MITHA M K

PES2UG23CS339

F

Task	Model	Success / Failure
Text Generation	BERT	Failure
Text Generation	RoBERTa	Failure
Text Generation	BART	Success
Fill Mask	BERT	Success
Fill Mask	RoBERTa	Success
Fill Mask	BART	Partial
Question Answering	BERT	Partial
Question Answering	RoBERTa	Partial
Question Answering	BART	Partial

e	What I observed
	BERT could not generate proper text and showed errors.
	RoBERTa also failed to generate meaningful text.
	BART generated a meaningful continuation of the prompt.
	BERT correctly predicted words like create and generate.
	It worked correctly after using `<mask>` instead of `[MASK]`.
	BART filled the mask but predictions were less accurate.
	BERT gave an answer but it was not very accurate.
	RoBERTa produced weak or incomplete answers.
	BART returned partial answers with low confidence.

	Why it happened
.	BERT is an encoder-only model and not designed for text generation.
	RoBERTa is encoder-only and lacks a decoder.
ɔt.	BART is an encoder–decoder model designed for generation.
e.	BERT is trained using masked language modeling.
sK].	RoBERTa uses a different mask token but is MLM-trained.
	BART is not mainly trained for masked language modeling.
	The model is not fine-tuned for question answering.
	QA fine-tuning is required for better performance.
	The QA head was not fine-tuned.

In this hands-on assignment, I worked with three transformer models: **BERT**, **RoBERTa**, and **BART** using the Hugging Face transformers library. The aim was to understand how different model architectures behave on different NLP tasks.

For **text generation**, BERT and RoBERTa failed to generate meaningful text, while BART successfully generated a coherent output. This showed that encoder-only models are not suitable for text generation, whereas encoder–decoder models perform better.

In the **fill-mask** task, BERT and RoBERTa performed well and predicted correct words. RoBERTa initially failed due to using a different mask token, but worked after correction. BART was able to fill the mask but with less accurate predictions.

For **question answering**, all three models produced weak or partial answers because the base models were not fine-tuned for this task.

Overall, this experiment helped me understand that transformer models work best on tasks they are specifically designed and trained for, and that architecture plays an important role in model performance.