# TF-NLP Model Garden

## Introduction

The TF-NLP library provides a collection of scripts for training and evaluating transformer-based models, on various tasks such as sentence classification, question answering, and translation. Additionally, we provide checkpoints of pretrained models which can be finetuned on downstream tasks.

### How to Train Models

Model Garden can be easily installed with `pip install tf-models-nightly`. After installation, check out [this instruction]() on how to train models with this codebase.

By default, the experiment runs on GPUs. To run on TPUs, one should overwrite `runtime.distribution_strategy` and set the tpu address. See [RuntimeConfig]() for details.

In general, the experiments can run with the folloing command by setting the corresponding `${TASK}`, `${TASK_CONFIG}`, `${MODEL_CONFIG}`.

```
EXPERIMENT=???
TASK_CONFIG=???
MODEL_CONFIG=???
EXRTRA_PARAMS=???
MODEL_DIR=???  # a-folder-to-hold-checkpoints-and-logs
python3 train.py \
  --experiment=${EXPERIMENT} \
  --mode=train_and_eval \
  --model_dir=${MODEL_DIR} \
  --config_file=${TASK_CONFIG} \
  --config_file=${MODEL_CONFIG} \
  --params_override=${EXRTRA_PARAMS}
```

- `EXPERIMENT` can be found under `configs/`
- `TASK_CONFIG` can be found under `configs/experiments/`
- `MODEL_CONFIG` can be found under `configs/models/`

**Order of params override:**

1. `train.py` looks up the registered `ExperimentConfig` with `${EXPERIMENT}`
2. Overrides params in `TaskConfig` in `${TASK_CONFIG}`
3. Overrides params `model` in `TaskConfig` with `${MODEL_CONFIG}`
4. Overrides any params in `ExperimentConfig` with `${EXTRA_PARAMS}`

Note that

1. `${TASK_CONFIG}`, `${MODEL_CONFIG}`, `${EXTRA_PARAMS}` can be optional when EXPERIMENT default is enough.
2. `${TASK_CONFIG}`, `${MODEL_CONFIG}`, `${EXTRA_PARAMS}` are only guaranteed to be compatible to it's `${EXPERIMENT}` that defines it.

## Experiments

| NAME | EXPERIMENT | TASK_CONFIG | MODEL_CONFIG | EXRT |
|---|---|---|---|---|
| BERT-base GLUE/MNLI-matched finetune | bert/sentence_prediction | glue_mnli_matched.yaml | bert_en_uncased_base.yaml | ▶ dat base |
| BERT-base GLUE/MNLI-matched finetune | bert/sentence_prediction | glue_mnli_matched.yaml | bert_en_uncased_base.yaml | ▶ dat base |
| BERT-base SQuAD v1.1 finetune | bert/squad | squad_v1.yaml | bert_en_uncased_base.yaml | ▶ dat base |
| ALBERT-base SQuAD v1.1 finetune | bert/squad | squad_v1.yaml | albert_base.yaml | ▶ dat alber init |
| Transformer-large WMT14/en-de scratch | wmt_transformer/large | | | ▶ enc sente |

## Useful links

How to Train Models

List of Pretrained Models for finetuning

How to Publish Models

TensorFlow blog on Model Garden.