Summarization (Seq2Seq model) training examples

The following example showcases how to finetune a sequence-to-sequence model for summarization using the JAX/Flax backend.

JAX/Flax allows you to trace pure functions and compile them into efficient, fused accelerator code on both GPU and TPU. Models written in JAX/Flax are **immutable** and updated in a purely functional way which enables simple and efficient model parallelism.

`run_summarization_flax.py` is a lightweight example of how to download and preprocess a dataset from the 🤗 Datasets library or use your own files (jsonlines or csv), then fine-tune one of the architectures above on it.

For custom datasets in `jsonlines` format please see: https://huggingface.co/docs/datasets/loading_datasets.html#json-files and you also will find examples of these below.

Train the model

Next we can run the example script to train the model:

```
python run_summarization_flax.py \
    --output_dir ./bart-base-xsum \
    --model_name_or_path facebook/bart-base \
    --tokenizer_name facebook/bart-base \
    --dataset_name="xsum" \
    --do_train --do_eval --do_predict --predict_with_generate \
    --num_train_epochs 6 \
    --learning_rate 5e-5 --warmup_steps 0 \
    --per_device_train_batch_size 64 \
    --per_device_eval_batch_size 64 \
    --overwrite_output_dir \
    --max_source_length 512 --max_target_length 64 \
    --push_to_hub
```

This should finish in 37min, with validation loss and ROUGE2 score of 1.7785 and 17.01 respectively after 6 epochs. training statistics can be accessed on tfhub.de.

> *Note that here we used default* `generate` *arguments, using arguments specific for* `xsum` *dataset should give better ROUGE scores.*