

# GENAI

Name :Y KUSUMITA

SRN: PES2UG23CS718

Section: K

Email ID: pes2ug23cs718@pesu.ped.edu

Mobile Number: 8431820485

Github Link: [https://github.com/kusumita-ai/GenAI\\_Assignment.git](https://github.com/kusumita-ai/GenAI_Assignment.git)

### ASSIGNMENT 1 SCREENSHOT:

## Experiment -1 (Text Generation)

```
0 prompt = "The future of Artificial Intelligence is #PES2Q6ZC3J8I8 KUSMIA Y"

for name, model in models.items():
    print(f'<!--name--> OUTPUT:')
    try:
        generator = pipeline("text-generation", model=model)
        output = generator(prompt, max_length=he)
        print(output)
    except Exception as e:
        print(f'TAIL-ED!'), e)

---

BEST OUTPUT:
/usr/local/lib/python3.11/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 (<a href='\"https://huggingface.co/settings/tokens\"')>https://huggingface.co/settings/tokens</a>, 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.
wandb.termlog
config.json 100% ██████████ 570570 [00:00<00.0, 27.0kB/s]
model.safetensors 100% ██████████ 4608440M [00:04<00.0, 75.1MB/s]
If you want to use `bert4keras` as a standalone, add `is_decoder=True`.
tokenizer_config.json 100% ██████████ 48.04k [00:00<00.0, 3.84kB/s]
vocab.txt 100% ██████████ 232k/232k [00:00<00.0, 5.78MB/s]
wandb.termlog
config.json 100% ██████████ 480x480 [00:00<00.0, 15.8kB/s]
Device set to use GPU
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 encode pairs of sequences (BOS-style) with the tokenizer you can select this strategy more precisely by providing a specific strat
Both `max_new_tokens` (=256) and `max_length` (=48) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (<a href='\"https://huggingface.co/docs/transformers/main/en/main_classes/text_generation\"')>https://huggingface.co/docs/transformers/main/en/main_classes/text_generation</a>)
[{"generated_text": "The future of artificial intelligence is....."}]

DEBUG: OUTPUT:
(HEADLESSDONTOR(HTTPSCONNECTIONPOOL(host='huggingface.co', port=443)): Read timed out. (read timeout=30)'), ('request id: 384f6231-8add-47ef-d32a-b0b7080e68d')') ' thrown while requesting GET <a href='\"https://huggingface.co/roberta-base/resolve/main/config.json\"')>https://huggingface.co/roberta-base/resolve/main/config.json</a>
WARNING:huggingface_hub.utils._http:(HEADLESSDONTOR(HTTPSCONNECTIONPOOL(host='huggingface.co', port=443)): Read timed out. (read timeout=30)'), ('request id: 384f6231-8add-47ef-d32a-b0b7080e68d')') ' thrown while requesting GET <a href='\"https://huggingface.co/roberta-base/resolve/main/config.json\"')>https://huggingface.co/roberta-base/resolve/main/config.json</a>
Retrying in s [retry 1/5].
WARNING:huggingface_hub.utils._http:Retrying in s [retry 1/5].
config.json 100% ██████████ 481481 [00:00<00.0, 40.7kB/s]
model.safetensors 100% ██████████ 4980498 [00:05<00.0, 98.8MB/s]
If you want to use `RobertaHeadModel` as a standalone, add `is_decoder=True`.
tokenizer_config.json 100% ██████████ 25.92k [00:00<00.0, 2.43kB/s]
vocab.txt 100% ██████████ 390k/390k [00:00<00.0, 12.3kB/s]
merges.txt 100% ██████████ 455kx455k [00:00<00.0, 8.3kB/s]
wandb.termlog
config.json 100% ██████████ 138Mx138M [00:00<00.0, 15.4kB/s]
(HEADLESSDONTOR(HTTPSCONNECTIONPOOL(host='huggingface.co', port=443)): Read timed out. (read timeout=30)'), ('request id: 433ae9ff-81cd-4e34-94bd-2b67d24c8038')') ' thrown while requesting HEAD <a href='\"https://huggingface.co/roberta-base/resolve/main/add_tokens.json\"')>https://huggingface.co/roberta-base/resolve/main/add_tokens.json</a>
WARNING:huggingface_hub.utils._http:(HEADLESSDONTOR(HTTPSCONNECTIONPOOL(host='huggingface.co', port=443)): Read timed out. (read timeout=30)'), ('request id: 433ae9ff-81cd-4e34-94bd-2b67d24c8038')') ' thrown while requesting HEAD <a href='\"https://huggingface.co/roberta-base/resolve/main/add_tokens.json\"')>https://huggingface.co/roberta-base/resolve/main/add_tokens.json</a>
Retrying in s [retry 1/5].
WARNING:huggingface_hub.utils._http:Retrying in s [retry 1/5].
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 strategy. If you encode pairs of sequences (BOS-style) with the tokenizer you can select this strategy more precisely by providing a specific strat
Both `max_new_tokens` (=256) and `max_length` (=48) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (<a href='\"https://huggingface.co/docs/transformers/main/en/main_classes/text_generation\"')>https://huggingface.co/docs/transformers/main/en/main_classes/text_generation</a>)
[{"generated_text": "The future of artificial intelligence is"}]

---

BEST OUTPUT:
config.json 100% ██████████ 172k/172k [00:00<00.0, 118kB/s]
model.safetensors 100% ██████████ 558Mx558M [00:06<00.0, 72.2MB/s]
Some weights of BartForCausalLM were not initialized from the model checkpoint at facebook/bart-base and are newly initialized: ['ln_attn_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 100% ██████████ 88k/87 [00:00<00.0, 19.2kB/s]
wandb.termlog
config.json 100% ██████████ 450k/450 [00:00<00.0, 18.9kB/s]
merges.txt 100% ██████████ 138Mx138M [00:00<00.0, 18.9kB/s]
tokenizer.json 100% ██████████ 138Mx138M [00:00<00.0, 40.2kB/s]
Device set to use GPU
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 encode pairs of sequences (BOS-style) with the tokenizer you can select this strategy more precisely by providing a specific strat
Both `max_new_tokens` (=256) and `max_length` (=48) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (<a href='\"https://huggingface.co/docs/transformers/main/en/main_classes/text_generation\"')>https://huggingface.co/docs/transformers/main/en/main_classes/text_generation</a>)
[{"generated_text": "The future of artificial intelligence is..."}]
```

## Experiment-2 (Fill-Mask)

```
EXPERIMENT 2: FILL-MASK

1 masked_text = "The goal of generative AI is to [MASK] new content." #pes2ug2kcs718 kusumita y

for name, model in models.items():
    print(f"<div>name: {name}</div> OUTPUT:")
    try:
        fill_mask = pipeline("fill-mask", model=model)
        output = fill_mask(masked_text)
        print(output)
    except Exception as e:
        print(f"FAILED: {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.seq_relationship.weight'].
  This architecture is from a model trained on another task or with another architecture (e.g., initializing a BertForSequenceClassification model from a BertForMaskedLM model).
  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 BertForSequenceClassification model).

BERT OUTPUT:
Device set to use GPU
{'score': 0.53983292439338813, 'token': 3440, 'token_str': 'create', 'sequence': 'the goal of generative ai is to create new content.'}, {'score': 0.1557528036851526, 'token': 9699, 'token_str': 'generate', 'sequence': 'the goal of generative ai is to generate new content.'}, {'score': 0.8548588898728624, 'token': 3965, 'token_str': 'produce', 'sequence': 'the goal of generative ai is to produce new content.'}

GPT2 OUTPUT:
Device set to use GPU
FAILED: No mask_token (<div>mask</div>) found on the input

BART OUTPUT:
Device set to use GPU
FAILED: No mask_token (<div>mask</div>) found on the input
```

### Experiment-3( Question and Answering)

```

question = "What are the risks?"
context = "Generative AI poses significant risks such as hallucinations, bias, and deepfakes." #PES2UG23CS718 KUSUMITA Y

for name, model in models.items():
    print(f"{name} OUTPUT:")
    try:
        qa = pipeline("question-answering", model=model)
        output = qa(question=question, context=context)
        print(output)
    except Exception as e:
        print(f"FAILED: {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  
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': 0.015888878632827282, 'start': 46, 'end': 81, 'answer': 'hallucinations, bias, and deepfakes'})

RoBERTa OUTPUT:  
Device set to use cpu  
Some weights of BertForQuestionAnswering 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': 0.00828557787463069, 'start': 0, 'end': 67, 'answer': 'Generative AI poses significant risks such as hallucinations, bias, and'})

BART OUTPUT:  
Device set to use cpu  
({'score': 0.03820175491273403, 'start': 20, 'end': 81, 'answer': 'significant risks such as hallucinations, bias, and deepfakes'})

## Final Table

Task	Model	Classification (Success/Failure)	Observation (What actually happened?)	Why did this happen? (Architectural Reason)
Generation	BERT	Failure	Generated nonsense or random symbols.	BERT is an encoder it is not trained to predict the next-word.
Generation	RoBERTa	Failure	Repeated the input prompt without any continuation.	RoBERTa is also encoder and focuses on understanding text not generating new tokens.
Generation	BART	Partial Failure	Produced incoherent and random text with warnings.	BART has an encoder-decoder architecture, but the base model is not fine-tuned for causal text generation.
Fill-Mask	BERT	Success	predicted 'create' generate'.	BERT is trained on Masked Language Modeling (MLM).
Fill-Mask	RoBERTa	Success	Produced meaningful predictions similar to BERT.	RoBERTa is an optimized version of BERT and is also trained with MLM objectives.
Fill-Mask	BART	Partial Failure	Predictions were weak or inconsistent.	BART is trained for denoising sequences not specifically for word-level masking.
QA	BERT	Partial Failure	Returned incomplete or weak answers.	The base BERT model is not fine-tuned on Question Answering datasets.
QA	RoBERTa	Partial Failure	Produced weak or incorrect answers.	RoBERTa base model lacks task-specific fine-tuning for QA.
QA	BART	Failure	Failed to extract the correct answer from context.	Although BART can generate text it is not fine-tuned for extractive Question Answering.

## ASSIGNMENT-2 (PROJECT)

### Smart Resume Parser

```

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

```

Warning: /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% ██████████ 998/998 [00:00<00:00, 87.3kB/s]

model.safetensors: 100% ██████████ 1.33G/1.33G [00:14<00:00, 256MB/s]

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.weight']  
- 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 BertForSequenceClassification model from a  
- 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 from a BertForSeque

tokenizer\_config.json: 100% ██████████ 60.0/60.0 [00:00<00:00, 2.05kB/s]

vocab.txt: 213k/7 [00:00<00:00, 3.14MB/s]

Device set to use cpu

```

resume_text = """
John doe
Email: abc@gmail.com

Education:
B.Tech in Computer Science from PES University

Experience:
Software Intern at Apple
Worked on iOS application testing and debugging.
"""

```

```
16]
↳ entities = ner(resume_text)
entities
```

```
... [{ 'entity_group': 'PER',
      'score': np.float32(0.74730295),
      'word': 'John',
      'start': 1,
      'end': 5},
     { 'entity_group': 'MISC',
      'score': np.float32(0.39845026),
      'word': 'Computer Science',
      'start': 53,
      'end': 69},
     { 'entity_group': 'ORG',
      'score': np.float32(0.9776699),
      'word': 'PES University',
      'start': 75,
      'end': 89},
     { 'entity_group': 'ORG',
      'score': np.float32(0.9964496),
      'word': 'Apple',
      'start': 122,
      'end': 127},
     { 'entity_group': 'MISC',
      'score': np.float32(0.9849535),
      'word': 'iOS',
      'start': 138,
      'end': 141}]
```

```
17]
↳ names = [e['word'] for e in entities if e['entity_group'] == 'PER']
organizations = [e['word'] for e in entities if e['entity_group'] == 'ORG']

print("Extracted Name(s):", names)
print("Extracted Organization(s):", organizations)
```

```
Extracted Name(s): ['John']
Extracted Organization(s): ['PES University', 'Apple']
```