Unit 1 Hands-on: Generative AI & NLP Fundamentals

Name: Bidisha Paul
SRN: PES2UG23CS131
6B CSE

Class Notes:

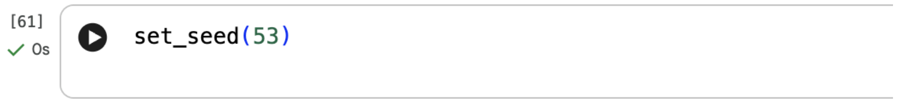
Q) What is Hugging Face?

Ans. Open source platform where you can find multiple ML models.

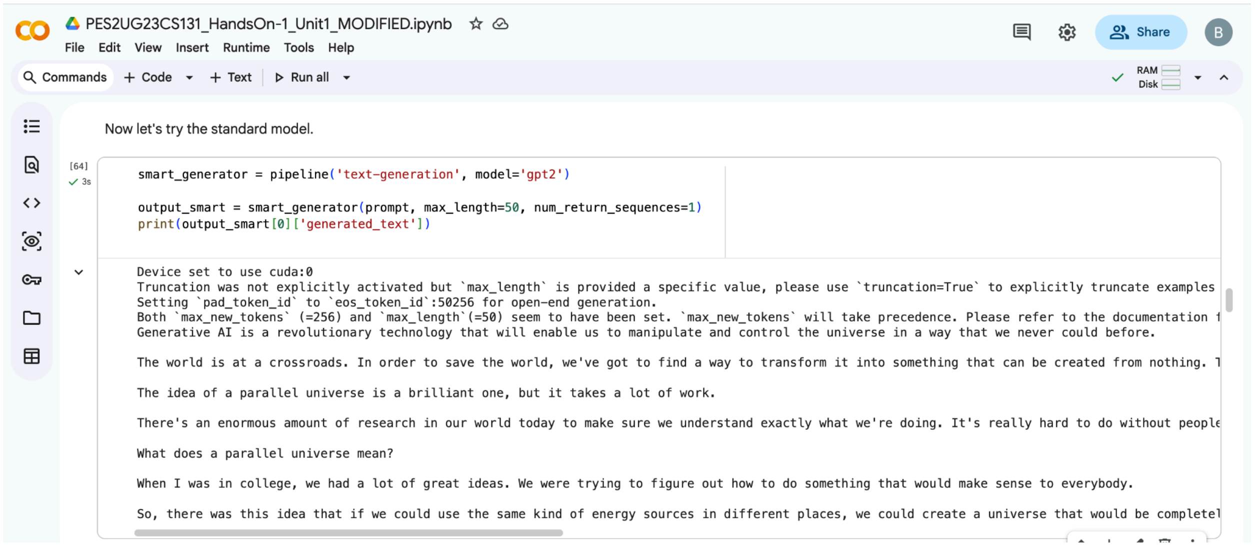
Q) SEED value

Ans. In this notebook, a seed value is used to make the random results of the text generation reproducible. When a seed is set, the random number generator starts from the same point every time, producing the same sequence of random values. This means that if you run the text generation code multiple times with the same seed, you will get the same output. If you change the seed, the output will change.

For seed value **53**(gpt2 model):



```
[61]
✓ 0s
set_seed(53)
```



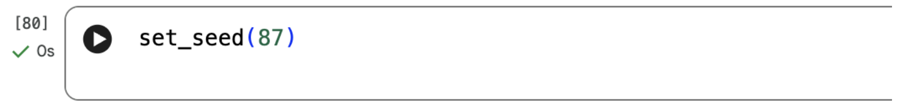
PES2UG23CS131_HandsOn-1_Unit1_MODIFIED.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code
+ Text
Run all
Share
RAM
Disk

Now let's try the standard model.

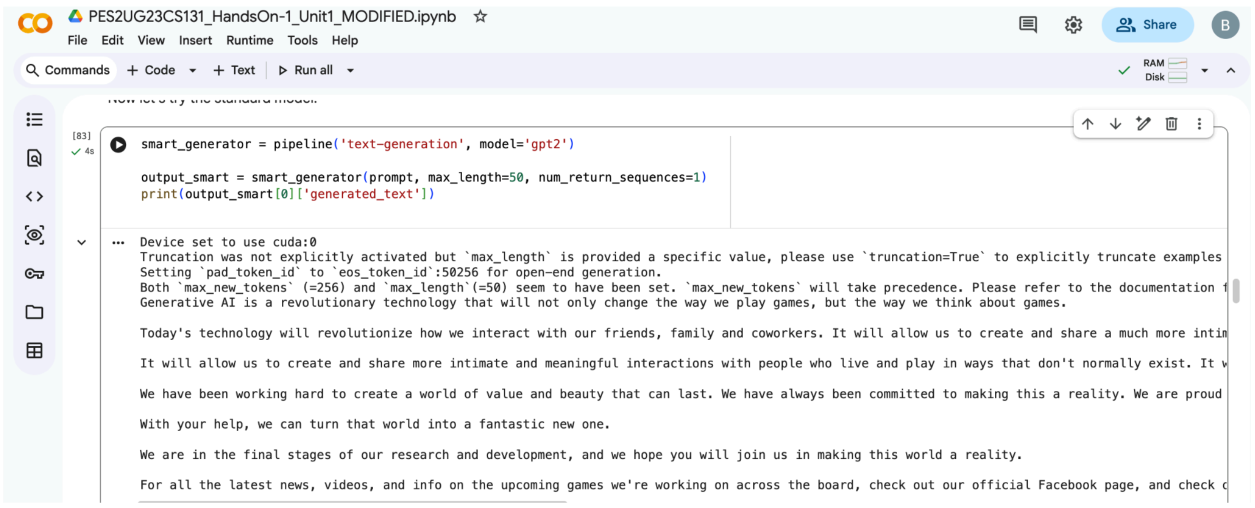
```
[64]
✓ 3s
smart_generator = pipeline('text-generation', model='gpt2')

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

```
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples
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 f
Generative AI is a revolutionary technology that will enable us to manipulate and control the universe in a way that we never could before.

The world is at a crossroads. In order to save the world, we've got to find a way to transform it into something that can be created from nothing. T

The idea of a parallel universe is a brilliant one, but it takes a lot of work.

There's an enormous amount of research in our world today to make sure we understand exactly what we're doing. It's really hard to do without people

What does a parallel universe mean?

When I was in college, we had a lot of great ideas. We were trying to figure out how to do something that would make sense to everybody.

So, there was this idea that if we could use the same kind of energy sources in different places, we could create a universe that would be completel
```

For seed value **87**(for gpt model):



```
[80]
✓ 0s
set_seed(87)
```


PES2UG23CS131_HandsOn-1_Unit1_MODIFIED.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

```
[83]
✓ 4s
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 cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples
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 f
Generative AI is a revolutionary technology that will not only change the way we play games, but the way we think about games.

Today's technology will revolutionize how we interact with our friends, family and coworkers. It will allow us to create and share a much more intim

It will allow us to create and share more intimate and meaningful interactions with people who live and play in ways that don't normally exist. It w

We have been working hard to create a world of value and beauty that can last. We have always been committed to making this a reality. We are proud

With your help, we can turn that world into a fantastic new one.

We are in the final stages of our research and development, and we hope you will join us in making this world a reality.

For all the latest news, videos, and info on the upcoming games we're working on across the board, check out our official Facebook page, and check c
```

Q)Analysis: Compare the two outputs. Does the standard model stay more on topic? Does the fast model drift into nonsense?

Ans. `distilgpt2` (Fast Model): The output from `distilgpt2` is highly repetitive, constantly repeating phrases like "It's the latest in a long line of technology that's revolutionizing the everyday lives of all." This indicates that it quickly loses coherence and drifts into a kind of repetitive "nonsense" or a loop.

`gpt2` (Standard Model): In contrast, the `gpt2` output continues to elaborate on the theme of how Generative AI will change interactions and experiences, discussing its impact on games, friends, family, and coworkers. While it also generates a longer text, it maintains a much higher level of coherence and stays more on topic, presenting a more meaningful and diverse continuation of the prompt.

Conclusion: The standard `gpt2` model indeed stays more on topic and generates more coherent text, whereas the faster `distilgpt2` model quickly drifts into repetitive and less meaningful content, confirming the difference in quality expected from smaller vs. larger models.

Q) NLP Fundatmentals:

Ans.

- **Tokenization:** It's the process of breaking down raw text into smaller units called 'tokens' (words, subwords, or characters) and assigning each a unique numerical ID that models can understand.
- **Named Entity Recognition (NER):** This is the task of identifying and classifying named entities in text into predefined categories such as person names,

organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. It helps in extracting structured information from unstructured text.

Q) Linguistics. What is POS tagging?
Ans) POS tagging is a classic **Natural Language Processing (NLP)** task where each word in a sentence is labeled with its **grammatical role**.

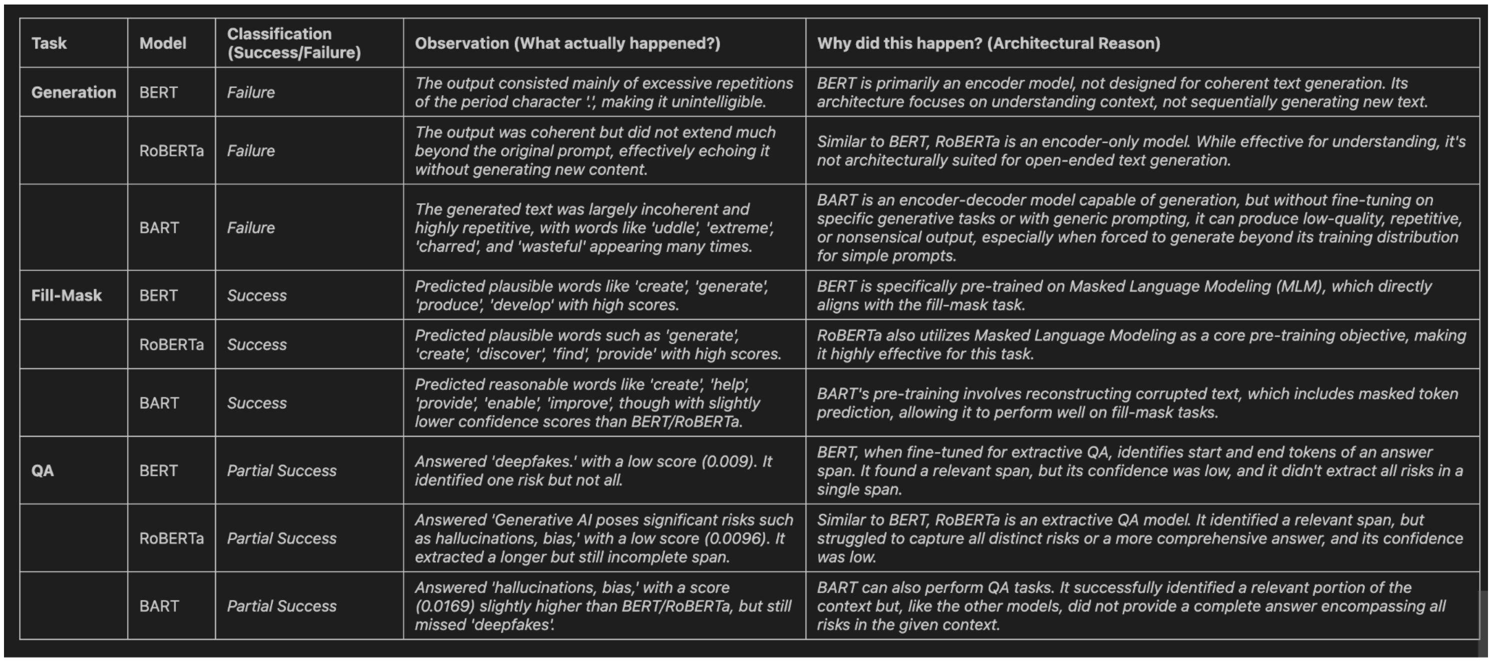
Common POS tag meanings (Penn Treebank)

- **NN** → noun (singular)
- **NNS** → noun (plural)
- **NNP** → proper noun
- **VB / VBD / VBG / VBN / VBP / VBZ** → verb forms
- **JJ** → adjective
- **RB** → adverb

Q) What is temperature?

Ans. Temperature in generative AI controls the randomness of the output. A higher temperature leads to more creative and diverse, but potentially less coherent, text. A lower temperature results in more predictable and focused, but less creative, output.

Unit1_Benchmark Observation Table



Task	Model	Classification (Success/Failure)	Observation (What actually happened?)	Why did this happen? (Architectural Reason)
Generation	BERT	*Failure*	*The output consisted mainly of excessive repetitions of the period character '.', making it unintelligible.*	*BERT is primarily an encoder model, not designed for coherent text generation. Its architecture focuses on understanding context, not sequentially generating new text.*
	RoBERTa	*Failure*	*The output was coherent but did not extend much beyond the original prompt, effectively echoing it without generating new content.*	*Similar to BERT, RoBERTa is an encoder-only model. While effective for understanding, it's not architecturally suited for open-ended text generation.*
	BART	*Failure*	*The generated text was largely incoherent and highly repetitive, with words like 'uddle', 'extreme', 'charred', and 'wasteful' appearing many times.*	*BART is an encoder-decoder model capable of generation, but without fine-tuning on specific generative tasks or with generic prompting, it can produce low-quality, repetitive, or nonsensical output, especially when forced to generate beyond its training distribution for simple prompts.*
Fill-Mask	BERT	*Success*	*Predicted plausible words like 'create', 'generate', 'produce', 'develop' with high scores.*	*BERT is specifically pre-trained on Masked Language Modeling (MLM), which directly aligns with the fill-mask task.*
	RoBERTa	*Success*	*Predicted plausible words such as 'generate', 'create', 'discover', 'find', 'provide' with high scores.*	*RoBERTa also utilizes Masked Language Modeling as a core pre-training objective, making it highly effective for this task.*
	BART	*Success*	*Predicted reasonable words like 'create', 'help', 'provide', 'enable', 'improve', though with slightly lower confidence scores than BERT/RoBERTa.*	*BART's pre-training involves reconstructing corrupted text, which includes masked token prediction, allowing it to perform well on fill-mask tasks.*
QA	BERT	*Partial Success*	*Answered 'deepfakes.' with a low score (0.009). It identified one risk but not all.*	*BERT, when fine-tuned for extractive QA, identifies start and end tokens of an answer span. It found a relevant span, but its confidence was low, and it didn't extract all risks in a single span.*
	RoBERTa	*Partial Success*	*Answered 'Generative AI poses significant risks such as hallucinations, bias,' with a low score (0.0096). It extracted a longer but still incomplete span.*	*Similar to BERT, RoBERTa is an extractive QA model. It identified a relevant span, but struggled to capture all distinct risks or a more comprehensive answer, and its confidence was low.*
	BART	*Partial Success*	*Answered 'hallucinations, bias,' with a score (0.0169) slightly higher than BERT/RoBERTa, but still missed 'deepfakes'.*	*BART can also perform QA tasks. It successfully identified a relevant portion of the context but, like the other models, did not provide a complete answer encompassing all risks in the given context.*

General Observation

Model performance depends more on architecture + pretraining objective than on model size or name.
When you force models to do tasks they weren't designed for, weaknesses appear immediately.

1. TEXT GENERATION

What happened:

- BERT → Repetitive symbols / nonsense
- RoBERTa → Echoed prompt, didn't really continue meaningfully
- BART → Generated text but incoherent and repetitive

What we learn:

Model	What This Reveals
BERT	Pure encoder → understands text but cannot generate sequences autoregressively
RoBERTa	Same encoder-only limitation as BERT
BART	Has decoder, *can* generate — but base model isn't instruction-tuned → produces messy output

Observation:
Understanding ≠ Generating.
Encoder-only models break when forced to produce text.

2. FILL MASK

What happened:

All three predicted sensible words like *create, generate, produce, develop*

What we learn:

Model	Why it worked
BERT	Trained exactly for MLM (Masked Language Modeling)
RoBERTa	Improved MLM training → strong performance
BART	Denoising autoencoder training includes mask reconstruction

Observation:
When the task matches the pretraining objective, performance is strong even without fine tuning.

3. QUESTION ANSWERING

What happened:

- Models found some risks (hallucinations, bias, deepfakes)
- Low confidence scores
- None gave a complete multi-risk answer

What we learn:

Model	Why partial
All	These are base models, not fine-tuned on QA datasets (like SQuAD)

BERT/RoBERTa	Extractive QA ability exists in architecture, but needs QA fine-tuning
BART	Can do QA, but again lacks task-specific tuning

Observation:
Architecture allows the task, but fine tuning determines quality.

Final Conclusion

This experiment shows:

Architecture dictates capability

- Encoder → Understanding tasks
- Encoder-Decoder → Generation tasks

Pretraining objective shapes strengths

- MLM → Fill-mask strong
- Denoising → Reconstruction ability
- No next-token training → Poor generation

Fine-tuning matters

Even if architecture supports a task, base models perform poorly without task-specific training (like QA).

Unit1 Project - Legal Jargon Classifier

Overview:

This project focuses on building an AI system that converts complex legal language found in Terms & Conditions (T&C), privacy policies, and agreements into simple, easy-to-understand explanations. Legal documents from companies like Apple, Samsung, and other services are often difficult for normal users to understand.

Using a transformer-based Large Language Model (LLM), the system reads legal text and rewrites it in everyday language while preserving the original meaning. The goal is to improve user understanding and make legal information more accessible.

Methodology:

1. **User Input:**
   The user provides a paragraph or multi line text containing legal terms or T&C clauses.
2. **Text Processing:**
   If the input text is long, it is split into smaller chunks so that it fits within the model's input size.
3. **Prompt Engineering:**
   An instruction such as *"Rewrite this legal text in clear and simple language"* is added before the text to guide the model toward simplification instead of just shortening.
4. **Model Processing:**
   The system uses the **BART (facebook/bart-large-cnn)** transformer model through Hugging Face pipelines.
   The model analyzes the legal language and generates a simplified explanation.
5. **Output Generation:**
   Simplified outputs from each chunk are combined to produce a final easy-to-read paragraph.

Output of the System:

The system produces:

- A **clear explanatory paragraph** in layman terms
- The explanation keeps the original meaning but removes complex legal jargon
- Output is human readable and easier for non legal users to understand

Example Output Style:

Instead of:

> "The company shall not be liable for any indirect or consequential damages…"

The system outputs:

> This means the company is not responsible for extra losses or problems that happen as a result of using the service.

Conclusion Key Learnings from the Project

- We understood how **transformer based Large Language Models (LLMs)** like BART can interpret complex text and generate meaningful, human-readable explanations.

- We learned how **Hugging Face pipelines and prompt engineering** can guide a model to perform specific NLP tasks such as legal text simplification.

- We observed how AI can be applied to a **real world problem**, making complicated legal information more accessible and improving user understanding.