Bert

in submitting a resource to be included here, please feel free to open a Pull Request and we’ll review it! The resource should ideally demonstrate something new instead of duplicating an existing resource.

Text Classification

* A blog post on [BERT Text Classification in a different language](https://www.philschmid.de/bert-text-classification-in-a-different-language).
* A notebook for [Finetuning BERT (and friends) for multi-label text classification](https://colab.research.google.com/github/NielsRogge/Transformers-Tutorials/blob/master/BERT/Fine_tuning_BERT_(and_friends)_for_multi_label_text_classification.ipynb).
* A notebook on how to [Finetune BERT for multi-label classification using PyTorch](https://colab.research.google.com/github/abhimishra91/transformers-tutorials/blob/master/transformers_multi_label_classification.ipynb). 🌎
* A notebook on how to [warm-start an EncoderDecoder model with BERT for summarization](https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/BERT2BERT_for_CNN_Dailymail.ipynb).
* [BertForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.BertForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification.ipynb).
* [TFBertForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.TFBertForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification-tf.ipynb).
* [FlaxBertForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.FlaxBertForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification_flax.ipynb).
* [Text classification task guide](https://huggingface.co/docs/transformers/tasks/sequence_classification)

Token Classification

* A blog post on how to use [Hugging Face Transformers with Keras: Fine-tune a non-English BERT for Named Entity Recognition](https://www.philschmid.de/huggingface-transformers-keras-tf).
* A notebook for [Finetuning BERT for named-entity recognition](https://colab.research.google.com/github/NielsRogge/Transformers-Tutorials/blob/master/Custom_Named_Entity_Recognition_with_BERT_only_first_wordpiece.ipynb) using only the first wordpiece of each word in the word label during tokenization. To propagate the label of the word to all wordpieces, see this [version](https://github.com/NielsRogge/Transformers-Tutorials/blob/master/BERT/Custom_Named_Entity_Recognition_with_BERT.ipynb) of the notebook instead.
* [BertForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.BertForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/token-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification.ipynb).
* [TFBertForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.TFBertForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/token-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification-tf.ipynb).
* [FlaxBertForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.FlaxBertForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/token-classification).
* [Token classification](https://huggingface.co/course/chapter7/2?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Token classification task guide](https://huggingface.co/docs/transformers/tasks/token_classification)

Fill-Mask

* [BertForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.BertForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling#robertabertdistilbert-and-masked-language-modeling) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling.ipynb).
* [TFBertForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.TFBertForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/language-modeling#run_mlmpy) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling-tf.ipynb).
* [FlaxBertForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.FlaxBertForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/language-modeling#masked-language-modeling) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/masked_language_modeling_flax.ipynb).
* [Masked language modeling](https://huggingface.co/course/chapter7/3?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Masked language modeling task guide](https://huggingface.co/docs/transformers/tasks/masked_language_modeling)

Question Answering

* [BertForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.BertForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/question_answering.ipynb).
* [TFBertForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.TFBertForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/question-answering) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/question_answering-tf.ipynb).
* [FlaxBertForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.FlaxBertForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/question-answering).
* [Question answering](https://huggingface.co/course/chapter7/7?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Question answering task guide](https://huggingface.co/docs/transformers/tasks/question_answering)

Multiple choice

* [BertForMultipleChoice](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.BertForMultipleChoice) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/multiple-choice) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/multiple_choice.ipynb).
* [TFBertForMultipleChoice](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/bert#transformers.TFBertForMultipleChoice) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/multiple-choice) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/multiple_choice-tf.ipynb).
* [Multiple choice task guide](https://huggingface.co/docs/transformers/tasks/multiple_choice)

⚡️ Inference

* A blog post on how to [Accelerate BERT inference with Hugging Face Transformers and AWS Inferentia](https://huggingface.co/blog/bert-inferentia-sagemaker).
* A blog post on how to [Accelerate BERT inference with DeepSpeed-Inference on GPUs](https://www.philschmid.de/bert-deepspeed-inference).

⚙️ Pretraining

* A blog post on [Pre-Training BERT with Hugging Face Transformers and Habana Gaudi](https://www.philschmid.de/pre-training-bert-habana).

🚀 Deploy

* A blog post on how to [Convert Transformers to ONNX with Hugging Face Optimum](https://www.philschmid.de/convert-transformers-to-onnx).
* A blog post on how to [Setup Deep Learning environment for Hugging Face Transformers with Habana Gaudi on AWS](https://www.philschmid.de/getting-started-habana-gaudi#conclusion).
* A blog post on [Autoscaling BERT with Hugging Face Transformers, Amazon SageMaker and Terraform module](https://www.philschmid.de/terraform-huggingface-amazon-sagemaker-advanced).
* A blog post on [Serverless BERT with HuggingFace, AWS Lambda, and Docker](https://www.philschmid.de/serverless-bert-with-huggingface-aws-lambda-docker).
* A blog post on [Hugging Face Transformers BERT fine-tuning using Amazon SageMaker and Training Compiler](https://www.philschmid.de/huggingface-amazon-sagemaker-training-compiler).
* A blog post on [Task-specific knowledge distillation for BERT using Transformers & Amazon SageMaker](https://www.philschmid.de/knowledge-distillation-bert-transformers).

Gpt

Text Classification

* A blog post on [outperforming OpenAI GPT-3 with SetFit for text-classification](https://www.philschmid.de/getting-started-setfit).
* See also: [Text classification task guide](https://huggingface.co/docs/transformers/tasks/sequence_classification)

Text Generation

* A blog on how to [Finetune a non-English GPT-2 Model with Hugging Face](https://www.philschmid.de/fine-tune-a-non-english-gpt-2-model-with-huggingface).
* A blog on [How to generate text: using different decoding methods for language generation with Transformers](https://huggingface.co/blog/how-to-generate) with GPT-2.
* A blog on [Training CodeParrot 🦜 from Scratch](https://huggingface.co/blog/codeparrot), a large GPT-2 model.
* A blog on [Faster Text Generation with TensorFlow and XLA](https://huggingface.co/blog/tf-xla-generate) with GPT-2.
* A blog on [How to train a Language Model with Megatron-LM](https://huggingface.co/blog/megatron-training) with a GPT-2 model.
* A notebook on how to [finetune GPT2 to generate lyrics in the style of your favorite artist](https://colab.research.google.com/github/AlekseyKorshuk/huggingartists/blob/master/huggingartists-demo.ipynb). 🌎
* A notebook on how to [finetune GPT2 to generate tweets in the style of your favorite Twitter user](https://colab.research.google.com/github/borisdayma/huggingtweets/blob/master/huggingtweets-demo.ipynb). 🌎
* [Causal language modeling](https://huggingface.co/course/en/chapter7/6?fw=pt#training-a-causal-language-model-from-scratch) chapter of the 🤗 Hugging Face Course.
* [OpenAIGPTLMHeadModel](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/openai-gpt#transformers.OpenAIGPTLMHeadModel) is supported by this [causal language modeling example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling#gpt-2gpt-and-causal-language-modeling), [text generation example script](https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-generation/run_generation.py) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling.ipynb).
* [TFOpenAIGPTLMHeadModel](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/openai-gpt#transformers.TFOpenAIGPTLMHeadModel) is supported by this [causal language modeling example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/language-modeling#run_clmpy) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling-tf.ipynb).
* See also: [Causal language modeling task guide](https://huggingface.co/docs/transformers/tasks/language_modeling)

Token Classification

* A course material on [Byte-Pair Encoding tokenization](https://huggingface.co/course/en/chapter6/5).

LLama-2

A list of official Hugging Face and community (indicated by 🌎) resources to help you get started with LLaMA2. If you’re interested in submitting a resource to be included here, please feel free to open a Pull Request and we’ll review it! The resource should ideally demonstrate something new instead of duplicating an existing resource.

* [Llama 2 is here - get it on Hugging Face](https://huggingface.co/blog/llama2), a blog post about Llama 2 and how to use it with 🤗 Transformers and 🤗 PEFT.
* [LLaMA 2 - Every Resource you need](https://www.philschmid.de/llama-2), a compilation of relevant resources to learn about LLaMA 2 and how to get started quickly.

Text Generation

* A [notebook](https://colab.research.google.com/drive/1PEQyJO1-f6j0S_XJ8DV50NkpzasXkrzd?usp=sharing) on how to fine-tune Llama 2 in Google Colab using QLoRA and 4-bit precision. 🌎
* A [notebook](https://colab.research.google.com/drive/134o_cXcMe_lsvl15ZE_4Y75Kstepsntu?usp=sharing) on how to fine-tune the “Llama-v2-7b-guanaco” model with 4-bit QLoRA and generate Q&A datasets from PDFs. 🌎

Text Classification

* A [notebook](https://colab.research.google.com/drive/1ggaa2oRFphdBmqIjSEbnb_HGkcIRC2ZB?usp=sharing) on how to fine-tune the Llama 2 model with QLoRa, TRL, and Korean text classification dataset. 🌎🇰🇷

⚗️ Optimization

* [Fine-tune Llama 2 with DPO](https://huggingface.co/blog/dpo-trl), a guide to using the TRL library’s DPO method to fine tune Llama 2 on a specific dataset.
* [Extended Guide: Instruction-tune Llama 2](https://www.philschmid.de/instruction-tune-llama-2), a guide to training Llama 2 to generate instructions from inputs, transforming the model from instruction-following to instruction-giving.
* A [notebook](https://colab.research.google.com/drive/1SYpgFpcmtIUzdE7pxqknrM4ArCASfkFQ?usp=sharing) on how to fine-tune the Llama 2 model on a personal computer using QLoRa and TRL. 🌎

⚡️ Inference

* A [notebook](https://colab.research.google.com/drive/1TC56ArKerXUpbgRy5vM3woRsbTEVNq7h?usp=sharing) on how to quantize the Llama 2 model using GPTQ from the AutoGPTQ library. 🌎
* A [notebook](https://colab.research.google.com/drive/1X1z9Q6domMKl2CnEM0QGHNwidLfR4dW2?usp=sharing) on how to run the Llama 2 Chat Model with 4-bit quantization on a local computer or Google Colab. 🌎

🚀 Deploy

* [Fine-tune LLaMA 2 (7-70B) on Amazon SageMaker](https://www.philschmid.de/sagemaker-llama2-qlora), a complete guide from setup to QLoRA fine-tuning and deployment on Amazon SageMaker.
* [Deploy Llama 2 7B/13B/70B on Amazon SageMaker](https://www.philschmid.de/sagemaker-llama-llm), a guide on using Hugging Face’s LLM DLC container for secure and scalable deployment.

Gpt-2

* A blog on how to [Finetune a non-English GPT-2 Model with Hugging Face](https://www.philschmid.de/fine-tune-a-non-english-gpt-2-model-with-huggingface).
* A blog on [How to generate text: using different decoding methods for language generation with Transformers](https://huggingface.co/blog/how-to-generate) with GPT-2.
* A blog on [Training CodeParrot 🦜 from Scratch](https://huggingface.co/blog/codeparrot), a large GPT-2 model.
* A blog on [Faster Text Generation with TensorFlow and XLA](https://huggingface.co/blog/tf-xla-generate) with GPT-2.
* A blog on [How to train a Language Model with Megatron-LM](https://huggingface.co/blog/megatron-training) with a GPT-2 model.
* A notebook on how to [finetune GPT2 to generate lyrics in the style of your favorite artist](https://colab.research.google.com/github/AlekseyKorshuk/huggingartists/blob/master/huggingartists-demo.ipynb). 🌎
* A notebook on how to [finetune GPT2 to generate tweets in the style of your favorite Twitter user](https://colab.research.google.com/github/borisdayma/huggingtweets/blob/master/huggingtweets-demo.ipynb). 🌎
* [Causal language modeling](https://huggingface.co/course/en/chapter7/6?fw=pt#training-a-causal-language-model-from-scratch) chapter of the 🤗 Hugging Face Course.
* [GPT2LMHeadModel](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/gpt2#transformers.GPT2LMHeadModel) is supported by this [causal language modeling example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling#gpt-2gpt-and-causal-language-modeling), [text generation example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-generation), and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling.ipynb).
* [TFGPT2LMHeadModel](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/gpt2#transformers.TFGPT2LMHeadModel) is supported by this [causal language modeling example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/language-modeling#run_clmpy) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling-tf.ipynb).
* [FlaxGPT2LMHeadModel](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/gpt2#transformers.FlaxGPT2LMHeadModel) is supported by this [causal language modeling example script](https://github.com/huggingface/transformers/tree/main/examples/flax/language-modeling#causal-language-modeling) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/causal_language_modeling_flax.ipynb).
* [Text classification task guide](https://huggingface.co/docs/transformers/tasks/sequence_classification)
* [Token classification task guide](https://huggingface.co/docs/transformers/tasks/token_classification)
* [Causal language modeling task guide](https://huggingface.co/docs/transformers/tasks/language_modeling)

Roberta

Text Classification

* A blog on [Getting Started with Sentiment Analysis on Twitter](https://huggingface.co/blog/sentiment-analysis-twitter) using RoBERTa and the [Inference API](https://huggingface.co/inference-api).
* A blog on [Opinion Classification with Kili and Hugging Face AutoTrain](https://huggingface.co/blog/opinion-classification-with-kili) using RoBERTa.
* A notebook on how to [finetune RoBERTa for sentiment analysis](https://colab.research.google.com/github/DhavalTaunk08/NLP_scripts/blob/master/sentiment_analysis_using_roberta.ipynb). 🌎
* [RobertaForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.RobertaForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification.ipynb).
* [TFRobertaForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.TFRobertaForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification-tf.ipynb).
* [FlaxRobertaForSequenceClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.FlaxRobertaForSequenceClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/text-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification_flax.ipynb).
* [Text classification task guide](https://huggingface.co/docs/transformers/tasks/sequence_classification)

Token Classification

* [RobertaForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.RobertaForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/token-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification.ipynb).
* [TFRobertaForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.TFRobertaForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/token-classification) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification-tf.ipynb).
* [FlaxRobertaForTokenClassification](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.FlaxRobertaForTokenClassification) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/token-classification).
* [Token classification](https://huggingface.co/course/chapter7/2?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Token classification task guide](https://huggingface.co/docs/transformers/tasks/token_classification)

Fill-Mask

* A blog on [How to train a new language model from scratch using Transformers and Tokenizers](https://huggingface.co/blog/how-to-train) with RoBERTa.
* [RobertaForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.RobertaForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling#robertabertdistilbert-and-masked-language-modeling) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling.ipynb).
* [TFRobertaForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.TFRobertaForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/language-modeling#run_mlmpy) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/language_modeling-tf.ipynb).
* [FlaxRobertaForMaskedLM](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.FlaxRobertaForMaskedLM) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/language-modeling#masked-language-modeling) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/masked_language_modeling_flax.ipynb).
* [Masked language modeling](https://huggingface.co/course/chapter7/3?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Masked language modeling task guide](https://huggingface.co/docs/transformers/tasks/masked_language_modeling)

Question Answering

* A blog on [Accelerated Inference with Optimum and Transformers Pipelines](https://huggingface.co/blog/optimum-inference) with RoBERTa for question answering.
* [RobertaForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.RobertaForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/question-answering) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/question_answering.ipynb).
* [TFRobertaForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.TFRobertaForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/question-answering) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/question_answering-tf.ipynb).
* [FlaxRobertaForQuestionAnswering](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.FlaxRobertaForQuestionAnswering) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/flax/question-answering).
* [Question answering](https://huggingface.co/course/chapter7/7?fw=pt) chapter of the 🤗 Hugging Face Course.
* [Question answering task guide](https://huggingface.co/docs/transformers/tasks/question_answering)

Multiple choice

* [RobertaForMultipleChoice](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.RobertaForMultipleChoice) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/pytorch/multiple-choice) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/multiple_choice.ipynb).
* [TFRobertaForMultipleChoice](https://huggingface.co/docs/transformers/v4.35.2/en/model_doc/roberta#transformers.TFRobertaForMultipleChoice) is supported by this [example script](https://github.com/huggingface/transformers/tree/main/examples/tensorflow/multiple-choice) and [notebook](https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/multiple_choice-tf.ipynb).
* [Multiple choice task guide](https://huggingface.co/docs/transformers/tasks/multiple_choice)