XTREME-S benchmark examples

Maintainers: Anton Lozhkov and Patrick von Platen

The Cross-lingual TRansfer Evaluation of Multilingual Encoders for Speech (XTREME-S) benchmark is a benchmark designed to evaluate speech representations across languages, tasks, domains and data regimes. It covers XX typologically diverse languages and seven downstream tasks grouped in four families: speech recognition, translation, classification and retrieval.

XTREME-S covers speech recognition with Fleurs, Multilingual LibriSpeech (MLS) and VoxPopuli, speech translation with CoVoST-2, speech classification with LangID (Fleurs) and intent classification (MInds-14) and finally speech(-text) retrieval with Fleurs. Each of the tasks covers a subset of the 102 languages included in XTREME-S (shown here with their ISO 3166-1 codes): afr, amh, ara, asm, ast, azj, bel, ben, bos, cat, ceb, ces, cmn, cym, dan, deu, ell, eng, spa, est, fas, ful, fin, tgl, fra, gle, glg, guj, hau, heb, hin, hrv, hun, hye, ind, ibo, isl, ita, jpn, jav, kat, kam, kea, kaz, khm, kan, kor, ckb, kir, ltz, lug, lin, lao, lit, luo, lav, mri, mkd, mal, mon, mar, msa, mlt, mya, nob, npi, nld, nso, nya, oci, orm, ory, pan, pol, pus, por, ron, rus, bul, snd, slk, slv, sna, som, srp, swe, swh, tam, tel, tgk, tha, tur, ukr, umb, urd, uzb, vie, wol, xho, yor, yue and zul.

Paper: XTREME-S: Evaluating Cross-lingual Speech Representations

Dataset: https://huggingface.co/datasets/google/xtreme_s

Fine-tuning for the XTREME-S tasks

Based on the `run_xtreme_s.py` script.

This script can fine-tune any of the pretrained speech models on the hub on the XTREME-S dataset tasks.

XTREME-S is made up of 7 different tasks. Here is how to run the script on each of them:

```
export TASK_NAME=mls.all

python run_xtreme_s.py \
    --model_name_or_path="facebook/wav2vec2-xls-r-300m" \
    --task="${TASK_NAME}" \
    --output_dir="xtreme_s_xlsr_${TASK_NAME}" \
    --num_train_epochs=100 \
    --per_device_train_batch_size=32 \
    --learning_rate="3e-4" \
    --target_column_name="transcription" \
    --save_steps=500 \
    --eval_steps=500 \
    --gradient_checkpointing \
```

```
    --fp16 \
    --group_by_length \
    --do_train \
    --do_eval \
    --do_predict \
    --push_to_hub
```

where `TASK_NAME` can be one of: `mls`, `voxpopuli`, `covost2`, `fleurs-asr`, `fleurs-lang_id`, `minds14`.

We get the following results on the test set of the benchmark's datasets. The corresponding training commands for each dataset are given in the sections below:

Task	Dataset	Result	Fine-tuned model & logs	Training time	GPUs
Speech Recognition	MLS	30.33 WER	here	18:47:25	8xV100
Speech Recognition	VoxPopuli	-	-	-	-
Speech Recognition	FLEURS	-	-	-	-
Speech Translation	CoVoST-2	-	-	-	-
Speech Classification	Minds-14	90.15 F1 / 90.33 Acc.	here	2:54:21	2xA100
Speech Classification	FLEURS	-	-	-	-
Speech Retrieval	FLEURS	-	-	-	-

Speech Recognition with MLS

The following command shows how to fine-tune the XLS-R model on XTREME-S MLS using 8 GPUs in half-precision.

```
python -m torch.distributed.launch \
    --nproc_per_node=8 \
    run_xtreme_s.py \
    --task="mls" \
```

```
    --language="all" \
    --model_name_or_path="facebook/wav2vec2-xls-r-300m" \
    --output_dir="xtreme_s_xlsr_300m_mls" \
    --overwrite_output_dir \
    --num_train_epochs=100 \
    --per_device_train_batch_size=4 \
    --per_device_eval_batch_size=1 \
    --gradient_accumulation_steps=2 \
    --learning_rate="3e-4" \
    --warmup_steps=3000 \
    --evaluation_strategy="steps" \
    --max_duration_in_seconds=20 \
    --save_steps=500 \
    --eval_steps=500 \
    --logging_steps=1 \
    --layerdrop=0.0 \
    --mask_time_prob=0.3 \
    --mask_time_length=10 \
    --mask_feature_prob=0.1 \
    --mask_feature_length=64 \
    --freeze_feature_encoder \
    --gradient_checkpointing \
    --fp16 \
    --group_by_length \
    --do_train \
    --do_eval \
    --do_predict \
    --metric_for_best_model="wer" \
    --greater_is_better=False \
    --load_best_model_at_end \
    --push_to_hub
```

On 8 V100 GPUs, this script should run in ~19 hours and yield a cross-entropy loss of **0.6215** and word error rate of **30.33**

Speech Classification with Minds-14

The following command shows how to fine-tune the XLS-R model on XTREME-S MLS using 2 GPUs in half-precision.

```
python -m torch.distributed.launch \
    --nproc_per_node=2 \
    run_xtreme_s.py \
    --task="minds14" \
    --language="all" \
    --model_name_or_path="facebook/wav2vec2-xls-r-300m" \
    --output_dir="xtreme_s_xlsr_300m_minds14" \
```

```
--overwrite_output_dir \
--num_train_epochs=50 \
--per_device_train_batch_size=32 \
--per_device_eval_batch_size=8 \
--gradient_accumulation_steps=1 \
--learning_rate="3e-4" \
--warmup_steps=1500 \
--evaluation_strategy="steps" \
--max_duration_in_seconds=30 \
--save_steps=200 \
--eval_steps=200 \
--logging_steps=1 \
--layerdrop=0.0 \
--mask_time_prob=0.3 \
--mask_time_length=10 \
--mask_feature_prob=0.1 \
--mask_feature_length=64 \
--freeze_feature_encoder \
--gradient_checkpointing \
--fp16 \
--group_by_length \
--do_train \
--do_eval \
--do_predict \
--metric_for_best_model="f1" \
--greater_is_better=True \
--load_best_model_at_end \
--push_to_hub
```

On 2 A100 GPUs, this script should run in ~5 hours and yield a cross-entropy loss of **0.4119** and F1 score of **90.15**