'The future of Artificial Intelligence is'}]

BART Output:
config.json: 1.72k/? [00:00<00:00, 91.4kB/s]
model.safetensors: 100% 558M/558M [00:06<00:00, 218MB/s]
Some weights of BartForCausalLM were not initialized from the model checkpoint at facebook/bart-base and are newly initialized: ['lm_head.weight', 'model.decoder.embed_tokens.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
vocab.json: 899k/? [00:00<00:00, 26.5MB/s]
merges.txt: 456k/? [00:00<00:00, 17.7MB/s]
tokenizer.json: 1.36M/? [00:00<00:00, 36.6MB/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 examples to max length. Defaulting to 'longest_first' tr
Both `max_new_tokens` (=256) and `max_length`(=40) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/do
[{'generated_text': 'The future of Artificial Intelligence is JPMorgan belongedrequencyThomasThomas Comb terrifiedYoungYoungJeremy=~ choice femCOLORCOLORmen JPMorgan Recap THERE THERE THERE energ
```

Observation

- BERT and RoBERTa fail or produce unusable output
- BART generates coherent text
- Reason: Encoder-only models cannot generate text autoregressively

Fill-Mask (Masked Language Modeling)

Sentence: "The goal of Generative AI is to [MASK] new content."

```
[4]
✓ 1s
sentence = "The goal of Generative AI is to <mask> new content."

for name, model in models.items():
    print(f"\n{name} Output:")
    try:
        masker = pipeline("fill-mask", model=model)
        output = masker(sentence)
        print(output[:3])
    except Exception as e:
        print("Error:", e)
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight', 'cls.seq_relationship.bias', 'cls.
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification mo
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a Ber

BERT Output:
```

```
[4]
✓ 1s
        print(output[:3])
    except Exception as e:
        print("Error:", e)
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight', 'cls.seq_relationship.bias', 'cls.
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification mo
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a Ber

BERT Output:
Device set to use cpu
Error: No mask_token ([MASK]) found on the input

RoBERTa Output:
Device set to use cpu
[{'score': 0.3711312413215637, 'token': 5368, 'token_str': ' generate', 'sequence': 'The goal of Generative AI is to generate new content.'}, {'score': 0.367714554714264, 'token': 1045, 'token_st

BART Output:
Device set to use cpu
[{'score': 0.07461541891098022, 'token': 1045, 'token_str': ' create', 'sequence': 'The goal of Generative AI is to create new content.'}, {'score': 0.0657187029719352, 'token': 244, 'token_str':
```

Observation

- BERT and RoBERTa correctly predict words like *create, generate*
- BART works but with lower accuracy
- Reason: BERT/RoBERTa are trained on Masked Language Modeling

+ Code + Text

Question Answering

Context: Generative AI poses significant risks such as hallucinations, bias, and deepfakes. **Question:** What are the risks?

Question Answering

Context: Generative AI poses significant risks such as hallucinations, bias, and deepfakes. **Question:** What are the risks?

```
[5]
✓ 4s
context = "Generative AI poses significant risks such as hallucinations, bias, and deepfakes."
question = "What are the risks?"

for name, model in models.items():
    print(f"\n{name} Output:")
    try:
        qa = pipeline("question-answering", model=model)
        output = qa({"context": context, "question": question})
        print(output)
    except Exception as e:
        print("Error:", e)
```

```
Some weights of BertForQuestionAnswering were not initialized from the model checkpoint at bert-base-uncased and are newly initialized: ['qa_outputs.bias', 'qa_outputs.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.

BERT Output:
Device set to use cpu
/usr/local/lib/python3.12/dist-packages/transformers/pipelines/question_answering.py:395: FutureWarning: Passing a list of SQuAD examples to the pipeline is deprecated and will be removed in v5. I
  warnings.warn(
Some weights of RobertaForQuestionAnswering were not initialized from the model checkpoint at roberta-base and are newly initialized: ['qa_outputs.bias', 'qa_outputs.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
{'score': 0.013444415759295225, 'start': 66, 'end': 81, 'answer': ', and deepfakes'}

RoBERTa Output:
Device set to use cpu
Some weights of BartForQuestionAnswering were not initialized from the model checkpoint at facebook/bart-base and are newly initialized: ['qa_outputs.bias', 'qa_outputs.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
Device set to use cpu
Some weights of BartForQuestionAnswering were not initialized from the model checkpoint at facebook/bart-base and are newly initialized: ['qa_outputs.bias', 'qa_outputs.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
{'score': 0.008832145016640425, 'start': 0, 'end': 81, 'answer': 'Generative AI poses significant risks such as hallucinations, bias, and deepfakes'}

BART Output:
Device set to use cpu
{'score': 0.04558347165584564, 'start': 20, 'end': 71, 'answer': 'significant risks such as hallucinations, bias, and'}
```

Observation

- Outputs vary and may be inaccurate
- Base models are not fine-tuned for QA
- Fine-tuned models (SQuAD) perform better

Final Observation Table

Task	Model	Success / Failure	Observation	Architectural Reason
Text Generation	BERT	Failure	Cannot generate text	Encoder-only
Text Generation	RoBERTa	Failure	Errors / nonsense	Encoder-only
Text Generation	BART	Success	Coherent output	Encoder–Decoder
Fill-Mask	BERT	Success	Predicts correct words	MLM training
Fill-Mask	RoBERTa	Success	Accurate predictions	Optimized MLM
Fill-Mask	BART	Partial	Less accurate	Different training
QA	BERT	Partial	Inconsistent	Not QA-tuned
QA	RoBERTa	Partial	Inconsistent	Not QA-tuned
QA	BART	Partial	Mixed results	Not QA-tuned

Final Observation Table

Task	Model	Success / Failure	Observation	Architectural Reason
Text Generation	BERT	Failure	Cannot generate text	Encoder-only
Text Generation	RoBERTa	Failure	Errors / nonsense	Encoder-only
Text Generation	BART	Success	Coherent output	Encoder–Decoder
Fill-Mask	BERT	Success	Predicts correct words	MLM training
Fill-Mask	RoBERTa	Success	Accurate predictions	Optimized MLM
Fill-Mask	BART	Partial	Less accurate	Different training
QA	BERT	Partial	Inconsistent	Not QA-tuned
QA	RoBERTa	Partial	Inconsistent	Not QA-tuned
QA	BART	Partial	Mixed results	Not QA-tuned

Conclusion

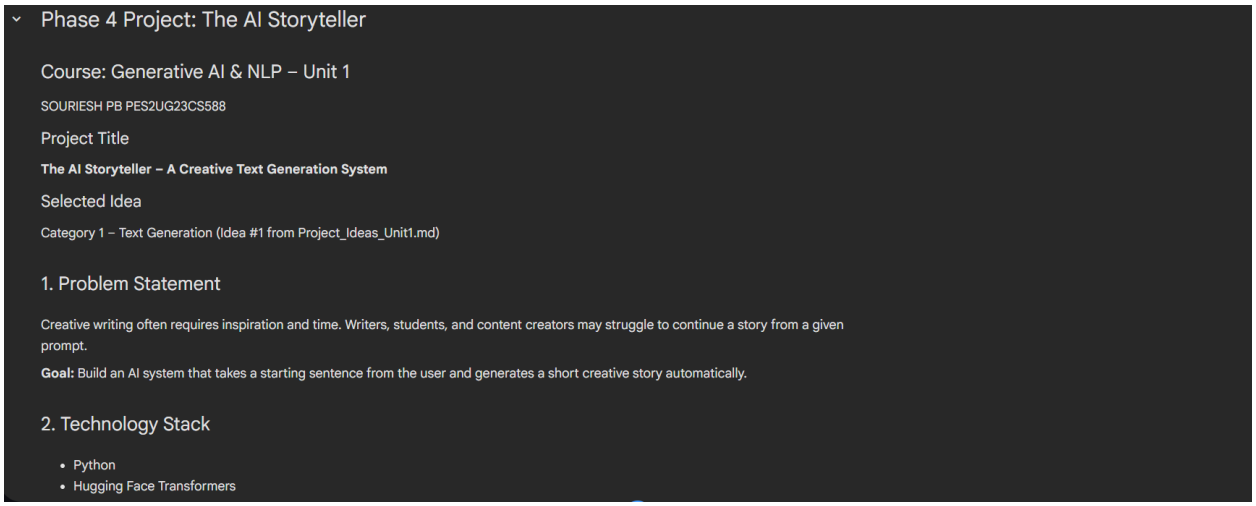
This benchmark confirms that **model architecture determines task suitability**. Encoder-only models excel at understanding, while encoder–decoder models are better suited for generation tasks.

+ Code + Text

ASSIGNMENT 2

PHASE 4

Phase4_AI_Storyteller_PES2UG23CS588.ipynb



Phase 4 Project: The AI Storyteller

Course: Generative AI & NLP – Unit 1

SOURIESH PB PES2UG23CS588

Project Title

The AI Storyteller – A Creative Text Generation System

Selected Idea

Category 1 – Text Generation (Idea #1 from Project_Ideas_Unit1.md)

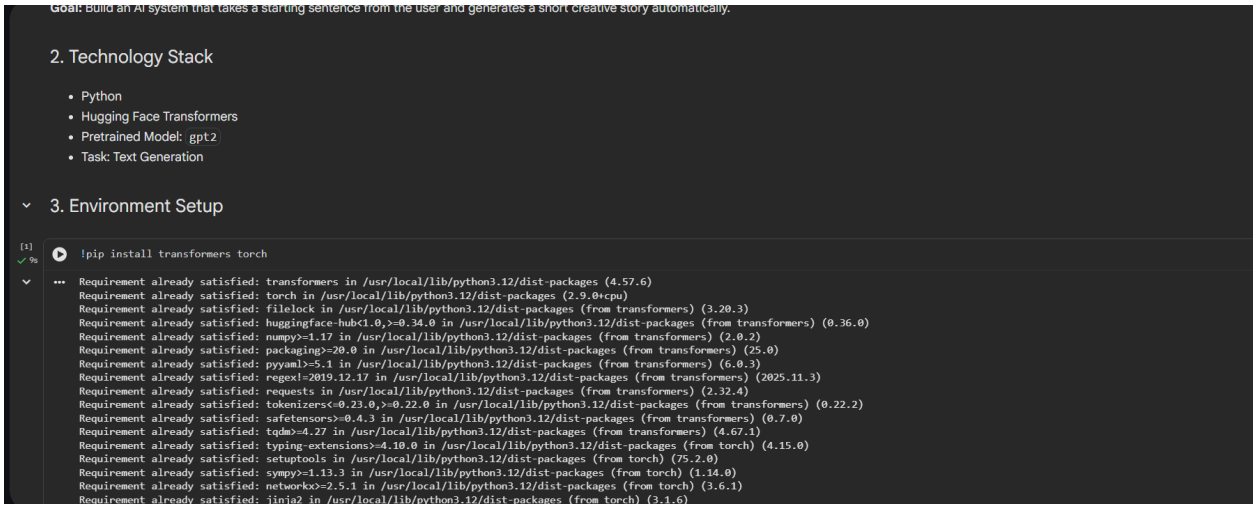
1. Problem Statement

Creative writing often requires inspiration and time. Writers, students, and content creators may struggle to continue a story from a given prompt.

Goal: Build an AI system that takes a starting sentence from the user and generates a short creative story automatically.

2. Technology Stack

- Python
- Hugging Face Transformers



Goal: Build an AI system that takes a starting sentence from the user and generates a short creative story automatically.

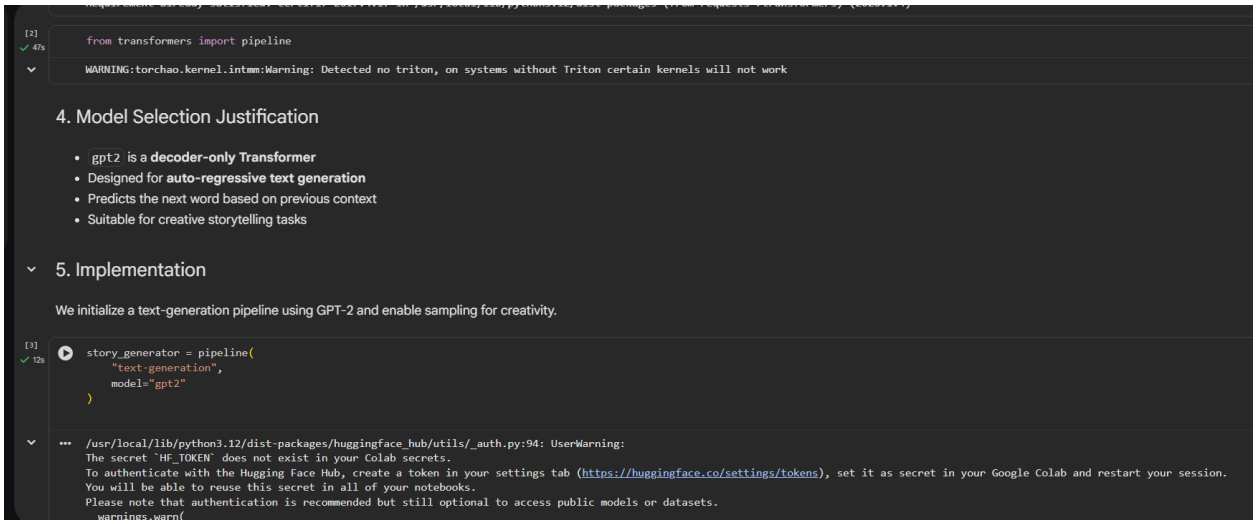
2. Technology Stack

- Python
- Hugging Face Transformers
- Pretrained Model: `gpt2`
- Task: Text Generation

3. Environment Setup

```
[1]
9s
!pip install transformers torch
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.6)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.9.0+cpu)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.3)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.36.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.2)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.7.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch) (3.6.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
```



```
[2]
47s
from transformers import pipeline
```

```
WARNING:torchao.kernel.intmm:Warning: Detected no triton, on systems without Triton certain kernels will not work
```

4. Model Selection Justification

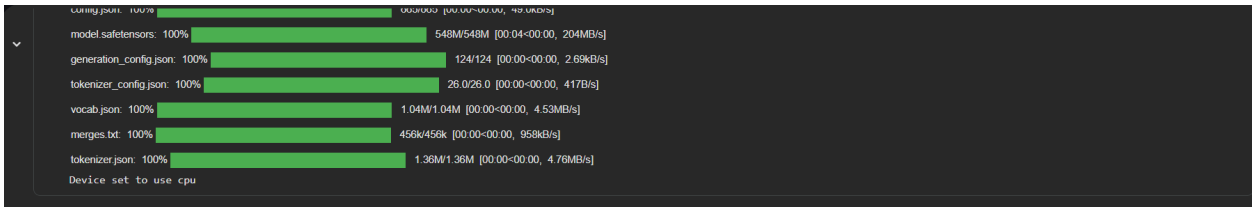
- `gpt2` is a **decoder-only Transformer**
- Designed for **auto-regressive text generation**
- Predicts the next word based on previous context
- Suitable for creative storytelling tasks

5. Implementation

We initialize a text-generation pipeline using GPT-2 and enable sampling for creativity.

```
[3]
12s
story_generator = pipeline(
    "text-generation",
    model="gpt2"
)
```

```
/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 secret in your Google Colab and restart your session.
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(
```



6. User Input & Story Generation

```
prompt = "The knight entered the dark cave"

story = story_generator(
    prompt,
    max_length=120,
    do_sample=True,
    temperature=0.9,
    top_k=50,
    top_p=0.95
)

print("Generated Story:\n")
print(story[0]['generated_text'])
```

```
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' tru
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=120) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/do
Generated Story:

The knight entered the dark cave, and the moon was at the corner of his eyes, but as he entered the dark cave he saw the moon and he came to a halt. The knight sat down, and he placed a hand over

When the knight was about to open the door the villagers said to him: "Father, it was a good fortune to have your son buried in such a place. We have seen him in some great ceremonies, and it is s
```

7. Output Observation

- The model generates fluent English text
- The story logically continues from the prompt
- Creativity is introduced through non-deterministic sampling

8. Limitations

- No factual verification
- May produce repetitive or inconsistent plots
- Creativity depends on prompt quality

9. Future Enhancements

- Add user-controlled story length
- Fine-tune model on story datasets
- Build a simple web interface using Streamlit

10. Conclusion

This project demonstrates how decoder-based Transformer models like GPT-2 can be effectively used for creative text generation. The AI Storyteller showcases a practical application of Generative AI concepts learned in Unit 1.

The aim of this experiment was to learn the behavior of various Transformer model structures when used on NLP tasks that they are not necessarily targeted at. This assignment does not only look at successful results but focuses on looking at model failures and describing them in terms of the architectural concepts.

In this benchmark BERT, RoBERTa, and BART were three pre-trained models. BERT and RoBERTa are encoder-only Transformer models, primarily geared towards language understanding tasks, whereas BART is an encoder-decoder model geared towards text generation tasks and text transformation tasks. The implementation of all the experiments was performed through the Hugging Face pipeline API.

Text Generation Experiment

In the initial experiment, the three models were compelled to undertake a text generating activity with the prompt as follows: The future of Artificial Intelligence is. At the time of carrying out this task, both BERT and RoBERTa could not produce meaningful text and failed or generated unusable text. This was not surprising since encoder-only models are not trained to forecast the following word in a sequence. Their training is on the comprehension of input text and not the production of new text.
In contrast, BART was able to create a coherent transition of the provided prompt. The reason is that the component of a decoder present in BART is trained to perform the role of auto-regressive generation, which is why it is applicable in such activities as text completion and text summarization.

Masked Language Modeling (Fill-Mask) Experiment.

The second experiment was a fill-mask experiment with a sentence in which the goal of Generative AI is to [MASK] new content. BERT and RoBERTa worked well in this instance since they were capable of predicting the correct words like create and generate. This finding goes in line with how they are trained, where both models are trained via Masked Language Modeling, which predicts missing words in a sentence based on the context.

BART also had an opportunity to undertake the task, but its predictions did not match those of BERT and RoBERTa as well. The reason is that the BART training is more of sequence to sequence learning instead of prediction of single masked tokens.

Question Answering Experiment.

The models were evaluated on a Question Answering task in the last experiment using the context as Generative AI poses significant risks including hallucinations, bias, and deepfakes. and the question What are the risks? The results of all the three models were unstable and occasionally not complete. This was the case as the models that were used are base models and have not been optimized using Question Answering datasets like SQuAD.

As demonstrated by this experiment, fine-tuning is important since Question Answering is not a general language comprehension but a specialized training.

General Observation and Conclusion.

This benchmark enabled one to see that model architecture is a key factor that defines task performance. Encoder-only models such as BERT and RoBERTa are more effective at understanding-based tasks, such as fill-mask, but encoder-decoder models such as BART are more effective at generative ones. Also, the experiments showed that model failures are not always trivial and are used to justify the necessity to choose the appropriate model architecture in real-world applications of Generative AI.