Source:

github.com/prdwb/attentive_history_selection

Scripts

- cqa_run_his_atten.py. Entry code.
- cqa_supports.py. Utility functions.
- cqa_gen_batches.py. Generate batches.
- cqa_model.py. Our models.
- scorer.py. Official evaluation script for QuAC.

More useful shit:

link to Huggingface Medium article on TF -> PyTorch:

https://medium.com/huggingface/from-tensorflow-to-pytorch-265f40ef2a28

Maybe to start:

study cqa_run_his_atten.py script and start from there; this is the main code

FIRST STEP: get the stuff from cqa_supports up and running

To re-implement in PyTorch:

- cqa_run_his_atten.py
  - main script
- cqa_flags.py
  - can just use opt.argparse in the main script, cqa_run_his_atten.py
- cqa_gen_batches.py
  - generates batches (no duh)
  - train batches: cqa_gen_example_aware_batches_v2
    - this is the only function that cqa_run_his_atten.py uses ?
- cqa_model.py
  - the models
  - used functions:
    - bert_rep - 1
    - bert_segment_rep - 1
    - cqa_model - 0
    - aux_cqa_model - 0
    - yesno_model - 0
    - followup_model - 0
    - history_attention_net - 1
    - disable_history_attention_net - 0
    - fine_grained_history_attention_net - 1
- cqa_supports.py
  - contains the "convert_examples_to_features" function, which I have seen before elsewhere
  - transformers/examples/utils_*.py /
  - what does 'reformulated question' refer to????
  - rewrite:
    - convert_examples_to_example_variations
    - convert_examples_to_features

        - convert_examples_to_variations_and_then_features

Can reuse PyTorch script:

- modeling.py
    - we can import all of this stuff from pytorch
        - BertConfig
        - BertModel
        - scripts:
            - modeling_bert.py
            - configuration_bert.py
            - configuration_utils.py
    - modeling.BertConfig
    - if FLAGS.init_checkpoint:
        - (assignment_map, initialized_variable_names) = modeling.get_assigment_map_from_checkpoint(tvars, FLAGS.init_checkpoint)
        - tf.train.init_from_checkpoint(FLAGS.init_checkpoint, assignment_map)
    - seq_length = modeling.get_shape_list(input_ids)[1]

- optimization.py
    - this just uses Adam? I think we can just import Adam in PyTorch....
- reindent.py
    - this doesn't even matter...?
- scorer.py
    - only uses 'external_call' function
    - val_eval_res = external_call(val_file_json, output_prediction_file)
    - some sort of standard scoring file
- tokenization.py
    - tokenizer = tokenization.FullTokenizer(vocab_file=FLAGS.vocab_file, do_lower_case=FLAGS.do_lower_case)
    - Pretty sure we can just import the bert tokenizer from PyTorch

Calendar:

- Done with coding/model by 19 March (paper due 30 March)
- Week 1: cqa_support.py script running which includes the convert_to_features... functions
- Week 2: cqa_gen_batches.py (228 lines) + ***!!! SET UP GITHUB !!!*** + cqa_model.py
- Week 3: cqa_model.py - getting the encoder working (BERT + PosHAE)
- Week 4: cqa_model.py - get the encoder working (BERT + PosHAE) + adding cqa_flags.py added to cqa_run_his_atten.py
- Week 5:
- Week 6:
- Week 7:
- Week 8:
- Week 9: